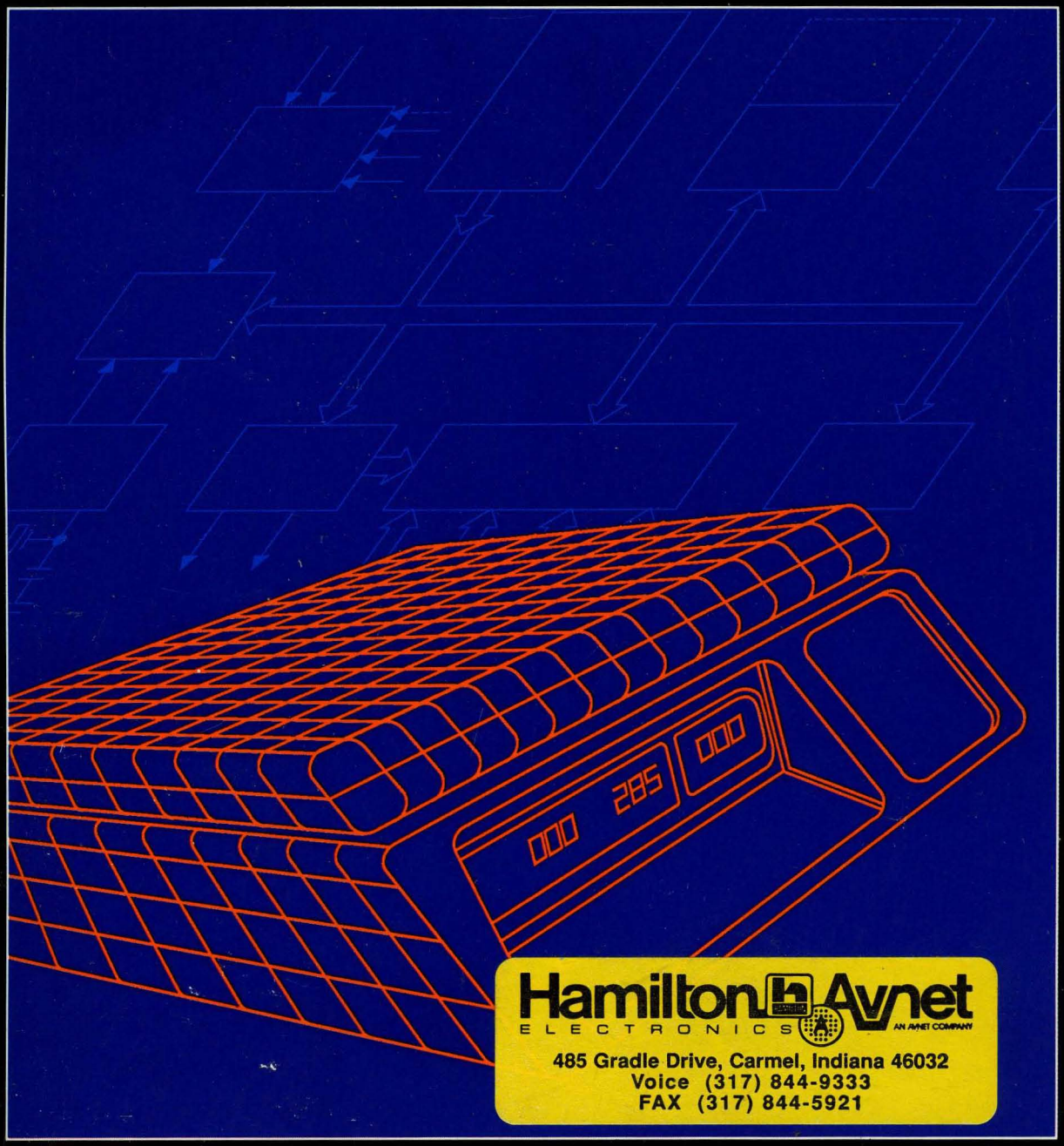
intel

8-Bit Embedded Controller Handbook



Order Number: 270645-001

intel
8-Bit Embedded Controller Handbook

1989

LITERATURE

To order Intel Literature or obtain literature pricing information in the U.S. and Canada call or write Intel Literature Sales. In Europe and other international locations, please contact your local sales office or distributor.

INTEL LITERATURE SALES
P.O. BOX 58130
SANTA CLARA, CA 95052-8130

In the U.S. and Canada
call toll free
(800) 548-4725

CURRENT HANDBOOKS

Product line handbooks contain data sheets, application notes, article reprints and other design information.

TITLE	LITERATURE ORDER NUMBER
COMPLETE SET OF HANDBOOKS (Available in U.S. and Canada only)	**231003**
AUTOMOTIVE PRODUCTS HANDBOOK (Not included in handbook set)	231792
COMPONENTS QUALITY/RELIABILITY HANDBOOK	210997
EMBEDDED CONTROL APPLICATIONS HANDBOOK	270648
8-BIT EMBEDDED CONTROLLER HANDBOOK	270645
16-BIT EMBEDDED CONTROLLER HANDBOOK	270646
32-BIT EMBEDDED CONTROLLER HANDBOOK	270647
MEMORY COMPONENTS HANDBOOK	210830
MICROCOMMUNICATIONS HANDBOOK	231658
MICROCOMPUTER PROGRAMMABLE LOGIC HANDBOOK	296083
MICROPROCESSOR AND PERIPHERAL HANDBOOK (2 volume set)	230843
MILITARY PRODUCTS HANDBOOK (2 volume set. Not included in handbook set)	210461
OEM BOARDS AND SYSTEMS HANDBOOK	280407
PRODUCT GUIDE (Overview of Intel's complete product lines)	210846
SYSTEMS QUALITY/RELIABILITY HANDBOOK	231762
INTEL PACKAGING OUTLINES AND DIMENSIONS (Packaging types, number of leads, etc.)	231369
LITERATURE PRICE LIST (U.S. and Canada) (Comprehensive list of current Intel Literature)	210620
INTERNATIONAL LITERATURE GUIDE	E00029

CG/LIT/100188

About Our Cover:

As the inventor of the 8-bit embedded architecture that leads the market, Intel has provided the basis for over 50% of all 8-bit embedded controllers today. From simple consumer products to complex peripherals, our 8-bit controllers provide new venues for mold-breaking technological development.

intel

U.S. and CANADA LITERATURE ORDER FORM

NAME: _____

COMPANY: _____

ADDRESS: _____

CITY: _____ STATE: _____ ZIP: _____

COUNTRY: _____

PHONE NO.: ( ) _____

ORDER NO.	TITLE	QTY.	PRICE	TOTAL

Subtotal _____

Must Add Your Local Sales Tax _____

Postage: add 10% of subtotal → Postage _____

Total _____

Pay by check, money order, or include company purchase order with this form ($100 minimum).We also accept VISA, MasterCard or American Express. Make payment to Intel Literature Sales. Allow 2-4 weeks for delivery.

☐ VISA ☐ MasterCard ☐ American Express Expiration Date _____

Account No. _____

Signature _____

Mail To: Intel Literature Sales
P.O. Box 58130
Santa Clara, CA 95052-8130

International Customers outside the U.S. and Canada should use the International order form or contact their local Sales Office or Distributor.

For phone orders in the U.S. and Canada
Call Toll Free: (800) 548-4725

Prices good until 12/31/89.

Source HB

intel

INTERNATIONAL LITERATURE ORDER FORM

NAME: _____

COMPANY: _____

ADDRESS: _____

CITY: _____ STATE: _____ ZIP: _____

COUNTRY: _____

PHONE NO.: ( ) _____

ORDER NO.	TITLE	QTY.	PRICE	TOTAL

Subtotal _____

Must Add Your Local Sales Tax _____

Total _____

PAYMENT

Cheques should be made payable to your local Intel Sales Office (see inside back cover.)

Other forms of payment may be available in your country. Please contact the Literature Coordinator at your local Intel Sales Office for details.

The completed form should be marked to the attention of the LITERATURE COORDINATOR and returned to your local Intel Sales Office.

intel

Intel the Microcomputer Company:

When Intel invented the microprocessor in 1971, it created the era of microcomputers. Whether used as microcontrollers in automobiles or microwave ovens, or as personal computers or supercomputers, Intel's microcomputers have always offered leading-edge technology. In the second half of the 1980s, Intel architectures have held at least a 75% market share of microprocessors at 16 bits and above. Intel continues to strive for the highest standards in memory, microcomputer components, modules, and systems to give its customers the best possible competitive advantages.

8-BIT EMBEDDED CONTROLLER HANDBOOK

1989

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 58130
Santa Clara, CA 95052-8130

CUSTOMER SUPPORT

INTEL'S COMPLETE SUPPORT SOLUTION WORLDWIDE

Customer Support is Intel's complete support service that provides Intel customers with hardware support, software support, customer training, consulting services and network management services. For detailed information contact your local sales offices.

After a customer purchases any system hardware or software product, service and support become major factors in determining whether that product will continue to meet a customer's expectations. Such support requires an international support organization and a breadth of programs to meet a variety of customer needs. As you might expect, Intel's customer support is quite extensive. It can start with assistance during your development effort to network management. 100 Intel sales and service offices are located worldwide — in the U.S., Canada, Europe and the Far East. So wherever you're using Intel technology, our professional staff is within close reach.

HARDWARE SUPPORT SERVICES

Intel's hardware maintenance service, starting with complete on-site installation will boost your productivity from the start and keep you running at maximum efficiency. Support for system or board level products can be tailored to match your needs, from complete on-site repair and maintenance support economical carry-in or mail-in factory service.

Intel can provide support service for not only Intel systems and emulators, but also support for equipment in your development lab or provide service on your product to your end-user/customer.

SOFTWARE SUPPORT SERVICES

Software products are supported by our Technical Information Phone Service (TIPS) that has a special toll free number to provide you with direct, ready information on known, documented problems and deficiencies, as well as work-arounds, patches and other solutions.

Intel's software support consists of two levels of contracts. Standard support includes TIPS (Technical Information Phone Service), updates and subscription service (product-specific troubleshooting guides and; *COMMENTS Magazine*). Basic support consists of updates and the subscription service. Contracts are sold in environments which represent product groupings (e.g., iRMX® environment).

CONSULTING SERVICES

Intel provides field system engineering consulting services for any phase of your development or application effort. You can use our system engineers in a variety of ways ranging from assistance in using a new product, developing an application, personalizing training and customizing an Intel product to providing technical and management consulting. Systems Engineers are well versed in technical areas such as microcommunications, real-time applications, embedded microcontrollers, and network services. You know your application needs; we know our products. Working together we can help you get a successful product to market in the least possible time.

CUSTOMER TRAINING

Intel offers a wide range of instructional programs covering various aspects of system design and implementation. In just three to ten days a limited number of individuals learn more in a single workshop than in weeks of self-study. For optimum convenience, workshops are scheduled regularly at Training Centers worldwide or we can take our workshops to you for on-site instruction. Covering a wide variety of topics, Intel's major course categories include: architecture and assembly language, programming and operating systems, BITBUS™ and LAN applications.

NETWORK MANAGEMENT SERVICES

Today's networking products are powerful and extremely flexible. The return they can provide on your investment via increased productivity and reduced costs can be very substantial.

Intel offers complete network support, from definition of your network's physical and functional design, to implementation, installation and maintenance. Whether installing your first network or adding to an existing one, Intel's Networking Specialists can optimize network performance for you.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

Alphanumeric Index

Alphanumeric Index (Continued)

Any of the following products may appear in this publication. If so, it must be noted that such products have counterparts manufactured by Intel Puerto Rico, Inc., Intel Puerto Rico II, Inc., and/or Intel Singapore, Ltd. The product codes/part numbers of these counterpart products are listed below next to the corresponding Intel Corporation product codes/part numbers.

Intel Corporation Product Codes/ Part Numbers	Intel Puerto Rico, Inc. Intel Puerto Rico II, Inc. Product Codes/ Part Numbers	Intel Singapore, Ltd. Product Codes/ Part Numbers
376SKIT	p376SKIT	
903	p903	
904	p904	
913	p913	
914	p914	
923	p923	
924	p924	
952	p952	
953	p953	
954	p954	
ADAICE	pADAICE	
B386M1	pB386M1	
B386M2	pB386M2	
B386M4	pB386M4	
B386M8	pB386M8	
C044KIT	pC044KIT	
C252KIT	pC252KIT	
C28	pC28	
C32	pC32	
C452KIT	pC452KIT	
D86ASM	pD86ASM	
D86C86	pD86C86	
D86EDI	pD86EDI	
DCM9111	pDCM9111	
DOSNET	pDOSNET	
F1	pF1	
GUPILOGICIID	pGUPILOGICIID	
H4	pH4	
I044	pI044	
I252KIT	pI252KIT	
I452KIT	pI452KIT	
I86ASM	pI86ASM	
ICE386	pICE386	
III010	pIII010	
III086	pIII086	
III086	TIII086	
III111	pIII111	
III186	pIII186	
III186	TIII186	
III198	pIII198	
III212	pIII212	
III286	pIII286	
III286	TIII286	
III515	pIII515	
III520	TIII520	
III520	pIII520	
III531	pIII531	
III532	pIII532	
III533	pIII533	
III621	pIII621	
III707	pIII707	
III707	TIII707	
III815	pIII815	
INA961	pINA961	
IPAT86	pIPAT86	
KAS	pKAS	
KC	pKC	
KH	pKH	
KM1	pKM1	
KM2	pKM2	
KM4	pKM4	
KM8	pKM8	
KNLAN	pKNLAN	
KT60	pKT60	
KW140	pKW140	
KW40	pKW40	
KW80	pKW80	
M1	pM1	
M2	pM2	
M4	pM4	
M8	pM8	
MDS610	pMDS610	
MDX3015	pMDX3015	
MDX3015	pMDX3015	
MDX3016	pMDX3016	
MDX3016	pMDX3016	
MDX457	pMDX457	
MDX457	pMDX457	
MDX458	pMDX458	
MDX458	pMDX458	
MSA96	pMSA96	
NLAN	pNLAN	
PCLINK		sPCLINK
PCX344A	pPCX344A	
R286ASM	pR286ASM	
R286EDI	pR286EDI	
R286PLM	pR286PLM	
R286SSC	pR286SSC	
R86FOR	pR86FOR	
RCB4410		sRCB4410
RCX920	pRCX920	
RMX286	pRMX286	
RMXNET	pRMXNET	
S301	pS301	
S386	pS386	
SBC010	pSBC010	
SBC012	pSBC012	sSBC012
SBC020	pSBC020	
SBC028	pSBC028	
SBC040	pSBC040	
SBC056	pSBC056	
SBC108	pSBC108	
SBC116	pSBC116	
SBC18603	pSBC18603	sSBC18603
SBC186410	pSBC186410	
SBC18651	pSBC18651	sSBC18651
SBC186530	pSBC186530	
SBC18678	pSBC18678	
SBC18848	pSBC18848	sSBC18848
SBC18856	pSBC18856	sSBC18856
SBC208	pSBC208	sSBC208
SBC214	pSBC214	
SBC215	pSBC215	
SBC220	pSBC220	sSBC220
SBC221	pSBC221	
SBC28610	pSBC28610	sSBC28610
SBC28612	pSBC28612	
SBC28614	pSBC28614	

Intel Corporation Product Codes/ Part Numbers	Intel Puerto Rico, Inc. Intel Puerto Rico II, Inc. Product Codes/ Part Numbers	Intel Singapore, Ltd. Product Codes/ Part Numbers
SBC28616	pSBC28616	
SBC300	pSBC300	
SBC301	pSBC301	
SBC302	pSBC302	
SBC304	pSBC304	
SBC307	pSBC307	
SBC314	pSBC314	
SBC322	pSBC322	
SBC324	pSBC324	
SBC337	pSBC337	
SBC341	pSBC341	
SBC386	pSBC386	sSBC386
SBC386116	pSBC386116	
SBC386120	pSBC386120	
SBC38621	pSBC38621	
SBC38622	pSBC38622	
SBC38624	pSBC38624	
SBC38628	pSBC38628	
SBC38631	pSBC38631	
SBC38632	pSBC38632	
SBC38634	pSBC38634	
SBC38638	pSBC38638	
SBC428	pSBC428	sSBC428
SBC464	pSBC464	
SBC517	pSBC517	
SBC519	pSBC519	sSBC519
SBC534	pSBC534	sSBC534
SBC548	pSBC548	
SBC550	TSBC550	
SBC550	pSBC550	
SBC550	pSBC550	
SBC552	pSBC552	
SBC556	pSBC556	sSBC556
SBC569	pSBC569	
SBC589	pSBC589	
SBC604	pSBC604	
SBC608	pSBC608	
SBC614	pSBC614	
SBC618	pSBC618	
SBC655	pSBC655	
SBC6611	pSBC6611	
SBC8010	pSBC8010	
SBC80204	pSBC80204	
SBC8024	pSBC8024	sSBC8024
SBC8030	pSBC8030	
SBC8605	pSBC8605	sSBC8605
SBC8612	pSBC8612	
SBC8614	pSBC8614	
SBC8630	pSBC8630	sSBC8630
SBC8635	pSBC8635	sSBC8635
SBC86C38		sSBC86C38
SBC8825	pSBC8825	sSBC8825
SBC8840	pSBC8840	
SBC8845	pSBC8845	sSBC8845
SBC905	pSBC905	
SBCLNK001	pSBCLNK001	

Intel Corporation Product Codes/ Part Numbers	Intel Puerto Rico, Inc. Intel Puerto Rico II, Inc. Product Codes/ Part Numbers	Intel Singapore, Ltd. Product Codes/ Part Numbers
SBCMEM310	pSBCMEM310	
SBCMEM312	pSBCMEM312	
SBCMEM320	pSBCMEM320	
SBCMEM340	pSBCMEM340	
SBE96	pSBE96	
SBX217	pSBX217	
SBX218	pSBX218	
SBX270	pSBX270	
SBX311	pSBX311	
SBX328	pSBX328	
SBX331	pSBX331	
SBX344	pSBX344	
SBX350	pSBX350	
SBX351	pSBX351	
SBX354	pSBX354	
SBX488	pSBX488	
SBX586		sSBX586
SCHEMAIIPLD	pSCHEMAIIPLD	
SCOM	pSCOM	
SDK51	pSDK51	
SDK85	pSDK85	
SDK86	pSDK86	
SXM217	pSXM217	
SXM28612	pSXM28612	
SXM386	pSXM386	
SXM544	pSXM544	
SXM552	pSXM552	
SXM951	pSXM951	
SXM955	pSXM955	
SYP120	pSYP120	
SYP301	pSYP301	
SYP302	pSYP302	
SYP31090	pSYP31090	
SYP311	pSYP311	
SYP3847	pSYP3847	
SYR286	pSYR286	
SYR86	pSYR86	
SYS120	pSYS120	
SYS310	pSYS310	
SYS311	pSYS311	
T60	pT60	
TA096	pTA096	
TA252	pTA252	
TA452	pTA452	
W140	pW140	
W280	pW280	
W40	pW40	
W80	pW80	
XNX286DOC	pXNX286DOC	
XNX286DOCB	pXNX286DOCB	
XNXIBASE	pXNXIBASE	
XNXIDB	pXNXIDB	
XNXIDESK	pXNXIDESK	
XNXIPLAN	pXNXIPLAN	
XNXIWORD	pXNXIWORD	

MCS®-48 Single Component System

1

THE SINGLE COMPONENT MCS®-48 SYSTEM

1.0 INTRODUCTION

Sections 2 through 5 describe in detail the functional characteristics of the 8748H and 8749H EPROM, 8048AH/8049AH/8050AH ROM, and 8035AHL/8039AHL/8040-AHL CPU only single component microcomputers. Unless otherwise noted, details within these sections apply to all versions. This chapter is limited to those functions useful in single-chip implementations of the MCS®-48. The Chapter on the Expanded MCS®-48 System discusses functions which allow expansion of program memory, data memory, and input output capability.

2.0 ARCHITECTURE

The following sections break the MCS-48 Family into functional blocks and describe each in detail. The following description will use the 8048AH as the representative product for the family. See Figure 1.

2.1 Arithmetic Section

The arithmetic section of the processor contains the basic data manipulation functions of the 8048AH and can be divided into the following blocks:

- Arithmetic Logic Unit (ALU)
- Accumulator
- Carry Flag
- Instruction Decoder

In a typical operation data stored in the accumulator is combined in the ALU with data from another source on the internal bus (such as a register or I/o port) and the result is stored in the accumulator or another register.

The following is more detailed description of the function of each block.

INSTRUCTION DECODER

The operation code (op code) portion of each program instruction is stored in the Instruction Decoder and converted to outputs which control the function of each of the blocks of the Arithmetic Section. These lines control the source of data and the destination register as well as the function performed in the ALU.

ARITHMETIC LOGIC UNIT

The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under control of the Instruction Decoder. The ALU can perform the following functions:

- Add With or Without Carry
- AND, OR, Exclusive OR
- Increment/Decrement
- Bit Complement
- Rotate Left, Right
- Swap Nibbles
- BCD Decimal Adjust

If the operation performed by the ALU results in a value represented by more than 8 bits (overflow of most significant bit), a Carry Flag is set in the Program Status Word.

ACCUMULATOR

The accumulator is the single most important data register in the processor, being one of the sources of input to the ALU and often the destination of the result of operations performed in the ALU. Data to and from I/O ports and memory also normally passes through the accumulator.

2.2 Program Memory

Resident program memory consists of 1024, 2048, or 4096 words eight bits wide which are addressed by the program counter. In the 8748H and the 8749H this memory is user programmable and erasable EPROM; in the 8048AH/8049AH/8050AH the memory is ROM which is mask programmable at the factory. The 8035AHL/8039AHL/8040AHL has no internal program memory and is used with external memory devices. Program code is completely interchangeable among the various versions. To access the upper 2K of program memory in the 8050AH, and other MCS-48 devices, a select memory bank and a JUMP or CALL instruction must be executed to cross the 2K boundary.

There are three locations in Program Memory of special importance as shown in Figure 2.

LOCATION 0
Activating the Reset line of the processor causes the first instruction to be fetched from location 0.

LOCATION 3
Activating the Interrupt input line of the processor (if interrupt is enabled) causes a jump to subroutine at location 3.

LOCATION 7
A timer/counter interrupt resulting from timer counter overflow (if enabled) causes a jump to subroutine at location 7.

Therefore, the first instruction to be executed after initialization is stored in location 0, the first word of an external interrupt service subroutine is stored in location 3, and the first word of a timer/counter service routines

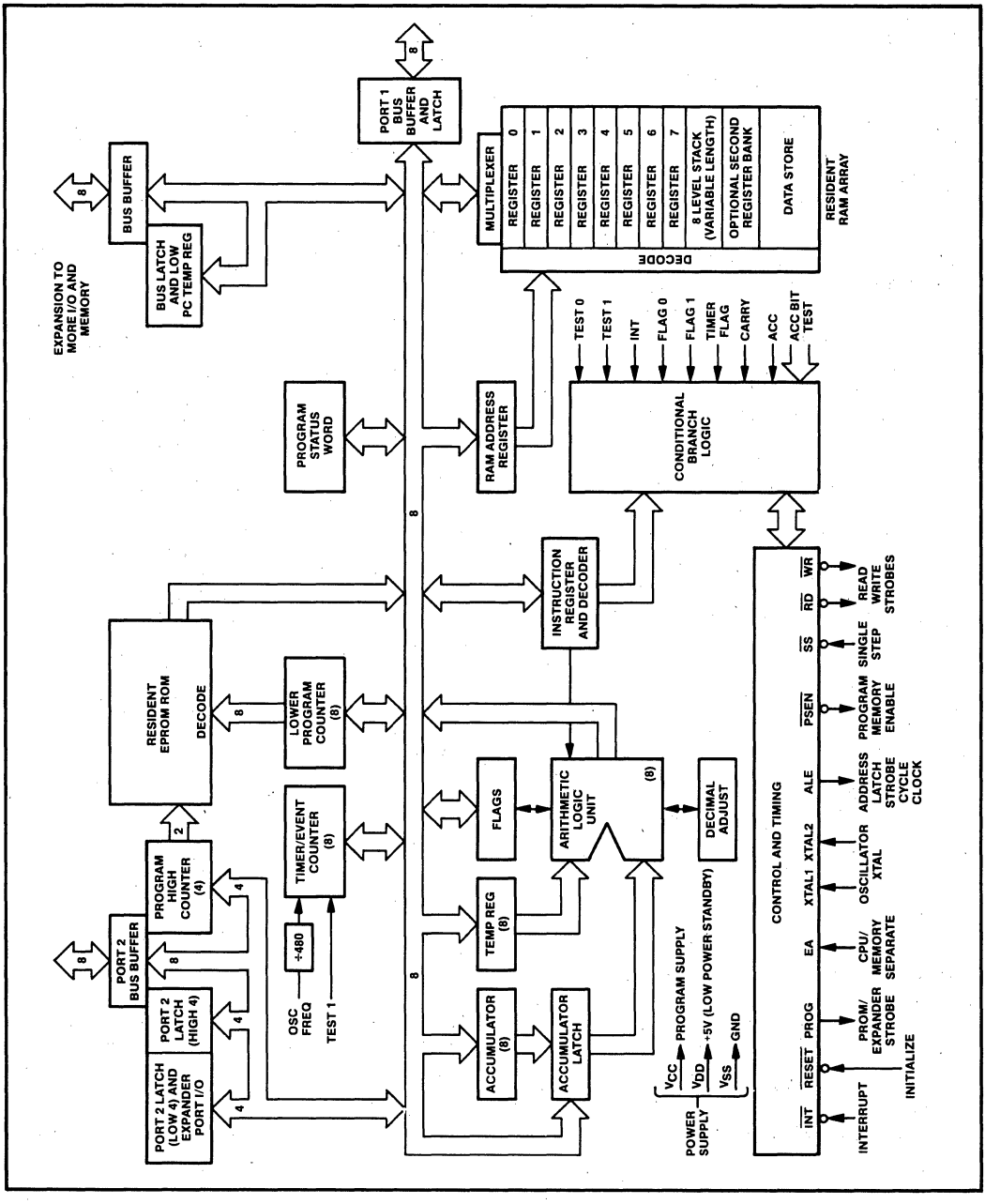


Figure 1. 8748H/8048H/8749AH/8050AH Block Diagram

is stored in location 7. Program memory can be used to store constants as well as program instructions. Instructions such as MOVP and MOVP3 allow easy access to data "lookup" tables.

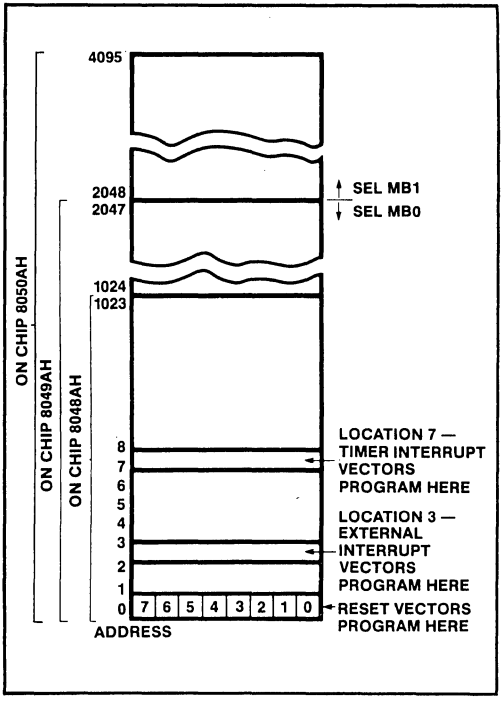


Figure 2. Program Memory Map

2.3 Data Memory

Resident data memory is organized as 64, 128, or 256 by 8-bits wide in the 8048AH, 8049AH and 8050AH. All locations are indirectly addressable through either of two RAM Pointer Registers which reside at address 0 and 1 of the register array. In addition, as shown in Figure 3, the first 8 locations (0–7) of the array are designated as working registers and are directly addressable by several instructions. Since these registers are more easily addressed, they are usually used to store frequently accessed intermediate results. The DJNZ instruction makes very efficient use of the working registers as program loop counters by allowing the programmer to decrement and test the register in a single instruction.

By executing a Register Bank Switch instruction (SEL RB) RAM locations 24–31 are designated as the working registers in place of locations 0–7 and are then directly addressable. This second bank of working registers may be used as an extension of the first bank or reserved for use during interrupt service subroutines allowing the registers of Bank 0 used in the main program to be instantly "saved" by a Bank Switch. Note that if this second bank is not used, locations 24–31 are still addressable as general purpose RAM. Since the two RAM pointer Registers R0 and R1 are a part of the working register array, bank switching effectively creates two more pointer registers (R0/and R1/) which can be used with R0 and R1 to easily access up to four separate working areas in RAM at one time. RAM locations (8–23) also serve a dual role in that they contain the program counter stack as explained in Section 2.6. These locations are addressed by the Stack Pointer during subroutine calls as well as by RAM Pointer Registers R0 and R1. If the level of subroutine nesting is less than 8, all stack registers are not required and can be used as general purpose RAM locations. Each level of subroutine nesting not used provides the user with two additional RAM locations.

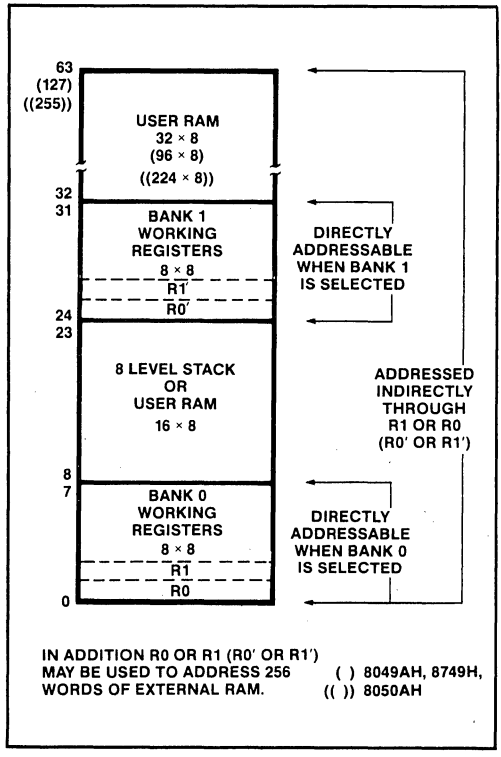
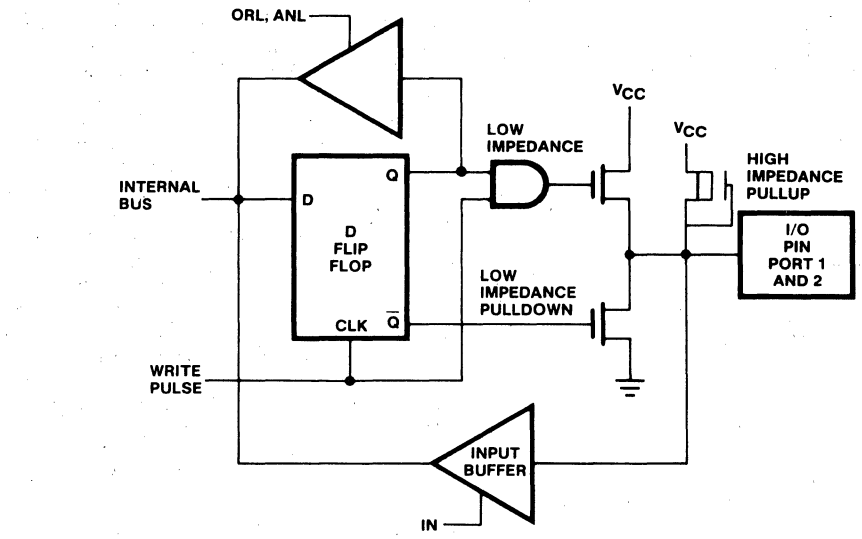
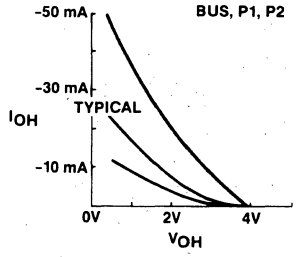
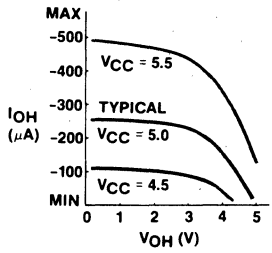


Figure 3. Data Memory Map

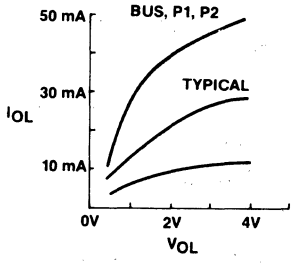




LOW IMPEDANCE PULLUP



HIGH IMPEDANCE PULLUP



LOW IMPEDANCE PULLDOWN

These graphs are for informational purposes only and are not guaranteed minimums or maximums.

Figure 4. "Quasi-bidirectional" Port Structure

2.4 Input/Output

The 8048AH has 27 lines which can be used for input or output functions. These lines are grouped as 3 ports of 8 lines each which serve as either inputs, outputs or bidirectional ports and 3 "test" inputs which can alter program sequences when tested by conditional jump instructions.

PORTS 1 AND 2

Ports 1 and 2 are each 8 bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, i.e., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, and output, or both even though outputs are statically latched. Figure 4 shows the circuit configuration in detail. Each line is continuously pulled up to V_{CC} through a resistive device of relatively high impedance.

This pullup is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. To provide fast switching times in a "0" to "1" transition a relatively low impedance device is switched in momentarily ($\approx$ 1/5 of a machine cycle) whenever a "1" is written to the line. When a "0" is written to the line a low impedance device overcomes the light pullup and provides TTL current sinking capability. Since the pulldown transistor is a low impedance device a "1" must first be written to any line which is to be used as an input. Reset initializes all lines to the high impedance "1" state.

It is important to note that the ORL and the ANL are read/write operations. When executed, the μC "reads" the port, modifies the data according to the instruction, then "writes" the data back to the port. The "writing" (essentially an OUTL instruction) enables the low impedance pull-up momentarily again even if the data was unchanged from a "1." This specifically applies to configurations that have inputs and outputs mixed together on the same port. See also section 8 in the Expanded MCS-48 System chapter.

BUS

Bus is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, Bus can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed however.

As a static port, data is written and latched using the OUTL instruction and inputted using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{WR}$ output line and output data is valid at the trailing edge of $\overline{WR}$. A read of the port generates a pulse on the $\overline{RD}$ output line and input data must be valid at the trailing edge of $\overline{RD}$. When not being written or read, the BUS lines are in a high impedance state. See also sections 7 and 8 in the Expanded MCS-48 System chapter.

2.5 Test and INT Inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These are T0, T1, and $\overline{INT}$. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1, and $\overline{INT}$ pins have other possible functions as well. See the pin description in Section 3.

2.6 Program Counter and Stack

The Program Counter is an independent counter while the Program Counter Stack is implemented suing pairs of registers in the Data Memory Array. Only 10, 11, or 12 bits of the Program Counter are used to address the 1024, 2048, or 4096 words of on-board program memory of the 8048AH, 8049AH, or 8050AH, while the most significant bits can be used for external Program Memory fetches. See Figure 5. The Program Counter is initialized to zero by activating the Reset line.

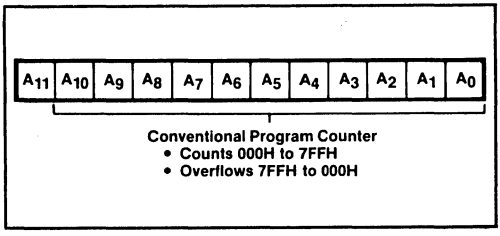


Figure 5. Program Counter

An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the Program Counter Stack as shown in Figure 6. The pair to be used is determined by a 3-bit Stack Pointer which is part of the Program Status Word (PSW).

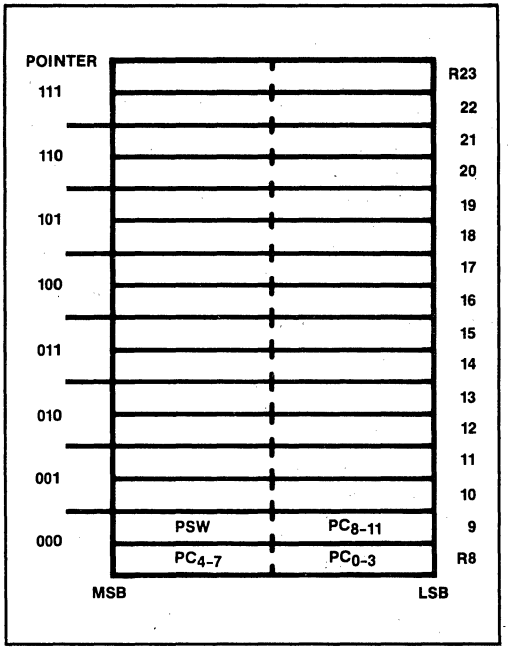


Figure 6. Program Counter Stack

Data RAM locations 8-23 are available as stack registers and are used to store the Program Counter and 4 bits of PSW as shown in Figure 6. The Stack Pointer .when initialized to 000 points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines wihtin subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the Stack Pointer to be decremented and the contents of the resulting register pair to be transferred to the Program Counter.

2.7 Program Status Word

An 8-bit status word which can be loaded to and from the accumulator exists called the Program Status Word (PSW). Figure 7 shows the information available in the word. The Program Status Word is actually a collection of flip-flops throughout the machine which can be read or written as a whole. The ability to write to PSW allows for easy restoration of machine status after a power down sequence.

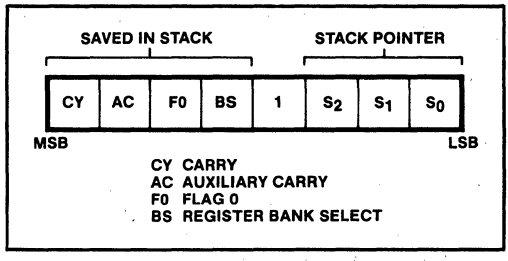


Figure 7. Program Status Word (PSW)

The upper four bits of PSW are stored in the Program Counter Stack with every call to subroutine or interrupt vector and are optionally restored upon return with the RETR instruction. The RET return instruction does not update PSW.

The PSW bit definitions are as follows:

Bits 0–2: Stack Pointer bits (S_0, S_1, S_2)

Bit 3: Not used (''1'' level when read)

Bit 4: Working Register Bank Switch Bit (BS)
0 = Bank 0
1 = Bank 1

Bit 5: Flag 0 bit (F0) user controlled flag which can be complemented or cleared, and tested with the conditional jump instruction JF0.

Bit 6: Auxiliary Carry (AC) carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A.

Bit 7: Carry (CY) carry flag which indicates that the previous operation has resulted in overflow of the accumulator.

2.8 Conditional Branch Logic

The conditional branch logic within the processsor enables several conditions internal and external to the processor to be tested by the users program. By using the conditional jump instruction the conditions that are listed in Table 1 can effect a change in the sequence of the program execution.

Table 1

Device Testable	Jump Conditions (Jump On)	
Accumulator	All zeros	not all zeros
Accumulator Bit	—	1
Carry Flag	0	1
User Flags (F0, F1)	—	1
Timer Overflow Flag	—	1
Test Inputs (T0, T1)	0	1
Interrupt Input ($\overline{INT}$)	0	—

2.9 Interrupt

An interrupt sequence is initiated by applying a low "0" level input to the INT pin. Interrupt is level triggered and active low to allow "WIRE ORing" of several interrupt sources at the input pin. Figure 8 shows the interrupt logic of the 8048AH. The Interrupt line is sampled every instruction cycle and when detected causes a "call to subroutine" at location 3 in program memory as soon as all cycles of the current instruction are complete. On 2-cycle instructions the interrupt line is sampled on the 2nd cycle only. $\overline{INT}$ must be held low for at least 3 machine cycles to ensure proper interrupt operations. As in any CALL to subroutine, the Program Counter and Program Status word are saved in the stack. For a description of this operation see the previous section, Program Counter and Stack. Program Memory location 3 usually contains an unconditional jump to an interrupt service subroutine elsewhere in program memory. The end of an interrupt service subroutine is signalled by the execution of a Return and Restore Status instruction RETR. The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR reenables the interrupt input logic. This occurs at the beginning of the second cycle of the RETR instruction. This sequence holds true also for an internal interrupt generated by timer overflow. If an internal timer/counter generated interrupt and an external interrupt are detected at the same time, the external source will be recognized. See the following Timer/Counter section for a description of timer interrupt. If needed, a second external interrupt can be created by enabling the timer/counter interrupt, loading FFH in the Counter (ones less than terminal count), and enabling the event counter mode. A "1" to "0" transition on the T1 input will then cause an interrupt vector to location 7.

INTERRUPT TIMING

The interrupt input may be enabled or disabled under Program Control using the EN I and DIS I instructions. Interrupts are disabled by Reset and remain so until enabled by the users program. An interrupt request must be removed before the RETR instruction is executed upon return from the service routine otherwise the processor will re-enter the service routine immediately. Many peripheral devices prevent this situation by resetting their interrupt request line whenever the processor accesses (Reads or Writes) the peripherals data buffer register. If the interrupting device does not require access by the processor, one output line of the 8048AH may be designated as an "interrupt acknowledge" which is activated by the service subroutine to reset the interrupt request. The $\overline{INT}$ pin may also be tested using the conditional jump instruction JNI. This instruction may be used to detect the presence of a pending interrupt before interrupts are enabled. If interrupt is left disabled, $\overline{INT}$ may be used as another test input like T0 and T1.

2.10 Timer/Counter

The 8048AH contains a counter to aid the user in counting external events and generating accurate time delays without placing a burden on the processor for these functions. In both modes the counter operation is the same, the only difference being the source of the input to the counter. The timer/event counter is shown in Figure 9.

COUNTER

The 8-bit binary counter is presettable and readable with two MOV instructions which transfer the contents of the accumulator to the counter and vice versa. The counter content may be affected by Reset and should be initialized by software. The counter is stopped by a Reset or STOP TCNT instruction and remains stopped until started as a timer by a START T instruction or as an event counter by a START CNT instruction. Once started the counter will increment to this maximum count (FF) and overflow to zero continuing its count until stopped by a STOP TCNT instruction or Reset.

The increment from maximum count to zero (overflow) results in the setting of an overflow flag flip-flop and in the generation of an interrupt request. The state of the overflow flag is testable with the conditional jump instruction JTF. The flag is reset by executing a JTF or by Reset. The interrupt request is stored in a latch and then ORed with the external interrupt input INT. The timer interrupt may be enabled or disabled independently of external interrupt by the EN TCNTI and DIS TCNTI instructions. If enabled, the counter overflow will cause a subroutine call to location 7 where the timer or counter service routine may be stored.

If timer and external interrupts occur simultaneously, the external source will be recognized and the Call will be to

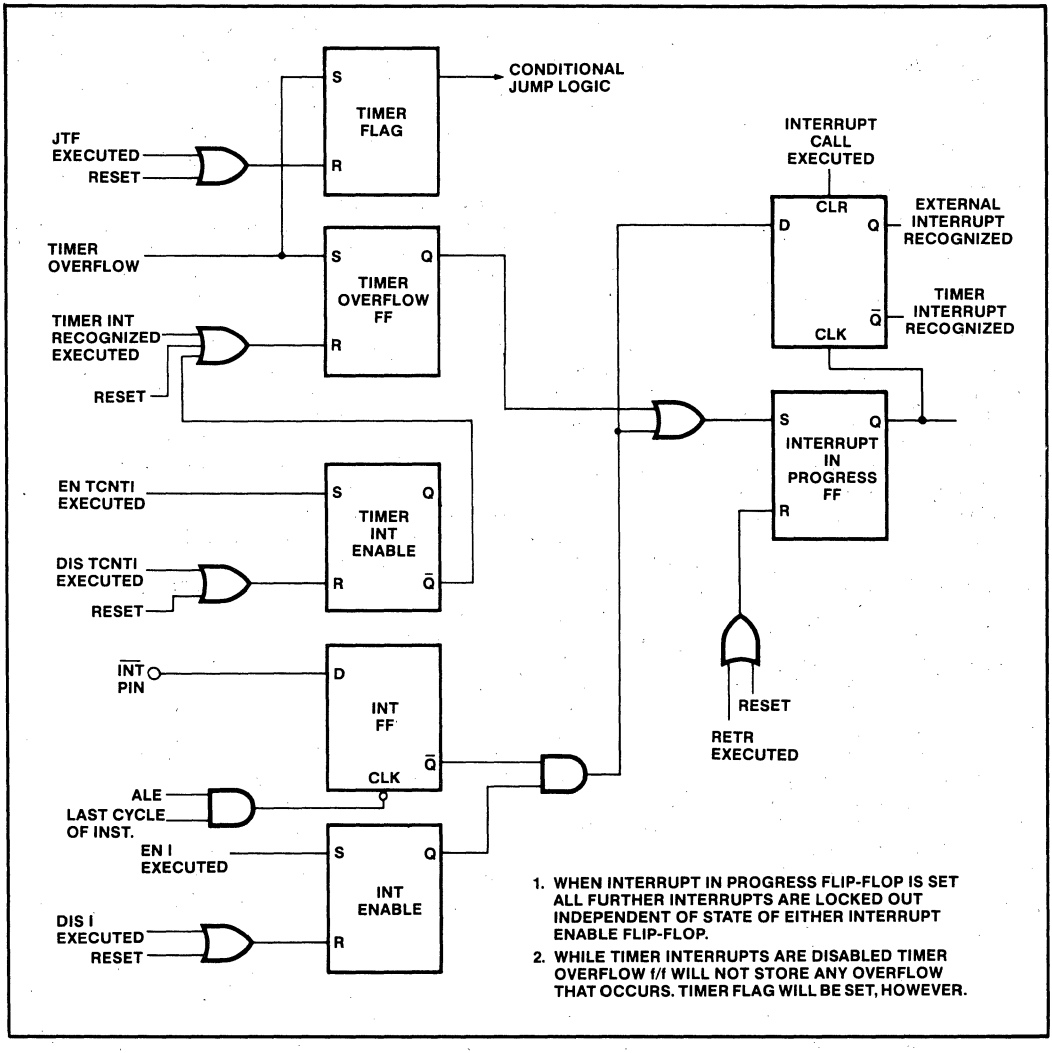


Figure 8. Interrupt Logic

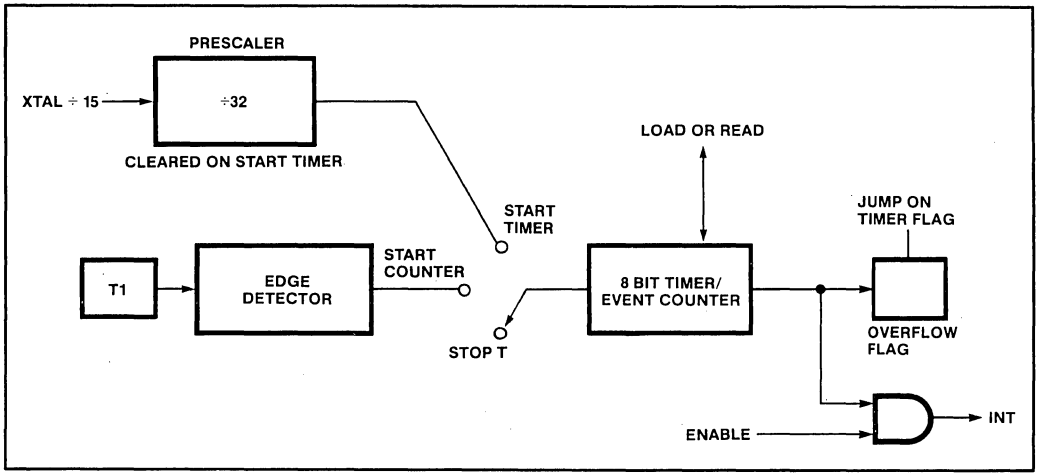


Figure 9. Timer/Event Counter

location 3. Since the timer interrupt is latched it will remain pending until the external device is serviced and immediately be recognized upon return from the service routine. The pending timer interrupt is reset by the Call to location 7 or may be removed by executing a DIS TCNT1 instruction.

AS AN EVENT COUNTER

Execution of a START CNT instruction connects the T1 input pin to the counter input and enables the counter. The T1 input is sampled at the beginning of state 3 or in later MCS-48 devices in state time 4. Subsequent high to low transitions on T1 will cause the counter to increment. T1 must be held low for at least 1 machine cycle to insure it won't be missed. The maximum rate at which the counter may be incremented is once per three instruction cycles (every 5.7 μsec when using an 8 MHz crystal) — there is no minimum frequency. T1 input must remain high for at least 1/5 machine cycle after each transition.

AS A TIMER

Eexcution of a START T instruction connects an internal clock to the counter input and enables the counter. The internal clock is derived bypassing the basic machine cycle clock through a ÷32 prescaler. The prescaler is reset during the START T instruction. The resulting clock increments the counter every 32 machine cycles. Various delays from 1 to 256 counts can be obtained by presetting the counter and detecting overflow. Times longer than 256 counts may be achieved by accumulating multiple overflows in a register under software control. For time resolution less than 1 count an external clock can be applied to the T1 input and the counter operated in the event counter mode. ALE divided by 3 or more can serve as this external clock. Very small delays or "fine tuning" of larger delays can be easily accomplished by software delay loops.

Often a serial link is desirable in an MCS-48 family member. Table 2 lists the timer counts and cycles needed for a specific baud rate given a crystal frequency.

2.11 Clock and Timing Circuits

Timing generation for the 8048AH is completely selfcontained with the exeception of a frequency reference which can be XTAL, ceramic resonator, or external clock source. The Clock and Timing circuitry can be divided into the following functional blocks.

OSCILLATOR

The on-board oscillator is a high gain parallel resonant circuit with a frequency range of 1 to 11 MHz. The X1 external pin is the input to the amplifier stage while X2 is the output. A crystal or ceramic resonator connected between X1 and X2 provides the feedback and phase shift required for oscillation. If an accurate frequency reference is not required, ceramic resonator may be used in place of the crystal.

For accurate clocking, a crystal should be used. An externally generated clock may also be applied to X1–X2 as the frequency source. See the data sheet for more information.

Table 2. Baud Rate Generation

	Frequency (MHz)	T_{cy}	T0 Prr(1/5 T_{cy})	Timer Prescaler (32 T_{cy})
	4	3.75μs	750ns	120μs
	6	2.50μs	500ns	80μs
	8	1.88μs	375ns	60.2μs
	11	1.36μs	275ns	43.5μs
Baud Rate	4 MHz Timer Counts + Instr. Cycles	6 MHz Timer Counts + Instr. Cycles	8 MHz Timer Counts + Instr. Cycles	11 MHz Timer Counts + Instr. Cycles
110	75 + 24 Cycles .01% Error	113 + 20 Cycles .01% Error	151 + 3 Cycles .01% Error	208 + 28 Cycles .01% Error
300	27 + 24 Cycles .1% Error	41 + 21 Cycles .03% Error	55 + 13 Cycles .01% Error	76 + 18 Cycles .04% Error
1200	6 + 30 Cycles .1% Error	10 + 13 Cycles .1% Error	12 + 27 Cycles .06% Error	19 + 4 Cycles .12% Error
1800	4 + 20 Cycles .1% Error	6 + 30 Cycles .1% Error	9 + 7 Cycles .17% Error	12 + 24 Cycles .12% Error
2400	3 + 15 Cycles .1% Error	5 + 6 Cycles .4% Error	6 + 24 Cycles .29% Error	9 + 18 Cycles .12% Error
4800	1 + 23 Cycles 1.0% Error	2 + 19 Cycles .4% Error	3 + 14 Cycles .74% Error	4 + 25 Cycles .12% Error

STATE COUNTER

The output of the oscillator is divided by 3 in the State Counter to create a clock which defines the state times of the machine (CLK). CLK can be made available on the external pin T0 by executing an ENTO CLK instruction. The output of CLK on T0 is disabled by Reset of the processor.

CYCLE COUNTER

CLK is then divided by 5 in the Cycle Counter to provide a clock which defines a machine cycle consisting of 5 machine states as shown in Figure 10. Figure 11 shows the different internal operations as divided into the machine states. This clock is called Address Latch Enable (ALE) because of its function in MCS-48 systems with external memory. It is provided continuously on the ALE output pin.

2.12 Reset

The reset input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pull-up device which in combination with an external 1 μ fd capacitor provides an internal reset pulse of sufficient length to guarantee all circuitry is reset, as shown in Figure 12. If the reset pulse is generated externally the RESET pin must be held low for at least 10 milliseconds after the power supply is within tolerance.Only 5 machine cycles (6.8 μs @ 11 MHz) are required if power is already on and the oscillator has stabilized. ALE and PSEN (if EA = 1) are active while in Reset.

Reset performs the following functions:

1) Sets program counter to zero.

2) Sets stack pointer to zero.

3) Selects register bank 0.

4) Selects memory bank 0.

5) Sets BUS to high impedance state (except when EA = 5V).

6) Sets Ports 1 and 2 to input mode.

7) Disables interrupts (timer and external).

8) Stops timer.

9) Clears timer flag.

10) Clears F0 and F1.

11) Disables clock output from T0.

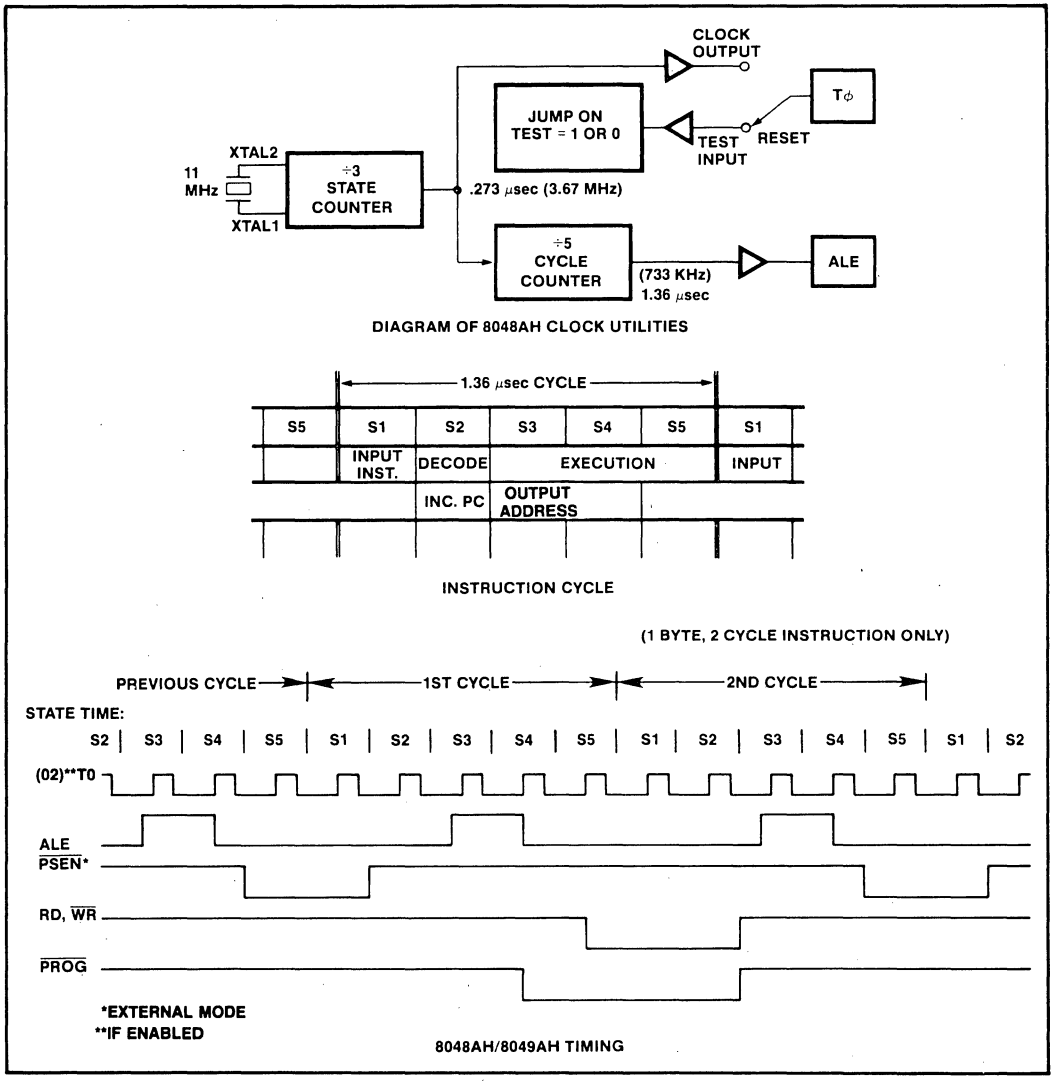


Figure 10. MCS®-48 Timing Generation and Cycle Timing

2.13 Single-Step

This feature, as pictured in Figure 13, provides the user with a debug capability in that the processor can be stepped through the program one instruction at a time. While stopped, the address of the next instruction to be fetched is available concurrently on BUS and the lower half of Port 2. The user can therefore follow the program through each of the instruction steps. A timing diagram, showing the interaction between output ALE and input $\overline{SS}$, is shown. The BUS buffer contents are lost during single step; however, a latch may be added to reestablish the lost I/O capability if needed. Data is valid at the leading edge of ALE.

	CYCLE 1					CYCLE 2				
INSTRUCTION	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	*INCREMENT TIMER	—	—	READ PORT	—	* —	—
OUTL P,A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	*INCREMENT TIMER	OUTPUT TO PORT	—	—	—	* —	—
ANL P, = DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	*INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	—	INCREMENT PROGRAM COUNTER	*OUTPUT TO PORT	—
ORL P, = DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	*INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	—	INCREMENT PROGRAM COUNTER	*OUTPUT TO PORT	—
INS A, BUS	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	—	—	READ PORT	—	* —	—
OUTL BUS, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	INCREMENT TIMER	OUTPUT TO PORT	—	—	—	* —	—
ANL BUS, = DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	*INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	—	INCREMENT PROGRAM COUNTER	*OUTPUT TO PORT	—
ORL BUS, = DATA	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	*INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	—	INCREMENT PROGRAM COUNTER	*OUTPUT TO PORT	—
MOVX @ R,A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	OUTPUT DATA TO RAM	—	—	—	* —	—
MOVX A,@R	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	—	—	READ DATA	—	* —	—
MOVD A,P_i	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	—	—	READ P2 LOWER	—	* —	—
MOVD P_i,A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA TO P2 LOWER	—	—	—	* —	—
ANLD P,A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA	—	—	—	* —	—
ORLD P,A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA	—	—	—	* —	—
J(CONDITIONAL)	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	SAMPLE CONDITION	*INCREMENT SAMPLE	—	FETCH IMMEDIATE DATA	—	UPDATE PROGRAM COUNTER	* —	—
STRT T STRT CNT	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	* —	START COUNTER					
STOP TCNT	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	* —	STOP COUNTER					
ENI	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	* ENABLE INTERRUPT	—					
DIS I	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	—	* DISABLE INTERRUPT	—					
ENTO CLK	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER		* ENABLE CLOCK	—					

***VALID INSTRUCTION ADDRESSES ARE OUTPUT AT THIS TIME IF EXTERNAL PROGRAM MEMORY IS BEING ACCESSED.**

(1) IN LATER MCS-48 DEVICES T1 IS SAMPLED IN S4.

Figure 11. 8048AH/8049AH Instruction Timing Diagram

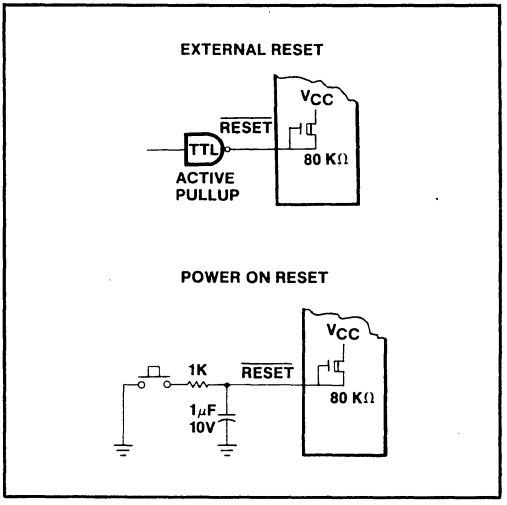


Figure 12.

TIMING

The 8048AH operates in a single-step mode as follows:

1) The processor is requested to stop by applying a low level on $\overline{SS}$.
2) The processor responds by stopping during the address fetch portion of the next instruction. If a double cycle instruction is in progress when the single step command is received, both cycles will be completed before stopping.
3) The processor acknowledges it has entered the stopped state by raising ALE high. In this state (which can be maintained indefinitely) the address of the next instruction to be fetched is present on BUS and the lower half of port 2.
4) $\overline{SS}$ is then raised high to bring the processor out of the stopped mode allowing it to fetch the next instruction. The exit from stop is indicated by the processor bringing ALE low.
5) To stop the processor at the next instruction $\overline{SS}$ must be brought low again soon after ALE goes low. If $\overline{SS}$ is left high the processor remains in a "Run" mode.

A diagram for implementing the single-step function of the 8748H is shown in Figure 13. D-type flip-flop with preset and clear is used to generate $\overline{SS}$. In the run mode SS is held high by keeping the flip-flop preset (preset has precedence over the clear input). To enter single step, preset is removed allowing ALE to bring $\overline{SS}$ low via the clear input. ALE should be buffered since the clear input of an SN7474 is the equivalent of 3 TTL loads. The processor is now in the stopped state. The next instruction is initiated by clocking a "1" into the flip-flop. This "1" will not appear on $\overline{SS}$ unless ALE is high removing clear from the flip-flop. In response to $\overline{SS}$ going high the processor begins an instruction fetch which brings ALE low resetting $\overline{SS}$ through the clear input and causing the processor to again enter the stopped state.

2.14 Power Down Mode (8048AH, 8049AH, 8050AH, 8039AHL, 8035AHL, 8040AHL)

Extra circuitry has been added to the 8048AH/8049AH/8050AH ROM version to allow power to be removed from all but the data RAM array for low power standby operation. In the power down mode the contents of data RAM can be maintained while drawing typically 10% to 15% of normal operating power requirements.

V_{CC} serves as the 5V supply pin for the bulk of circuitry while the V_{DD} pin supplies only the RAM array. In normal operation both pins are a 5V while in standby, V_{CC} is at ground and V_{DD} is maintained at its standby value. Applying Reset to the processor through the $\overline{RESET}$ pin inhibits any access to the RAM by the processor and guarantees that RAM cannot be inadvertently altered as power is removed from V_{CC}.

A typical power down sequence (Figure 14) occurs as follows:

1) Imminent power supply failure is detected by user defined circuitry. Signal must be early enough to allow 8048AH to save all necessary data before V_{CC} falls below normal operating limits.
2) Power fail signal is used to interrupt processor and vector it to a power fail service routine.
3) Power fail routine saves all important data and machine status in the internal data RAM array. Routine may also initiate transfer of backup supply to the V_{DD} pin and indicate to external circuitry that power fail routine is complete.
4) Reset is applied to guarantee data will not be altered as the power supply falls out of limits. Reset must be held low until V_{CC} is at ground level.

Recovery from the Power Down mode can occur as any other power-on sequence with an external capacitor on the Reset input providing the necessary delay. See the previous section on Reset.

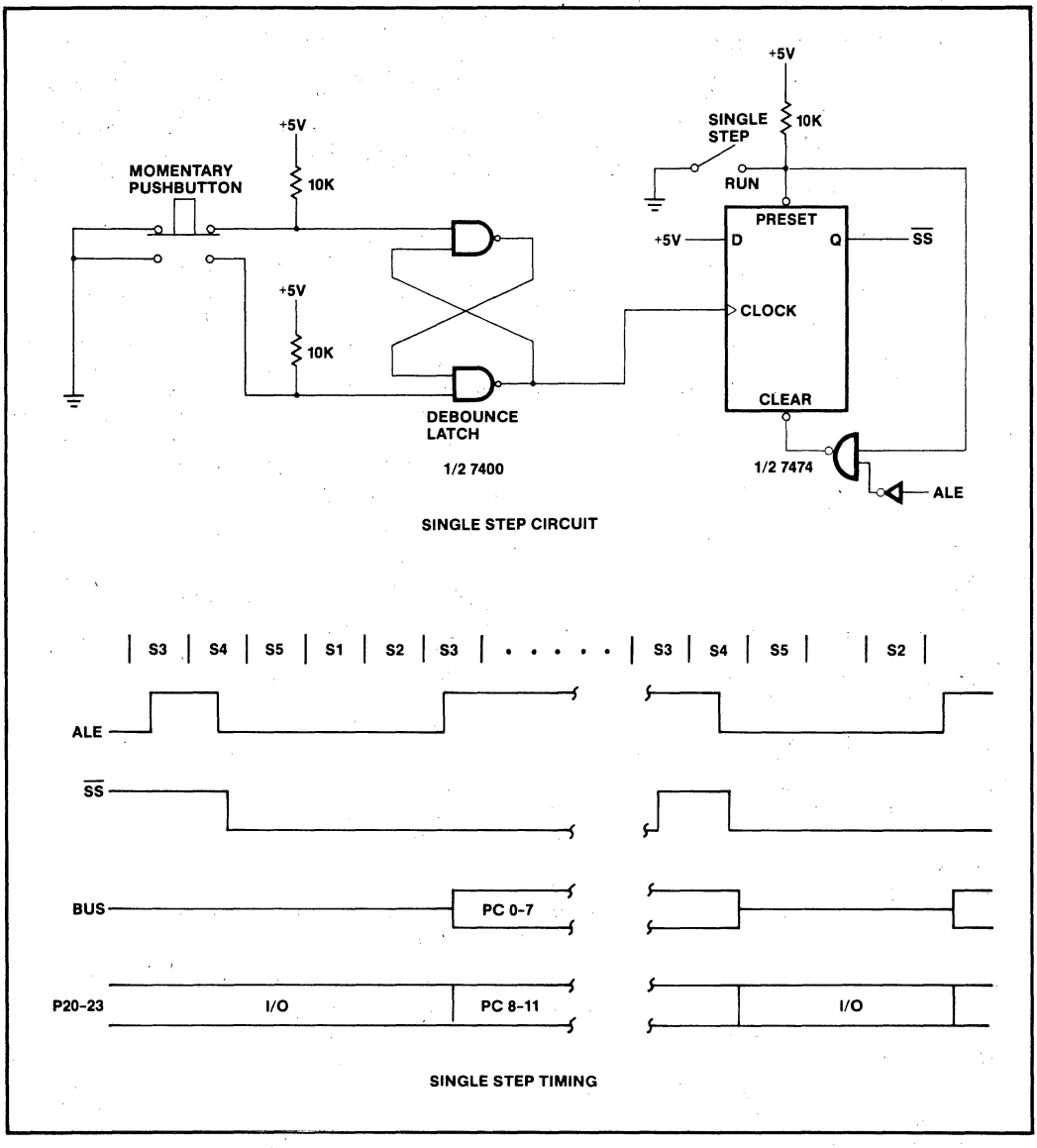


Figure 13. Single Step Operation

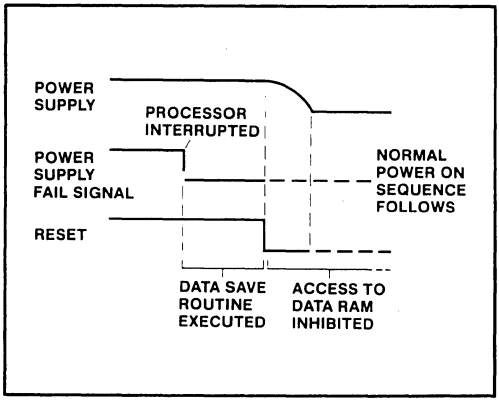


Figure 14. Power Down Sequence

2.15 External Access Mode

Normally the first 1K (8048AH), 2K (8049AH), or 4K (8050AH) words of program memory are automatically fetched from internal ROM or EPROM. The EA input pin however allows the user to effectively disable internal program memory by forcing all program memory fetches to reference external memory. The following chapter explains how access to external program memory is accomplished.

The External Access mode is very useful in system test and debug because it allows the user to disable his internal applications program and substitute an external program of his choice — a diagnostic routine for instance. In addition, the date sheet shows how internal program memory can be read externally, independent of the processor. A "1" level on EA initiates the external accesss mode. For proper operation, Reset should be applied while the EA input is changed.

2.16 Sync Mode

The 8048AH, 8049AH, 8050AH has incorporated a new SYNC mode. The Sync mode is provided to ease the design of multiple controller circuits by allowing the designer to force the device into known phase and state time. The SYNC mode may also be utilized by automatic test equipment (ATE) for quick, easy, and efficient synchronizing between the tester and the DUT (device under test).

SYNC mode is enabled when SS′ pin is raised to high voltage level of +12 volts. To begin synchronization, T0 is raised to 5 volts at least four clocks cycles after SS′. T0 must be high for at least four X1 clock cycles to fully reset the prescaler and time state generators. T0 may then be brought down with the rising edge of X1. Two clock cycles later, with the rising edge of X1, the device enters into Time State 1, Phase 1, SS′ is then brought down to 5 volts 4 clocks later after T0. RESET′ is allowed to go high 5 tCY (75 clocks) later for normal execution of code. See Figure 15.

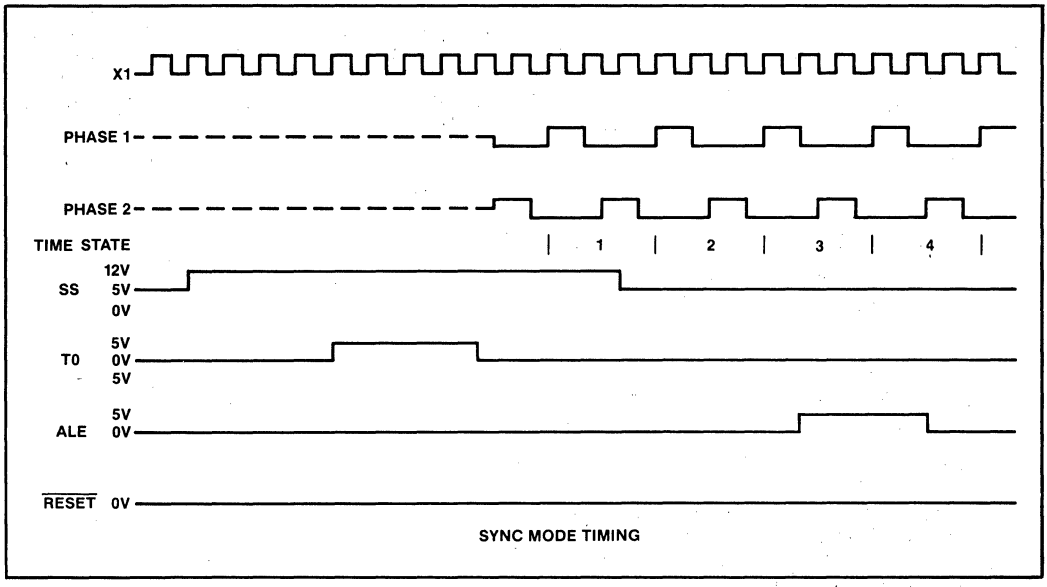


Figure 15. Sync Mode Timing

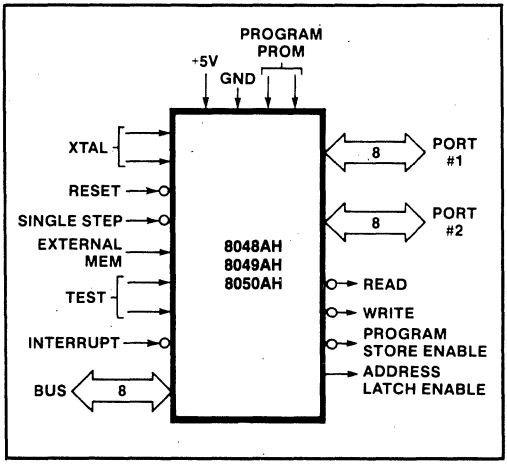


Figure 16. 8048AH and 8049AH Logic Symbol

3.0 PIN DESCRIPTION

The MCS-48 processors are packaged in 40 pin Dual In-Line Packages (DIP's). Table 3 is a summary of the functions of each pin. Figure 16 is the logic symbol for the 8048AH product family. Where it exists, the second paragraph describes each pin's function in an expanded MCS-48 system. Unless otherwise specified, each input is TTL compatible and each output will drive one standard TTL load.

Table 3. Pin Description

Designation	Pin Number*	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	Programming power supply; 21V during program for the 8748H/8749H; +5V during operation for both ROM and EPROM. Low power standby pin in 8048AH and 8049AH/8050AH ROM versions.
V_{CC}	40	Main power supply; +5V during operation and during 8748H and 8749H programming.
PROG	25	Program pulse; +18V input pin during 8748H/8749H programming. Output strobe for 8243 I/O expander.
P10-P17 (Port 1)	27-34	8-bit quasi-bidirectional port. (Internal Pullup ≈ 50KΩ)
P20-P27 (Port 2)	21-24 35-38	8-bit quasi-bidirectional port. (Internal Pullup ≈ 50KΩ) P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
D0-D7 (BUS)	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENTO CLK instruction. T0 is also used during programming and sync mode.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the event counter input using the STRT CNT instruction. (See Section 2.10).
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. (Active low) Interrupt must remain low for at least 3 machine cycles to ensure proper operation.
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the BUS from an external device. (Active low) Used as a Read Strobe to External Data Memory.
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during EPROM programming and verification. (Active low) (Internal pullup ≈ 80K Ω)
$\overline{WR}$	10	Output strobe during a BUS write. (Active low) Used as write strobe to external data memory.
ALE	11	Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.

Table 3. Pin Description (Continued)

Designation	Pin Number*	Function
$\overline{\text{PSEN}}$	9	Program Store Enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{\text{SS}}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low) (Internal pullup ≈ 300KΩ) +12V for sync modes (See 2.16).
EA	7	External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high) +12V for 8048AH/8049AH/8050AH program verification and +18V for 8748H/8749H program verification (Internal pullup ≈ 10MΩ on 8048AH/8049AH/8035AHL/8039AHL/8050AH/8040AHL)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source.
XTAL2	3	Other side of crystal/external source input.

*Unless otherwise stated, inputs do not have internal pullup resistors. 8048AH, 8748H, 8049AH, 8050AH, 8040AHL

4.0 PROGRAMMING, VERIFYING AND ERASING EPROM

The internal Program Memory of the 8748H and the 8749H may be erased and reprogrammed by the user as explained in the following sections. See also the 8748H and 8749H data sheets.

4.1 Programming/Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. This programming algorithm applies to both the 8748H and 8749H. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a descsription of their functions:

Pin	Function
XTAL 1	Clock Input (3 to 4 MHz)
Reset	Initialization and Address Latching
Test 0	Selection of Program (0V) or Verify (5V) Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P20-1	Address Input for 8748H
P20-2	Address Input for 8749H
V_{DD}	Programming Power Supply
PROG	Program Pulse Input
P10–P11	Tied to ground (8749H only)

8748H AND 8749H ERASURE CHARACTERISTICS

The erasure characteristics of the 8748H and 8749H are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (A). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000A range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8748H and 8749H in approximately 3 years while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 8748H or 8749H is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the 8748H window to prevent unintentional erasure.

When erased, bits of the 8748H and 8749H Program Memory are in the logic "0" state.

The recommended erasure procedure for the 8748H and 8749H is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (A). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000μW/cm^2 power rating. The 8748H and 8749H should be placed within one inch from the lamp tubes during erasure. Some lamps have a filter in their tubes and this filter should be removed before erasure.

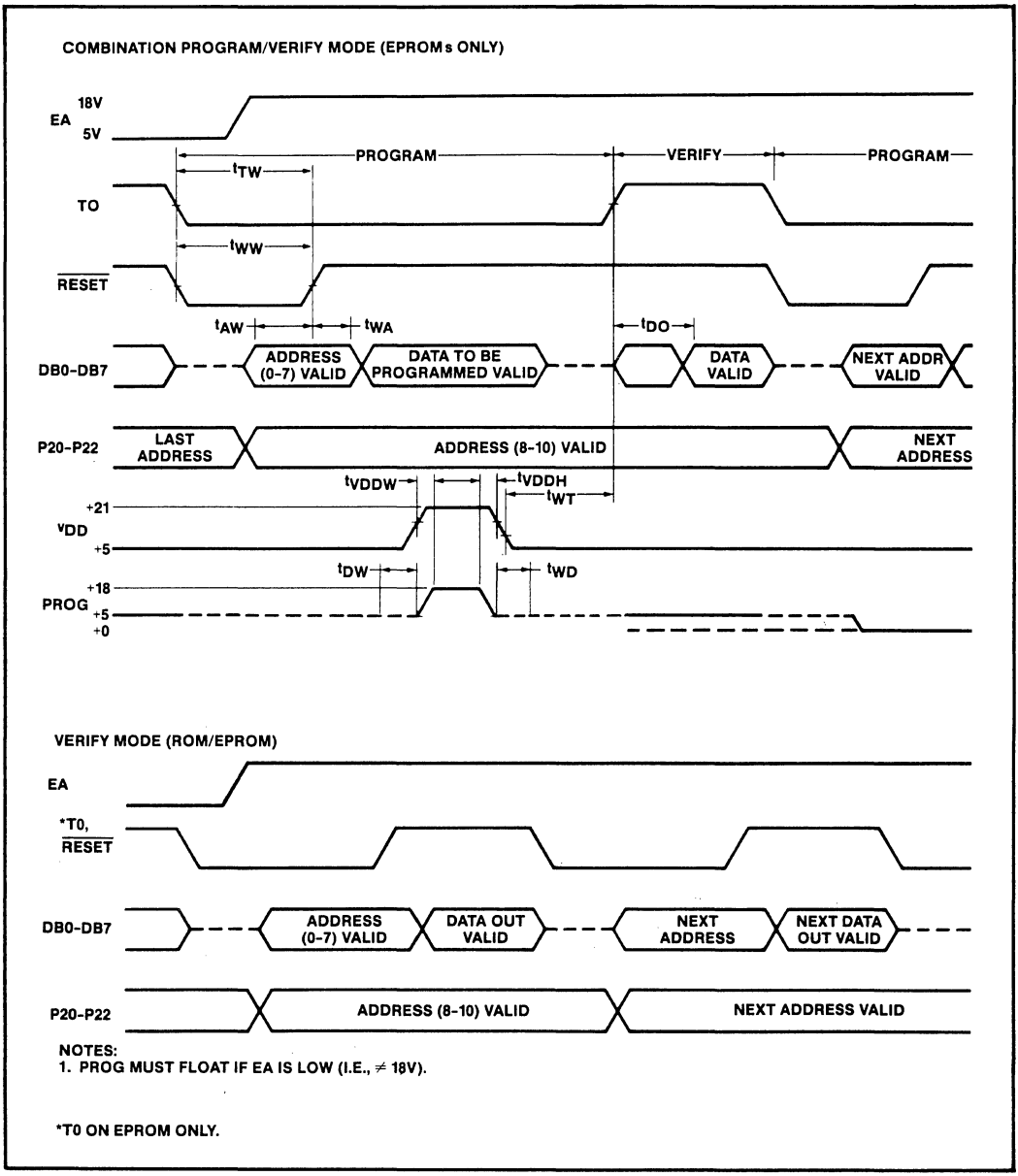


Figure 17. Program/Verify Sequence for 8749H/8748H

MCS®-48 Expanded System

2

EXPANDED MCS®-48 SYSTEM

1.0 INTRODUCTION

If the capabilities resident on the single-chip 8048AH/8748H/8035AHL/8049AH/8749H/8039AHL are not sufficient for your system requirements, special on-board circuitry allows the addition of a wide variety of external memory, I/O, or special peripherals you may require. The processors can be directly and simply expanded in the following areas:

- Program Memory to 4K words
- Data Memory to 320 words (384 words with 8049AH)
- I/O by unlimited amount
- Special Functions using 8080/8085AH peripherals

By using bank switching techniques, maximum capability is essentially unlimited. Bank switching is discussed later in the chapter. Expansion is accomplished in two ways:

1) Expander I/O — A special I/O Expander circuit, the 8243, provides for the addition of four 4-bit Input/Output ports with the sacrifice of only the lower half (4-bits) of port 2 for inter-device communication. Multiple 8243's may be added to this 4-bit bus by generating the required "chip select" lines.
2) Standard 8085 Bus — One port of the 8048AH/8049AH is like the 8-bit bidirectional data bus of the 8085 microcomputer system allowing interface to the numerous standard memories and peripherals of the MCS®-80/85 microcomputer family.

MCS-48 systems can be configured using either or both of these expansion features to optimize system capabilities to the application.

Both expander devices and standard memories and peripherals can be added in virtually any number and combination required.

2.0 EXPANSION OF PROGRAM MEMORY

Program Memory is expanded beyond the resident 1K or 2K words by using the 8085 BUS feature of the MCS®-48. All program memory fetches from the addresses less than 1024 on the 8048AH and less than 2048 on the 8049AH occur internally with no external signals being generated (except ALE which is always present). At address 1024 on the 8048AH, the processor automatically initiates external program memory fetches.

2.1 Instruction Fetch Cycle (External)

As shown in Figure 1, for all instruction fetches from addresses of 1024 (2048) or greater, the following will occur:

1) The contents of the 12-bit program counter will be output on BUS and the lower half of port 2.
2) Address Latch Enable (ALE) will indicate the time at which address is valid. The trailing edge of ALE is used to latch the address externally.
3) Program Store Enable ($\overline{PSEN}$) indicates that an external instruction fetch is in progress and serves to enable the external memory device.
4) BUS reverts to input (floating) mode and the processor accepts its 8-bit contents as an instruction word.

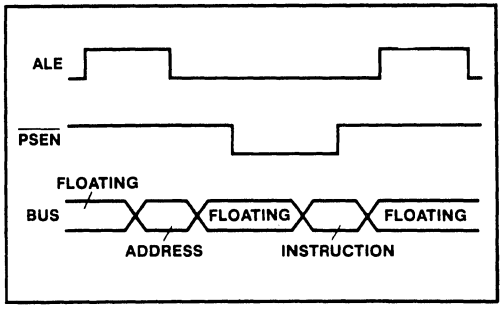


Figure 1. Instruction Fetch from External Program Memory

All instruction fetches, including internal addresses, can be forced to be external by activating the EA pin of the 8048AH/8049AH/8050AH. The 8035AHL/8039AHL/8040AHL processors without program memory always operate in the external program memory mode (EA = 5V).

2.2 Extended Program Memory Addressing (Beyond 2K)

For programs of 2K words or less, the 8048AH/8049AH addresses program memory in the conventional manner. Addresses beyond 2047 can be reached by executing a program memory bank switch instruction (SEL MB0, SEL MB1) followed by a branch instruction (JMP or CALL). The bank switch feature extends the range of branch instructions beyond their normal 2K range and at the same time prevents the user from inadvertently crossing the 2K boundary.

PROGRAM MEMORY BANK SWITCH

The switching of 2K program memory banks is accomplished by directly setting or resetting the most significant bit of the program counter (bit 11); see Figure 2. Bit 11 is not altered by normal incrementing of the program counter but is loaded with the contents of a special flip-flop each time a JMP or CALL instruction is executed. This special flip-flop is set by executing an SEL MB1

instruction and reset by SEL MB0. Therefore, the SEL MB instruction may be executed at any time prior to the actual bank switch which occurs during the next branch instruction encountered. Since all twelve bits of the program counter, including bit 11, are stored in the stack, when a Call is executed, the user may jump to subroutines across the 2K boundary and the proper bank will be restored upon return. However, the bank switch flip-flop will not be altered on return.

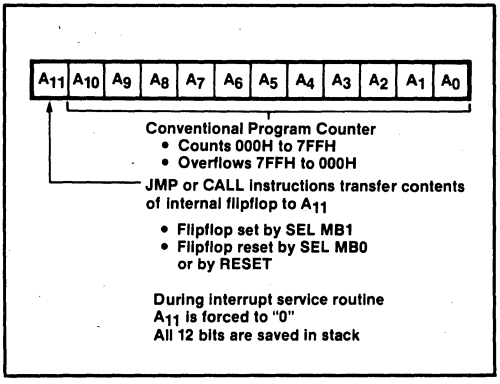


Figure 2. Program Counter

INTERRUPT ROUTINES

Interrupts always vector the program counter to location 3 or 7 in the first 2K bank, and bit 11 of the program counter is held at "0" during the interrupt service routine. The end of the service routine is signalled by the execution of an RETR instruction. Interrupt service routines should therefore be contained entirely in the lower 2K words of program memory. The execution of a SEL MB0 or SEL MB1 instruction within an interrupt routine is not recommended since it will not alter PC11 while in the routine, but will change the internal flip-flop.

2.3 Restoring I/O Port Information

Although the lower half of Port 2 is used to output the four most significant bits of address during an external program memory fetch, the I/O information is still outputed during certain portions of each machine cycle. I/O information is always present on Port 2's lower 4 bits at the rising edge of ALE and can be sampled or latched at this time.

2.4 Expansion Examples

Shown in Figure 3 is the addition of 2K words of program memory using an 2716A 2K x 8 ROM to give a total of 3K words of program memory. In this case no chip select decoding is required and $\overline{PSEN}$ enables the memory directly through the chip select input. If the system requires only 2K of program memory, the same configuration can be used with an 8035AHL substituted for the 8048AH. The 8049AH would provide 4K of program memory with the same configuration.

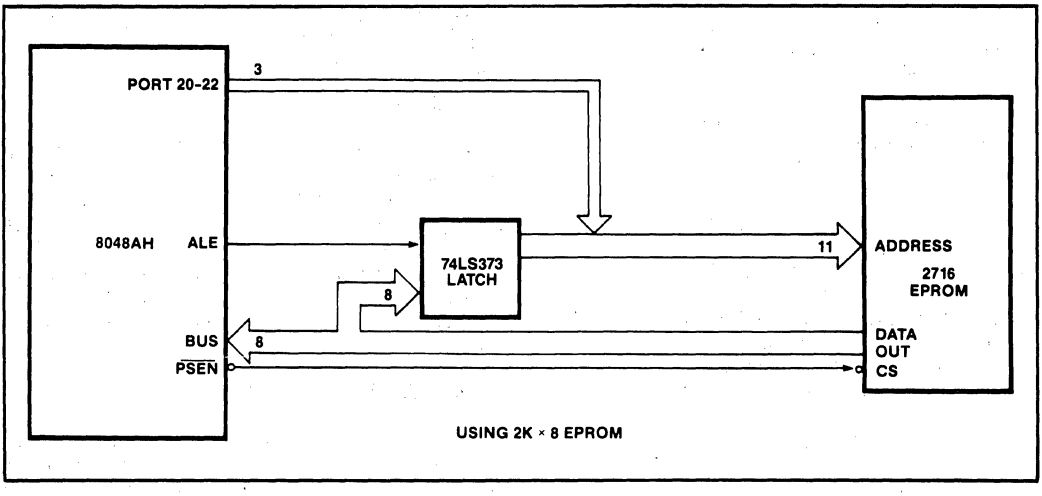


Figure 3. Expanding MCS®-48 Program Memory Using Standard Memory Products

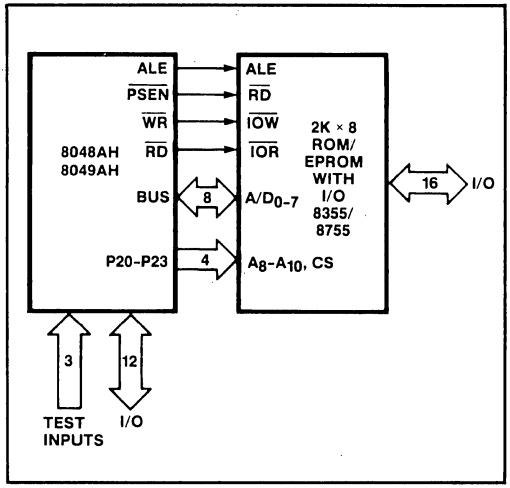


Figure 4. External Program Memory Interface

Figure 4 shows how the 8755/8355 EPROM/ROM with I/O interfaces directly to the 8048AH without the need for an address latch. The 8755/8355 contains an internal 8-bit address latch eliminating the need for an 8212 latch. In addition to a 2K x 8 program memory, the 8755/8355 also contains 16 I/O lines addressable as two 8-bit ports. These ports are addressed as external RAM; therefore the $\overline{RD}$ and $\overline{WR}$ outputs of the 8048AH are required. See the following section on data memory expansion for more detail. The subsequent section on I/O expansion explains the operation of the 16 I/O lines.

3.0 EXPANSION OF DATA MEMORY

Data Memory is expanded beyond the resident 64 words by using the 8085AH type bus feature of the MCS®-48.

3.1 Read/Write Cycle

All address and data is transferred over the 8 lines of BUS. As shown in Figure 5, a read or write cycle occurs as follows:

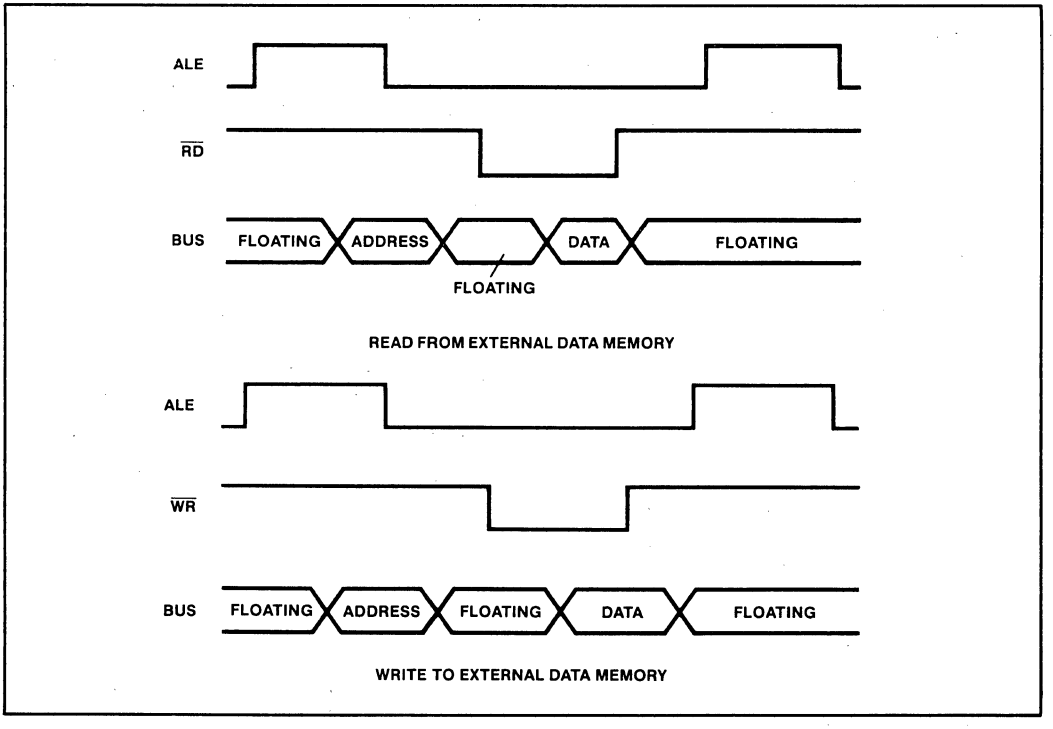


Figure 5. External Data Memory Timings

1) The contents of register R0 or R1 is outputed on BUS.
2) Address Latch Enable (ALE) indicates addresss is valid. The trailing edge of ALE is used to latch the address externally.
3) A read ($\overline{RD}$) or write ($\overline{WR}$) pulse on the corresponding output pins of the 8048AH indicates the type of data memory access in progress. Output data is valid at the trailing edge of $\overline{WR}$ and input data must be valid at the trailing edge of $\overline{RD}$.
4) Dat (8 bits) is transferred in or out over BUS.

3.2 Addressing External Data Memory

External Data Memory is accessed with its own two-cycle move instructions. MOVXA, @R and MOVX@R, A, which transfer 8 bits of data between the accumulator and the external memory location addressed by the contents of one of the RAM Pointer Registers R0 and R1. This allows 256 locations to be addressed in addition to the resident locations. Additional pages may be added by "bank switching" with extra output lines of the 8048AH.

3.3 Examples of Data Memory Expansion

Figure 6 shows how the 8048-AH can be expanded using the 8155 memory and I/O expanding device. Since the 8155 has an internal 8-bit address latch, it can interface directly to the 8048AH without the use of an external latch. The 8155 provides an additional 256 words of static data memory and also includes 22 I/O lines and a 14-bit timer. See the following section on I/O expansion and the 8155 data sheet for more details on these additional features.

4.0 EXPANSION OF INPUT/OUTPUT

There are four possible modes of I/O expansion with the 8048AH: one using a special low-cost expander, the 8243; another using standard MCS-80/85 I/O devices; and a third using the combination memory I/O expander devices the 8155, 8355, and 8755. It is also possible to expand using standard TTL devices.

4.1 I/O Expander Device

The most efficient means of I/O expansion for small systems is the 8243 I/O Expander Device which requires only 4 port lines (lower half of Port 2) for communication with the 8048AH. The 8243 contains four 4-bit I/O ports which serve as an extension of the on-chip I/O and are addressed as ports #4–7 (see Figure 13-7). The following operations may be performed on these ports:

- Transfer Accumulator to Port
- Transfer Port to Accumulator
- AND Accumulator to Port
- OR Accumulator to Port

A 4-bit transfer from a port to the lower half of the Accumulator sets the most significant four bits to zero. All communication between the 8048AH and the 8243 occurs over Port 2 lower (P20–P23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles: The first containing the "op code" and port address, and the second containing the actual 4 bits of data.

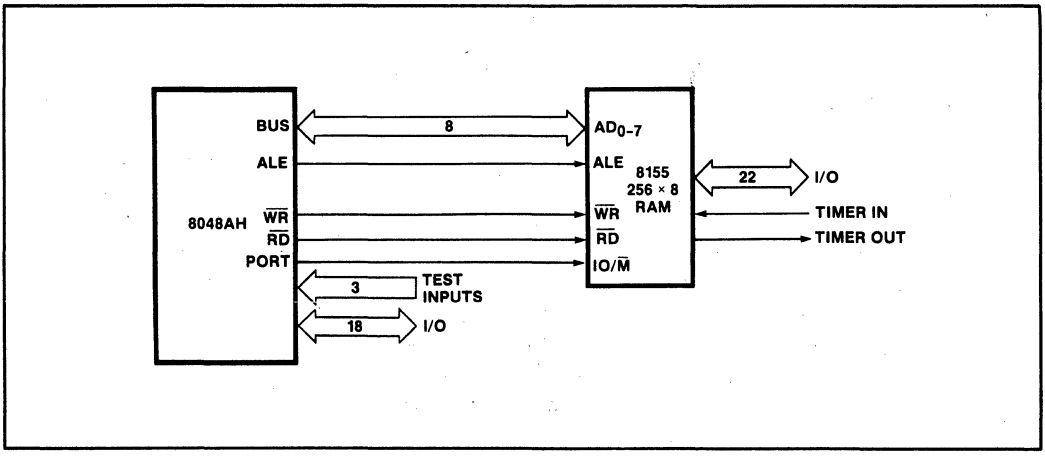


Figure 6. 8048AH Interface to 256 x 8 Standard Memories

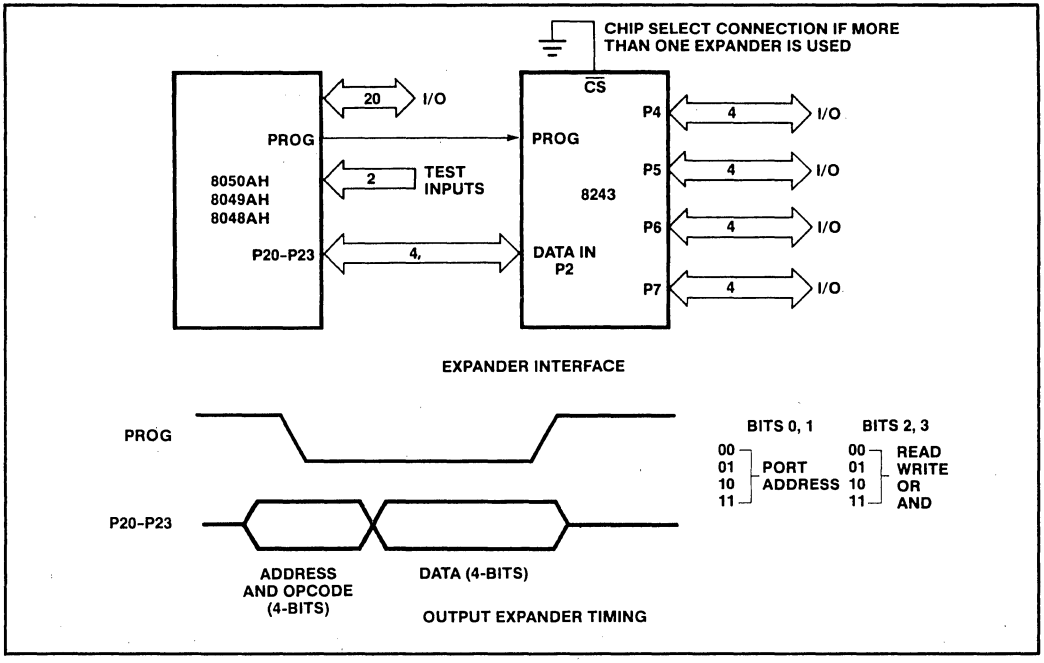
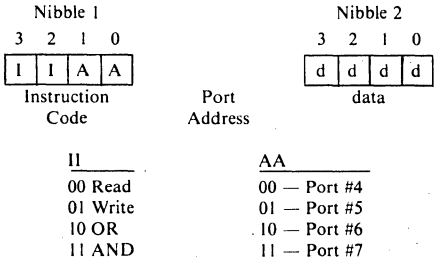


Figure 7. 8243 Expander I/O Interface



A high to low transition of the PROG line indicates that address is present, while allow to high transition indicates the presence of data. Additional 8243's may be added to the four-bit bus and chip selected using additional output lines from the 8048AH/8748H.

I/O PORT CHARACTERISTICS

Each of the four 4-bit ports of the 8243 can serve as either input or output and can provide high drive capability in both the high and low state.

4.2 I/O Expansion with Standard Peripherals

Standard MCS-80/85 type I/O devices may be added to the MCS®-48 using the same bus and timing used for Data Memory expansion. Figure 8 shows an example of how an 8048AH can be connected to an MCS-85 peripheral. I/O devices reside on the Data Memory bus and in the data memory address space and are accessed with the same MOVX instructions. (See the previous section on data memory expansion for a description of timing.) The following are a few of the Standard MCS-80 devices which are very useful in MCS®-48 systems:

- 8214 Priority Interrupt Encoder
- 8251 Serial Communications Interface
- 8255 General Purpose Programmable I/O
- 8279 Keyboard/Display Interface
- 8254 Interval Timer

4.3 Combination Memory and I/O Expanders

As mentioned in the sections on program and data memory expansion, the 8355/8755 and 8155 expanders also contain I/O capability.

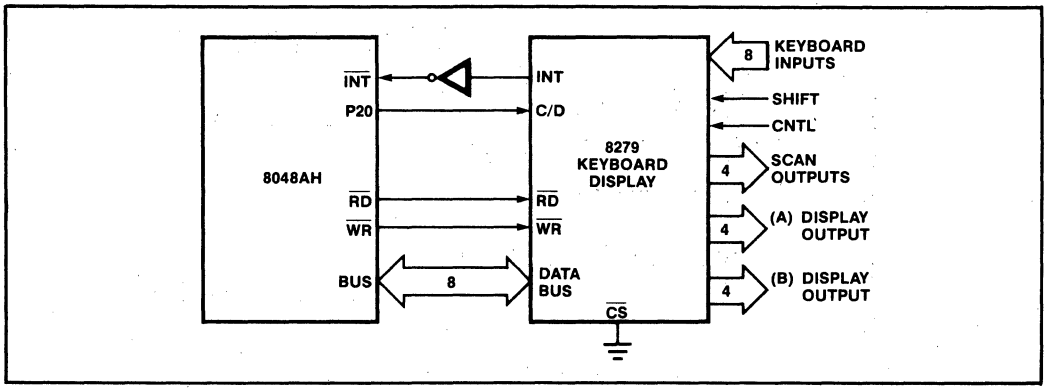


Figure 8. Keyboard/Display Interface

8355/8755: These two parts of ROM and EPROM equivalents and therefore contain the same I/O structure. I/O consists of two 8-bit ports which normally reside in the external data memory address space and are accessed with MOVX instructions. Associated with each port is an 8-bit Data Direction Register which defines each bit in the port as either an input or an output. The data direction registers are directly addressable, thereby allowing the user to define under software control each individual bit of the ports as either input or output. All outputs are statically latched and double buffered. Inputs are not latched.

8155/8156: I/O on the 8155/8156 is configured as two 8-bit programmable I/O ports and one 6-bit programmable port. These three registers and a Control/Status register are accessible as external data memory with the MOVX instructions. The contents of the control register determines the mode of the three ports. The ports can be programmed as input or output with or without associated handshake communication lines. In the handshake mode, lines of the six-bit port become input and output strobes for the two 8-bit ports. Also included in the 8155 is a 14-bit programmable timer. The clock input to the timer and the timer overflow output are available on external pins. The timer can be programmed to stop on terminal count or to continuously reload itself. A square wave or pulse output on terminal count can also be specified.

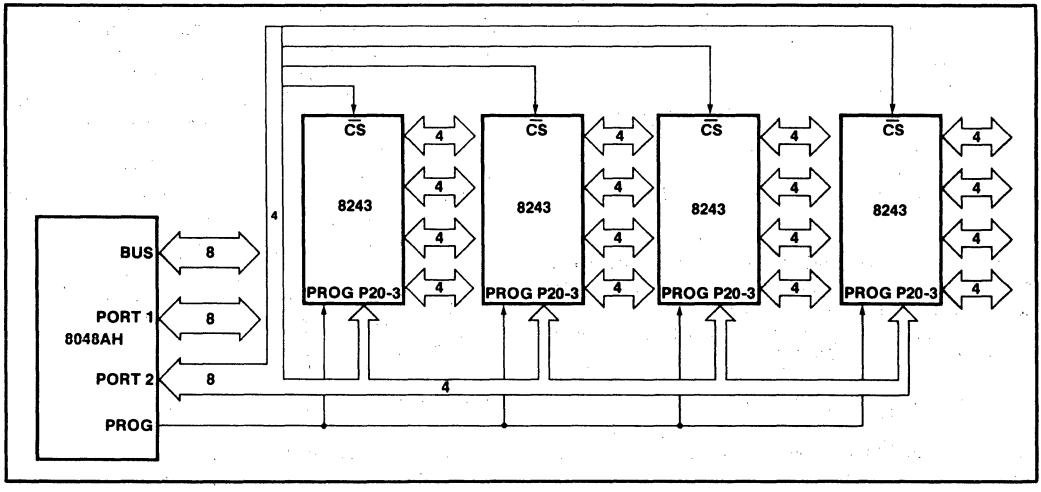


Figure 9. Low Cost I/O Expansion

I/O EXPANSION EXAMPLES

Figure 9 shows the expansion of I/O using multiple 8243's. The only difference from a single 8243 system is the addition of chip selects provided by additional 8048AH output lines. Two output liens and a decoder could also be used to address the four chips. Large numbers of 8243's would require a chip select decoder chip such as the 8205 to save I/O pins.

Figure 10 shows the 8048AH interface to a standard MCS®-80 peripheral; in this case, the 8255 Programmable Peripheral Interface, a 40-pin part which provides three 8-bit programmable I/O ports. The 8255 bus interface is typical of programmable MCS®-80 peripherals with an 8-bit bidirectional data bus, a RD and WR input for Read/Write control, a CS (chip select) input used to enable the Read/Write control logic and the address inputs used to select various internal registers.

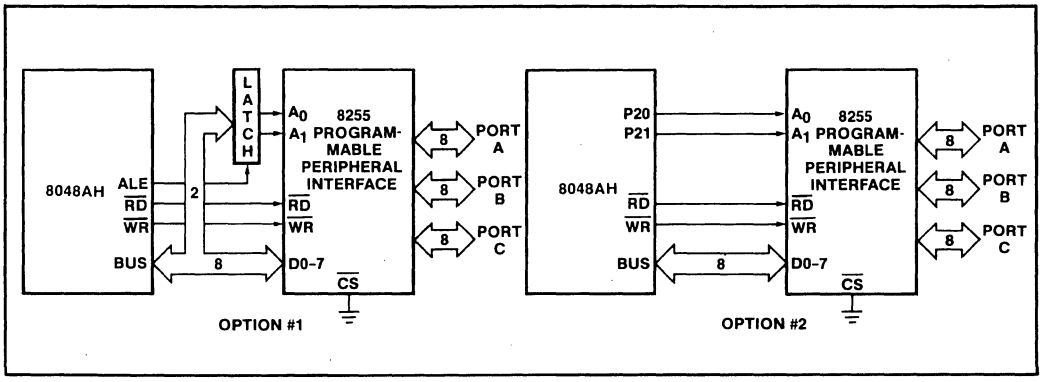


Figure 10. Interface to MCS®-80 Peripherals

Interconnection to the 8048AH is very straightforward with BUS, $\overline{RD}$, and $\overline{WR}$ connecting directly to the corresponding pins on the 8255. The only design consideration is the way in which the internal registers of the 8255 are to be addressed. If the registers are to be addressed as external data memory using the MOVX instructions, the appropriate number of address bits (in this case, 2) must be latched on BUS using ALE as described in the section on external data memories. If only a single device is connected to BUS, the 8255 may be continuously selected by grounding $\overline{CS}$. If multiple 8255's are used, additional address bits can be latched and used as chip selects.

A second addressing method eliminates external latches and chip select decoders by using output port lines as address and chip select lines directly. This method, of course, requires the setting of an output port with address information prior to executing a MOVX instruction.

5.0 MULTI-CHIP MCS®-48 SYSTEMS

Figure 11 shows the addition of two memory expanders to the 8048AH, one 8355/8755 ROM and one 8156 RAM. The main consideration in designing such a system is the addressing of the various memories and I/O ports. Note that in this configuration address lines A_{10} and A_{11} have been ORed to chip select the 8355. This ensures that the chip is active for all external program memory fetches in the 1K to 3K range and is disabled for all other addresses. This gating has been added to allow the I/O port of the 8355 to be used. If the chip was left selected all the time, there would be conflict between these ports and the RAM and I/O of the 8156. The NOR gate could be eliminated and A_{11} connected directly to the CE (instead of $\overline{CE}$) input of the 8355; however, this would create a 1K word "hole" in the program memory by causing the 8355 to be active in the 2K and 4K range instead of the normal 1K to 3K range.

In this system the various locations are addressed as follows:

- Data RAM — Addresses 0 to 255 when Port 2 Bit 0 has been previously set = 1 and Bit 1 set = 0
- RAM I/O — Addresses 0 to 3 when Port 2 Bit 0 = 1 and Bit 1 = 1
- ROM I/O — Addresses 0 to 3 when Port 2 Bit 2 or Bit 3 = 1

See the memory map in Figure 12.

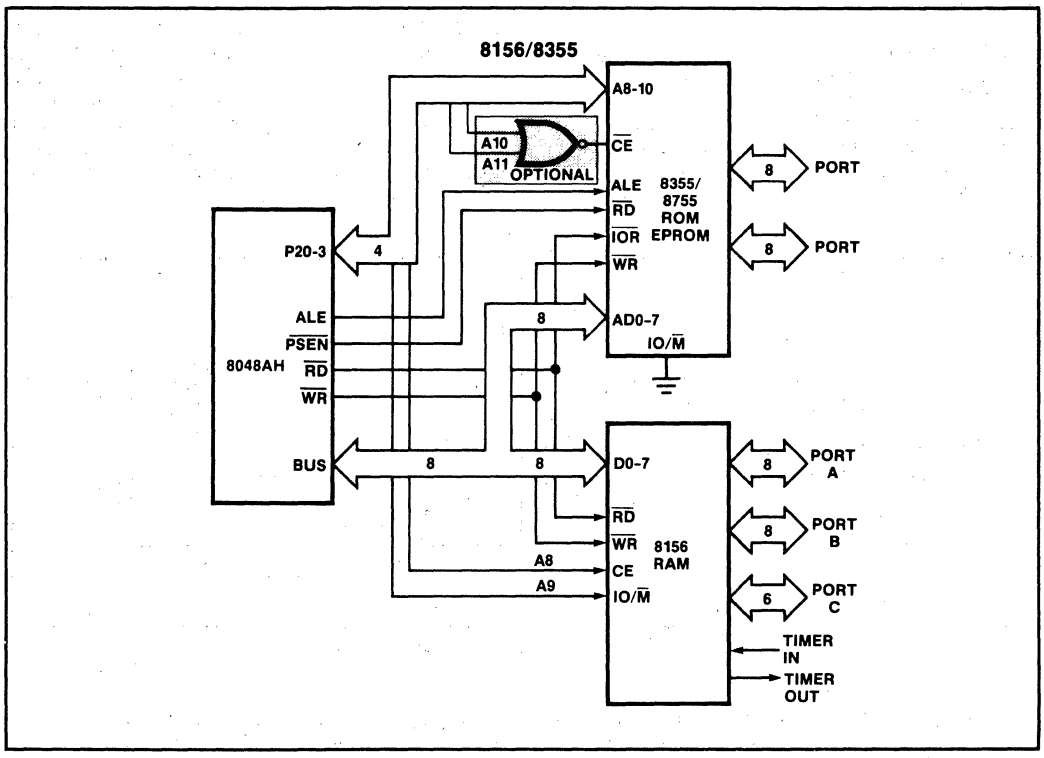


Figure 11. The Three-Component MCS®-48 System

6.0 MEMORY BANK SWITCHING

Certain systems may require more than the 4K words of program memory which are directly addressable by the program counter or more than the 256 data memory and I/O locations directly addressable by the pointer registers R0 and R1. These systems can be achieved using "bank switching" techniques. Bank switching is merely the selection of various blocks of "banks" of memory using dedicated output port lines from the processor. In the case of the 8048AH, program memory is selected in blocks of 4K words at a time, while data memory and I/O are enabled 256 words at a time.

The most important consideration in implementing two or more banks is the software required to cross the bank boundaries. Each crossing of the boundary requires that the processor first write a control bit to an output port before accessing memory or I/O in the new bank. If program memory is being switched, programs should be organized to keep boundary crossings to a minimum.

Jumping to subroutines across the boundary should be avoided when possible since the programmer must keep track of which bank to return to after completion of the subroutine. If these subroutines are to be nested and accessed from either bank, a software "stack" should be implemented to save the bank switch bit just as if it were another bit of the program counter.

From a hardware standpoint bank switching is very straightforward and involves only the connection of an I/O line or lines as bank enable signals. These enables are ANDed with normal memory and I/O chip select signals to activate the proper bank.

7.0 CONTROL SIGNAL SUMMARY

Table 1 summarizes the instructions which activate the various control outputs of the MCS®-48 processors. During all other instructions these outputs are driven to the active state.

Table 1. MCS®-48 Control Signals

Control Signal	When Active
$\overline{RD}$	During MOVX, A, @R or INS Bus
$\overline{WR}$	During MOVX @R, A or OUTL Bus
ALE	Every Machine Cycle
$\overline{PSEN}$	During Fetch of external program memory (instruction or immediate data)
PROG	During MOVD, A,P ANLD P,A MOVD P,A ORLD P,A

8.0 PORT CHARACTERISTICS

8.1 BUS Port Operations

The BUS port can operate in three different modes: as a latched I/O port, as a bidirectional bus port, or as a program memory address output when external memory is used. The BUS port lines are either active high, active low, or high impedance (floating).

The latched mode (INS, OUTL) is intended for use in the single-chip configuration where BUS is not begin used as an expander port. OUTL and MOVX instructions can be mixed if necessary. However, a previously latched output will be destroyed by executing a MOVX instruction and BUS will be left in the high impedance state. INS does not put the BUS in a high impedance state. Therefore, the use of MOVX after OUTL to put the BUS in a high impedance state is necessary before an INS instruction intended to read an external word (as opposed to the previously latched value).

OUTL should never be used in a system with external program memory, since latching BUS can cause the next instruction, if external, to be fetched improperly.

8.2 Port 2 Operations

The lower half of Port 2 can be used in three different ways: as a quasi-bidirectional static port, as an 8243 expander port, and to adddress external program memory.

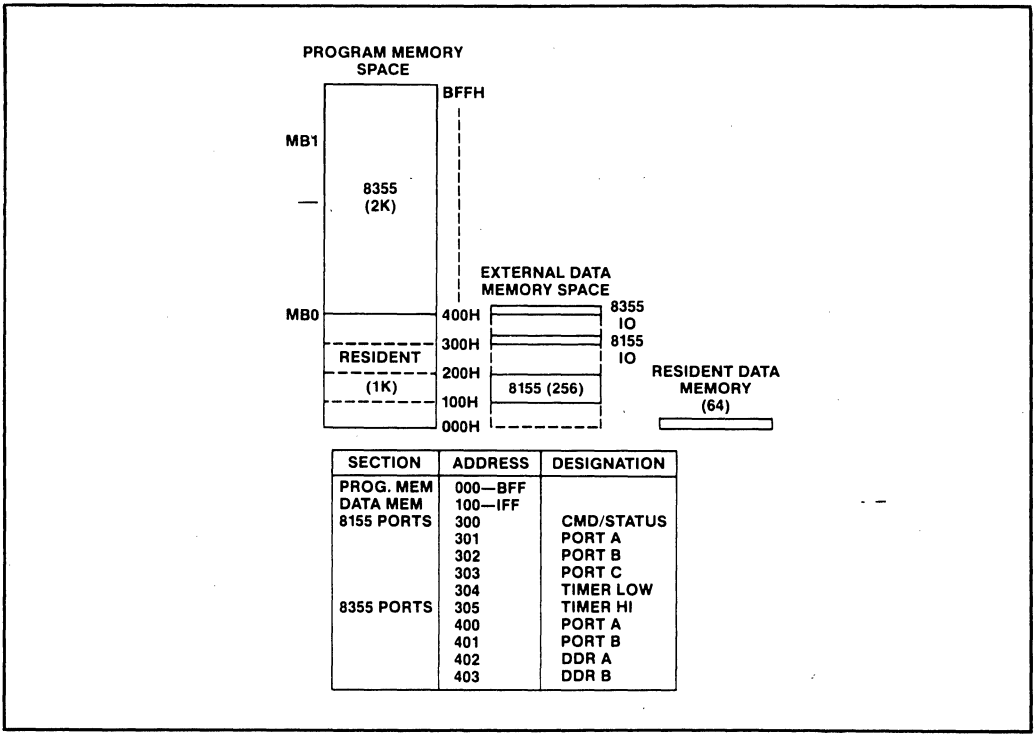


Figure 12. Memory Map for Three-Component MCS®-48 Family

In all cases outputs are driven low by an active device and driven high momentarily by a low impedance device and held high by a high impedance device to VCC.

The port may contain latched I/O data prior to its use in another mode without affecting operation of either. If lower Port 2 (P20-3) is used to output address for an external program memory fetch, the I/O information previously latched will be automatically removed temporarily while address is present, then retored when the fetch is complete. However, if lower Port 2 is used to communicate with an 8243, previously latched I/O information will be removed and not restored. After an input from the 8243, P20-3 will be left in the input mode (floating). After an output to the 8243, P20-3 will contain the value written, ANDed, or ORed to the 8243 port.

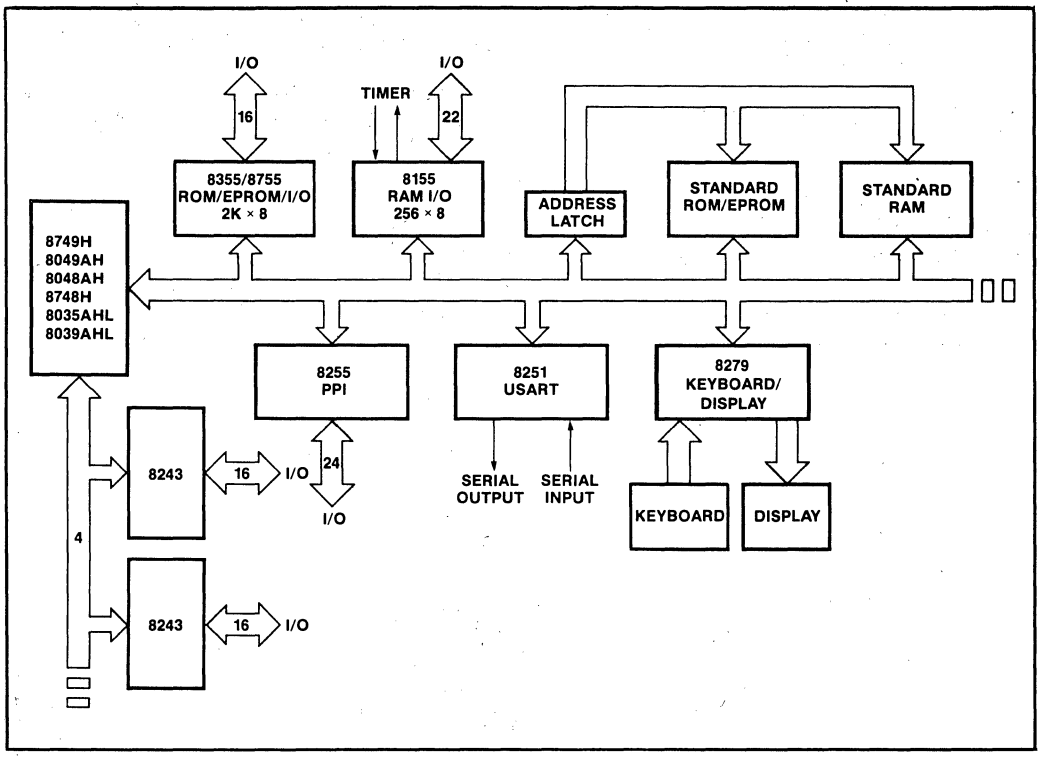


Figure 13. MCS®-48 Expansion Capability

MCS®-48 Instruction Set

3

MCS®-48 INSTRUCTION SET

1.0 INTRODUCTION

The MCS®-48 instruction set is extensive for a machine of its size and has been tailored to be straightforward and very efficient in its use of program memory. All instructions are either one or two bytes in length and over 80% are only one byte long. Also, all instructions execute in either one or two cycles and over 50% of all instructions execute in a single cycle. Double cycle instructions include all immediate instructions, and all I/O instructions.

The MCS-48 microcomputers have been designed to handle arithmetic operations efficiently in both binary and BCD as well as handle the single-bit operations required in control applications. Special instructions have also been included to simplify loop counters, table look-up routines, and N-way branch routines.

1.1 Data Transfers

As can be seen in Figure 1 the 8-bit accumulator is the central point for all data transfers within the 8048. Data can be transferred between the 8 registers of each working register bank and the accumulator directly, i.e., the source or destination register is specified by the instruction. The remaining locations of the internal RAM array are referred to as Data Memory and are addressed indirectly via an address stored in either R0 or R1 of the active register bank. R0 and R1 are also used to indirecly address external data memory when it is present. Transfers to and from internal RAM require one cycle, while transfers to external RAM require two. Constants stored in Program Memory can be loaded directly to the accumulator and to the 8 working registers. Data can also be transferred directly between the accumulator and the on-

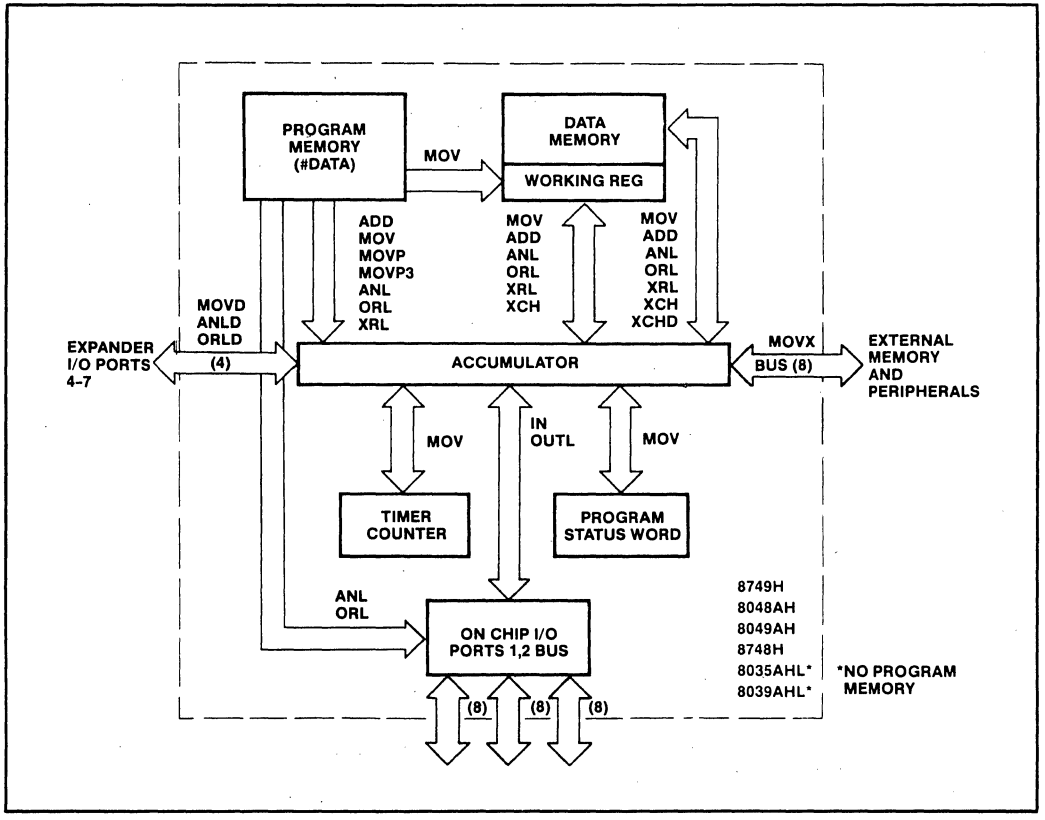


Figure 1. Data Transfer Instructions

board timer counter or the accumulator and the Program Status word (PSW). Writing to the PSW alters machine status accordingly and provides a means of restoring status after an interrupt or of altering the stack pointer if necessary.

1.2 Accumulator Operations

Immediate data, data memory, or the working registers can be added with or without carry to the accumulator. These sources can also be ANDed, ORed, or Exclusive ORed to the accumulator. Data may be moved to or from the accumulator and working registers or data memory. The two values can also be exchanged in a single operation.

In addition, the lower 4 bits of the accumulator can be exchanged with the lower 4-bits of any of the internal RAM locations. This instruction, along with an instruction which swaps the upper and lower 4-bit halves of the accumulator, provides for easy handling of 4-bit quantities, including BCD numbers. To facilitate BCD arithmetic, a Decimal Adjust instruction is included. This instruction is used to correct the result of the binary addition of two 2-digit BCD numbers. Performing a decimal adjust on the result in the accumulator produces the required BCD result.

Finally, the accumulator can be incremented, decremented, cleared, or complemented and can be rotated left or right 1 bit at a time with or without carry.

Although there is no subtract instruction in the 8048AH, this operation can be easily implemented with three single-byte single-cycle instructions.

A value may be subtracted from the accumulator with the result in the accumulator by:

- Complementing the accumulator
- Adding the value to the accumulator
- Complementing the accumulator

1.3 Register Operations

The working registers can be accessed via the accumulator as explained above, or can be loaded immediate with constants from program memory. In addition, they can be incremented or decremented or used as loop counters using the decrement and jump, if not zero instruction, as explained under branch instructions.

All Data Memory including working registers can be accessed with indirect instructions via R0 and R1 and can be incremented.

1.4 Flags

There are four user-accessible flags in the 8048AH: Carry, Auxiliary Carry, F0 and F1. Carry indicates overflow of the accumulator, and Auxiliary Carry is used to indiate overflow between BCD digits and is used during decimal-adjust operation. Both Carry and Auxiliary Carry are accessible as part of the program status word and are stored on the stack during subroutines. F0 and F1 are undedicated general-purpose flags to be used as the programmer desires. Both flags can be cleared or complemented and tested by conditional jump instructions. F0 is also accessible via the Program Status word and is stored on the stack with the carry flags.

1.5 Branch Instructions

The unconditional jump instruction is two bytes and allows jumps anywhere in the first 2K words of program memory. Jumps to the second 2K of memory (4K words are directly addressable) are made first by executing a select memory bank instruction, then executing the jump instruction. The 2K boundary can only be crossed via a jump or subroutine call instruction, i.e., the bank switch does not occur until a jump is executed. Once a memory bank has been selected all subsequent jumps will be to the selected bank until another select memory bank instruction is executed. A subroutine in the opposite bank can be accessed by a select memory bank instruction followed by a call instruction. Upon completion of the subroutine, execution will automatically return to the original bank; however, unless the original bank is reselected, the next jump instruction encountered will again transfer execution to the opposite bank.

Conditional jumps can test the following inputs and machine status:

- T0 Input Pin
- T1 Input Pin
- $\overline{\text{INT}}$ Input Pin
- Accumulator Zero
- Any bit of Accumulator
- Carry Flag
- F0 Flag
- F1 Flag

Conditional jumps allow a branch to any address within the current page (256 words) of execution. The conditions tested are the instantaneous values at the time the conditional jump is executed. For instance, the jump on accumulator zero instruction tests the accumulator itself, not an intermediate zero flag.

The decrement register and jump if not zero instruction combines a decrement and a branch instruction to create an instruction very useful in implementing a loop counter. This instruction can designate any one of the 8 working registers as a counter and can effect a branch to any address within the current page of execution.

A single-byte indirect jump instruction allows the program to be vectored to any one of several different locations based on the contents of the accumulator. The contents of the accumulator points to a location in program memory which contains the jump address. The 8-bit jump address refers to the current page of execution. This instruction could be used, for instance, to vector to any one of several routines based on an ASCII character which has been loaded in the accumulator. In this way ASCII key inputs can be used to initiate various routines.

1.6 Subroutines

Subroutines are entered by executing a call instruction. Calls can be made like unconditional jumps to any address in a 2K word bank, and jumps across the 2K boundary are executed in the same manner. Two separate return instructions determine whether or not status (upper 4-bits of PSW) is restored upon return from the subroutine.

The return and restore status instruction also signals the end of an interrupt service routine if one has been in progress.

1.7 Timer Instructions

The 8-bit on board timer/counter can be loaded or read via the accumulator while the counter is stopped or while counting. The counter can be started as a timer with an internal clock source or an event counter or timer with an external clock applied to the T1 input pin. The instruction executed determines which clock source is used. A single instruction stops the counter whether it is operating with an internal or an external clock source. In addition, two instructions allow the timer interrupt to be enabled or disabled.

1.8 Control Instructions

Two instructions allow the external interrupt source to be enabled or disabled. Interrupts are initially disabled and are automatically disabled while an interrupt service routines is in progress and re-enabled afterward.

There are four memory bank select instructions, two to designate the active working register bank and two to control program memory banks. The operation of the program memory bank switch is explained in Section 2.2 in the Expanded MCS-48 System chapter.

The working register bank switch instructions allow the programmer to immediately substitute a second 8-register working register bank for the one in use. This effectively provides 16 working registers or it can be used as a means of quickly saving the contents of the registers in response to an interrupt. The user has the option to switch or not to switch banks on interrupt. However, if the banks are switched, the original bank will be automatically restored upon execution of a return and restore status instruction at the end of the interrupt service routine.

A special instruction enables an internal clock, which is the XTAL frequency divided by three to be output on pin T0. This clock can be used as a general-purpose clock in the user's system. This instruction should be used only to initialize the system since the clock output can be disabled only by application of system reset.

1.9 Input/Output Instructions

Ports 1 and 2 are 8-bit static I/O ports which can be loaded to and from the accumulator. Outputs are statically latched but inputs are not latched and must be read while inputs are present. In addition, immediate data from program memory can be ANDed or ORed directly to Port 1 and Port 2 with the result remaining on the port. This allows "masks" stored in program memory to selectively set or reset individual bits of the I/O ports. Ports 1 and 2 are configured to allow input on a given pin by first writing a "1" out to the pin.

An 8-bit port called BUS can also be accessed via the accumulator and can have statically latched outputs as well. It too can have immediate data ANDed or ORed directly to its outputs, however, unlike ports 1 and 2, all eight lines of BUS must be treated as either input or output at any one time. In addition to being a static port, BUS can be used as a true synchronous bi-directional port using the Move External instructions used to access external data memory. When these instructions are executed, a corresponding $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ pulse is generated and data is valid only at that time. When data is not being transferred, BUS is in a high impedance state. Note that the OUTL, ANL, and the ORL instructions for the BUS are for use with internal program memory only.

The basic three on-board I/O ports can be expanded via a 4-bit expander bus using half of port 2. I/O expander devices on this bus consist of four 4-bit ports which are addressed as ports 4 through 7. These ports have their own AND and OR instructions like the on-board ports as well as move instructions to transfer data in or out. The expander AND and OR instructions, however, combine the contents of accumulator with the selected port rather than immediate data as is done with the on-board ports.

I/O devices can also be added externally using the BUS port as the expansion bus. In this case the I/O ports become "memory mapped", i.e., they are addressed in the same way as external data memory and exist in the external data memory address space addressed by pointer register R0 or R1.

2.0 INSTRUCTION SET DESCRIPTION

The following pages describe the MCS®-48 instruction set in detail. The instruction set is first summarized with instructions grouped functionally. This summary page is followed by a detailed description listed alphabetically by mnemonic opcode.

The alphabetical listing includes the following information.

- Mnemonic
- Machine Code
- Verbal Description
- Symbolic Description
- Assembly Language Example

The machine code is represented with the most significant bit (7) to the left and two byte instructions are represented with the first byte on the left. The assembly language examples are formulated as follows:

Arbitrary

Label: Mnemonic, Operand;

Descriptive Comment

8048AH/8748H/8049AH/8050AH/8749H
Instruction Set Summary

Mnemonic	Description	Bytes	Cycle
Accumulator			
ADD A, R	Add register to A	1	1
ADD A, @R	Add data memory to A	1	1
ADD A, # data	Add immediate to A	2	2
ADDC A, R	Add register with carry	1	1
ADDC A, @R	Add data memory with carry	1	1
ADDC A, # data	Add immediate with carry	2	2
ANL A, R	And register to A	1	1
ANL A, @R	And data memory to A	1	1
ANL A, # data	And immediate to A	2	2
ORL A, R	Or register to A	1	1
ORL A @R	Or data memory to A	1	1
ORL A, # data	Or immediate to A	2	2
XRL A, R	Exclusive Or register to A	1	1
XRL A, @R	Exclusive or data memory to A	1	1
XRL, A, # data	Exclusive or immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
Input/Output			
IN A, P	Input port to A	1	2
OUTL P, A	Output A to port	1	2
ANL P, # data	And immediate to port	2	2
ORL P, # data	Or immediate to port	2	2
*INS A, BUS	Input BUS to A	1	2
*OUTL BUS, A	Output A to BUS	1	2
*ANL BUS, # data	And immediate to BUS	2	2
*ORL BUS, # data	Or immediate to BUS	2	2
MOVD A, P	Input Expander port to A	1	2
MOVD P, A	Output A to Expander port	1	2
ANLD P, A	And A to Expander port	1	2
ORLD P, A	Or A to Expander port	1	2

Mnemonic	Description	Bytes	Cycles
Registers			
INC R	Increment register	1	1
INC @R	Increment data memory	1	1
DEC R	Decrement register	1	1
Branch			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ R, addr	Decrement register and jump	2	2
JC addr	Jump on carry = 1	2	2
JNC addr	Jump on carry = 0	2	2
JZ addr	Jump on A Zero	2	2
JNZ addr	Jump on A not Zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 = 1	2	2
JF1 addr	Jump on F1 = 1	2	2
JTF addr	Jump on timer flag = 1	2	2
JNI addr	Jump on INT = 0	2	2
JBb addr	Jump on Accumulator Bit	2	2
Subroutine			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
Flags			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear Flag 1	1	1
CPL F1	Complement Flag 1	1	1
Data Moves			
MOV A, R	Move register to A	1	1
MOV A, @R	Move data memory to A	1	1
MOV A, # data	Move immediate to A	2	2
MOV R, A	Move A to register	1	1
MOV @R, A	Move A to data memory	1	1
MOV R, # data	Move immediate to register	2	2
MOV @R, # data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1

*For use with internal memory only.

8048AH/8748H/8049AH/8050AH/8749H
Instruction Set Summary (Con't)

Mnemonic	Description	Bytes	Cycle
Data Moves (Cont'd)			
XCH A, R	Exchange A and register	1	1
XCH A, @R	Exchange A and data memory	1	1
XCHD A, @R	Exchange nibble of A and register	1	1
MOVX A, @R	Move external data memory to A	1	2
MOVX @R, A	Move A to external data memory	1	2
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @A	Move to A from Page 3	1	2
Timer/Counter			
MOV A, T	Read Timer/Counter	1	1
MOV T, A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1

Mnemonic	Description	Bytes	Cycle
Control			
EN I	Enable external Interrupt	1	1
DIS I	Disable external Interrupt	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
ENT0 CLK	Enable clock output on T0	1	1
NOP	No Operation	1	1

MCS®-48 INSTRUCTION SET
Symbols and Abbreviations Used

A	Accumulator
AC	Auxiliary Carry
addr	12-Bit Program Memory Address
Bb	Bit Designator (b = 0–7)
BS	Bank Switch
BUS	BUS Port
C	Carry
CLK	Clock
CNT	Event Counter
CRR	Conversion Result Register
D	Mnemonic for 4-Bit Digit (Nibble)
data	8-Bit Number or Expression
DBF	Memory Bank Flip-Flop
F0, F1	Flag 0, Flag 1
I	Interrupt
P	Mnemonic for "in-page" Operation
PC	Program Counter
Pp	Port Designator (p = 1, 2 or 4–7)
PSW	Program Status Word
Ri	Data memory Pointer (i = 0, or 1)
Rr	Register Designator (r = 0–7)
SP	Stack Pointer
T	Timer
TF	Timer Flag
T0, T1	Test 0, Test 1
X	Mnemonic for External RAM
#	Immediate Data Prefix
@	Indirect Address Prefix
$	Current Value of Program Counter
(X)	Contents of X
((X))	Contents of Location Addressed by X
←	Is Replaced by

ADD A,R_r Add Register Contents to Accumulator

Encoding: | 0 1 1 0 | 1 r r r | 68H-6FH

Description: The contents of register 'r' are added to the accumulator. Carry is affected.

Operation: (A) ← (A) + (Rr) r = 0-7

Example:
```
ADDREG: ADD A,R6      ;ADD REG 6 CONTENTS
                      ;TO ACC
```

ADD A,@R_i Add Data Memory Contents to Accumulator

Encoding; | 0 1 1 0 | 0 0 0 i | 60H-61H

Description: The contents of the resident data memory location addressed by register 'i' bits 0–5** are added to the accumulator. Carry is affected.

Operation: (A) ← (A) + ((Ri)) i = 0-1

Example:
```
ADDM: MOV R0, #01FH   ;MOVE '1F' HEX TO REG 0
      ADD A, @R0      ;ADD VALUE OF LOCATION
                      ;31 TO ACC
```

ADD A,#data Add Immediate Data to Accumulator

Encoding: | 0 0 0 0 | 0 0 1 1 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 03H

Description: This is a 2-cycle instruction. The specified data is added to the accumulator. Carry is affected.

Operation: (A) ← (A) + data

Example:
```
ADDID: ADD A,#ADDER:  ;ADD VALUE OF SYMBOL
                      ;ADDER' TO ACC
```

ADDC A,R_r Add Carry and Register Contents to Accumulator

Encoding: | 0 1 1 1 | 1 r r r | 78H-7FH

Description: The content of the carry bit is added to accumulator location 0 and the carry bit cleared. The contents of register 'r' are then added to the accumulator. Carry is affected.

Operation: (A) ← (A) + (Rr) + (C) r = 0-7

Example:
```
ADDRGC: ADDC A,R4     ;ADD CARRY AND REG 4
                      ;CONTENTS TO ACC
```

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

ADDC A,@R_i Add Carry and Data Memory Contents to Accumulator

Encoding: | 0 1 1 1 | 0 0 0 i | 70H-71H

Description: The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the contents of the resident data memory location addressed by register 'i' bits 0–5** are added to the accumulator. Carry is affected.

Operation: (A) ← (A) + ((Ri)) + (C) i = 0-1

Example:
```
ADDMC:  MOV R1,#40      ;MOVE '40' DEC TO REG 1
        ADDC A,@R1      ;ADD CARRY AND LOCATION 40
                        ;CONTENTS TO ACC
```

ADDC A,@data Add Carry and Immediate Data to Accumulator

Encoding: | 0 0 0 1 | 0 0 1 1 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 13H

Description: This is a 2-cycle instruction. The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the specified data is added to the accumulator. Carry is affected.

Operation: (A) ← (A) + data + (C)

Example:
```
ADDC A,#225             ;ADD CARRY AND '225' DEC
                        ;TO ACC
```

ANL A,R_r Logical AND Accumulator with Register Mask

Encoding: | 0 1 0 1 | 1 r r r | 58H-5FH

Description: Data in the accumulator is logically ANDed with the mask contained in working register 'r'.

Operation: (A) ← (A) AND (Rr) r = 0-7

Example:
```
ANDREG: ANL A,R3        ;'AND' ACC CONTENTS WITH MASK
                        ;IN REG 3
```

ANL A,@R_i Logical AND Accumulator with memory Mask

Encoding: | 0 1 0 1 | 0 0 0 i | 50H-51H

Description: Data in the accumulator is logically ANDed with the mask contained in the data memory location referenced by register 'i' bits 0-5**.

Operation: (A) ← (A) AND ((Ri)) i = 0-1

Example:
```
ANDDM:  MOV R0,#03FH    ;MOVE '3F' HEX TO REG 0
        ANL A, @R0      ;'AND' ACC CONTENTS WITH
                        ;MASK IN LOCATION 63
```

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

ANL A,#data Logical AND Accumulator with Immediate Mask

Encoding: | 0 1 0 1 | 0 0 1 1 | | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 53H

Description: This is a 2-cycle instruction. Data in the accumulator is logically ANDed with an immediately-specified mask.

Operation: (A) ← (A) AND data

Examples:
```
ANDID: ANL A,#0AFH      ;'AND' ACC CONTENTS
                        ;WITH MASK 10101111
       ANL A,#3 + X/Y   ;'AND' ACC CONTENTS
                        ;WITH VALUE OF EXP
                        ;'3 + XY/Y'
```

ANL BUS,#data* Logical AND BUS with Immediate Mask

Encoding: | 1 0 0 1 | 1 0 0 0 | | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 98H

Description: This is a 2-cycle instruction. Data on the BUS port is logically ANDed with an immediately-specified mask. This instruction assumes prior specification of an 'OUTL BUS, A' instruction.

Operation: (BUS) ← (BUS) AND data

Example:
```
ANDBUS: ANL BUS,#MASK   ;'AND' BUS CONTENTS
                        ;WITH MASK EQUAL VALUE
                        ;OF SYMBOL 'MASK'
```

ANL Pp,#data Logical AND Port 1-2 with Immediate Mask

Encoding: | 1 0 0 1 | 1 0 p p | | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 99H-9AH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ANDed with an immediately-specified mask.

Operation: (Pp) ← (Pp) AND DATA p = 1-2

Example:
```
ANDP2: ANL P2,#0F0H     ;'AND' PORT 2 CONTENTS
                        ;WITH MASK 'F0' HEX
                        ;(CLEAR P20-23)
```

* For use with internal program memory ONLY.

ANLD Pp,A Logical AND Port 4-7 with Accumulator Mask

Encoding: | 1 0 0 1 | 1 1 p p | 9CH-9FH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ANDed with the digit mask contained in accumulator bits 0-3.

Operation: (Pp) ← (Pp) AND (A0-3) p = 4-7

Note: The mapping of port 'p' to opcode bits 0-1 is as follows:

1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example:
```
ANDP4: ANLD P4,A        ;'AND' PORT 4 CONTENTS
                        ;WITH ACC BITS 0-3
```

CALL address Subroutine Call

Encoding: | a_{10} a_9 a_8 1 | 0 1 0 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

Page	Hex Op Code
0	14
1	34
2	54
3	74
4	94
5	B4
6	D4
7	F4

Description: This is a 2-cycle instruction. The program counter and PSW bits 4-7 are saved in the stack. The stack pointer (PSW bits 0-2) is updated. Program control is then passed to the location specified by 'address'. PC bit 11 is determined by the most recent SEL MB instruction.

A CALL cannot begin in locations 2046-2047 or 4094-4095. Execution continues at the instruction following the CALL upon return from the subroutine.

Operation:
$((SP)) \leftarrow (PC), (PSW_{4-7})$
$(SP) \leftarrow (SP) + 1$
$(PC_{8-10}) \leftarrow (addr_{8-10})$
$(PC_{0-7}) \leftarrow (addr_{0-7})$
$(PC_{11}) \leftarrow DBF$

Example: Add three groups of two numbers. Put subtotals in locations 50, 51 and total in location 52.

```
        MOV R0,#50      ;MOVE '50' DEC TO ADDRESS
                        ;REG 0
BEGADD: MOV A,R1        ;MOVE CONTENTS OF REG 1
                        ;TO ACC
        ADD A,R2        ;ADD REG 2 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'
        ADDC A R3       ;ADD REG 3 TO ACC
        ADDC A,R4       ;ADD REG 4 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'
        ADDC A,R5       ;ADD REG 5 TO ACC
        ADDC A,R6       ;ADD REG 6 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'
SUBTOT: MOV @R0,A       ;MOVE CONTENTS OF ACC TO
                        ;LOCATION ADDRESSED BY
                        ;REG 0
        INC R0          ;INCREMENT REG 0
        RET             ;RETURN TO MAIN PROGRAM
```

CLR A Clear Accumulator

Encoding: | 0 0 1 0 | 0 1 1 1 | 27H

Description: The contents of the accumulator are cleared to zero.

Operation: A ← 0

CLR C Clear Carry Bit

Encoding: | 1 0 0 1 | 0 1 1 1 | 97H

Description: During normal program execution, the carry bit can be set to one by the ADD, ADDC, RLC, CPL C, RRC, and DAA insructions. This instruction resets the carry bit to zero.

Operation: C ← 0

CLR F1 Clear Flag 1

Encoding: | 1 0 1 0 | 0 1 0 1 | A5H

Description: Flag 1 is cleared to zero.

Operation: (F1) ← 0

CLR F0 Clear Flag 0

Encoding: | 1 0 0 0 | 0 1 0 1 | 85H

Description: Flag 0 is cleared to zero.

Operation: (F0) ← 0

CPL A Complement Accumulator

Encoding: | 0 0 1 1 | 0 1 1 1 | 37H

Description: The contents of the accumulator are complemented. This is strictly a one's complement. Each one is changed to zero and vice-versa.

Operation: (A) ← NOT (A)

Example: Assume accumulator contains 01101010.

```
CPLA: CPL A          ;ACC CONTENTS ARE COMPLE-
                     ;MENTED TO 10010101
```

CPL C Complement Carry Bit

Encoding: | 1 0 1 0 | 0 1 1 1 | A7H

Description: The setting of the carry bit is complemented; one is changed to zero, and zero is changed to one.

Operation: (C) ← NOT (C)

Example: Set C to one; current setting is unknown.

```
CTO1: CLR C          ;C IS CLEARED TO ZERO
      CPL C          ;C IS SET TO ONE
```

CPL F0 Complement Flag 0

Encoding: | 1 0 0 1 | 0 1 0 1 | 95H

Description: The setting of flag 0 is complemented; one is changed to zero, and zero is changed to one.

Operation: F0 ← NOT (F0)

CPL F1 Complement Flag 1

Encoding: | 1 0 1 1 | 0 1 0 1 | B5H

Description: The setting of flag 1 is complemented; one is changed to zero, and zero is changed to one.

Operation: (F1) ← NOT (F1)

DA A Decimal Adjust Accumulator

Encoding: | 0 1 0 1 | 0 1 1 1 | 57H

Description: The 8-bit accumulator value is adjusted to form two 4-bit Binary Coded Decimal (BCD) digits following the binary addition of BCD numbers. The carry bit C is affected. If the contents of bits 0-3 are greater than nine, or if AC is one, the accumulator is incremented by six.

The four high-order bits are then checked. If bits 4-7 exceed nine, or if C is one, these bits are increased by six. If an overflow occurs, C is set to one.

Example: Assume accumulator contains 10011011.

```
DA A                                ;ACC Adjusted to 00000001
                                    ;WITH C SET
C AC 7     4 3     0
0  0 1 0 0 1 1 0 1 1
     0 0 0 0 0 1 1 0        ADD SIX TO BITS 0-7
0  1 1 0 1 0 0 0 0 1
     0 1 1 0                ADD SIX TO BITS 4-7
1  0 0 0 0 0 0 0 0 1        OVERFLOW TO C
```

DEC A Decrement Accumulator

Encoding: | 0 0 0 0 | 0 1 1 1 | 07H

Description: The contents of the accumulator are decremented by one. The carry flag is not affected.

Operation: (A) ← (A) -1

Example: Decrement contents of external data memory location 63.

```
MOV R0,#3FH                 ;MOVE '3F' HEX TO REG 0
MOVX A, @R0                 ;MOVE CONTENTS OF
                            ;LOCATION 63 TO ACC
DEC A                       ;DECREMENT ACC
MOVX @R0,A                  ;MOVE CONTENTS OF ACC TO
                            ;LOCATION 63 IN EXPANDED
                            ;MEMORY
```

DEC Rr Decrement Register

Encoding: | 1 1 0 0 | 1 r r r | C8H-CFH

Description: The contents of working register 'r' are decremented by one.

Operation: (Rr) ← (Rr) -1 r = 0-7

Example:

```
DECR1: DEC R1               ;DECREMENT CONTENTS OF REG 1
```

DIS I External Interrupt

Encoding: | 0 0 0 1 | 0 1 0 1 | 15H

Description: External interrupts are disabled. A low signal on the interrupt input pin has no effect.

DIS TCNTI Disable Timer/Counter Interrupt

Encoding: | 0 0 1 1 | 0 1 0 1 | 35H

Description: Timer/counter interrupts are disabled. Any pending timer interrupt request is cleared. The interrupt sequence is not initiated by an overflow, but the timer flag is set and time accumulation continues.

DJNZ R_r, address Decrement Register and Test

Encoding: | 1 1 1 0 | 1 r r r | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | E8H-EFH

Description: This is a 2-cycle instruction. Register 'r' is decremented, then tested for zero. If the register contains all zeros, program control falls through to the next instruction. If the register contents are not zero, control jumps to the specified 'address'.

The address in this case must evaluate to 8-bits, that is, the jump must be to a location within the current 256-location page.

Example: (Rr) ← (Rr) -1 r = 0-7
If Rr not 0
(PC_{0-7}) ← addr
Note: A 12-bit address specification does not cause an error if the DJNZ instruction and the jump target are on the same page. If the DJNZ instruction begins in location 255 of a page, it must jump to a target address on the following page.

Example: Increment values in data memory locations 50-54.

```
        MOV R0,#50          ;MOVE '50' DEC TO ADDRESS
                            ;REG 0
        MOV R3,#5           ;MOVE '5' DEC TO COUNTER
                            ;REG 3
INCRT:  INC @R0             ;INCREMENT CONTENTS OF
                            ;LOCATION ADDRESSED BY
                            ;REG 0
        INC R0              ;INCREMENT ADDRESS IN REG 0
        DJNZ R3, INCRT      ;DECREMENT REG 3 — JUMP TO
                            ;'INCRT' IF REG 3 NONZERO
        NEXT —              ;'NEXT' ROUTINE EXECUTED
                            ;IF R3 IS ZERO
```

EN I Enable External Interrupt

Encoding: | 0 0 0 0 | 0 1 0 1 | 05H

Description: External interrupts are enabled. A low signal on the interrupt input pin initiates the interrupt sequence.

EN TCNTI Enable Timer/Counter Interrupt

Encoding: | 0 0 1 0 | 0 1 0 1 | 25H

Description: Timer/counter interrupts are enabled. An overflow of the timer/counter initiates the interrupt sequence.

ENT0 CLK Enable Clock Output

Encoding: | 0 1 1 1 | 0 1 0 1 | 75H

Description: The test 0 pin is enabled to act as the clock output. This function is disabled by a system reset.

Example:
```
EMTST0: ENT0 CLK          ;ENABLE T0 AS CLOCK OUTPUT
```

IN A,Pp Input Port or Data to Accumulator

Encoding: | 0 0 0 0 | 1 0 p p | 09H-0AH

Description: This is a 2-cycle instruction. Data present on port 'p' is transferred (read) to the accumulator.

Operation: (A) ← (Pp)

```
                          p = 1-2
INP12: IN A,P1            ;INPUT PORT 1 CONTENTS TO ACC
       MOV R6,A           ;MOVE ACC CONTENTS TO REG 6
       IN A,P2            ;INPUT PORT 2 CONTENTS TO ACC
       MOV R7,A           ;MOVE ACC CONTENTS TO REG 7
```

INC A Increment Accumulator

Encoding: | 0 0 0 1 | 0 1 1 1 | 17H

Description: The contents of the accumulator are incremented by one. Carry is not affected.

Operation. (A) ← (A) +1

Example: Increment contents of location 100 in external data memory.

```
INCA:  MOV R0,#100      ;MOVE '100' DEC TO ADDRESS REG 0
       MOVX A,@R0       ;MOVE CONTENTS OF LOCATION
                        ;100 TO ACC
       INC A            ;INCREMENT A
       MOVX @R0,A       ;MOVE ACC CONTENTS TO
                        ;LOCATION 101
```

INC R_r Increment Register

Encoding: | 0 0 0 1 | 1 r r r | 18H-1FH

Description: The contents of working register 'r' are incremented by one.

Operation: (Rr) ← (Rr) + 1 r = 0-7

Example:
```
INCR0: INC R0           ;INCREMENT CONTENTS OF REG 0
```

INC @R_i Increment Data Memory Location

Encoding: | 0 0 0 1 | 0 0 0 i | 10H-11H

Description: The contents of the resident data memory location addressed by register 'i' bits 0–5** are incremented by one.

Operation: ((Ri)) ← ((Ri)) + 1 i = 0-1

Example:
```
INCDM: MOV R1,#03FH     ;MOVE ONES TO REG 1
       INC @R1          ;INCREMENT LOCATION 63
```

INS A,BUS* Strobed Input of BUS Data to Accumulator

Encoding: | 0 0 0 0 | 1 0 0 0 | 08H

Description: This is a 2-cycle instruction. Data present on the BUS port is transferred (read) to the accumulator when the RD pulse is dropped. (Refer to section on programming memory expansion for details.)

Operation: (A) ← (BUS)

Example:
```
INPBUS: INS A,BUS       ;INPUT BUS CONTENTS TO ACC
```

* For use with internal program memory ONLY.

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

JBb address Jump If Accumulator Bit Is Set

Encoding: | b_2 b_1 b_0 1 | 0 0 1 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

Accumulator Bit	Hex Op Code
0	12
1	32
2	52
3	72
4	92
5	B2
6	D2
7	F2

Description: This is a 2-cycle instruction. Control passes to the specified address if accumulator bit 'b' is set to one.

Operation:

	b = 0–7
$(PC_{0-7}) \leftarrow$ addr	If Bb = 1
(PC) = (PC) + 2	If Bb = 0

Example:

```
JB4IS1: JB4 NEXT    ;JUMP TO 'NEXT' ROUTINE
                    ;IF ACC BIT 4 = 1
```

JC address Jump If Carry Is Set

Encoding: | 1 1 1 1 | 0 1 1 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | F6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the carry bit is set to one.

Operation:

$(PC_{0-7}) \leftarrow$ addr	If C = 1
(PC) = (PC) + 2	If C = 0

Example:

```
JC1: JC OVFLOW      ;JUMP TO 'OVFLOW' ROUTINE
                    ;IF C = 1
```

JF0 address Jump If Flag 0 Is Set

Encoding: | 1 0 1 1 | 0 1 1 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | B6H

Description: This is a 2-cycle instruction. Control passes to the specified address if flag 0 is set to one.

Operation:

$(PC_{0-7}) \leftarrow$ addr	If F0 = 1
(PC) =(PC) + 2	If F0 = 0

Example:

```
JF0IS1: JF0 TOTAL   ;JUMP TO 'TOTAL' ROUTINE IF F0 = 1
```

JF1 address Jump If Flag 1 Is Set

Encoding: | 0 1 1 1 | 0 1 1 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 76H

Description: This is a 2-cycle instruction. Control passes to the specified address if flag 1 is set to one.

Operation: $(PC_{0-7}) \leftarrow$ addr If F1 = 1
(PC) =(PC + 2) If F1 = 0

Example:
```
JF1IS1: JF1 FILBUF      ;JUMP TO 'FILBUF'
                        ;ROUTINE IF F1 = 1
```

JMP address Direct Jump within 2K Block

Encoding: | a_{10} a_9 a_8 0 | 0 1 0 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 |

Page	Hex Op Code
0	04
1	24
2	44
3	64
4	84
5	A4
6	C4
7	E4

Description: This is a 2-cycle instruction. Bits 0-10 of the program counter are replaced with the directly-specified address. The setting of PC bit 11 is determined by the most recent SELECT MB instruction.

Operation: $(PC_{8-10}) \leftarrow$ addr 8-10
$(PC_{0-7}) \leftarrow$ addr 0-7
$(PC_{11}) \leftarrow$ DBF

Example:
```
JMP SUBTOT              ;JUMP TO SUBROUTINE 'SUBTOT'
JMP $-6                 ;JUMP TO INSTRUCTION SIX
                        ;LOCATIONS BEFORE CURRENT
                        ;LOCATION
JMP 2FH                 ;JUMP TO ADDRESS '2F' HEX
```

JMPP @A Indirect Jump within Page

Encoding: | 1 0 1 1 | 0 0 1 1 | B3H

Description: This is a 2-cycle insruction. The contents of the program memory location pointed to by the accumulator are substituted for the 'page' portion of the program counter (PC bits 0-7).

Operation: $(PC_{0-7}) \leftarrow ((A))$

Example: Assume accumulator contains 0FH.

```
JMPPAG: JMPP @A          ;JUMP TO ADDRESS STORED IN
                         ;LOCATION 15 IN CURRENT PAGE
```

JNC address Jump If Carry Is Not Set

Encoding:

1 1 1 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	E6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the carry bit is not set, that is, equals zero.

Operation:

$(PC_{0-7}) \leftarrow$ addr If C = 0
(PC) = (PC) + 2 If C = 1

Example:

```
JC0: JNC NOVFLO          ;JUMP TO 'NOVFLO' ROUTINE
                         ;IF C = 0
```

JNI address Jump If Interrupt Input Is Low

Encoding:

1 0 0 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	86H

Description: This is a 2-cycle instruction. Control passes to the specified address if the interrupt input signal is low (= 0), that is, an external interrupt has been signaled. (This signal initiates an interrupt service sequence if the external interrupt is enabled.)

Operation:

$(PC_{0-7}) \leftarrow$ addr If I = 0
(PC) = (PC) + 2 If I = 1

Example:

```
LOC 3: JNI EXTINT        ;JUMP TO 'EXTINT' ROUTINE
                         ;IF I = 0
```

JNT0 address Jump If Test 0 Is Low

Encoding:

0 0 1 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	26H

Description: This is a 2-cycle instruction. Control passes to the specified address, if the test 0 signal is low.

Operation:

$(PC_{0-7}) \leftarrow$ addr If T0 = 0
(PC) = (PC) + 2 If T0 = 1

Example:

```
JT0LOW: JNT0 60          ;JUMP TO LOCATION 60 DEC
                         ;IF T0 = 0
```

JNT1 address Jump If Test 1 Is Low

Encoding: | 0 1 0 0 | 0 1 1 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 46H

Description: This is a 2-cycle instruction. Control passes to the specified address, if the test 1 signal is low.

Operation: (PC_{0-7}) ← addr — If T1 = 0
(PC) = (PC) + 2 — If T1 = 1

JNZ Address Jump If Accumulator Is Not Zero

Encoding: | 1 0 0 1 | 0 1 1 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 96H

Description: This is a 2-cycle instruction. Control passes to the specified address if the accumulator contents are nonzero at the time this instruction is executed.

Operation: (PC_{0-7}) ← addr — If A ≠ 0
(PC) = (PC) + 2 — If A = 0

Example:
```
JACCN0: JNZ 0ABH      ;JUMP TO LOCATION 'AB' HEX
                      ;IF ACC VALUE IS NONZERO
```

JTF address Jump If Timer Flag Is Set

Encoding: | 0 0 0 1 | 0 1 1 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 16H

Description: This is a 2-cycle instruction. Control passes to the specified address if the timer flag is set to one, that is, the timer/counter register has overflowed. Testing the timer flag resets it to zero. (This overflow initiates an interrupt service sequence if the timer-overflow interrupt is enabled.)

Operation: (PC_{0-7}) ← addr — If TF = 1
(PC) = (PC) + 2 — If TF = 0

Example:
```
JTF1: JTF TIMER       ;JUMP TO 'TIMER' ROUTINE
                      ;IF TF = 1
```

JT0 address Jump If Test 0 Is High

Encoding: | 0 0 1 1 | 0 1 1 0 | | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 36H

Description: This is a 2-cycle instruction. Control passes to the specified address if the test 0 signal is high (= 1).

Operation: (PC_{0-7}) ← addr — If T0 = 1
(PC) = (PC) + 2 — If T0 = 0

Example:
```
JT0HI: JT0 53         ;JUMP TO LOCATION 53 DEC
                      ;IF T0 = 1
```

JT1 address Jump If Test 1 Is High

Encoding: | 0 1 0 1 | 0 1 1 0 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 56H

Description: This is a 2-cycle instruction. Control passes to the specified address if the test 1 signal is high (= 1).

Operation: $(PC_{0-7}) \leftarrow$ addr — If T1 = 1
(PC) = (PC) + 2 — If T1 = 0

Example:
```
JT1HI: JT1 COUNT        ;JUMP TO 'COUNT' ROUTINE
                        ;IF T1 = 1
```

JZ address Jump If Accumulator Is Zero

Encoding: | 1 1 0 0 | 0 1 1 0 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | C6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the accumulator contains all zeros at the time this instruction is executed.

Operation: $(PC_{0-7}) \leftarrow$ addr — If A = 0
(PC) = (PC) + 2 — If A $\neq$ 1

Example:
```
JACCO: JZ 0A3H          ;JUMP TO LOCATION 'A3' HEX
                        ;IF ACC VALUE IS ZERO
```

MOV A,#data Move Immediate Data to Accumulator

Encoding: | 0 0 1 0 | 0 0 1 1 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 23H

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is loaded in the accumulator.

Operation: (A) $\leftarrow$ data

Example:
```
MOV A,#0A3H             ;MOVE 'A3' HEX TO ACC
```

MOV A,PSW Move PSW Contents to Accumulator

Encoding: | 1 1 0 0 | 0 1 1 1 | C7H

Description: The contents of the program status word are moved to the accumulator.

Operation: (A) $\leftarrow$ (PSW)

Example: Jump to 'RB1SET' routine if PSW bank switch, bit 4, is set.
```
BSCHK: MOV A,PSW        ;MOVE PSW CONTENTS TO ACC
       JB4 RB1SET       ;JUMP TO 'RB1SET' IF ACC BIT 4 = 1
```

MOV A,R_r Move Register Contents to Accumulator

Encoding: | 1 1 1 1 | 1 r r r | F8H-FFH

Description: 8-bits of data are removed from working register 'r' into the accumulator.

Operation: (A) ← (Rr) r = 0-7

Example:
```
MAR: MOV A,R3          ;MOVE CONTENTS OF REG 3 TO ACC
```

MOV A,@R_i Move Data Memory Contents to Accumulator

Encoding | 1 1 1 1 | 0 0 0 i | F0H-F1H

Description: The contents of the resident data memory location addressed by bits 0–5** of register 'i' are moved to the accumulator. Register 'i' contents are unaffected.

Operation: (A) ← ((Ri)) i = 0-1

Example: Assume R1 contains 00110110.
```
MADM: MOV A,@R1        ;MOVE CONTENTS OF DATA MEM
                       ;LOCATION 54 TO ACC
```

MOV A,T Move Timer/Counter Contents to Accumulator

Encoding: | 0 1 0 0 | 0 0 1 0 | 42H

Description: The contents of the timer/event-counter register are moved to the accumulator.

Operation: (A) ← (T)

Example: Jump to "EXIT" routine when timer reaches '64', that is, when bit 6 set—assuming initialization 64,
```
TIMCHK: MOV A,T        ;MOVE TIMER CONTENTS TO ACC
        JB6 EXIT       ;JUMP TO 'EXIT' IF ACC BIT 6 = 1
```

MOV PSW,A Move Accumulator Contents to PSW

Encoding: | 1 1 0 1 | 0 1 1 1 | D7H

Description: The contents of the accumulator are moved into the progam status word. All condition bits and the stack pointer are affected by this move.

Operation: (PSW) ← (A)

Example: Move up stack pointer by two memory locations, that is, increment the pointer by one.
```
INCPTR: MOV A,PSW      ;MOVE PSW CONTENTS TO ACC
        INC A          ;INCREMENT ACC BY ONE
        MOV PSW,A      ;MOVE ACC CONTENTS TO PSW
```

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

MOV R_r,A Move Accumulator Contents to Register

Encoding: | 1 0 1 0 | 1 r r r | A8H-AFH

Description: The contents of the accumulator are moved to register 'r'.

Operation: (Rr) ← (A) r = 0-7

Example:
```
MRA:  MOV R0,A          ;MOVE CONTENTS OF ACC TO REG 0
```

MOV R_r,#data Move Immediate Data to Register

Encoding: | 1 0 1 1 | 1 r_2 r_1 r_0 | | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | B8H-BFH

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to register 'r'.

Operation: (Rr) ← data r = 0-7

Examples:
```
MIR4:  MOV R4,#HEXTEN      ;THE VALUE OF THE SYMBOL
                           ;'HEXTEN' IS MOVED INTO REG 4
MIR 5: MOV R5,#PI*(R*R)    ;THE VALUE OF THE EXPRESSION
                           ;'PI*(R*R)' IS MOVED INTO REG 5
MIR 6: MOV R6, #0ADH       ;'AD' HEX IS MOVED INTO REG 6
```

MOV @ R_i,A Move Accumulator Contents to Data Memory

Encoding: | 1 0 1 0 | 0 0 0 i | A0H-A1H

Description: The contents of the accumulator are moved to the resident data memory location whose address is specified by bits 0-5** of register 'i'. Register 'i' contents are unaffected.

Operation: ((Ri)) ← (A) i = 0-1

Example: Assume R0 contains 00000111.
```
MDMA:  MOV @R0,A           ;MOVE CONTENTS OF ACC TO
                           ;LOCATION 7 (REG 7)
```

MOV @ R_i,#data Move Immediate Data to Data memory

Encoding: | 1 0 1 1 | 0 0 0 i | | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | B0H-B1H

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to the resident data memory location addressed by register 'i', bits 0-5**.

Operation: ((Ri)) ← data i = 0-1

Examples: Move the hexadecimal value AC3F to locations 62-63.
```
MIDM:  MOV R0,#62          ;MOVE '62' DEC TO ADDR REG 0
       MOV @R0,#0ACH       ;MOVE 'AC' HEX TO LOCATION 62
       INC R0              ;INCREMENT REG 0 to '63'
       MOV @R0,#3FH        ;MOVE '3F' HEX TO LOCATION 63
```

** 0-5 in 8048AH/8748H
0-6 in 8049AH/8749H
0-7 in 8050AH

MOV T,A Move Accumulator Contents to Timer/Counter

Encoding: | 0 1 1 0 | 0 0 1 0 | 62H

Description: The contents of the accumulator are moved to the timer/event-counter register.

Operation: (T) ← (A)

Example: Initialize and start event counter.

```
INITEC: CLR A          ;CLEAR ACC TO ZEROS
        MOV T,A        ;MOVE ZEROS TO EVENT COUNTER
        START CNT      ;START COUNTER
```

MOVD A,Pp Move Port 4-7 Data to Accumulator

Encoding: | 0 0 0 0 | 1 1 p p | 0CH-0FH

Description: This is a 2-cycle instruction. Data on 8243 port 'p' is moved (read) to accumulator bits 0-3. Accumulator bits 4-7 are zeroed.

Operation: (0-3) ← (Pp) p = 4-7
(4-7) ← 0

Note: Bits 0-7 of the opcode are used to represent ports 4-7. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits 1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example:

```
INPPT5: MOVD A,P5      ;MOVE PORT 5 DATA TO ACC
                       ;BITS 0-3, ZERO ACC BITS 4-7
```

MOVD Pp,A Move Accumulator Data to Port 4-7

Encoding: | 0 0 1 1 | 1 1 p p | 3CH-3FH

Description: This is a 2-cycle instruction. Data in accumulator bits 0-3 is moved (written) to 8243 port 'p'. Accumulator bits 4-7 are unaffected. (See NOTE above regarding port mapping.)

Operation: (Pp) ← (A_{0-3}) P = 4-7

Example: Move data in accumulator to ports 4 and 5.

```
OUTP45: MOVD P4,A      ;MOVE ACC BITS 0-3 TO PORT 4
        SWAP A         ;EXCHANGE ACC BITS 0-3 and 4-7
        MOVD P5,A      ;MOVE ACC BITS 0-3 TO PORT 5
```

MOVP A,@A Move Current Page Data to Accumulator

Encoding: | 1 0 1 0 | 0 0 1 1 | A3H

Description: The contents of the program memory location addressed by the accumulator are moved to the accumulator. Only bits 0-7 of the program counter are affected, limiting the program memory reference to the current page. The program counter is restored *following* this operation.

Operation: $(PC_{0-7}) \leftarrow (A)$
$(A) \leftarrow ((PC))$
Note: This is a 1-byte, 2-cycle instruction. If it appears in location 255 of a program memory page, @A addresses a location in the *following* page.

Example:
```
MOV128: MOV A,#128      ;MOVE '128' DEC TO ACC
        MOVP A,@A       ;CONTENTS OF 129th LOCATION IN
                        ;CURRENT PAGE ARE MOVED TO ACC
```

MOVP3 A,@A Move Page 3 Data to Accumulator

Encoding: | 1 1 1 0 | 0 0 1 1 | E3H

Description: This is a 2-cycle instruction. The contents of the program memory location (within page 3) addressed by the accumulator are moved to the accumulator. The program counter is restored following this operation.

Operation: $(PC_{0-7}) \leftarrow (A)$
$(PC_{8-11}) \leftarrow 0011$
$(A) \leftarrow ((PC))$

Example: Look up ASCII equivalent of hexadecimal code in table contained at the beginning of page 3. Note that ASCII characters are designated by a 7-bit code; the eighth bit is always reset.

```
TABSCH: MOV A,#0B8H     ;MOVE 'B8' HEX TO ACC (10111000)
        ANL A,#7FH      ;LOGICAL AND ACC TO MASK BIT
                        ;7 (00111000)
        MOVP3 A,@A      ;MOVE CONTENTS OF LOCATION '38'
                        ;HEX IN PAGE 3 TO ACC (ASCII '8')
```

Access contents of location in page 3 labelled TAB1.
Assume current program location is not in page 3.

```
TABSCH: MOV A,#LOW TAB 1 ;ISOLATE BITS 0-7 OF LABEL
                         ;ADDRESS VALUE
        MOVP3 A,@A       ;MOVE CONTENTS OF PAGE 3
                         ;LOCATION LABELED 'TAB1' TO ACC
```

MOVX A,@R_i Move External-Data-Memory Contents to Accumulator

Encoding: | 1 0 0 0 | 0 0 0 i | 80H-81H

Description: This is a 2-cycle instruction. The contents of the external data memory location addressed by register 'i' are moved to the accumulator. Register 'i' contents are unaffected. A read pulse is generated.

Operation: (A) ← ((Ri)) i = 0-1

Example: Assume R1 contains 01110110.

```
MAXDM:  MOVX A,@R1      ;MOVE CONTENTS OF LOCATION
                        ;118 TO ACC
```

MOVX @R_i,A Move Accumulator Contents to External Data Memory

Encoding: | 1 0 0 1 | 0 0 0 i | 90H-91H

Description: This is a 2-cycle instruction. The contents of the accumulator are moved to the external data memory location addressed by register 'i'. Register 'i' contents are unaffected. A write pulse is generated.

Operation: ((Ri)) ← A i = 0-1

Example: Assume R0 contains 11000111.

```
MXDMA:  MOVX @R0,A      ;MOVE CONTENTS OF ACC TO
                        ;LOCATION 199 IN EXPANDED
                        ;DATA MEMORY
```

NOP The NOP Instruction

Encoding: | 0 0 0 0 | 0 0 0 0 | 00H

Description: No operation is performed. Execution continues with the following instruction.

ORL A,R_r Logical OR Accumulator With Register Mask

Encoding: | 0 1 0 0 | 1 r r r | 48H-4FH

Description: Data in the accumulator is logically ORed with the mask contained in working register 'r'.

Operation: (A) ← (A) OR (Rr) r = 0-7

Example:

```
ORREG:  ORL A,R4        ;'OR' ACC CONTENTS WITH
                        ;MASK IN REG 4
```

ORL A,@R_i Logical OR Accumulator With Memory Mask

Encoding: | 0 1 0 0 | 0 0 0 i | 40H-41H

Description: Data in the accumulator is logically ORed with the mask contained in the resident data memory location referenced by register "i", bits 0–5**.

Operation: (A) ← (A) OR ((Ri)) i = 0-1

Example:
```
ORDM:  MOV R0,#3FH      ;MOVE '3F' HEX TO REG 0
       ORL A,@R0        ;'OR' AC CONTENTS WITH MASK
                        ;IN LOCATION 63
```

ORL A,#data Logical OR Accumulator With Immediate Mask

Encoding: | 0 1 0 0 | 0 0 1 1 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 43H

Description: This is a 2-cycle instruction. Data in the accumulator is logically ORed with an immediately-specified mask.

Operation: (A) ← (A) OR data

Example:
```
ORID:  ORL A,#'X'       ;'OR' ACC CONTENTS WITH MASK
                        ;01011000 (ASCII VALUE OF 'X')
```

ORL BUS,#data* Logical OR BUS With Immediate Mask

Encoding: | 1 0 0 0 | 1 0 0 0 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 88H

Description: This is a 2-cycle instruction. Data on the BUS port is logically ORed with an immediately-specified mask. This instruction assumes prior specification on an 'OUTL BUS,A' instruction.

Operation: (BUS) ← (BUS) OR data

Example:
```
ORBUS: ORL BUS,#HEXMSK  ;'OR' BUS CONTENTS WITH MASK
                        ;EQUAL VALUE OF SYMBOL 'HEXMSK'
```

ORL Pp, #data Logical OR Port 1 or 2 With Immediate Mask

Encoding: | 1 0 0 0 | 1 0 p p | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 89H-8AH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ORed with an immediately-specified mask.

Operation: (Pp) ← (Pp) OR data p = 1-2

Example:
```
ORP1:  ORL P1, #0FFH    ;'OR' PORT 1 CONTENTS WITH MASK
                        ;'FF' HEX (SET PORT 1 TO ALL ONES)
```

* For use with internal program memory ONLY.

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

ORLD Pp,A Logical OR Port 4-7 With Accumulator Mask

Encoding: | 1 0 0 0 | 1 1 p p | 8CH-8FH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ORed with the digit mask contained in accumulator bits 0-3.

Operation: (Pp) ← (Pp) OR (A_{0-3}) p = 4-7

Example:
```
ORP7: ORLD P7,A      ;'OR' PORT 7 CONTENTS WITH ACC
                     ;BITS 0-3
```

OUTL BUS,A* Output Accumulator Data to BUS

Encoding: | 0 0 0 0 | 0 0 1 0 | 02H

Description: This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to the BUS port and latched. The latched data remains valid until altered by another OUTL instruction. Any other instruction requiring use of the BUS port (except INS) destroys the contents of the BUS latch. This includes expanded memory operations (such as the MOVX instruction). Logical operations on BUS data (AND, OR) assume the OUTL BUS,A instruction has been issued previously.

Operation: (BUS) ← (A)

Example:
```
OUTLBP: OUTL BUS, A      ;OUTPUT ACC CONTENTS TO BUS
```

OUTL Pp,A Output Accumulator Data to Port 1 or 2

Encoding: | 0 0 1 1 | 1 0 p p | 39H-3AH

Description: This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to port 'p' and latched.

Operation: (Pp) ← (A) p = 1-2

Example:
```
OUTLP: MOV A,R7      ;MOVE REG 7 CONTENTS TO ACC
       OUTL P2,A     ;OUTPUT ACC CONTENTS TO PORT 2
       MOV A, R6     ;MOV REG 6 CONTENTS TO ACC
       OUTL P1,A     ;OUTPUT ACC CONTENTS TO PORT 1
```

* For use with internal program memory ONLY.

RET Return Without PSW Restore

Encoding: | 1 0 0 0 | 0 0 1 1 | 83H

Description: This is a 2-cycle instruction. The stack pointer (PSW bits 0-2) is decremented. The program counter is then restored from the stack. PSW bits 4-7 are not restored.

Operation: (SP) ← (SP)-1
(PC) ← ((SP))

RETR Return with PSW Restore

Encoding: | 1 0 0 1 | 0 0 1 1 | 93H

Description: This is a 2-cycle instruction. The stack pointer is decremented. The program counter and bits 4-7 of the PSW are then restored from the stack. Note that RETR should be used to return from an interrupt, but should not be used within the interrupt service routine as it signals the end of an interrupt routine by resetting the Interrupt in Progress flip-flop.

Operation: (SP) ← (SP)-1
(PC) ← ((SP))
(PSW 4-7) ← ((SP))

RL A Rotate Left without Carry

Encoding: | 1 1 1 0 | 0 1 1 1 | E7H

Description: The contents of the accumulator are rotated left one bit. Bit 7 is rotated into the bit 0 position.

Operation: (An + 1) ← (An)
(A0) ← (A7) n = 0-6

Example: Assume accumulator contains 10110001.

```
RLNC: RL A          ;NEW ACC CONTENTS ARE 01100011
```

RLC A Rotate Left through Carry

Encoding: | 1 1 1 1 | 0 1 1 1 | F7H

Description: The contents of the accumulator are rotated left one bit. Bit 7 replaces the carry bit; the carry bit is rotated into the bit 0 position.

Operation: (An + 1) ← (An)
n = 0-6
(A0) ← (C)
(C) ← (A7)

Example: Assume accumulator contains a 'signed' number; isolate sign without changing value.

```
RLTC: CLR C         ;CLEAR CARRY TO ZERO
      RLC A         ;ROTATE ACC LEFT, SIGN
                    ;BIT(7) IS PLACED IN CARRY
      RR A          ;ROTATE ACC RIGHT — VALUE
                    ;(BITS 0-6) IS RESTORED,
                    ;CARRY UNCHANGED, BIT 7
                    ;IS ZERO
```

RR A Rotate Right without Carry

Encoding: | 0 1 1 1 | 0 1 1 1 | 77H

Description: The contents of the accumulator are rotated right one bit. Bit 0 is rotated into the bit 7 position.

Operation: (An) ← (An + 1) n = 0-6
(A7) ← (A0)

Example: Assume accumulator contains 10110001.

```
RRNC: RR A          ;NEW ACC CONTENTS ARE 11011000
```

RRC A Rotate Right through Carry

Encoding: | 0 1 1 0 | 0 1 1 1 | 67H

Description: The contents of the accumulator are rotated right one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 7 position.

Operation: (An) ← (An + 1) n = 0-6
(A7) ← (C)
(C) ← (A_0)

Example: Assume carry is not set and accumulator contains 10110001.

```
RRTC: RRC A          ;CARRY IS SET AND ACC
                     ;CONTAINS 01011000
```

SEL MB0 Select Memory Bank 0

Encoding: | 1 1 1 0 | 0 1 0 1 | E5H

Description: PC bit 11 is set to zero on next JMP or CALL instruction. All references to program memory addresses fall within the range 0-2047.

Operation: (DBF) ← 0

Example: Assume program counter contains 834 Hex.

```
SEL MB0              ;SELECT MEMORY BANK 0
JMP $+20             ;JUMP TO LOCATION 58 HEX
```

SEL MB1 Select Memory Bank 1

Encoding: | 1 1 1 1 | 0 1 0 1 | F5H

Description: PC bit 11 is set to one on next JMP or CALL instruction. All references to program memory addresses fall within the range 2048-4095.

Operation: (DBF) ← 1

SEL RB0 Select Register Bank 0

Encoding: | 1 1 0 0 | 0 1 0 1 | C5H

Description: PSW bit 4 is set to zero. References to working registers 0-7 address data memory locations 0-7. This is the recommended setting for normal program execution.

Operation: (BS) ← 0

SEL RB1 Select Register Bank 1

Encoding: | 1 1 0 1 | 0 1 0 1 | D5H

Description: PSW bit 4 is set to one. References to working registers 0-7 address data memory locations 24-31. This is the recommended setting for interrupt service routines, since locations 0-7 are left intact. The setting of PSW bit 4 in effect at the time of an interrupt is restored by the RETR instruction when the interrupt service routine is completed.

Operation: (BS) ← 1

Example: Assume an external interrupt has occurred, control has passed to program memory location 3, and PSW bit 4 was zero before the interrupt.

Operation:

```
LOC3: JNI INIT          ;JUMP TO ROUTINE 'INIT' IF
                        ;INTERRUPT INPUT IS ZERO
INIT: MOV R7,A          ;MOVE ACC CONTENTS TO
                        ;LOCATION 7
      SEL RB1           ;SELECT REG BANK 1
      MOV R7,#0FAH      ;MOVE 'FA' HEX TO LOCATION 31
      .
      .
      .
      SEL RB0           ;SELECT REG BANK 0
      MOV A,R7          ;RESTORE ACC FROM LOCATION 7
      RETR              ;RETURN — RESTORE PC AND PSW
```

STOP TCNT Stop Timer/Event-Counter

Encoding: | 0 1 1 0 | 0 1 0 1 | 65H

Description: This instruction is used to stop both time accumulation and event counting.

Example: Disable interrupt, but jump to interrupt routine after eight overflows and stop timer. Count overflows in register 7.

```
START: DIS TCNTI          ;DISABLE TIMER INTERRUPT
       CLR A              ;CLEAR ACC TO ZEROS
       MOV T,A            ;MOVE ZEROS TO TIMER
       MOV R7,A           ;MOVE ZEROS TO REG 7
       STRT T             ;START TIMER
MAIN:  JTF COUNT          ;JUMP TO ROUTINE 'COUNT'
                          ;IF TF = 1 AND CLEAR TIMER FLAG
       JMP MAIN           ;CLOSE LOOP
COUNT: INC R7             ;INCREMENT REG 7
       MOV A,R7           ;MOVE REG 7 CONTENTS TO ACC
       JB3 INT            ;JUMP TO ROUTINE 'INT' IF ACC
                          ;BIT 3 IS SET (REG 7 = 8)
       JMP MAIN           ;OTHERWISE RETURN TO ROUTINE
                          ;MAIN
        .
        .
        .
INT:   STOP TCNT          ;STOP TIMER
       JMP 7H             ;JUMP TO LOCATION 7 (TIMER)
                          ;INTERRUPT ROUTINE
```

STRT CNT Start Event Conter

Encoding: | 0 1 0 0 | 0 1 0 1 | 45H

Description: The test 1 (T1) pin is enabled as the event-counter input and the counter is started. The event-counter register is incremented with each high-to-low transition on the T1 pin.

Example: Initialize and start event counter. Assume overflow is desired with first T1 input.

```
STARTC: EN TCNTI          ;ENABLE COUNTER INTERRUPT
        MOV A,#0FFH       ;MOVE 'FF'HEX (ONES) TO ACC
        MOV T,A           ;MOVES ONES TO COUNTER
        STRT CNT          ;ENABLE T1 AS COUNTER
                          ;INPUT AND START
```

STRT T Start Timer

Encoding: | 0 1 0 1 | 0 1 0 1 | 55H

Description: Timer accumulation is initiated in the timer register. The register is incremented every 32 instruction cycles. The prescaler which counts the 32 cycles is cleared but the timer register is not.

Example: Initialize and start timer.

```
STARTT: CLR A          ;CLEAR ACC TO ZEROS
        MOV T,A        ;MOVE ZEROS TO TIMER
        EN TCNTI       ;ENABLE TIMER INTERRUPT
        STRT T         ;START TIMER
```

SWAP A Swap Nibbles within Accumulator

Encoding: | 0 1 0 0 | 0 1 1 1 | 47H

Description: Bits 0-3 of the accumulator are swapped with bits 4-7 of the accumulator.

Operation: $(A_{4-7}) \rightleftarrows (A_{0-3})$

Example: Pack bits 0-3 of locations 50-51 into location 50.

```
PCKDIG: MOV R0, #50     ;MOVE '50' DEC TO REG 0
        MOV R1, #51     ;MOVE '51' DEC TO REG 1
        XCHD A,@R0      ;EXCHANGE BITS 0-3 OF ACC
                        ;AND LOCATION 50
        SWAP A          ;SWAP BITS 0-3 AND 4-7 OF ACC
        XCHD A,@R1      ;EXCHANGE BITS 0-3 OF ACC AND
                        ;LOCATION 51
        MOV @R0,A       ;MOVE CONTENTS OF ACC TO
                        ;LOCATION 50
```

XCH A,R_r Exchange Accumulator-Register Contents

Encoding: | 0 0 1 0 | 1 r r r | 28H-2FH

Description: The contents of the accumulator and the contents of working register 'r' are exchanged.

Operation: (A) $\rightleftarrows$ (Rr) r = 0-7

Example: Move PSW contents to Reg 7 without losing accumulator contents.

```
XCHAR7: XCH A,R7        ;EXCHANGE CONTENTS OF REG 7
                        ;AND ACC
        MOV A, PSW      ;MOVE PSW CONTENTS TO ACC
        XCH A,R7        ;EXCHANGE CONTENTS OF REG 7
                        'AND ACC AGAIN
```

XCH A,@R_i Exchange Accumulator and Data Memory Contents

Encoding: | 0 0 1 0 | 0 0 0 i | 20H-21H

Description: The contents of the accumulator and the contents of the resident data memory location addressed by bits 0-5** of register 'i' are exchanged. Register 'i' contents are unaffected.

Operation: (A) ⇆ ((Ri)) i = 0-1

Example: Decrement contents of location 52.

```
DEC52: MOV R0,#52      ;MOVE '52' DEC TO ADDRESS REG 0
       XCH A,@R0       ;EXCHANGE CONTENTS OF ACC
                       ;AND LOCATION 52
       DEC A           ;DECREMENT ACC CONTENTS
       XCH A,@R0       ;EXCHANGE CONTENTS OF ACC
                       ;AND LOCATION 52 AGAIN
```

XCHD A,@R_i Exchange Accumulator and Data Memory 4-Bit Data

Encoding: | 0 0 1 1 | 0 0 0 i | 30H-31H

Description: This instruction exchanges bits 0-3 of the accumulator with bits 0-3 of the data memory location addressed by bits 0-5** of register 'i'. Bits 4-7 of the accumulator, bits 4-7 of the data memory location, and the contents of register 'i' are unaffected.

Operation: (A_{0-3}) ⇆ ((Ri0-3)) i = 0-1

Example: Assume program counter contents have been stacked in locations 22-23.

```
XCHNIB: MOV R0,#23     ;MOVE '23' DEC TO REG 0
        CLR A          ;CLEAR ACC TO ZEROS
        XCHD A,@R0     ;EXCHANGE BITS 0-3 OF ACC AND
                       ;LOCATION 23 (BTS 8-11 OF PC ARE
                       ;ZEROED, ADDRESS REFERS
                       :TO PAGE 0)
```

XRL A,R_r Logical XOR Accumulator With Register Mask

Encoding: | 1 1 0 1 | 1 r r r | D8H-DFH

Description: Data in the accumulator is EXCLUSIVE ORed with the mask contained in working register 'r'.

Operation: (A) ← (A) XOR (Rr) r = 0-7

Example:

```
XORREG: XRL A,R5        ;'XOR' ACC CONTENTS WITH
                        ;MASK IN REG 5
```

** 0-5 in 8048AH/8748H
0-6 in 8049AH/8749H
0-7 in 8050AH

XRL A,@R_i Logical XOR Accumulator With Memory Mask

Encoding: | 1 1 0 1 | 0 0 0 i | D0H-D1H

Description: Data in the accumulator is EXCLUSIVE ORed with the mask contained in the data memory location addressed by register 'i', bits 0–5.**

Operation: (A) ← (A) XOR ((Ri)) i = 0-1

Example:
```
XORDM: MOV R1,#20H      ;MOVE '20' HEX TO REG 1
       XRL A,@R1        ;'XOR' ACC CONTENTS WITH MASK
                        ;IN LOCATION 32
```

XRL A,#data Logical XOR Accumulator With Immediate Mask

Encoding: | 1 1 0 1 | 0 0 1 1 | | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | D3H

Description: This is a 2-cycle instruction. Data in the accumulator is EXCLUSIVE ORed with an immediately-specified mask.

Operation: (A) ← (A) XOR data

Example:
```
XORID: XOR A,#HEXTEN    ;XOR CONTENTS OF ACC WITH MASK
                        ;EQUAL VALUE OF SYMBOL 'HEXTEN'
```

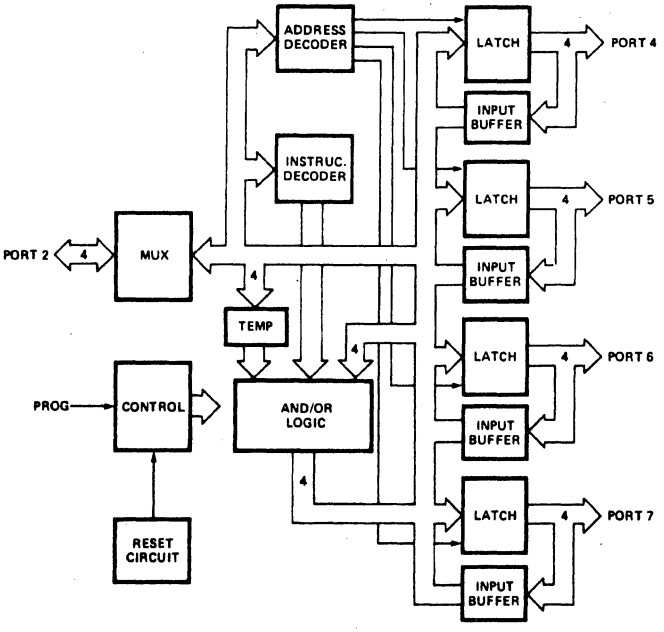
** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

MCS®-48 Data Sheets

4

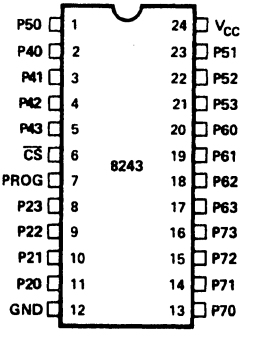
8243
MCS®-48 INPUT/OUTPUT EXPANDER

■ 0°C TO 70°C Operation



270161-1

Figure 1. 8243 Block Diagram



270161-2

Figure 2. 8243 Pin Configuration

October 1986
Order Number: 270161-001

Table 1. Pin Description

Symbol	Pin No.	Function
PROG	7	Clock Input. A high to low transition on PROG signifies that address and control are available on P20–P23, and a low to high transition signifies that data is available on P20–P23.
$\overline{CS}$	6	Chip Select Input. A high on CS inhibits any change of output or internal status.
P20–P23	11–8	Four (4) bit bi-directional port contains the address and control bits on a high to low transition of PROG. During a low to high transition, P2 contains the data for a selected output port if a write operation, or the data from a selected port before the low to high transition if a read operation.
GND	12	0V supply.
P40–P43	2–5	Four (4) bit bi-directional I/O ports.
P50–P53 P60–P63 P70–P73	1, 23–21 20–17 13–16	May be programmed to be input (during read), low impedance latched output (after write), or a tri-state (after read). Data on pins P20–P23 may be directly written, ANDed or ORed with previous data.
V_{CC}	24	+5V supply.

FUNCTIONAL DESCRIPTION

General Operation

The 8243 contains four 4-bit I/O ports which serve as an extension of the on-chip I/O and are addressed as Ports 4–7. The following operations may be performed on these ports:

- Transfer Accumulator to Port.
- Transfer Port to Accumulator.
- AND Accumulator to Port.
- OR Accumulator to Port.

All communication between the 8048 and the 8243 occurs over Port 2 (P20–P23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles:

The first containing the "op code" and port address and the second containing the actual 4-bits of data. A high to low transition of the PROG line indicates that address is present while a low to high transition indicates the presence of data. Additional 8243's may be added to the 4-bit bus and chip selected using additional output lines from the 8048/8748/8035.

Power On Initialization

Initial application of power to the device forces input/output Ports 4, 5, 6, and 7 to the tri-state and Port 2 to the input mode. The PROG pin may be either high or low when power is applied. The first high to low transition of PROG causes the device to exit power on mode. The power on sequence is initiated if V_{CC} drops below 1V.

P21	P20	Address Code	P23	P22	Instruction Code
0	0	Port 4	0	0	Read
0	1	Port 5	0	1	Write
1	0	Port 6	1	0	ORLD
1	1	Port 7	1	1	ANLD

Write Modes

The device has three write modes. MOVD Pi, A directly writes new data into the selected port and old data is lost. ORLD Pi, A takes new data, OR's it with the old data and then writes it to the port. ANLD Pi, A takes new data, AND's it with the old data and then writes it to the port. Operation code and port address are latched from the input Port 2 on the high to low transition of the PROG pin. On the low to high transition of PROG data on Port 2 is transferred to the logic block of the specified output port.

After the logic manipulation is performed, the data is latched and outputed. The old data remains latched until new valid outputs are entered.

Read Mode

The device has one read mode. The operation code and port address are latched from the input Port 2

on the high to low transition of the PROG pin. As soon as the read operation and port address are decoded, the appropriate outputs are tri-stated, and the input buffers switched on. The read operation is terminated by a low to high transition of the PROG pin. The port (4, 5, 6 or 7) that was selected is switched to the tri-stated mode while Port 2 is returned to the input mode.

Normally, a port will be in an output (write mode) or input (read mode). If modes are changed during operation, the first read following a write should be ignored; all following reads are valid. This is to allow the external driver on the port to settle after the first read instruction removes the low impedance drive from the 8243 output. A read of any port will leave that port in a high impedance state.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin
with Respect to Ground.....−0.5V to +7V
Power Dissipation .....1 Watt

Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 10%

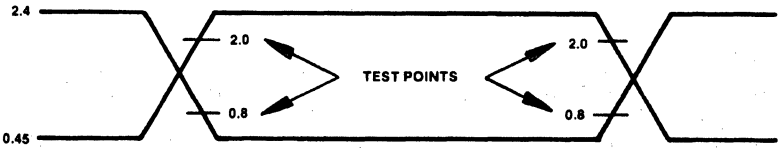
Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC} + 0.5	V	
V_{OL1}	Output Low Voltage Ports 4–7			0.45	V	I_{OL} = 4.5 mA*
V_{OL2}	Output Low Voltage Port 7			1	V	I_{OL} = 20 mA
V_{OH1}	Output High Voltage Ports 4–7	2.4			V	I_{OH} = 240 μA
I_{IL1}	Input Leakage Ports 4–7	−10		20	μA	V_{in} = V_{CC} to 0V
I_{IL2}	Input Leakage Port 2, CS, PROG	−10		10	μA	V_{in} = V_{CC} to 0V
V_{OL3}	Output Low Voltage Port 2			0.45	V	I_{OL} = 0.6 mA
I_{CC}	V_{CC} Supply Current		10	20	mA	(Note 1)
V_{OH2}	Output Voltage Port 2	2.4				I_{OH} = 100 μA
I_{OL}	Sum of All I_{OL} From 16 Outputs			72	mA	4.5 mA Each Pin

*Refer to Figure 3 for additional sink current capability.

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 10%

Symbol	Parameter	Min	Max	Units	Test Conditions
t_A	Code Valid before PROG	50		ns	80 pF Load
t_B	Code Valid after PROG	60		ns	20 pF Load
t_C	Data Valid before PROG	200		ns	80 pF Load
t_D	Data Valid after PROG	20		ns	20 pF Load
t_H	Floating after PROG	0	150	ns	20 pF Load
t_K	PROG Negative Pulse Width	700		ns	
t_{CS}	CS Valid before/after PROG	50		ns	
t_{PO}	Ports 4–7 Valid after PROG		700	ns	100 pF Load
t_{LP1}	Ports 4–7 Valid before/after PROG	100		ns	
t_{ACC}	Port 2 Valid after PROG		650	ns	80 pF Load

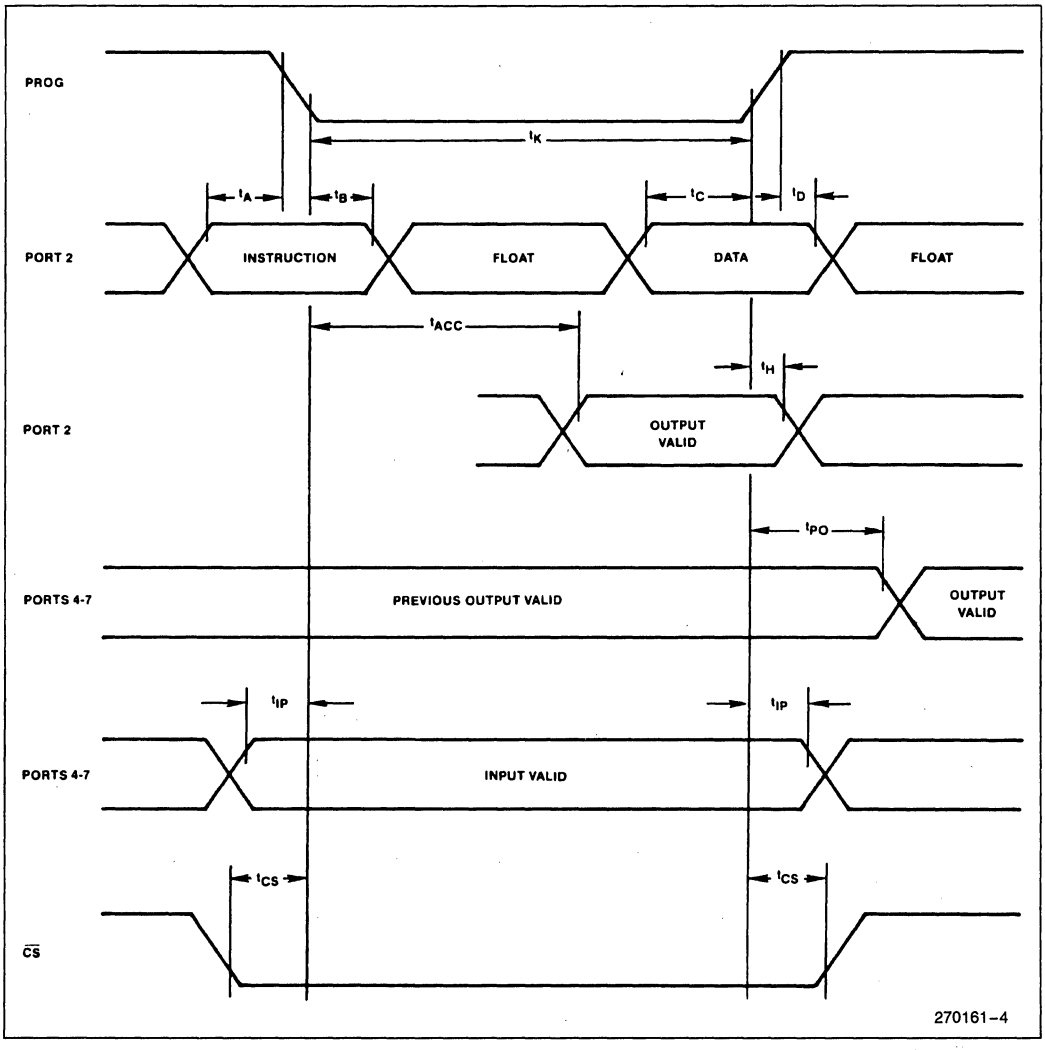
NOTE:
1. I_{CC} (−40°C to 85°C EXPRESS options) 15 mA typical/25 mA maximum.



270161-3

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Output timing measurements are made at 2.0V for Logic "1" and 0.8V for a Logic "0".

WAVEFORMS


PROG
t_K
t_A
t_B
t_C
t_D
PORT 2
INSTRUCTION
FLOAT
DATA
FLOAT
t_{ACC}
t_H
PORT 2
OUTPUT VALID
t_{PO}
PORTS 4-7
PREVIOUS OUTPUT VALID
OUTPUT VALID
t_{IP}
t_{IP}
PORTS 4-7
INPUT VALID
t_{CS}
t_{CS}
$\overline{CS}$
270161-4

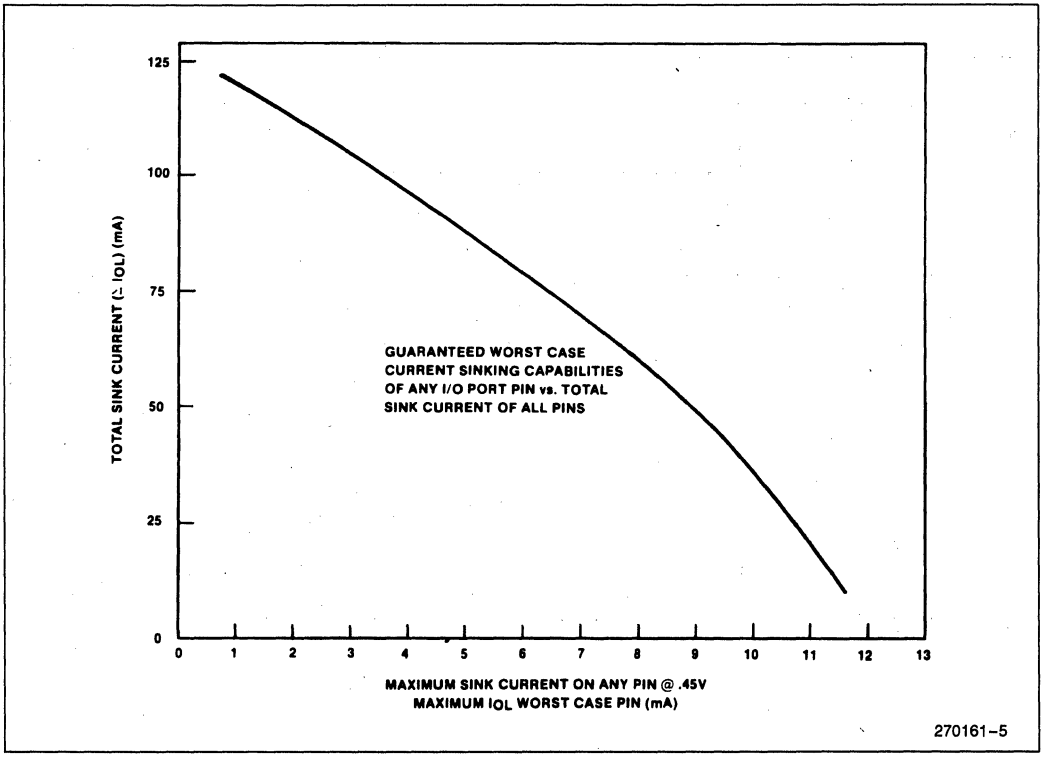


Figure 3. 8243 Current Sink Capability

Sink Capability

The 8243 can sink 5 mA @ 0.45V on each of its 16 I/O lines simultaneously. If, however, all lines are not sinking simultaneously or all lines are not fully loaded, the drive capability of any individual line increases as is shown by the accompanying curve.

For example, if only 5 of the 16 lines are to sink current at one time, the curve shows that each of those 5 lines is capable of sinking 9 mA @ 0.45V (if any lines are to sink 9 mA the total I_{OL} must not exceed 45 mA or five 9 mA loads).

Example: How many pins can drive 5 TTL loads (1.6 mA) assuming remaining pins are unloaded?

$I_{OL} = 5 \times 1.6\ mA = 8\ mA$

$\epsilon I_{OL} = 60$ mA from curve

pins = 60 mA ÷ 8 mA/pin = 7.5 = 7

In this case, 7 lines can sink 8 mA for a total of 56 mA. This leaves 4 mA sink current capability which can be divided in any way among the remaining 8 I/O lines of the 8243.

NOTE:

A10 to 50 KΩ pullup resistor to +5V should be added to 8243 outputs when driving to 5V CMOS directly.

Example: This example shows how the use of the 20 mA sink capability of Port 7 affects the sinking capability of the other I/O lines.

An 8243 will drive the following loads simultaneously.

2 loads—20 mA @ 1V (Port 7 only)

8 loads—4 mA @ 0.45V

6 loads—3.2 mA @ 0.45V

Is this within the specified limits?

$\epsilon I_{OL} = (2 \times 20) + (8 \times 4) + (6 \times 3.2) = 91.2$ mA.

From the curve: for $I_{OL} = 4$ mA, $\epsilon I_{OL} \cong 93$ mA. Since 91.2 mA < 93 mA the loads are within specified limits.

Although the 20 mA @ 1V loads are used in calculating ϵI_{OL}, it is the largest current required @ 0.45V which determines the maximum allowable ϵI_{OL}.

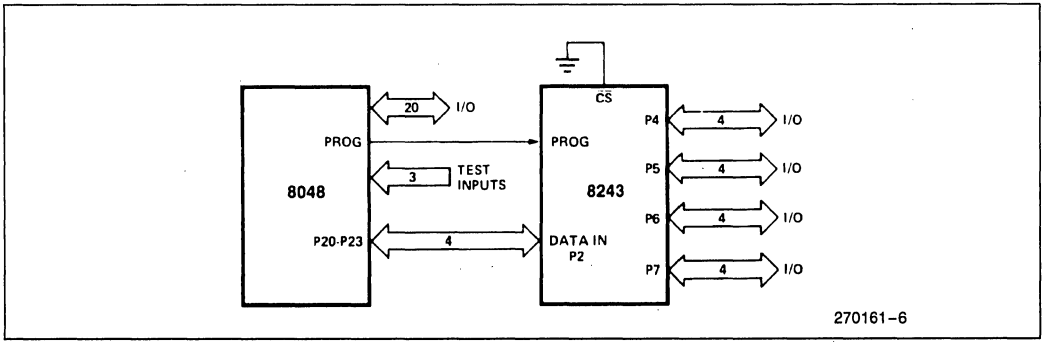


Figure 4. Expander Interface

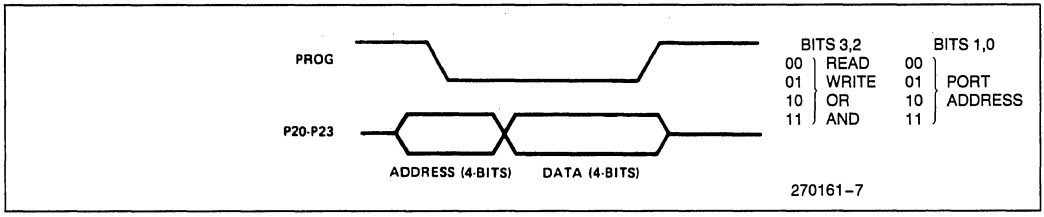


Figure 5. Output Expander Timing

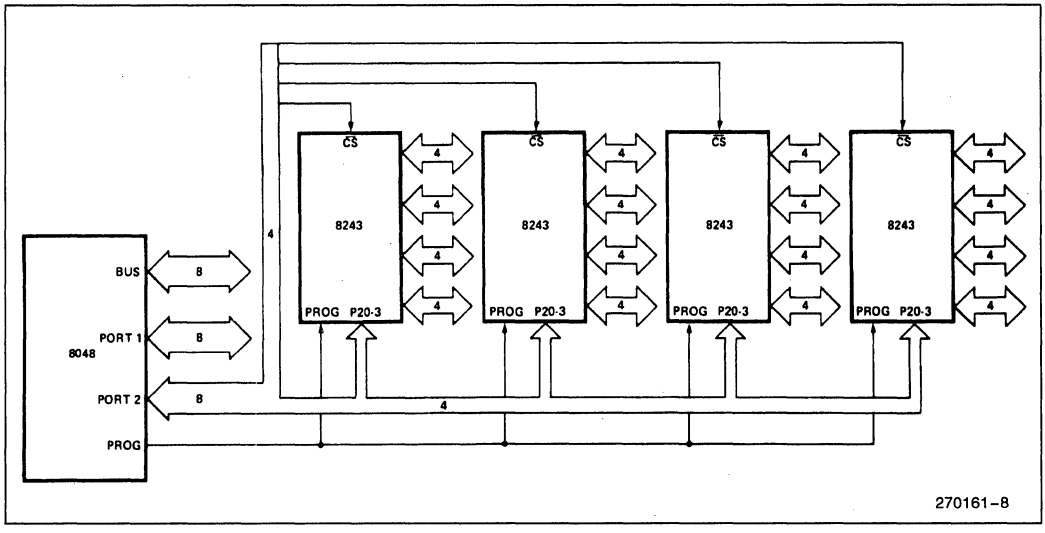


Figure 6. Using Multiple 8243's

P8748H/P8749H
8048AH/8035AHL/8049AH/8039AHL/8050AH/8040AHL
HMOS SINGLE-COMPONENT 8-BIT PRODUCTION MICROCONTROLLER

- **High Performance HMOS II**
- **Interval Time/Event Counter**
- **Two Single Level Interrupts**
- **Single 5-Volt Supply**
- **Over 96 Instructions; 90% Single Byte**
- **Programmable ROMs Using 21V**
- **Easily Expandable Memory and I/O**
- **Up to 1.36 μs Instruction Cycle All Instructions 1 or 2 Cycles**

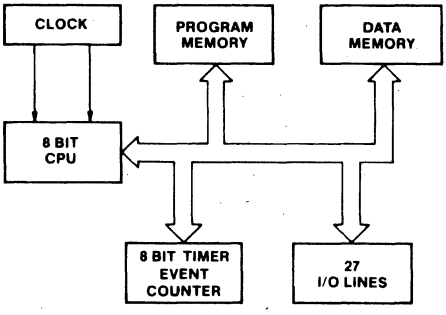
The Intel MCS®-48 family are totally self-sufficient, 8-bit parallel computers fabricated on single silicon chips using Intel's advanced N-channel silicon gate HMOS process.

The family contains 27 I/O lines, an 8-bit timer/counter, and on-board oscillator/clock circuits. For systems that require extra capability, the family can be expanded using MCS®-80/MCS®-85 peripherals.

These microcontrollers are available in both masked ROM and ROMless versions as well as a new version, The Programmable ROM. The Programmable ROM provides the user with the capability of a masked ROM while providing the flexibility of a device that can be programmed at the time of requirement and to the desired data. Programmable ROM's allow the user to lower inventory levels while at the same time decreasing delay times and code risks.

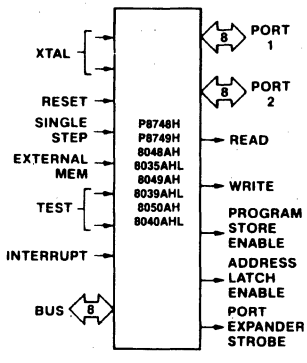
These microcomputers are designed to be efficient controllers as well as arithmetic processors. They have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting of mostly single byte instructions and no instructions over 2 bytes in length.

Device	Internal	Memory	RAM STANDBY
8050AH	4K x 8 ROM	256 x 8 RAM	yes
8049AH	2K x 8 ROM	128 x 8 RAM	yes
8048AH	1K x 8 ROM	64 x 8 RAM	yes
8040AHL	None	256 x 8 RAM	yes
8039AHL	None	128 x 8 RAM	yes
8035AHL	None	64 x 8 RAM	yes
P8749H	2K x 8 Programmable ROM	128 x 8 RAM	no
P8748H	1K x 8 Programmable ROM	64 x 8 RAM	no



270053-1

Figure 1. Block Diagram



270053-2

Figure 2. Logic Symbol

November 1987
Order Number: 270053-002

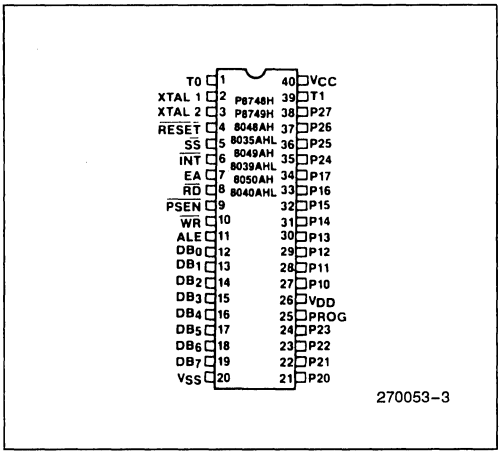


Figure 3. Pin Configuration

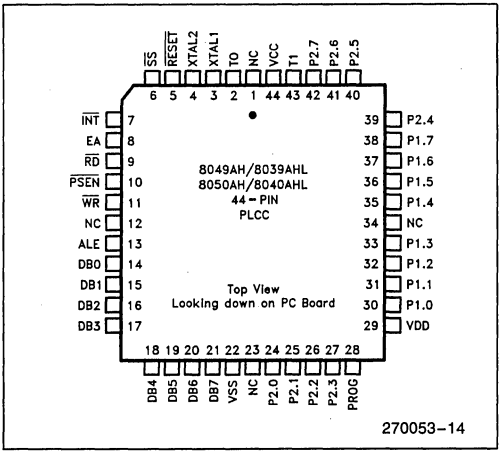


Figure 4. Pad Configuration

Table 1. Pin Description

Symbol	Pin No.	Function	Device
V_{SS}	20	Circuit GND potential.	All
V_{DD}	26	+5V during normal operation.	All
		Low power standby pin.	8048AH 8035AHL 8049AH 8039AHL 8050AH 8040AHL
		Programming power supply (+21V).	P8748H P8749H
V_{CC}	40	Main power supply; +5V during operation and programming.	All
PROG	25	Output strobe for 8243 I/O expander.	All
		Program pulse (+18V) input pin During Programming.	P8748H P8749H
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.	All
P20-P23 P24-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.	All
DB0-DB7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.	All
T0	1	Input pin testable using the conditional transfer instruction JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.	All
		Used during programming.	P8748H P8749H

Table 1. Pin Description (Continued)

Symbol	Pin No.	Function	Device
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.	All
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low) interrupt must remain low for at least 3 machine cycles for proper operation.	All
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device. Used as a read strobe to external data memory. (Active low)	All
$\overline{RESET}$	4	Input which is used to initialize the processor. (Active low) (Non TTL V_{IH})	All
		Used during power down.	8048AH 8035AHL 8049AH 8039AHL 8050AH 8040AHL
		Used during programming.	P8748H P8749H
		Used during ROM verification.	8048AH P8748H 8049AH P8749H 8050AH
$\overline{WR}$	10	Output strobe during a bus write. (Active low) Used as write strobe to external data memory.	All
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.	All
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low)	All
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction.	All
		(Active low) Used in sync mode.	8048AH 8035AHL 8049AH 8039AHL 8050AH 8040AHL
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug. (Active high)	All
		Used during (18V) programming.	P8748H P8749H
		Used during ROM verification (12V).	8048AH 8049AH 8050AH
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH})	All
XTAL2	3	Other side of crystal input.	All

Table 2. Instruction Set

Accumulator

Mnemonic	Description	Bytes	Cycles
ADD A, R	Add register to A	1	1
ADD A, @R	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, R	Add register with carry	1	1
ADDC A, @R	Add data memory with carry	1	1
ADDC A, #data	Add immediate with carry	2	2
ANL A, R	And register to A	1	1
ANL A, @R	And data memory to A	1	1
ANL A, #data	And immediate to A	2	2
ORL A, R	Or register to A	1	1
ORL A, @R	Or data memory to A	1	1
ORL A, #data	Or immediate to A	2	2
XRL A, R	Exclusive or register to A	1	1
XRL A, @R	Exclusive or data memory to A	1	1
XRL A, #data	Exclusive or immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1

Input/Output

Mnemonic	Description	Bytes	Cycles
IN A, P	Input port to A	1	2
OUTL P, A	Output A to port	1	2
ANL P, #data	And immediate to port	2	2
ORL P, #data	Or immediate to port	2	2
INS A, BUS	Input BUS to A	1	2
OUTL BUS, A	Output A to BUS	1	2
ANL BUS, #data	And immediate to BUS	2	2
ORL BUS, #data	Or immediate to BUS	2	2
MOVD A, P	Input expander port to A	1	2
MOVD P, A	Output A to expander port	1	2
ANLD P, A	And A to expander port	1	2
ORLD P, A	Or A to expander port	1	2

Registers

Mnemonic	Description	Bytes	Cycles
INC R	Increment register	1	1
INC @R	Increment data memory	1	1
DEC R	Decrement register	1	1

Branch

Mnemonic	Description	Bytes	Cycles
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ R, addr	Decrement register and skip	2	2
JC addr	Jump on carry = 1	2	2
JNC addr	Jump on carry = 0	2	2
JZ addr	Jump on A zero	2	2
JNZ addr	Jump on A not zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 = 1	2	2
JF1 addr	Jump on F1 = 1	2	2
JTF addr	Jump on timer flag	2	2
JNI addr	Jump on $\overline{INT}$ = 0	2	2
JBb addr	Jump on accumulator bit	2	2

Table 2. Instruction Set (Continued)

Subroutine

Mnemonic	Description	Bytes	Cycles
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2

Flags

Mnemonic	Description	Bytes	Cycles
CLR C	Clear carry	1	1
CPL C	Complement carry	1	1
CLR F0	Clear flag 0	1	1
CPL F0	Complement flag 0	1	1
CLR F1	Clear flag 1	1	1
CPL F1	Complement flag 1	1	1

Data Moves

Mnemonic	Description	Bytes	Cycles
MOV A, R	Move register to A	1	1
MOV A, @R	Move data memory to A	1	1
MOV A, #data	Move immediate to A	2	2
MOV R, A	Move A to register	1	1
MOV @R, A	Move A to data memory	1	1
MOV R, #data	Move immediate to register	2	2
MOV @R, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, R	Exchange A and register	1	1
XCH A, @R	Exchange A and data memory	1	1
XCHD A, @R	Exchange nibble of A and data memory	1	1
MOVX A, @R	Move external data memory to A	1	2
MOVX @R, A	Move A to external data memory	1	2
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @A	Move to A from page 3	1	2

Timer/Counter

Mnemonic	Description	Bytes	Cycles
MOV A, T	Read timer/counter	1	1
MOV T, A	Load timer/counter	1	1
STRT T	Start timer	1	1
STRT CNT	Start counter	1	1
STOP TCNT	Stop timer/counter	1	1
EN TCNTI	Enable timer/ counter interrupt	1	1
DIS TCNTI	Disable timer/ counter interrupt	1	1

Control

Mnemonic	Description	Bytes	Cycles
EN I	Enable external interrupt	1	1
DIS I	Disable external interrupt	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
ENT0 CLK	Enable clock output on T0	1	1

Mnemonic	Description	Bytes	Cycles
NOP	No operation	1	1

ABSOLUTE MAXIMUM RATINGS*

Case Temperature Under Bias . . . . . 0°C to +70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on any Pin with Respect to Ground . . . . . −0.5V to +7V
Power Dissipation 1.5W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS T_A = 0°C to +70°C; V_{CC} = V_{DD} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter	Limits			Unit	Test Conditions	Device
		Min	Typ	Max			
V_{IL}	Input Low Voltage (All Except $\overline{RESET}$, X1, X2)	−0.5		0.8	V		All
V_{IL1}	Input Low Voltage ($\overline{RESET}$, X1, X2)	−5		0.6	V		All
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.0		V_{CC}	V		All
V_{IH1}	Input High Voltage (X1, X2, $\overline{RESET}$)	3.8		V_{CC}	V		All
V_{OL}	Output Low Voltage (BUS)			0.45	V	I_{OL} = 2.0 mA	All
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			0.45	V	I_{OL} = 1.8 mA	All
V_{OL2}	Output Low Voltage (PROG)			0.45	V	I_{OL} = 1.0 mA	All
V_{OL3}	Output Low Voltage (All Other Outputs)			0.45	V	I_{OL} = 1.6 mA	All
V_{OH}	Output High Voltage (BUS)	2.4			V	I_{OH} = −400 μA	All
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	I_{OH} = −100 μA	All
V_{OH2}	Output High Voltage (All Other Outputs)	2.4			V	I_{OH} = −40 μA	All

D.C. CHARACTERISTICS T_A = 0°C to +70°C; V_{CC} = V_{DD} = 5V ±10%; V_{SS} = 0V (Continued)

Symbol	Parameter	Limits			Unit	Test Conditions	Device
		Min	Typ	Max			
I_{L1}	Leakage Current (T1, $\overline{INT}$)			±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$	All
I_{LI1}	Input Leakage Current (P10–P17, P20–P27, EA, $\overline{SS}$)			−500	μA	$V_{SS} + 0.45 \le V_{IN} \le V_{CC}$	All
I_{LI2}	Input Leakage Current $\overline{RESET}$	−10		−300	μA	$V_{SS} \le V_{IN} \le 3.8$	All
I_{LO}	Leakage Current (BUS, T0) (High Impedance State)			±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$	All
I_{DD}	V_{DD} Supply Current (RAM Standby)		3	5	mA		8048AH 8035AHL
			4	7	mA		8049AH 8039AHL
			5	10	mA		8050AH 8040AHL
$I_{DD} + I_{CC}$	Total Supply Current*		30	65	mA		8048AH 8035AHL
			35	70	mA		8049AH 8039AHL
			40	80	mA		8050AH 8040AHL
			30	100	mA		P8748H
			50	110	mA		P8749H
V_{DD}	RAM Standby Voltage	2.2		5.5	V	Standby Mode Reset $\le V_{IL1}$	8048AH 8035AH
		2.2		5.5	V		8049AH 8039AH
		2.2		5.5	V		8050AH 8040AHL

*I_{CC} + I_{DD} are measured with all outputs in their high impedance state; $\overline{RESET}$ low; 11 MHz crystal applied; $\overline{INT}$, $\overline{SS}$, and EA floating.

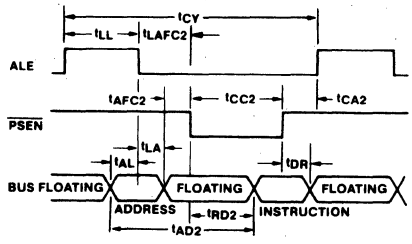
A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	f (t) (Note 3)	11 MHz		Unit	Conditions (Note 1)
			Min	Max		
t	Clock Period	1/xtal freq	90.9	1000	ns	(Note 3)
t_{LL}	ALE Pulse Width	3.5t−170	150		ns	
t_{AL}	Addr Setup to ALE	2t−110	70		ns	(Note 2)
t_{LA}	Addr Hold from ALE	t−40	50		ns	
t_{CC1}	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)	7.5t−200	480		ns	
t_{CC2}	Control Pulse Width ($\overline{PSEN}$)	6t−200	350		ns	
t_{DW}	Data Setup before $\overline{WR}$	6.5t−200	390		ns	
t_{WD}	Data Hold after $\overline{WR}$	t−50	40		ns	
t_{DR}	Data Hold ($\overline{RD}$, $\overline{PSEN}$)	1.5t−30	0	110	ns	
t_{RD1}	$\overline{RD}$ to Data in	6t−170		375	ns	
t_{RD2}	$\overline{PSEN}$ to Data in	4.5t−170		240	ns	
t_{AW}	Addr Setup to $\overline{WR}$	5t−150	300		ns	
t_{AD1}	Addr Setup to Data ($\overline{RD}$)	10.5t−220		730	ns	
t_{AD2}	Addr Setup to Data ($\overline{PSEN}$)	7.5t−200		460	ns	
t_{AFC1}	Addr Float to $\overline{RD}$, $\overline{WR}$	2t−40	140		ns	(Note 2)
t_{AFC2}	Addr Float to $\overline{PSEN}$	0.5t−40	10		ns	(Note 2)
t_{LAFC1}	ALE to Control ($\overline{RD}$, $\overline{WR}$)	3t−75	200		ns	
t_{LAFC2}	ALE to Control ($\overline{PSEN}$)	1.5t−75	60		ns	
t_{CA1}	Control to ALE ($\overline{RD}$, $\overline{WR}$, PROG)	t−65	25		ns	
t_{CA2}	Control to ALE ($\overline{PSEN}$)	4t−70	290		ns	
t_{CP}	Port Control Setup to PROG	1.5t−80	50		ns	
t_{PC}	Port Control Hold to PROG	4t−260	100		ns	
t_{PR}	PROG to P2 Input Valid	8.5t−120		650	ns	
t_{PF}	Input Data Hold from PROG	1.5t	0	140	ns	
t_{DP}	Output Data Setup	6t−290	250		ns	
t_{PD}	Output Data Hold	1.5t−90	40		ns	
t_{PP}	PROG Pulse Width	10.5t−250	700		ns	
t_{PL}	Port 2 I/O Setup to ALE	4t−200	160		ns	
t_{LP}	Port 2 I/O Hold to ALE	0.5t−30	15		ns	
t_{PV}	Port Output from ALE	4.5t+100		5.0	ns	
t_{0PRR}	T0 Rep Rate	3t	270		ns	
t_{CY}	Cycle Time	15t	1.36	15.0	μs	

NOTES:
1. Control outputs: $C_L = 80$ pF. BUS Outputs: $C_L = 150$ pF.
2. BUS High Impedance Load 20 pF
3. f(t) assumes 50% duty cycle on X1, X2. Max clock period is for a 1 MHz crystal input.

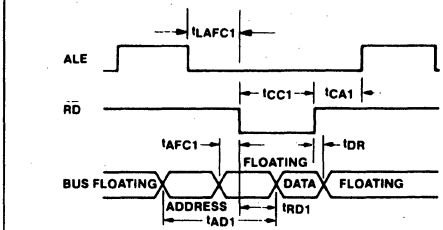
WAVEFORMS

INSTRUCTION FETCH FROM PROGRAM MEMORY



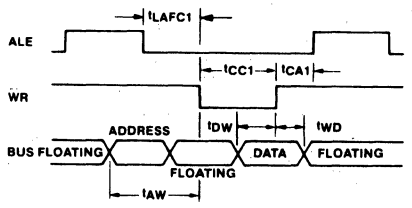
270053-4

READ FROM EXTERNAL DATA MEMORY



270053-5

WRITE TO EXTERNAL DATA MEMORY



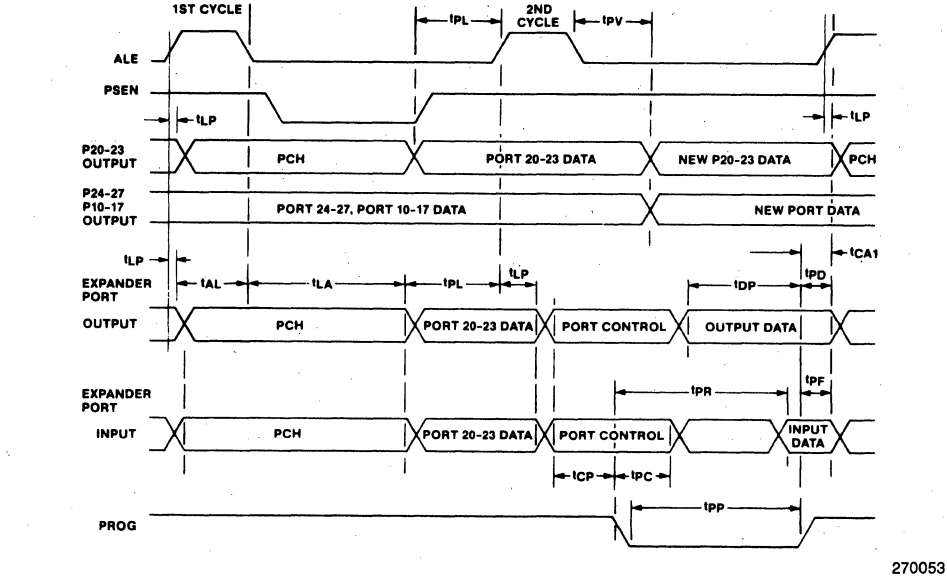
270053-6

INPUT AND OUTPUT FOR A.C. TESTS

270053-7

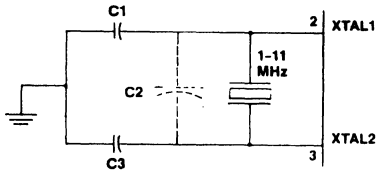
A.C. testing inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Output timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

PORT 1/PORT 2 TIMING



270053-8

CRYSTAL OSCILLATOR MODE

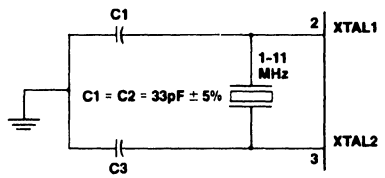


270053-9

C1 = 5 pF ±½ pF + (STRAY < 5 pF)
C2 = (CRYSTAL + STAY) < 8 pF
C3 = 20 pF ±1 pF + (STRAY < 5 pF)

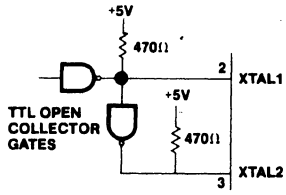
Crystal series resistance should be less than 30Ω at 11 MHz; less than 75Ω at 6 MHz; less than 180Ω at 3.6 MHz.

CERAMIC RESONATOR MODE



270053-10

DRIVING FROM EXTERNAL SOURCE



270053-11

For XTAL1 and XTAL2 define "high" as voltages above 1.6V and "low" as voltages below 1.6V. The duty cycle requirements for externally driving XTAL1 and XTAL2 using the circuits shown above are as follows: XTAL1 must be high 35–65% of the period and XTAL2 must be high 35–65% of the period. Rise and fall times must be faster than 20 ns.

PROGRAMMING AND VERIFYING THE P8749H/48H PROGRAMMABLE ROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL1	Clock Input (3 to 4.0 MHz)
XTAL2	
$\overline{\text{RESET}}$	Initialization and Address Latching
T0	Selection of Program or Verifying Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input
	Data Output During Verify
P20–P22	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:

An attempt to program a missocketed P8749H/48H will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. V_{DD} = 5V, Clock applied or internal oscillator operating, $\overline{\text{RESET}}$ = 0V, T0 = 5V, EA = 5V, BUS and PROG floating. P10 and P11 must be tied to ground.
2. Insert P8749H/48H in programming socket
3. T0 = 0V (select program mode)
4. EA = 18V (activate program mode)
5. Address applied to BUS and P20–22
6. $\overline{\text{RESET}}$ = 5V (latch address)
7. Data applied to BUS
8. V_{DD} = 21V (programming power)
9. PROG = V_{CC} or float followed by one 50 ms pulse to 18V
10. V_{DD} = 5V
11. T0 = 5V (verify mode)
12. Read and verify data on BUS
13. T0 = 0V
14. $\overline{\text{RESET}}$ = 0V and repeat from step 5
15. Programmer should be at conditions of step 1 when P8749H/48H is removed from socket.

NOTE:
Once programmed the P8749H/48H cannot be erased.

A.C. TIMING SPECIFICATION FOR PROGRAMMING P8748H/P8749H ONLY

T_A = 25°C ±5°C; V_{CC} = 5V ±5%; V_{DD} = 21 ±0.5V

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}$	$4t_{CY}$			
t_{WA}	Address Hold Time After $\overline{RESET}$	$4t_{CY}$			
t_{DW}	Data in Setup Time to PROG	$4t_{CY}$			
t_{WD}	Data in Hold Time After PROG	$4t_{CY}$			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	$4t_{CY}$			
t_{VDDW}	V_{DD} Hold Time Before PROG	0	1.0	ms	
t_{VDDH}	V_{DD} Hold Time After PROG	0	1.0	ms	
t_{PW}	Program Pulse Width	50	60	ms	
t_{TW}	T0 Setup Time for Program Mode	$4t_{CY}$			
t_{WT}	T0 Hold Time After Program Mode	$4t_{CY}$			
t_{DO}	T0 to Data Out Delay		$4t_{CY}$		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	$4t_{CY}$			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	100	μs	
t_{CY}	CPU Operation Cycle Time	3.75	5	μs	
t_{RE}	$\overline{RESET}$ Setup Time before EA	$4t_{CY}$			

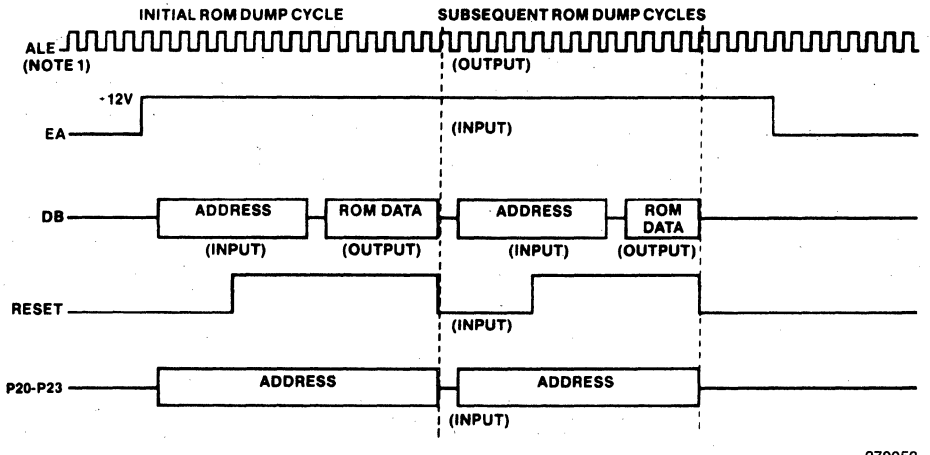
NOTE:
If Test 0 is high, t_{DO} can be triggered by $\overline{RESET}$.

D.C. CHARACTERISTICS FOR PROGRAMMING P8748H/P8749H ONLY

T_A = 25°C ±5°C; V_{CC} = 5V ±5%; V_{DD} = 21 ±0.5V

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{DDH}	V_{DD} Program Voltage High Level	20.5	21.5	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	17.5	18.5	V	
V_{PL}	PROG Voltage Low Level	4.0	V_{CC}	V	
V_{EAH}	EA Program or Verify Voltage High Level	17.5	18.5	V	
I_{DD}	V_{DD} High Voltage Supply Current		20.0	mA	
I_{PROG}	PROG High Voltage Supply Current		1.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

SUGGESTED ROM VERIFICATION ALGORITHM FOR ROM DEVICE ONLY



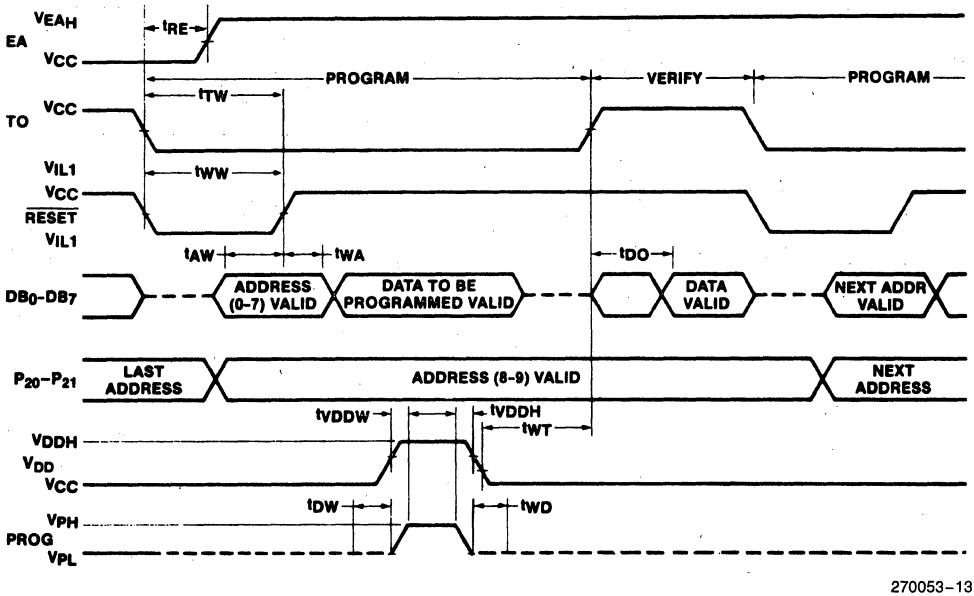
270053-12

	48H	49H	50H
A10	0	ADDR	ADDR
A11	0	0	ADDR

$V_{CC} = V_{DD} = +5V$
$V_{SS} = 0V$

NOTE:
ALE is function of X1, X2 inputs.

COMBINATION PROGRAM/VERIFY MODE (PROGRAMMABLE ROMS ONLY)



270053-13

D8748H/D8749H
HMOS-E SINGLE-COMPONENT 8-BIT MICROCOMPUTER

- **High Performance HMOS-E**
- **Interval Timer/Event Counter**
- **Two Single Level Interrupts**
- **Single 5-Volt Supply**
- **Over 96 Instructions; 90% Single Byte**
- **Compatible with 8080/8085 Peripherals**
- **Easily Expandable Memory and I/O**
- **Up to 1.35 μs Instruction Cycle; All Instructions 1 or 2 Cycles**

The Intel D8749H/D8748H are totally self-sufficient, 8-bit parallel computers fabricated on single silicon chips using Intel's advanced N-channel silicon gate HMOS-E process.

The family contains 27 I/O lines, an 8-bit timer/counter, on-chip RAM and on-board oscillator/clock circuits. For systems that require extra capability, the family can be expanded using MCS®-80/MCS®-85 peripherals.

These microcomputers are designed to be efficient controllers as well as arithmetic processors. They have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over 2 bytes in length.

Device	Internal Memory	
D8749H	2K x 8 EPROM	128 x 8 RAM
D8748H	1K x 8 EPROM	64 x 8 RAM

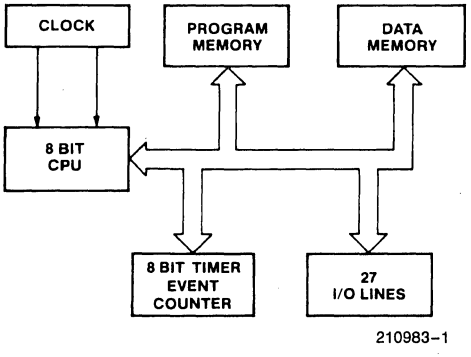


Figure 1. Block Diagram

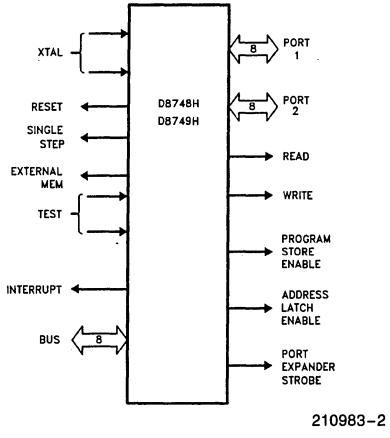


Figure 2. Logic Symbol

October 1987
Order Number: 210983-003

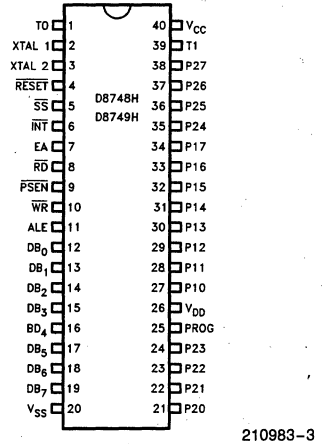


Figure 3. Pin Configuration

Table 1. Pin Description (40-Pin DIP)

Symbol	Pin No.	Function
V_{SS}	20	Circuit GND potential.
V_{DD}	26	+5V during normal operation.
		Programming power supply (+21V).
V_{CC}	40	Main power supply; +5V during operation and programming.
PROG	25	Output strobe for 8243 I/O expander.
		Program pulse (+18V) input pin during programming.
P10–P17 Port 1	27–34	8-bit quasi-bidirectional port.
P20–P23	21–24	8-bit quasi-bidirectional port. P20–P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
P24–P27 Port 2	35–38	
DB0–DB7 BUS	12–19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CKL instruction.
		Used during programming.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low) interrupt must remain low for at least 3 machine cycles for proper operation.
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device. Used as a read strobe to external data memory. (Active low)

Table 1. Pin Description (40-Pin DIP) (Continued)

Symbol	Pin No.	Function
$\overline{RESET}$	4	Input which is used to initialize the processor. (Active low) (Non TTL V_{IH})
		Used during programming.
$\overline{WR}$	10	Output strobe during a bus write. (Active low) Used as write strobe to external data memory.
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low.)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction.
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug. (Active high.)
		Used during (18V) programming.
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH}.)
XTAL2	3	Other side of crystal input.

Table 2. Instruction Set

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR			
ADD A, R	Add register to A	1	1
ADD A,@R	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	2
ADDC A, R	Add register with carry	1	1
ADDC A, @R	Add data memory with carry	1	1
ADDC A, #data	Add immediate with carry	2	2
ANL A, R	And register to A	1	1
ANL A, @R	And data memory to A	1	1
ANL A, #data	And immediate to A	2	2
ORL A, R	Or register to A	1	1
ORL A, @R	Or data memory to A	1	1
ORL A, #data	Or immediate to A	2	2
XRL A, R	Exclusive or register to A	1	1
XRL A, @R	Exclusive or data memory to A	1	1
XRL A, #data	Exclusive or immediate to A	2	2

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR (Continued)			
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A, P	Input port to A	1	2
OUTL P, A	Output A to port	1	2
ANL P, #data	And immediate to port	2	2
ORL P, #data	Or immediate to port	2	2
INS A, BUS	Input BUS to A	1	2
OUTL BUS, A	Output A to BUS	1	2
ANL BUS, #data	And immediate to BUS	2	2
ORL BUS, #data	Or immediate to BUS	2	2
MOVD A, P	Input expander port to A	1	2

Table 2. Instruction Set (Continued)

Mnemonic	Description	Bytes	Cycles
INPUT/OUTPUT (Continued)			
MOVD P, A	Output A to expander port	1	2
ANLD P, A	And A to expander port	1	2
ORLD P, A	Or A to expander port	1	2
REGISTERS			
INC R	Increment register	1	1
INC @R	Increment data memory	1	1
DEC R	Decrement register	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ R, addr	Decrement register and skip	2	2
JC addr	Jump on carry = 1	2	2
JNC addr	Jump on carry = 0	2	2
JZ addr	Jump on A zero	2	2
JNZ addr	Jump on A not zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 = 1	2	2
JF1 addr	Jump on F1 = 1	2	2
JTF addr	Jump on timer flag	2	2
JNI addr	Jump on INT = 0	2	2
JBb addr	Jump on accumulator bit	2	2
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear carry	1	1
CPL C	Complement carry	1	1
CLR F0	Clear flag 0	1	1
CPL F0	Complement flag 0	1	1
CLR F1	Clear flag 1	1	1
CPL F1	Complement flag 1	1	1
DATA MOVES			
MOV A, R	Move register to A	1	1
MOV A, @R	Move data memory to A	1	1
MOV A, #data	Move immediate to A	2	2

Mnemonic	Description	Bytes	Cycles
DATA MOVES (Continued)			
MOV R, A	Move A to register	1	1
MOV @R, A	Move A to data memory	1	1
MOV R, #data	Move immediate to register	2	2
MOV @R, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, R	Exchange A and register	1	1
XCH A, @R	Exchange A and data memory	1	1
XCHD A, @R	Exchange nibble of A and register	1	1
MOVX A, @R	Move external data memory to A	1	2
MOVX @R, A	Move A to external data memory	1	2
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A, T	Read timer/counter	1	1
MOV T, A	Load timer/counter	1	1
STRT T	Start timer	1	1
STRT CNT	Start counter	1	1
STOP TCNT	Stop timer/counter	1	1
EN TCNTI	Enable timer/ counter interrupt	1	1
DIS TCNTI	Disable timer/ counter interrupt	1	1
CONTROL			
EN I	Enable external interrupt	1	1
DIS I	Disable external interrupt	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
ENT0 CLK	Enable clock output on T0	1	1
NOP	No operation	1	1

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage On Any Pin With Respect to Ground.....−0.5V to +7V
Power Dissipation.....1.0 Watt

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS T_A = 0°C to +70°C; V_{CC} = V_{DD} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter	Limits			Unit	Test Conditions	Device
		Min	Typ	Max			
V_{IL}	Input Low Voltage (All Except RESET, X1, X2)	−0.5		0.8	V		All
V_{IL1}	Input Low Voltage ($\overline{RESET}$, X1, X2)	−0.5		0.6	V		All
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$	2.0		V_{CC}	V		All
V_{IH1}	Input High Voltage (X1, X2, $\overline{RESET}$)	3.8		V_{CC}	V		All
V_{OL}	Output Low Voltage (BUS)			0.45	V	I_{OL} = 2.0 mA	All
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			0.45	V	I_{OL} = 1.8 mA	All
V_{OL2}	Output Low Voltage (PROG)			0.45	V	I_{OL} = 1.0 mA	All
V_{OL3}	Output Low Voltage (All Other Outputs)			0.45	V	I_{OL} = 1.6 mA	All
V_{OH}	Output High Voltage (BUS)	2.4			V	I_{OH} = −400 μA	All
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	I_{OH} = −100 μA	All
V_{OH2}	Output High Voltage (All Other Outputs)	2.4			V	I_{OH} = −40 μA	All
I_{L1}	Leakage Current (T1, $\overline{INT}$)			±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$	All
I_{LI1}	Input Leakage Current (P10–P17, P20–P27, EA, $\overline{SS}$)			−500	μA	V_{SS} + 0.45 ≤ $V_{IN} \le V_{CC}$	All
I_{LI2}	Input Leakage Current $\overline{RESET}$	−10		−300	μA	$V_{SS} \le V_{IN} \le 3.8V$	All
I_{L0}	Leakage Current (BUS, T0) (High Impedance State)			±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$	All
I_{DD} + I_{CC}	Total Supply Current*		80	100	mA		8748H
			95	110	mA		8749H

NOTE:
*I_{CC} + I_{DD} is measured with all outputs disconnected; $\overline{SS}$, $\overline{RESET}$, and $\overline{INT}$ equal to V_{CC}; EA equal to V_{SS}.

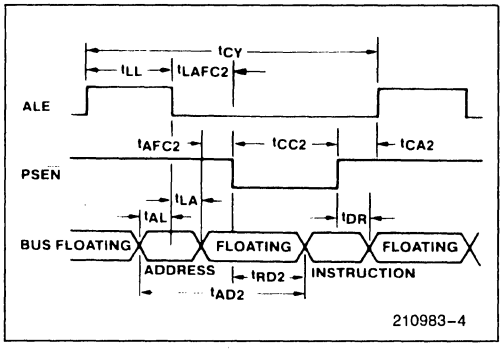
A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	f(t) (Note 3)	11 MHz		Unit	Conditions (Note 1)
			Min	Max		
t	Clock Period	1/xtal freq	90.9	1000	ns	(Note 3)
t_{LL}	ALE Pulse Width	3.5t − 170	150		ns	
t_{AL}	Addr Setup to ALE	2t − 110	70		ns	(Note 2)
t_{LA}	Addr Hold from ALE	t − 40	50		ns	
t_{CC1}	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)	7.5t − 200	480		ns	
t_{CC2}	Control Pulse Width ($\overline{PSEN}$)	6t − 200	350		ns	
t_{DW}	Data Setup before $\overline{WR}$	6.5t − 200	390		ns	
t_{WD}	Data Hold after $\overline{WR}$	t − 50	40		ns	
t_{DR}	Data Hold ($\overline{RD}$, $\overline{PSEN}$)	1.5t − 30	0	110	ns	
t_{RD1}	$\overline{RD}$ to Data In	6t − 170		375	ns	
t_{RD2}	$\overline{PSEN}$ to Data In	4.5t − 170		240	ns	
t_{AW}	Addr Setup to $\overline{WR}$	5t − 150	300		ns	
t_{AD1}	Addr Setup to Data ($\overline{RD}$)	10.5t − 220		730	ns	
t_{AD2}	Addr Setup to Data ($\overline{PSEN}$)	7.5t − 200		460	ns	
t_{AFC1}	Addr Float to $\overline{RD}$, $\overline{WR}$	2t − 40	140		ns	(Note 2)
t_{AFC2}	Addr Float to $\overline{PSEN}$	0.5t − 40	10		ns	(Note 2)
t_{LAFC1}	ALE to Control ($\overline{RD}$, $\overline{WR}$)	3t − 75	200		ns	
t_{LAFC2}	ALE to Control ($\overline{PSEN}$)	1.5t − 75	60		ns	
t_{CA1}	Control to ALE ($\overline{RD}$, $\overline{WR}$, PROG)	t − 65	25		ns	
t_{CA2}	Control to ALE ($\overline{PSEN}$)	4t − 70	290		ns	
t_{CP}	Port Control Setup to PROG	1.5t − 80	50		ns	
t_{PC}	Port Control Hold to PROG	4t − 260	100		ns	
t_{PR}	PROG to P2 Input Valid	8.5t − 120		650	ns	
t_{PF}	Input Data Hold from PROG	1.5t	0	140	ns	
t_{DP}	Output Data Setup	6t − 290	250		ns	
t_{PD}	Output Data Hold	1.5t − 90	40		ns	
t_{PP}	PROG Pulse Width	10.5t − 250	700		ns	
t_{PL}	Port 2 I/O Setup to ALE	4t − 200	160		ns	
t_{LP}	Port 2 I/O Hold to ALE	0.5t − 30	15		ns	
t_{PV}	Port Output from ALE	4.5t + 100		510	ns	
t_{0PRR}	T0 Rep Rate	3t	270		ns	
t_{CY}	Cycle Time	15t	1.36	15.0	μs	

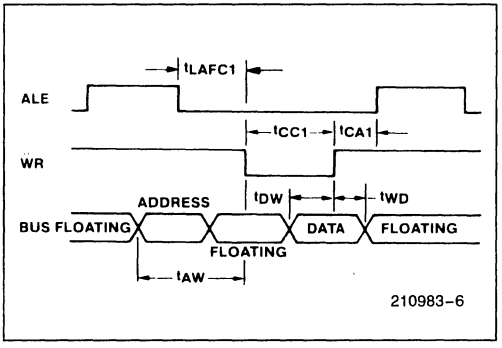
NOTES:
1. Control outputs CL = 80 pF; BUS outputs CL = 150 pF.
2. BUS High Impedance Load 20 pF.
3. f(t) assumes 50% duty cycle on X1, X2. Max clock period is for a 1 MHz crystal input.

WAVEFORMS

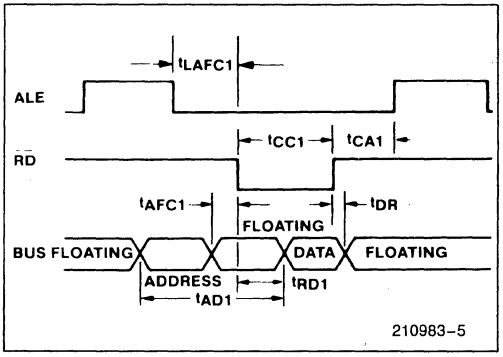
INSTRUCTION FETCH FROM PROGRAM MEMORY



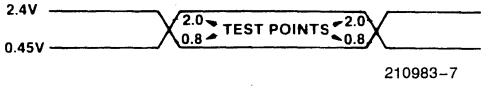
WRITE TO EXTERNAL DATA MEMORY



READ FROM EXTERNAL DATA MEMORY

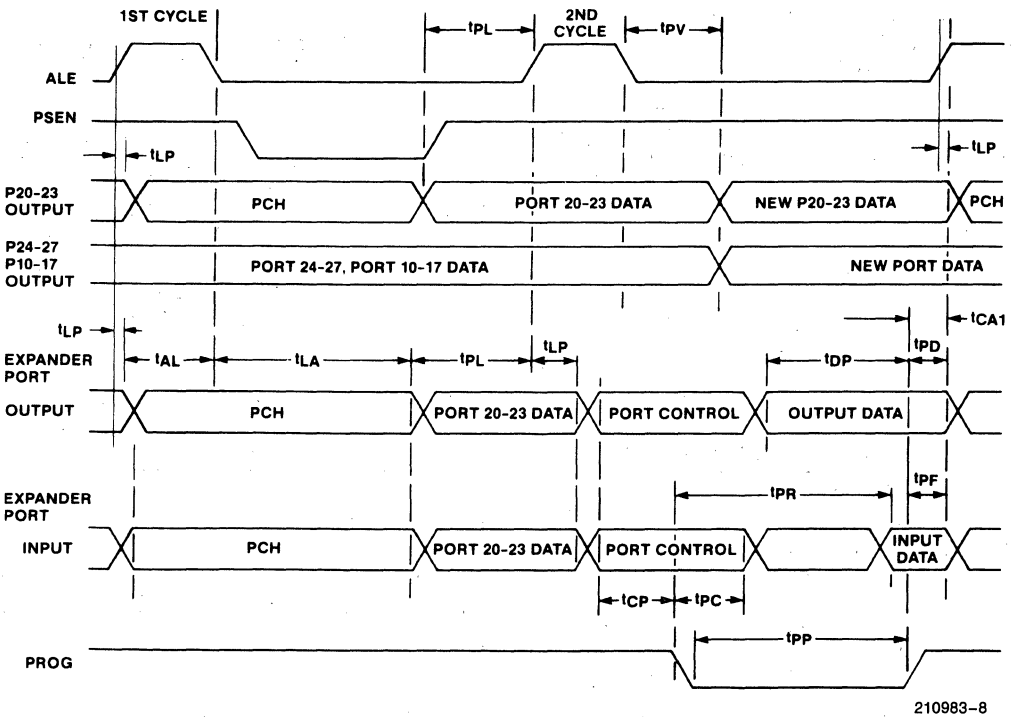


INPUT AND OUTPUT FOR A.C. TESTS

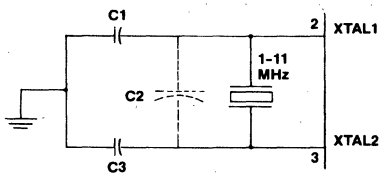


A.C. testing inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Output timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0."

PORT 1/PORT 2 TIMING



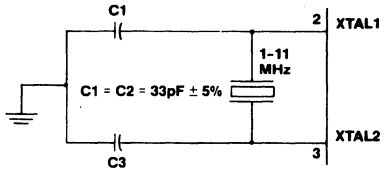
CRYSTAL OSCILLATOR MODE



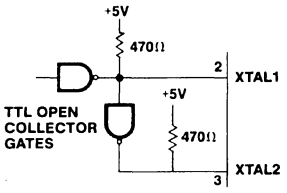
C1 = 5 pF ±½ pF + (STRAY < 5 pF)
C2 = (CRYSTAL + STRAY) < 8 pF
C3 = 20 pF ±1 pF (STRAY < 5 pF)

Crystal series resistance should be less than 30Ω at 11 MHz; less than 75Ω at 6 MHz; less than 180Ω at 3.6 MHz.

CERAMIC RESONATOR MODE



DRIVING FROM EXTERNAL SOURCE



210983-11

For XTAL1 and XTAL2 define "high" as voltages above 1.6V and "low" as voltages below 1.6V. The duty cycle requirements for externally driving XTAL1 and XTAL2 using the circuit shown above are as follows: XTAL1 must be high 35–65% of the period and XTAL2 must be high 36–65% of the period. Rise and fall times must be faster than 20 ns.

PROGRAMMING, VERIFYING AND ERASING THE 8749H (8748H) EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (3 to 4.0 MHz)
XTAL 2	
RESET	Initialization and Address Latching
TEST 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input
	Data Output During Verify
P20–P22	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING

An attempt to program a missocketed 8749H (8748H) will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1) V_{DD} = 5V, Clock applied or internal oscillator operating. RESET = 0V, TEST 0 = 5V, EA = 5V, BUS and PROG floating. P10 and P11 must be tied to ground.
2) Insert 8749H (8748H) in programming socket.
3) TEST 0 = 0V (select program mode)
4) EA = 18V (activate program mode)
5) Address applied to BUS and P20–22
6) RESET = 5V (latch address)
7) Data applied to BUS
8) V_{DD} = 21V (programming power)
9) PROG = V_{CC} or float followed by one 50 ms pulse to 18V
10) V_{DD} = 5V
11) TEST 0 = 5V (verify mode)
12) Read and verify data on BUS
13) TEST 0 = 0V
14) RESET = 0V and repeat from step 5
15) Programmer should be at conditions of step 1 when 8749H (8748H) is removed from socket.

A.C. TIMING SPECIFICATION FOR PROGRAMMING 8748H/8749H

T_A = 25°C ±5°C; V_{CC} = 5V ±5%; V_{DD} = 21V ±0.5V

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}$ ↑	$4t_{CY}$			
t_{WA}	Address Hold Time after $\overline{RESET}$ ↑	$4t_{CY}$			
t_{DW}	Data in Setup Time to PROG ↑	$4t_{CY}$			
t_{WD}	Data in Hold Time after PROG ↓	$4t_{CY}$			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	$4t_{CY}$			
t_{VDDW}	V_{DD} Hold Time before PROG ↑	0	1.0	ms	
t_{VDDH}	V_{DD} Hold Time after PROG ↓	0	1.0	ms	
t_{PW}	Program Pulse Width	50	60	ms	
t_{TW}	TEST 0 Setup Time for Program Mode	$4t_{CY}$			
t_{WT}	TEST 0 Hold Time after Program Mode	$4t_{CY}$			
t_{DO}	TEST 0 to Data Out Delay		$4t_{CY}$		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	$4t_{CY}$			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	100	μs	
t_{CY}	CPU Operation Cycle Time	3.75	5	μs	
t_{RE}	$\overline{RESET}$ Setup Time before EA ↑	$4t_{CY}$			

NOTE:
If TEST 0 is high, t_{DO} can be triggered by $\overline{RESET}$ ↑.

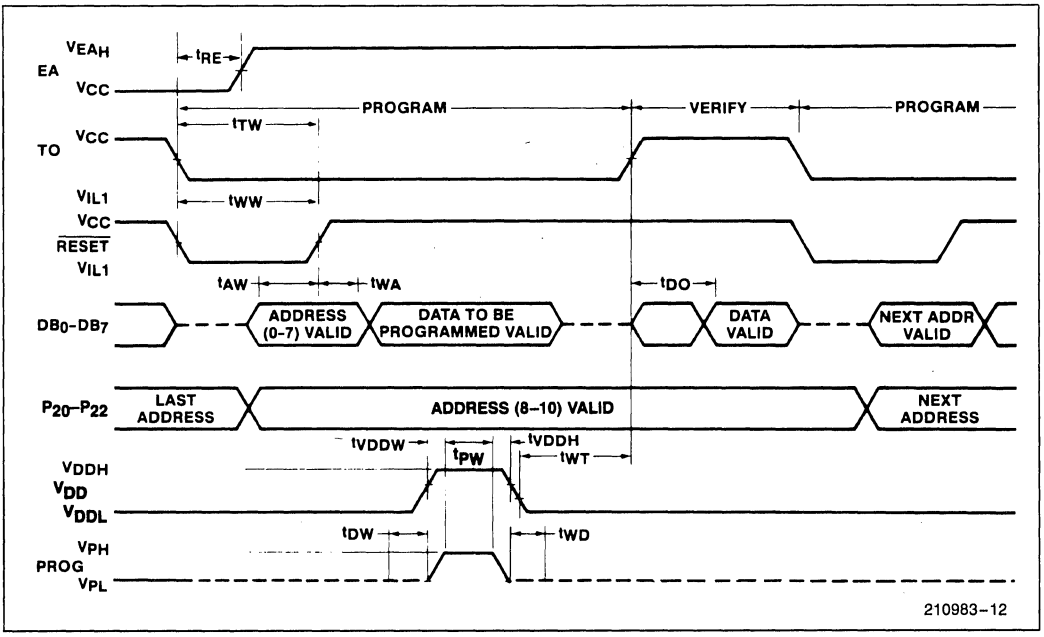
D.C. SPECIFICATION FOR PROGRAMMING 8748H/8749H

T_A = 25°C ±5°C; V_{CC} = 5V ±5%; V_{DD} = 21V ±0.5V

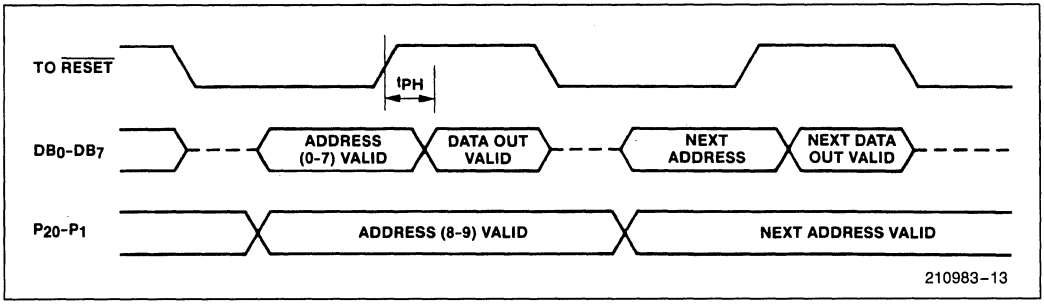
Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{DDH}	V_{DD} Program Voltage High Level	20.5	21.5	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	17.5	18.5	V	
V_{PL}	PROG Voltage Low Level	4.0	V_{CC}	V	
V_{EAH}	EA Program or Verify Voltage High Level	17.5	18.5	V	
I_{DD}	V_{DD} High Voltage Supply Current		20.0	mA	
I_{PROG}	PROG High Voltage Supply Current		1.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

WAVEFORMS

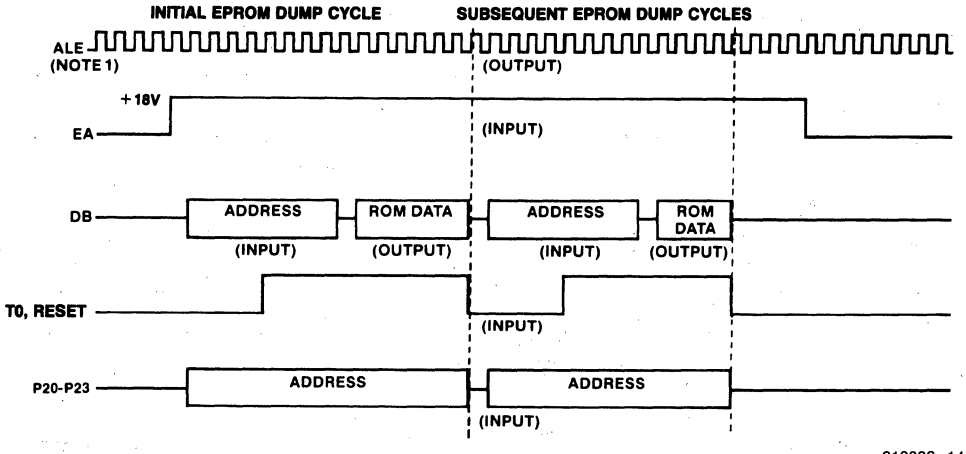
COMBINATION PROGRAM/VERIFY MODE (EPROMs ONLY)



VERIFY MODE



SUGGESTED EPROM VERIFICATION ALGORITHM FOR HMOS-E DEVICE ONLY



210983-14

	48H	49H
A10	0	ADDR
A11	0	0

$V_{CC} = V_{DD} = +5V$
$V_{SS} = 0V$

NOTE:
ALE is function of X1, X2 inputs.

MCS®-48
EXPRESS

- 0°C to 70°C Operation
- −40°C to +85°C Operation
- 168 Hr. Burn-In
- 8048AH/8035AHL
- 8049AH/8039AHL
- 8050AH/8040AHL
- 8748H
- 8243
- 8749H

The new Intel EXPRESS family of single-component 8-bit microcomputers offers enhanced processing options to the familiar 8048AH/8035AHL, 8748H, 8049AH/8039AHL, 8749H, 8050AH/8040AHL Intel components. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards, but fall short of military conditions.

The EXPRESS options include the commercial standard and −40°C to +85°C operation with or without 168 ±8 hours of dynamic burn-in at 125°C per MIL-STD-883, method 1015. Figure 1 summarizes the option marking designators and package selections.

For a complete description of 8048AH/8035AHL, 8748H, 8049AH/8309AHL, 8749H, 8040AHL and 8050AH features and operating characteristics, refer to the respective standard commercial grade data sheet. This document highlights only the electrical specifications which differ from the respective commercial part.

Temp Range °C	0–70	−40–+85	0–70	−40–+85
Burn In	0 Hrs	0 Hrs	168 Hrs	168 Hrs
	P8048AH	TP8048AH	QP8048AH	LP8048AH
	D8048AH	TD8048AH	QD8048AH	LD8048AH
	D8748H	TD8748H	QD8748H	LD8748H
	P8035AHL	TP8035AHL	QP8035AHL	LP8035AHL
	D8035AHL	TD8035AHL	QD8035AHL	LD8035AHL
	P8049AH	TP8049AH	QP8049AH	LP8049AH
	D8049AH	TD8049AH	QD8049AH	LD8049AH
	D8749H	TD8749AH	QD8749H	LD8749AH
	P8039AHL	TP8039AHL	QP8039AHL	LP8039AHL
	D8039AHL	TD8039AHL	QD8039AHL	LD8039AHL
	P8050AH	TP8050AH	QP8050AH	LP8050AH
	D8050AH	TD8050AH	QD8050AH	LD8050AH
	P8040AHL	TP8040AHL	QP8040AHL	LP8040AHL
	D8040AHL	TD8040AHL	QD8040AHL	LD8040AHL
	P8243	TP8243	QP8243	—
	D8243	TD8243	QD8243	LD8243

* Commercial Grade
P Plastic Package
D Cerdip Package

September 1987
Order Number: 270225-002

Extended Temperature Electrical Specification Deviations*

TP8048AH/TP8035AHL/LP8048AH/LP8035AHL
TD8048AH/TD8035AHL/LD8048AH/LD8035AHL

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.2		V_{CC}	V	
I_{DD}	V_{DD} Supply Current		4	8	mA	
$I_{DD} + I_{CC}$	Total Supply Current		40	80	mA	

TP8049AH/TP8039AHL/LP8049AH/LP8039AHL
TD8049AH/TD8039AHL/LD8049AH/LD8039AHL

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.2		V_{CC}	V	
I_{DD}	V_{DD} Supply Current		5	10	mA	
$I_{DD} + I_{CC}$	Total Supply Current		50	100	mA	

TP8050AH/TP8040AHL/LP8050AHL/LP8040AHL
TD8050AH/TD8040AHL/LD8050AH/LD8040AHL

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.2		V_{CC}	V	
I_{DD}	V_{DD} Supply Current		10	20	mA	
$I_{DD} + I_{CC}$	Total Supply Current		75	120	mA	

Extended Temperature Electrical Specification Deviations*

TD8748H/LD8748H

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.2		V_{CC}	V	
$I_{DD} + I_{CC}$	Total Supply Current		50	130	mA	

TD8749H/LD8749H

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.2		V_{CC}	V	
$I_{DD} + I_{CC}$	Total Supply Current		75	150	mA	

TP8743/TD8243/LD8243

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
I_{CC}	V_{CC} Supply Current		15	25	mA	

*Refer to individual commercial grade data sheet for complete operating characteristics.

MCS®-51 Architectural Overview

5

ARCHITECTURAL OVERVIEW OF THE MCS®-51 FAMILY OF MICROCONTROLLERS

INTRODUCTION

The 8051 is the original member of the MCS®-51 family, and is the core for all MCS-51 devices. The features of the 8051 core are:

- 8-bit CPU optimized for control applications
- Extensive Boolean processing (single-bit logic) capabilities
- 64K Program Memory address space
- 64K Data Memory address space
- 4K bytes of on-chip Program Memory
- 128 bytes of on-chip Data RAM
- 32 bidirectional and individually addressable I/O lines
- Two 16-bit timer/counters
- Full duplex UART
- 6-source/5-vector interrupt structure with two priority levels
- On-chip clock oscillator

The basic architectural structure of this 8051 core is shown in Figure 1.

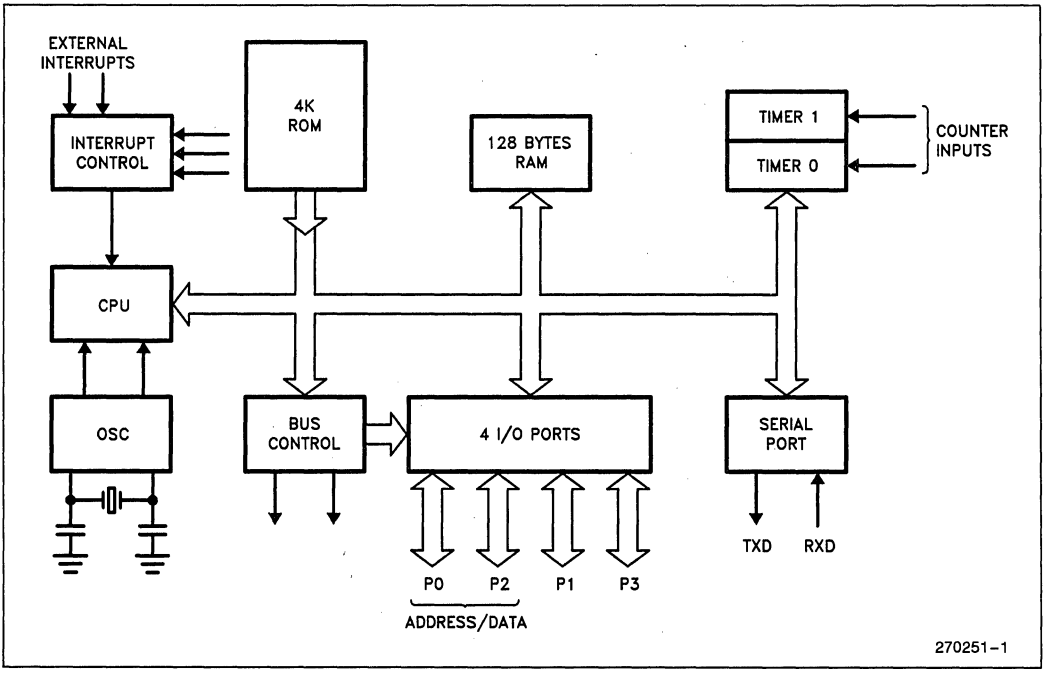


Figure 1. Block Diagram of the 8051 Core

Each device on the MCS-51 family consists of all the core features plus some additional features. A feature comparison of all the MCS-51 devices is shown in Table 1.

Table 1. The MCS®-51 Family of Microcontrollers

Device	ROMless Version	EPROM Version	ROM Bytes	RAM Bytes	8-Bit I/O Ports	16-Bit Timer/ Counters	Programmable Counter Array (PCA)	UART	Serial Expansion Port (SEP)	Global Serial Channel (GSC)	DMA Channels	A/D Channels	Interrupt Sources/ Vectors	Power Down and Idle Modes
8051	8031	—	4K	128	4	2		✓					6/5	
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓					6/5	
8052AH	8032AH	8752BH	8K	256	4	3		✓					8/6	
80C51BH	80C31BH	87C51	4K	128	4	2		✓					6/5	✓
83C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓					14/7	✓
83C51FB	80C51FA	87C51FB	16K	256	4	3	✓	✓					14/7	✓
83C51GA	80C51GA	87C51GA	4K	128	4	2		✓	✓			8	8/7	✓
83C152JA	80C152JA	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JB	—	—	256	7	2		✓		✓	2		19/11	✓
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		19/11	✓
83C451	80C451	—	4K	128	7	2		✓					6/5	✓
83C452	80C452	87C452P	8K	256	5	2		✓					9/8	✓

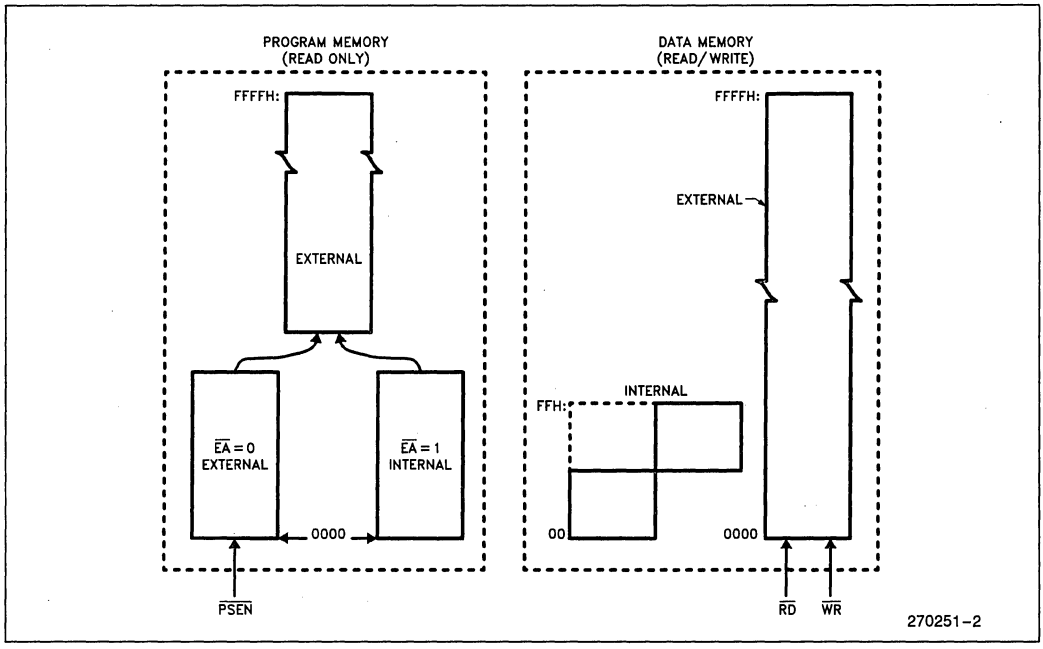


Figure 2. MCS®-51 Memory Structure

CHMOS Devices

Functionally, the CHMOS devices (designated with "C" in the middle of the device name) are all fully compatible with the 8051, but being CMOS, draw less current than an HMOS counterpart. To further exploit the power savings available in CMOS circuitry, two reduced power modes are added:

- Software-invoked Idle Mode, during which the CPU is turned off while the RAM and other on-chip peripherals continue operating. In this mode, current draw is reduced to about 15% of the current drawn when the device is fully active.
- Software-invoked Power Down Mode, during which all on-chip activities are suspended. The on-chip RAM continues to hold its data. In this mode the device typically draws less than 10 μA.

Although the 80C51BH is functionally compatible with its HMOS counterpart, specific differences between the two types of devices must be considered in the design of an application circuit if one wishes to ensure complete interchangeability between the HMOS and CHMOS devices. These considerations are discussed in the Application Note AP-252, "Designing with the 80C51BH".

For more information on the individual devices and features listed in Table 1, refer to the Hardware Descriptions and Data Sheets of the specific device.

MEMORY ORGANIZATION IN MCS®-51 DEVICES

Logical Separation of Program and Data Memory

All MCS-51 devices have separate address spaces for Program and Data Memory, as shown in Figure 2. The logical separation of Program and Data Memory allows the Data Memory to be accessed by 8-bit addresses, which can be more quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit Data Memory addresses can also be generated through the DPTR register.

Program Memory can only be read, not written to. There can be up to 64K bytes of Program Memory. In the ROM and EPROM versions of these devices the lowest 4K, 8K or 16K bytes of Program Memory are provided on-chip. Refer to Table 1 for the amount of on-chip ROM (or EPROM) on each device. In the ROMless versions all Program Memory is external. The read strobe for external Program Memory is the signal $\overline{\text{PSEN}}$ (Program Store Enable).

Data Memory occupies a separate address space from Program Memory. Up to 64K bytes of external RAM can be addressed in the external Data Memory space. The CPU generates read and write signals, $\overline{RD}$ and $\overline{WR}$, as needed during external Data Memory accesses.

External Program Memory and external Data Memory may be combined if desired by applying the $\overline{RD}$ and $\overline{PSEN}$ signals to the inputs of an AND gate and using the output of the gate as the read strobe to the external Program/Data memory.

Program Memory

Figure 3 shows a map of the lower part of the Program Memory. After reset, the CPU begins execution from location 0000H.

As shown in Figure 3, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

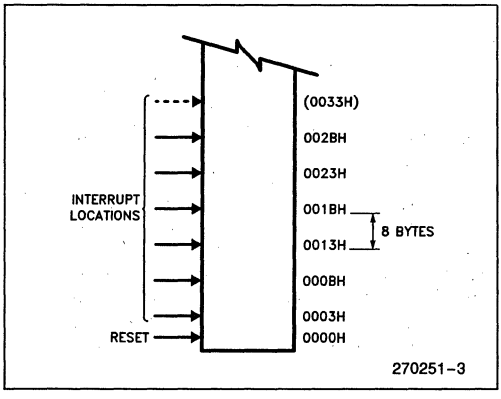


Figure 3. MCS®-51 Program Memory

The interrupt service locations are spaced at 8-byte intervals: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

The lowest 4K (or 8K or 16K) bytes of Program Memory can be either in the on-chip ROM or in an external ROM. This selection is made by strapping the $\overline{EA}$ (External Access) pin to either V_{CC} or V_{SS}.

In the 4K byte ROM devices, if the $\overline{EA}$ pin is strapped to V_{CC}, then program fetches to addresses 0000H through 0FFFH are directed to the internal ROM. Program fetches to addresses 1000H through FFFFH are directed to external ROM.

In the 8K byte ROM devices, $\overline{EA} = V_{CC}$ selects addresses 0000H through 1FFFH to be internal, and addresses 2000H through FFFFH to be external.

In the 16K byte ROM devices, $\overline{EA} = V_{CC}$ selects addresses 0000H through 3FFFH to be internal, and addresses 4000H through FFFFH to be external.

If the $\overline{EA}$ pin is strapped to V_{SS}, then all program fetches are directed to external ROM. The ROMless parts must have this pin externally strapped to V_{SS} to enable them to execute properly.

The read strobe to external ROM, $\overline{PSEN}$, is used for all external program fetches. $\overline{PSEN}$ is not activated for internal program fetches.

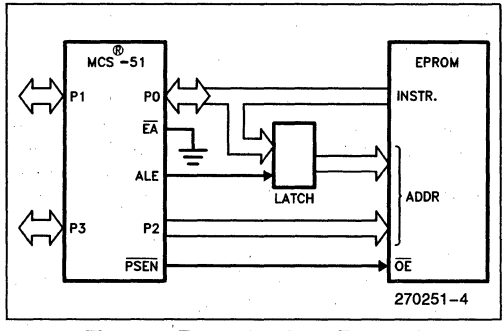


Figure 4. Executing from External Program Memory

The hardware configuration for external program execution is shown in Figure 4. Note that 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external Program Memory fetches. Port 0 (P0 in Figure 4) serves as a multiplexed address/data bus. It emits the low byte of the Program Counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the Program Memory. During the time that the low byte of the Program Counter is valid on P0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2 in Figure 4) emits the high byte of the Program Counter (PCH). Then $\overline{PSEN}$ strobes the EPROM and the code byte is read into the microcontroller.

Program Memory addresses are always 16 bits wide, even though the actual amount of Program Memory used may be less than 64K bytes. External program execution sacrifices two of the 8-bit ports, P0 and P2, to the function of addressing the Program Memory.

Data Memory

The right half of Figure 2 shows the internal and external Data Memory spaces available to the MCS-51 user.

Figure 5 shows a hardware configuration for accessing up to 2K bytes of external RAM. The CPU in this case is executing from internal ROM. Port 0 serves as a multiplexed address/data bus to the RAM, and 3 lines of Port 2 are being used to page the RAM. The CPU generates $\overline{RD}$ and $\overline{WR}$ signals as needed during external RAM accesses.

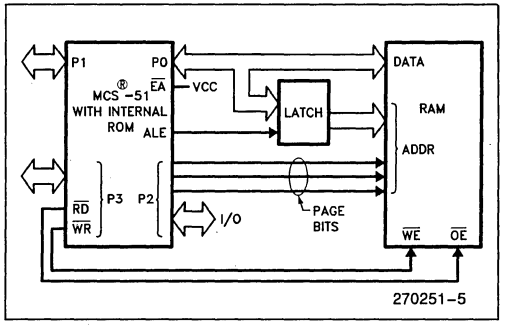


Figure 5. Accessing External Data Memory. If the Program Memory is Internal, the Other Bits of P2 are Available as I/O.

There can be up to 64K bytes of external Data Memory. External Data Memory addresses can be either 1 or 2 bytes wide. One-byte addresses are often used in conjunction with one or more other I/O lines to page the RAM, as shown in Figure 5. Two-byte addresses can also be used, in which case the high address byte is emitted at Port 2.

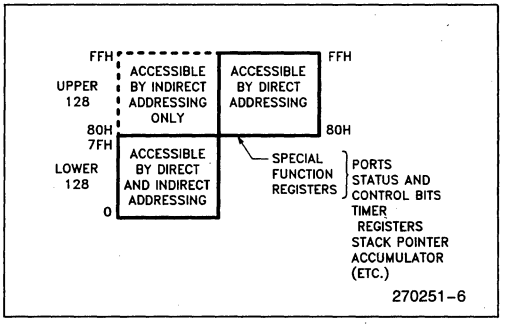


Figure 6. Internal Data Memory

Internal Data Memory is mapped in Figure 6. The memory space is shown divided into three blocks, which are generally referred to as the Lower 128, the Upper 128, and SFR space.

Internal Data Memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus Figure 6 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

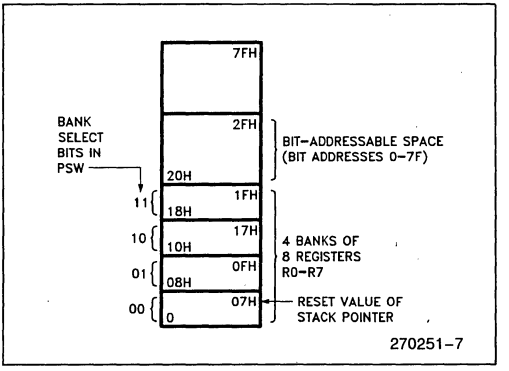


Figure 7. The Lower 128 Bytes of Internal RAM

The Lower 128 bytes of RAM are present in all MCS-51 devices as mapped in Figure 7. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

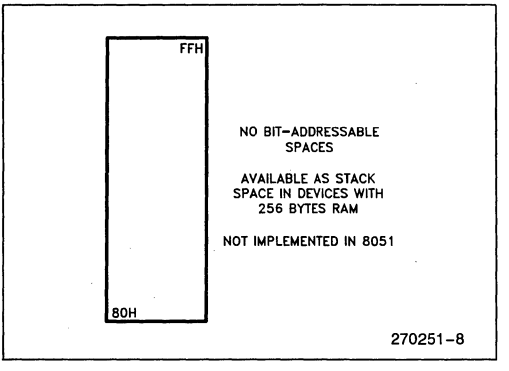


Figure 8. The Upper 128 Bytes of Internal RAM

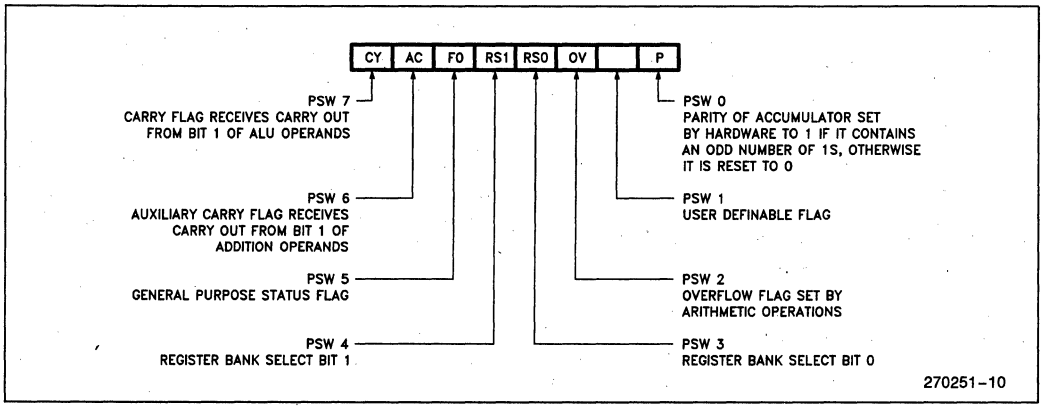


Figure 10. PSW (Program Status Word) Register in MCS®-51 Devices

The next 16 bytes above the register banks form a block of bit-addressable memory space. The MCS-51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 8) can only be accessed by indirect addressing. The Upper 128 bytes of RAM are not implemented in the 8051, but are in the devices with 256 bytes of RAM. (See Table 1).

Figure 9 gives a brief look at the Special Function Register (SFR) space. SFRs include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. In general, all MCS-51 microcontrollers have the same SFRs as the 8051, and at the same addresses in SFR space. However, enhancements to the 8051 have additional SFRs that are not present in the 8051, nor perhaps in other proliferations of the family.

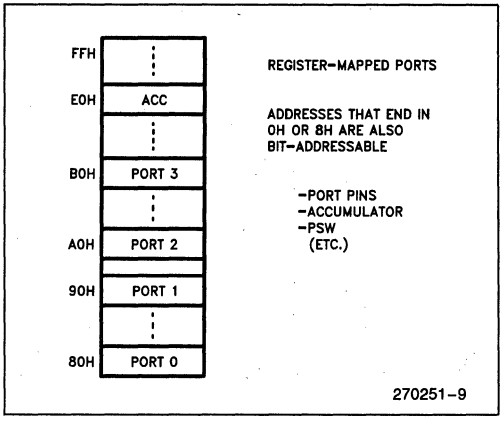


Figure 9. SFR Space

Sixteen addresses in SFR space are both byte- and bit-addressable. The bit-addressable SFRs are those whose address ends in 000B. The bit addresses in this area are 80H through FFH.

THE MCS®-51 INSTRUCTION SET

All members of the MCS-51 family execute the same instruction set. The MCS-51 instruction set is optimized for 8-bit control applications. It provides a variety of fast addressing modes for accessing the internal RAM to facilitate byte operations on small data structures. The instruction set provides extensive support for one-bit variables as a separate data type, allowing direct bit manipulation in control and logic systems that require Boolean processing.

An overview of the MCS-51 instruction set is presented below, with a brief description of how certain instructions might be used. References to "the assembler" in this discussion are to Intel's MCS-51 Macro Assembler, ASM51. More detailed information on the instruction set can be found in the MCS-51 Macro Assembler User's Guide (Order No. 9800937 for ISIS Systems, Order No. 122752 for DOS Systems).

Program Status Word

The Program Status Word (PSW) contains several status bits that reflect the current state of the CPU. The PSW, shown in Figure 10, resides in SFR space. It contains the Carry bit, the Auxiliary Carry (for BCD operations), the two register bank select bits, the Overflow flag, a Parity bit, and two user-definable status flags.

The Carry bit, other than serving the functions of a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

The bits RS0 and RS1 are used to select one of the four register banks shown in Figure 7. A number of instructions refer to these RAM locations as R0 through R7. The selection of which of the four banks is being referred to is made on the basis of the bits RS0 and RS1 at execution time.

The Parity bit reflects the number of 1s in the Accumulator: P = 1 if the Accumulator contains an odd number of 1s, and P = 0 if the Accumulator contains an even number of 1s. Thus the number of 1s in the Accumulator plus P is always even.

Two bits in the PSW are uncommitted and may be used as general purpose status flags.

Addressing Modes

The addressing modes in the MCS-51 instruction set are as follows:

DIRECT ADDRESSING

In direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs can be directly addressed.

INDIRECT ADDRESSING

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be R0 or R1 of the selected register bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit "data pointer" register, DPTR.

REGISTER INSTRUCTIONS

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits in the PSW.

REGISTER-SPECIFIC INSTRUCTIONS

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point to it. The opcode itself does that. Instructions that refer to the Accumlator as A assemble as accumulator-specific opcodes.

IMMEDIATE CONSTANTS

The value of a constant can follow the opcode in Program Memory. For example,

```
MOV A, #100
```

loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.

INDEXED ADDRESSING

Only Program Memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program Memory. A 16-bit base register (either DPTR or the Program Counter) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program Memory is formed by adding the Accumulator data to the base pointer.

Another type of indexed addressing is used in the "case jump" instruction. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

Arithmetic Instructions

The menu of arithmetic instructions is listed in Table 2. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A,<byte> instruction can be written as:

```
ADD     A,7FH       (direct addressing)
ADD     A,@R0       (indirect addressing)
ADD     A,R7        (register addressing)
ADD     A,#127      (immediate constant)
```

The execution times listed in Table 2 assume a 12 MHz clock frequency. All of the arithmetic instructions execute in 1 μs except the INC DPTR instruction, which takes 2 μs, and the Multiply and Divide instructions, which take 4 μs.

Note that any byte in the internal Data Memory space can be incremented or decremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

Table 2. A List of the MCS®-51 Arithmetic Instructions

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
ADD A,<byte>	A = A + <byte>	X	X	X	X	1
ADDC A,<byte>	A = A + <byte> + C	X	X	X	X	1
SUBB A,<byte>	A = A − <byte> − C	X	X	X	X	1
INC A	A = A + 1	Accumulator only				1
INC <byte>	<byte> = <byte> + 1	X	X	X		1
INC DPTR	DPTR = DPTR + 1	Data Pointer only				2
DEC A	A = A − 1	Accumulator only				1
DEC <byte>	<byte> = <byte> − 1	X	X	X		1
MUL AB	B:A = B x A	ACC and B only				4
DIV AB	A = Int [A/B] B = Mod [A/B]	ACC and B only				4
DA A	Decimal Adjust	Accumulator only				1

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

Oddly enough, DIV AB finds less use in arithmetic "divide" routines than in radix conversions and programmable shift operations. An example of the use of DIV AB in a radix conversion will be given later. In shift operations, dividing a number by 2^n shifts its n bits to the right. Using DIV AB to perform the division completes the shift in 4 μs and leaves the B register holding the bits that were shifted out.

The DA A instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DA A operation, to ensure that the result is also in BCD. Note that DA A will not convert a binary number to BCD. The DA A operation produces a meaningful result only as the second step in the addition of two BCD bytes.

Table 3. A List of the MCS®-51 Logical Instructions

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
ANL A,<byte>	A = A .AND. <byte>	X	X	X	X	1
ANL <byte>,A	<byte> = <byte> .AND. A	X				1
ANL <byte>,#data	<byte> = <byte> .AND. #data	X				2
ORL A,<byte>	A = A .OR. <byte>	X	X	X	X	1
ORL <byte>,A	<byte> = <byte> .OR. A	X				1
ORL <byte>,#data	<byte> = <byte> .OR. #data	X				2
XRL A,<byte>	A = A .XOR. <byte>	X	X	X	X	1
XRL <byte>,A	<byte> = <byte> .XOR. A	X				1
XRL <byte>,#data	<byte> = <byte> .XOR. #data	X				2
CRL A	A = 00H	Accumulator only				1
CPL A	A = .NOT. A	Accumulator only				1
RL A	Rotate ACC Left 1 bit	Accumulator only				1
RLC A	Rotate Left through Carry	Accumulator only				1
RR A	Rotate ACC Right 1 bit	Accumulator only				1
RRC A	Rotate Right through Carry	Accumulator only				1
SWAP A	Swap Nibbles in A	Accumulator only				1

Logical Instructions

Table 3 shows the list of MCS-51 logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and <byte> contains 01010011B, then

```
ANL    A,<byte>
```

will leave the Accumulator holding 00010001B.

The addressing modes that can be used to access the <byte> operand are listed in Table 3. Thus, the ANL A,<byte> instruction may take any of the forms

```
ANL    A,7FH      (direct addressing)
ANL    A,@R1      (indirect addressing)
ANL    A,R6       (register addressing)
ANL    A,#53H     (immediate constant)
```

All of the logical instructions that are Accumulator-specific execute in 1μs (using a 12 MHz clock). The others take 2 μs.

Note that Boolean operations can be performed on any byte in the lower 128 internal Data Memory space or the SFR space using direct addressing, without having to use the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in

```
XRL    P1,#0FFH
```

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to stack it in the service routine.

The Rotate instructions (RL A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This is a useful operation in BCD manipulations. For example, if the Accumulator contains a binary number which is known to be less than 100, it can be quickly converted to BCD by the following code:

```
MOV    B,#10
DIV    AB
SWAP   A
ADD    A,B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator, and the ones digit to the low nibble.

Data Transfers

INTERNAL RAM

Table 4 shows the menu of instructions that are available for moving data around within the internal memory spaces, and the addressing modes that can be used with each one. With a 12 MHz clock, all of these instructions execute in either 1 or 2 μs.

The MOV <dest>, <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator. Remember the Upper 128 byes of data RAM can be accessed only by indirect addressing, and SFR space only by direct addressing.

Note that in all MCS-51 devices, the stack resides in on-chip RAM, and grows upwards. The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored,

Table 4. A List of the MCS®-51 Data Transfer Instructions that Access Internal Data Memory Space

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
MOV A,<src>	A = <src>	X	X	X	X	1
MOV <dest>,A	<dest> = A	X	X	X		1
MOV <dest>, <src>	<dest> = <src>	X	X	X	X	2
MOV DPTR,#data16	DPTR = 16-bit immediate constant.				X	2
PUSH <src>	INC SP : MOV "@SP",<src>	X				2
POP <dest>	MOV <dest>, "@SP" : DEC SP	X				2
XCH A,<byte>	ACC and <byte> exchange data	X	X	X		1
XCHD A,@Ri	ACC and @Ri exchange low nibbles		X			1

but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128, if they are implemented, but not into SFR space.

In devices that do not implement the Upper 128, if the SP points to the Upper 128, PUSHed bytes are lost, and POPped bytes are indeterminate.

The Data Transfer instructions include a 16-bit MOV that can be used to initialize the Data Pointer (DPTR) for look-up tables in Program Memory, or for 16-bit external Data Memory accesses.

The XCH A, <byte> instruction causes the Accumulator and addressed byte to exchange data. The XCHD A,@Ri instruction is similar, but only the low nibbles are involved in the exchange.

To see how XCH and XCHD can be used to facilitate data manipulations, consider first the problem of shifting an 8-digit BCD number two digits to the right. Figure 11 shows how this can be done using direct MOVs, and for comparison how it can be done using XCH instructions. To aid in understanding how the code works, the contents of the registers that are holding the BCD number and the content of the Accumulator are shown alongside each instruction to indicate their status after the instruction has been executed.

		2A	2B	2C	2D	2E	ACC
MOV	A,2EH	00	12	34	56	78	78
MOV	2EH,2DH	00	12	34	56	56	78
MOV	2DH,2CH	00	12	34	34	56	78
MOV	2CH,2BH	00	12	12	34	56	78
MOV	2BH,#0	00	00	12	34	56	78

(a) Using direct MOVs: 14 bytes, 9 μs

		2A	2B	2C	2D	2E	ACC
CLR	A	00	12	34	56	78	00
XCH	A,2BH	00	00	34	56	78	12
XCH	A,2CH	00	00	12	56	78	34
XCH	A,2DH	00	00	12	34	78	56
XCH	A,2EH	00	00	12	34	56	78

(b) Using XCHs: 9 bytes, 5 μs

Figure 11. Shifting a BCD Number Two Digits to the Right

After the routine has been executed, the Accumulator contains the two digits that were shifted out on the right. Doing the routine with direct MOVs uses 14 code bytes and 9 μs of execution time (assuming a 12 MHz clock). The same operation with XCHs uses less code and executes almost twice as fast.

To right-shift by an odd number of digits, a one-digit shift must be executed. Figure 12 shows a sample of code that will right-shift a BCD number one digit, using the XCHD instruction. Again, the contents of the registers holding the number and of the Accumulator are shown alongside each instruction.

		2A	2B	2C	2D	2E	ACC
	MOV R1,#2EH	00	12	34	56	78	XX
	MOV R0,#2DH	00	12	34	56	78	XX
	loop for R1 = 2EH:						
LOOP:	MOV A,@R1	00	12	34	56	78	78
	XCHD A,@R0	00	12	34	58	78	76
	SWAP A	00	12	34	58	78	67
	MOV @R1,A	00	12	34	58	67	67
	DEC R1	00	12	34	58	67	67
	DEC R0	00	12	34	58	67	67
	CJNE R1,#2AH,LOOP						
	loop for R1 = 2DH:	00	12	38	45	67	45
	loop for R1 = 2CH:	00	18	23	45	67	23
	loop for R1 = 2BH:	08	01	23	45	67	01
	CLR A	08	01	23	45	67	00
	XCH A,2AH	00	01	23	45	67	08

Figure 12. Shifting a BCD Number One Digit to the Right

First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (Compare and Jump if Not Equal) is a loop control that will be described later.

The loop is executed from LOOP to CJNE for R1 = 2EH, 2DH, 2CH and 2BH. At that point the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the lost digit is moved to the Accumulator.

EXTERNAL RAM

Table 5 shows a list of the Data Transfer instructions that access external Data Memory. Only indirect addressing can be used. The choice is whether to use a one-byte address, @Ri, where Ri can be either R0 or R1 of the selected register bank, or a two-byte address, @DPTR. The disadvantage to using 16-bit addresses if only a few K bytes of external RAM are involved is that 16-bit addresses use all 8 bits of Port 2 as address bus. On the other hand, 8-bit addresses allow one to address a few K bytes of RAM, as shown in Figure 5, without having to sacrifice all of Port 2.

All of these instructions execute in 2 μs, with a 12 MHz clock.

Table 5. A List of the MCS®-51 Data Transfer Instructions that Access External Data Memory Space

Address Width	Mnemonic	Operation	Execution Time (μs)
8 bits	MOVX A,@Ri	Read external RAM @Ri	2
8 bits	MOVX @Ri,A	Write external RAM @Ri	2
16 bits	MOVX A,@DPTR	Read external RAM @DPTR	2
16 bits	MOVX @DPTR,A	Write external RAM @DPTR	2

Note that in all external Data RAM accesses, the Accumulator is always either the destination or source of the data.

The read and write strobes to external RAM are activated only during the execution of a MOVX instruction. Normally these signals are inactive, and in fact if they're not going to be used at all, their pins are available as extra I/O lines. More about that later.

LOOKUP TABLES

Table 6 shows the two instructions that are available for reading lookup tables in Program Memory. Since these instructions access only Program Memory, the lookup tables can only be read, not updated. The mnemonic is MOVC for "move constant".

If the table access is to external Program Memory, then the read strobe is $\overline{PSEN}$.

Table 6. The MCS®-51 Lookup Table Read Instructions

Mnemonic	Operation	Execution Time (μs)
MOVC A,@A+DPTR	Read Pgm Memory at (A+DPTR)	2
MOVC A,@A+PC	Read Pgm Memory at (A+PC)	2

The first MOVC instruction in Table 6 can accommodate a table of up to 256 entries, numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to beginning of the table. Then

```
MOVC    A,@A+DPTR
```

copies the desired table entry into the Accumulator.

The other MOVC instruction works the same way, except the Program Counter (PC) is used as the table base, and the table is accessed through a subroutine. First the number of the desired entry is loaded into the Accumulator, and the subroutine is called:

```
        MOV     A,ENTRY_NUMBER
        CALL    TABLE
```

The subroutine "TABLE" would look like this:

```
TABLE:  MOVC    A,@A+PC
        RET
```

The table itself immediately follows the RET (return) instruction in Program Memory. This type of table can have up to 255 entries, numbered 1 through 255. Number 0 can not be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

Boolean Instructions

MCS-51 devices contain a complete Boolean (single-bit) processor. The internal RAM contains 128 addressable bits, and the SFR space can support up to 128 other addressable bits. All of the port lines are bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR, and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

Table 7. A List of the MCS®-51 Boolean Instructions

Mnemonic		Operation	Execution Time (μs)
ANL	C,bit	C = C .AND. bit	2
ANL	C,/bit	C = C .AND. .NOT. bit	2
ORL	C,bit	C = C .OR. bit	2
ORL	C,/bit	C = C .OR. .NOT. bit	2
MOV	C,bit	C = bit	1
MOV	bit,C	bit = C	2
CLR	C	C = 0	1
CLR	bit	bit = 0	1
SETB	C	C = 1	1
SETB	bit	bit = 1	1
CPL	C	C = .NOT. C	1
CPL	bit	bit = .NOT. bit	1
JC	rel	Jump if C = 1	2
JNC	rel	Jump if C = 0	2
JB	bit,rel	Jump if bit = 1	2
JNB	bit,rel	Jump if bit = 0	2
JBC	bit,rel	Jump if bit = 1; CLR bit	2

The instruction set for the Boolean processor is shown in Table 7. All bit accesses are by direct addressing. Bit addresses 00H through 7FH are in the Lower 128, and bit addresses 80H through FFH are in SFR space.

Note how easily an internal flag can be moved to a port pin:

```
MOV    C,FLAG
MOV    P1.0,C
```

In this example, FLAG is the name of any addressable bit in the Lower 128 or SFR space. An I/O line (the LSB of Port 1, in this case) is set or cleared depending on whether the flag bit is 1 or 0.

The Carry bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry bit as C assemble as Carry-specific instructions (CLR C, etc). The Carry bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

Note that the Boolean instruction set includes ANL and ORL operations, but not the XRL (Exclusive OR) operation. An XRL operation is simple to implement in software. Suppose, for example, it is required to form the Exclusive OR of two bits:

C = bit1 .XRL. bit2

The software to do that could be as follows:

```
        MOV    C,bit1
        JNB    bit2,OVER
        CPL    C
OVER:   (continue)
```

First, bit1 is moved to the Carry. If bit2 = 0, then C now contains the correct result. That is, bit1 .XRL. bit2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1 C now contains the complement of the correct result. It need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a series of bit-test instructions which execute a jump if the addressed bit is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, bit2 is being tested, and if bit2 = 0 the CPL C instruction is jumped over.

JBC executes the jump if the addressed bit is set, and also clears the bit. Thus a flag can be tested and cleared in one operation.

All the PSW bits are directly addressable, so the Parity bit, or the general purpose flags, for example, are also available to the bit-test instructions.

RELATIVE OFFSET

The destination address for these jumps is specified to the assembler by a label or by an actual address in Program Memory. However, the destination address assembles to a **relative** offset byte. This is a signed (two's complement) offset byte which is added to the PC in two's complement arithmetic if the jump is executed.

The range of the jump is therefore −128 to +127 Program Memory bytes relative to the first byte following the instruction.

Jump Instructions

Table 8 shows the list of unconditional jumps.

Table 8. Unconditional Jumps in MCS®-51 Devices

Mnemonic	Operation	Execution Time (μs)
JMP addr	Jump to addr	2
JMP @A+DPTR	Jump to A+DPTR	2
CALL addr	Call subroutine at addr	2
RET	Return from subroutine	2
RETI	Return from interrupt	2
NOP	No operation	1

The Table lists a single "JMP addr" instruction, but in fact there are three—SJMP, LJMP and AJMP—which differ in the format of the destination address. JMP is a generic mnemonic which can be used if the programmer does not care which way the jump is encoded.

The SJMP instruction encodes the destination address as a relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to a range of −128 to +127 bytes relative to the instruction following the SJMP.

The LJMP instruction encodes the destination address as a 16-bit constant. The instruction is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64K Program Memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substituted for the low 11 bits in the PC. The high 5 bits stay the same. Hence the destination has to be within the same 2K block as the instruction following the AJMP.

In all cases the programmer specifies the destination address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the destination address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a "Destination out of range" message is written into the List file.

The JMP @A+DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically, DPTR is set up with the address of a jump table, and the Accumulator is given an index to the table. In a 5-way branch, for example, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows:

```
MOV     DPTR,#JUMP_TABLE
MOV     A,INDEX_NUMBER
RL      A
JMP     @A+DPTR
```

The RL A instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long:

```
JUMP_TABLE:
        AJMP    CASE_0
        AJMP    CASE_1
        AJMP    CASE_2
        AJMP    CASE_3
        AJMP    CASE_4
```

Table 8 shows a single "CALL addr" instruction, but there are two of them—LCALL and ACALL—which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64K Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2K block as the instruction following the ACALL.

In any case the programmer specifies the subroutine address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done. If there is no interrupt in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 9 shows the list of conditional jumps available to the MCS-51 user. All of these jumps specify the destination address by the relative offset method, and so are limited to a jump distance of −128 to +127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual destination address the same way as the other jumps: as a label or a 16-bit constant.

Table 9. Conditional Jumps in MCS®-51 Devices

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
JZ rel	Jump if A = 0	Accumulator only				2
JNZ rel	Jump if A ≠ 0	Accumulator only				2
DJNZ <byte>,rel	Decrement and jump if not zero	X		X		2
CJNE A,<byte>,rel	Jump if A ≠ <byte>	X			X	2
CJNE <byte>,#data,rel	Jump if <byte> ≠ #data		X	X		2

There is no Zero bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10:

```
        MOV     COUNTER,#10
LOOP:   (begin loop)
          *
          *
          *
        (end loop)
        DJNZ    COUNTER,LOOP
        (continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Figure 12. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 12, the two bytes were the data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop was executed, R1 was decremented, and the looping was to continue until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

CPU TIMING

All MCS-51 microcontrollers have an on-chip oscillator which can be used if desired as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller, and capacitors to ground as shown in Figure 13.

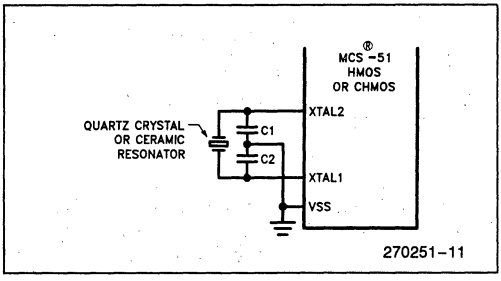


Figure 13. Using the On-Chip Oscillator

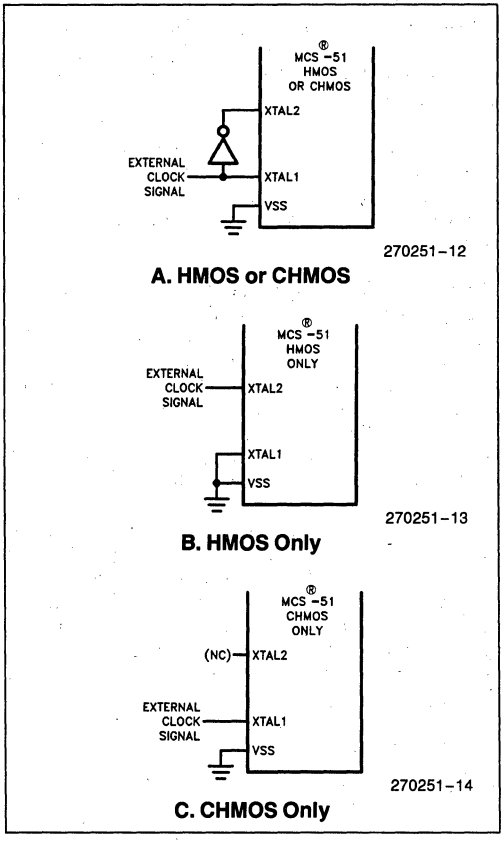


Figure 14. Using an External Clock

Examples of how to drive the clock with an external oscillator are shown in Figure 14. Note that in the HMOS devices (8051, etc.) the signal at the XTAL2 pin actually drives the internal clock generator. In the CHMOS devices (80C51BH, etc.) the signal at the XTAL1 pin drives the internal clock generator. If only one pin is going to be driven with the external oscillator signal, make sure it is the right pin.

The internal clock generator defines the sequence of states that make up the MCS-51 machine cycle.

Machine Cycles

A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1 μs if the oscillator frequency is 12 MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 15 shows the fetch/execute sequences in

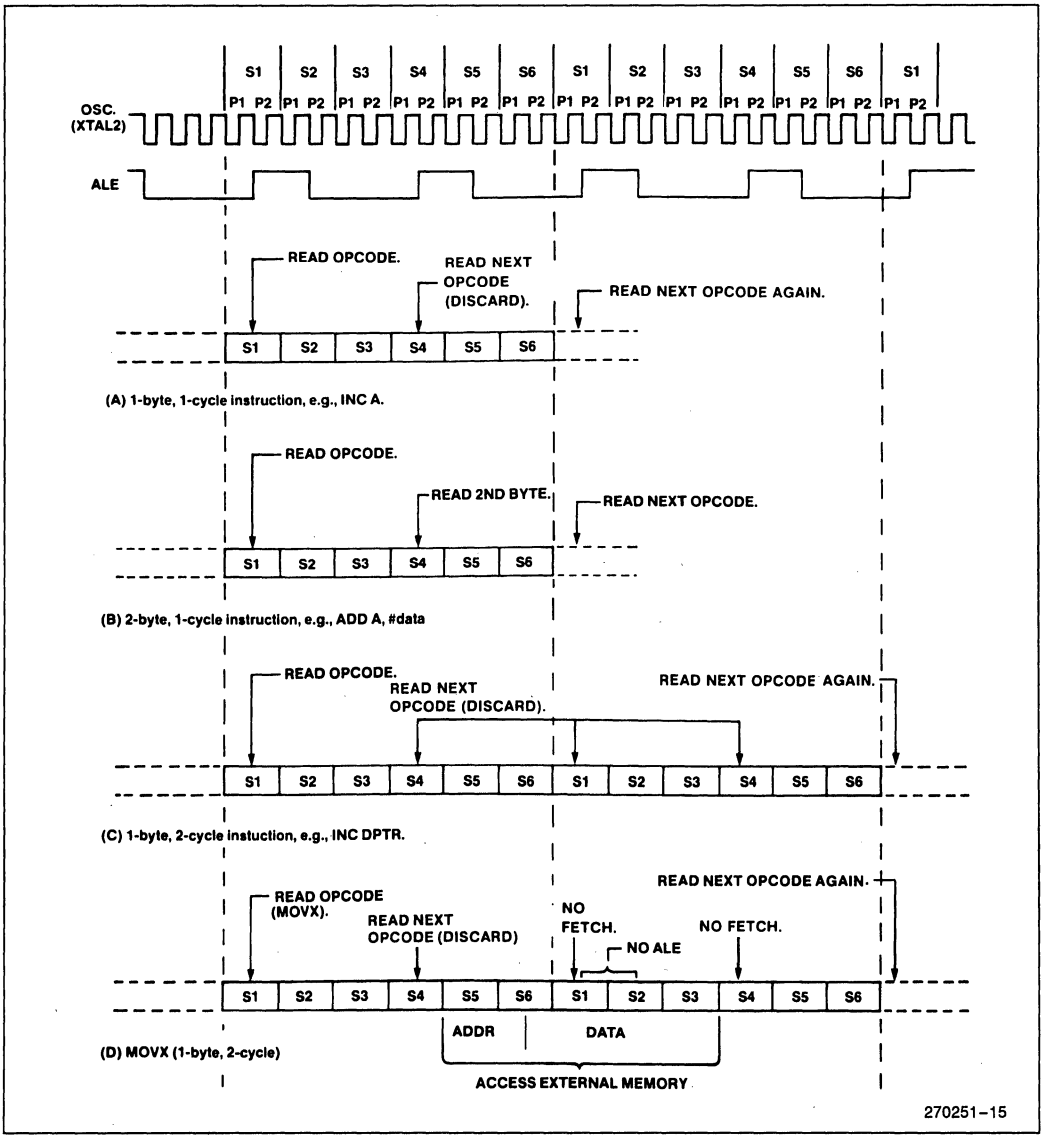


Figure 15. State Sequences in MCS®-51 Devices

states and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed doesn't require it. If the instruction being executed doesn't need more code bytes, the CPU simply ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figure 15A and B) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is complete at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program fetch is generated during the second cycle of a MOVX instruction. This is the only time program fetches are skipped. The fetch/execute sequence for MOVX instructions is shown in Figure 15(D).

The fetch/execute sequences are the same whether the Program Memory is internal or external to the chip. Execution times do not depend on whether the Program Memory is internal or external.

Figure 16 shows the signals and timing involved in program fetches when the Program Memory is external. If Program Memory is external, then the Program Memory read strobe $\overline{PSEN}$ is normally activated twice per machine cycle, as shown in Figure 16(A).

If an access to external Data Memory occurs, as shown in Figure 16(B), two $\overline{PSEN}$s are skipped, because the address and data bus are being used for the Data Memory access.

Note that a Data Memory bus cycle takes twice as much time as a Program Memory bus cycle. Figure 16 shows the relative timing of the addresses being emitted at Ports 0 and 2, and of ALE and $\overline{PSEN}$. ALE is used to latch the low address byte from P0 into the address latch.

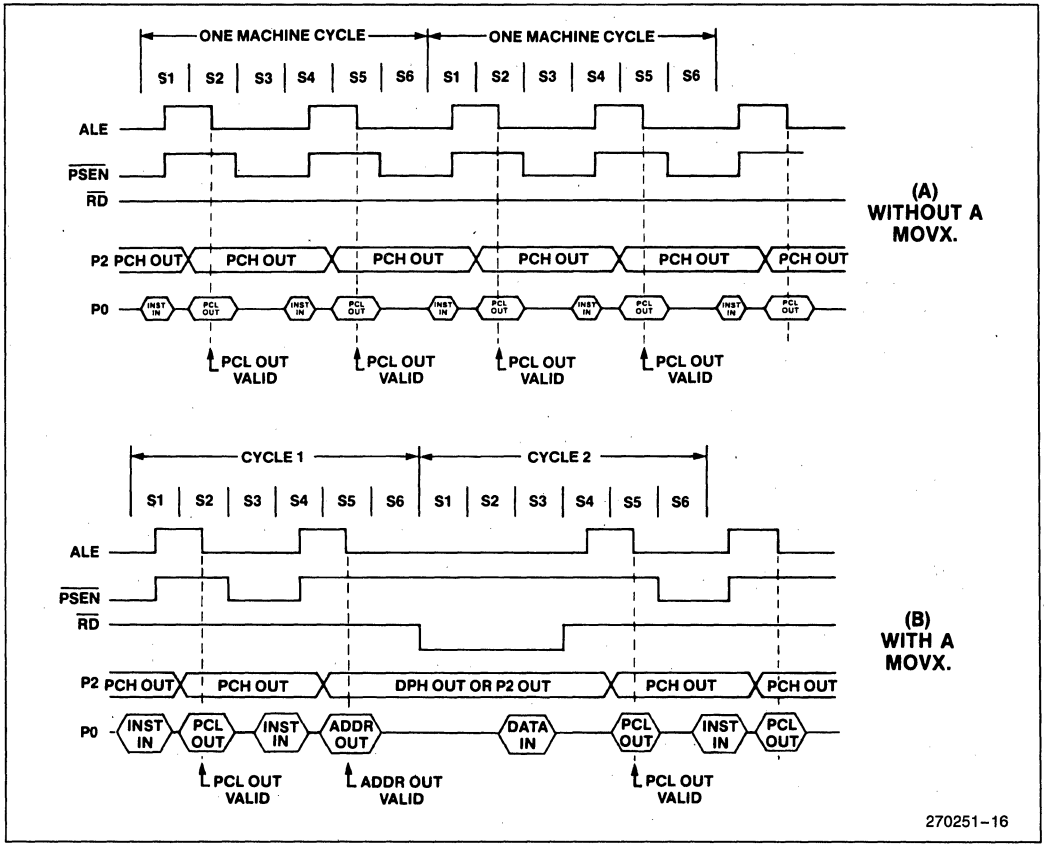


Figure 16. Bus Cycles in MCS®-51 Devices Executing from External Program Memory

When the CPU is executing from internal Program Memory, $\overline{PSEN}$ is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and so is available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

Interrupt Structure

The 8051 core provides 5 interrupt sources: 2 external interrupts, 2 timer interrupts, and the serial port interrupt. What follows is an overview of the interrupt structure for the 8051. Other MCS-51 devices have additional interrupt sources and vectors as shown in Table 1. Refer to the appropriate chapters on other devices for further information on their interrupts.

INTERRUPT ENABLES

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the SFR named IE (Interrupt Enable). This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 17 shows the IE register for the 8051.

(MSB)							(LSB)
EA	—	—	ES	ET1	EX1	ET0	EX0

Enable bit = 1 enables the interrupt.
Enable bit = 0 disables it.

Symbol	Position	Function
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	reserved*
—	IE.5	reserved*
ES	IE.4	Serial Port Interrupt enable bit.
ET1	IE.3	Timer 1 Overflow Interrupt enable bit.
EX1	IE.2	External Interrupt 1 enable bit.
ET0	IE.1	Timer 0 Overflow Interrupt enable bit.
EX0	IE.0	External Interrupt 0 enable bit.

*These reserved bits are used in other MCS-51 devices.

Figure 17. IE (Interrupt Enable) Register in the 8051

INTERRUPT PRIORITIES

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in the SFR named IP (Interrupt Priority). Figure 18 shows the IP register in the 8051.

A low-priority interrrupt can be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence.

Figure 19 shows, for the 8051, how the IE and IP registers and the polling sequence work to determine which if any interrupt will be serviced.

(MSB)							(LSB)
—	—	—	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority.
Priority bit = 0 assigns low priority.

Symbol	Position	Function
—	IP.7	reserved*
—	IP.6	reserved*
—	IP.5	reserved*
PS	IP.4	Serial Port interrupt priority bit.
PT1	IP.3	Timer 1 interrupt priority bit.
PX1	IP.2	External Interrupt 1 priority bit.
PT0	IP.1	Timer 0 interrupt priority bit.
PX0	IP.0	External Interrupt 0 priority bit.

*These reserved bits are used in other MCS-51 devices.

Figure 18. IP (Interrupt Priority) Register in the 8051

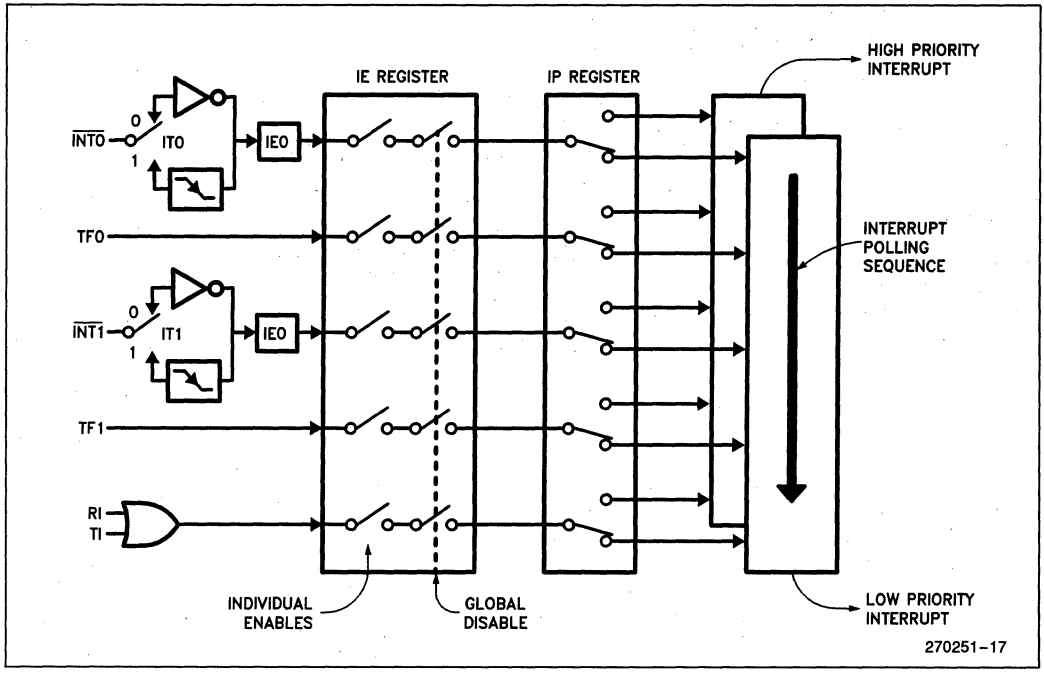


Figure 19. 8051 Interrupt Control System

In operation, all the interrupt flags are latched into the interrupt control system during State 5 of every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is found to be set (1), the interrupt system generates an LCALL to the appropriate location in Program Memory, unless some other condition blocks the interrupt. Several conditions can block an interrupt, among them that an interrupt of equal or higher priority level is already in progress.

The hardware-generated LCALL causes the contents of the Program Counter to be pushed onto the stack, and reloads the PC with the beginning address of the service routine. As previously noted (Figure 3), the service routine for each interrupt begins at a fixed location.

Only the Program Counter is automatically pushed onto the stack, not the PSW or any other register. Having only the PC be automatically saved allows the programmer to decide how much time to spend saving which other registers. This enhances the interrupt response time, albeit at the expense of increasing the programmer's burden of responsibility. As a result, many interrupt functions that are typical in control applications—toggling a port pin, for example, or reloading a timer, or unloading a serial buffer—can often be completed in less time than it takes other architectures to commence them.

SIMULATING A THIRD PRIORITY LEVEL IN SOFTWARE

Some applications require more than the two priority levels that are provided by on-chip hardware in MCS-51 devices. In these cases, relatively simple software can be written to produce the same effect as a third priority level.

First, interrupts that are to have higher priority than 1 are assigned to priority 1 in the IP (Interrupt Priority) register. The service routines for priority 1 interrupts that are supposed to be interruptible by "priority 2" interrupts are written to include the following code:

```
        PUSH    IE
        MOV     IE,#MASK
        CALL    LABEL
          *******
(execute service routine)
          *******
        POP     IE
        RET
LABEL:  RETI
```

As soon as any priority 1 interrupt is acknowledged, the IE (Interrupt Enable) register is re-defined so as to disable all but "priority 2" interrupts. Then, a CALL to LABEL executes the RETI instruction, which clears the priority 1 interrupt-in-progress flip-flop. At this point any priority 1 interrupt that is enabled can be serviced, but only "priority 2" interrupts are enabled.

POPping IE restores the original enable byte. Then a normal RET (rather than another RETI) is used to terminate the service routine. The additional software adds 10 μs (at 12 MHz) to priority 1 interrupts.

ADDITIONAL REFERENCES

The following application notes are found in the *Embedded Control Applications* handbook. (Order Number: 270648)

1. AP-69 "An Introduction to the Intel MCS®-51 Single-Chip Microcomputer Family"
2. AP-70 "Using the Intel MCS®-51 Boolean Processing Capabilities"

MCS®-51 Programmer's Guide and Instruction Set

6

MCS®-51 PROGRAMMER'S GUIDE AND INSTRUCTION SET

The information presented in this chapter is collected from the MCS®-51 Architectural Overview and the Hardware Description of the 8051, 8052 and 80C51 chapters of this book. The material has been selected and rearranged to form a quick and convenient reference for the programmers of the MCS-51. This guide pertains specifically to the 8051, 8052 and 80C51.

The following list should make it easier to find a subject in this chapter.

MEMORY ORGANIZATION

PROGRAM MEMORY

The 8051 has separate address spaces for Program Memory and Data Memory. The Program Memory can be up to 64K bytes long. The lower 4K (8K for the 8052) may reside on-chip.

Figure 1 shows a map of the 8051 program memory, and Figure 2 shows a map of the 8052 program memory.

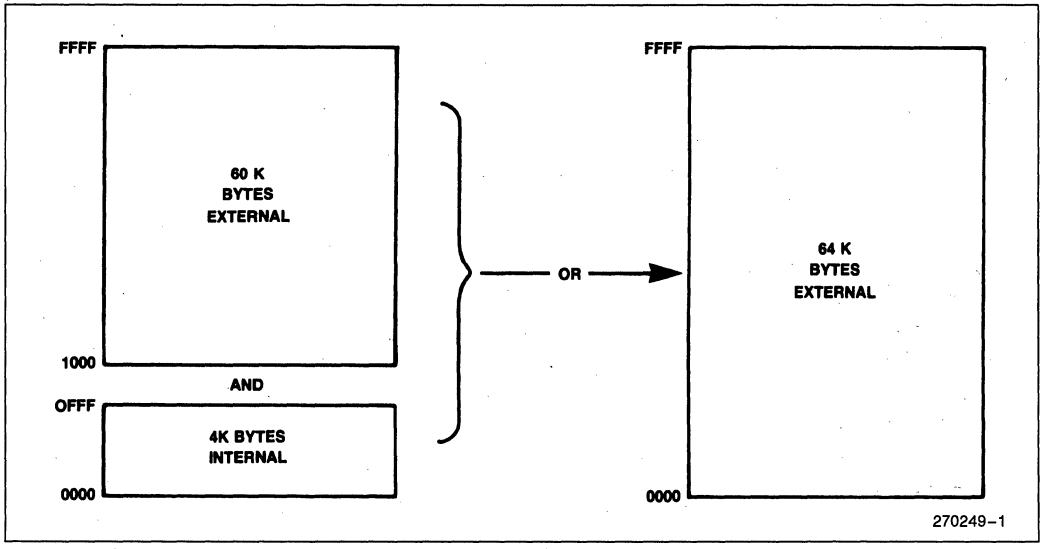


Figure 1. The 8051 Program Memory

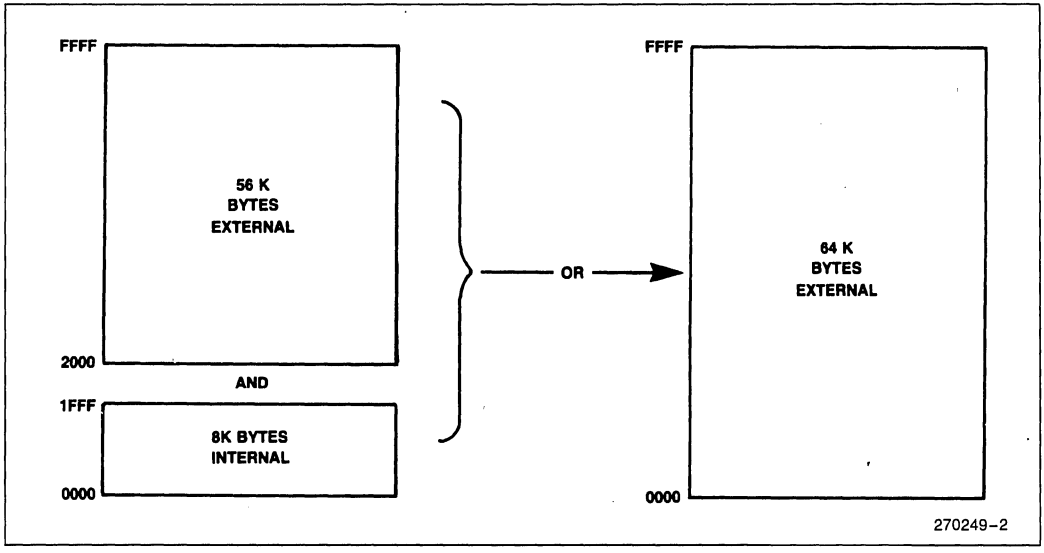


Figure 2. The 8052 Program Memory

Data Memory:

The 8051 can address up to 64K bytes of Data Memory external to the chip. The "MOVX" instruction is used to access the external data memory. (Refer to the MCS-51 Instruction Set, in this chapter, for detailed description of instructions).

The 8051 has 128 bytes of on-chip RAM (256 bytes in the 8052) plus a number of Special Function Registers (SFRs). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr) or by indirect addressing (MOV @Ri). Figure 3 shows the 8051 and the 8052 Data Memory organization.

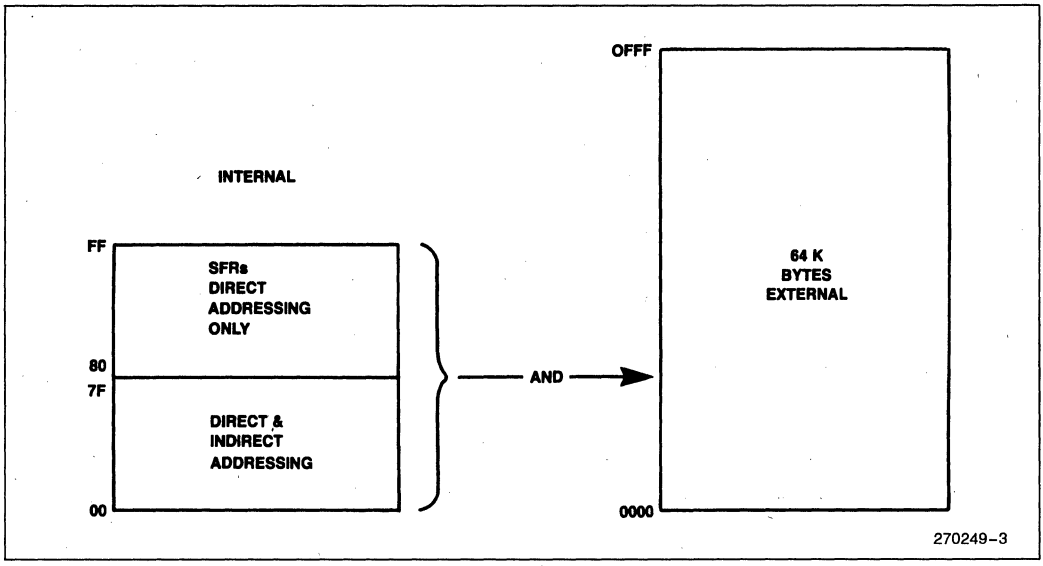


Figure 3a. The 8051 Data Memory

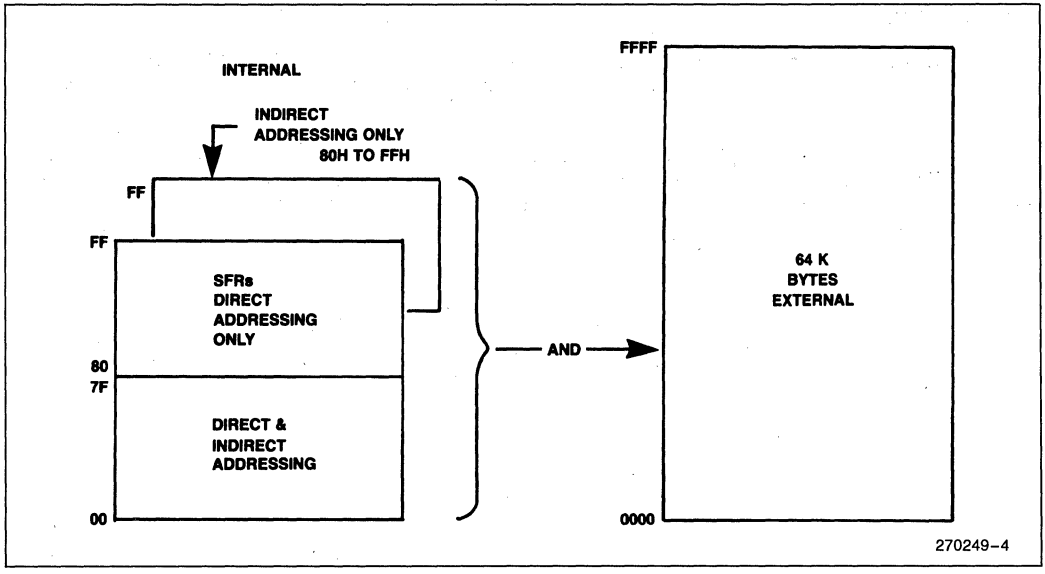


Figure 3b. The 8052 Data Memory

INDIRECT ADDRESS AREA:

Note that in Figure 3b the SFRs and the indirect address RAM have the same addresses (80H–0FFH). Nevertheless, they are two separate areas and are accessed in two different ways.

For example the instruction

```
MOV    80H,#0AAH
```

writes 0AAH to Port 0 which is one of the SFRs and the instruction

```
MOV    R0,#80H

MOV    @R0,#0BBH
```

writes 0BBH in location 80H of the data RAM. Thus, after execution of both of the above instructions Port 0 will contain 0AAH and location 80 of the RAM will contain 0BBH.

Note that the stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space in those devices which implement 256 bytes of internal RAM.

DIRECT AND INDIRECT ADDRESS AREA:

The 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into 3 segments as listed below and shown in Figure 4.

1. Register Banks 0-3: Locations 0 through 1FH (32 bytes). ASM-51 and the device after reset default to register bank 0. To use the other register banks the user must select them in the software (refer to the MCS-51 Micro Assembler User's Guide). Each register bank contains 8 one-byte registers, 0 through 7.

Reset initializes the Stack Pointer to location 07H and it is incremented once to start from location 08H which is the first register (RO) of the second register bank. Thus, in order to use more than one register bank, the SP should be intialized to a different location of the RAM where it is not used for data storage (ie, higher part of the RAM).

2. Bit Addressable Area: 16 bytes have been assigned for this segment, 20H-2FH. Each one of the 128 bits of this segment can be directly addressed (0-7FH).

The bits can be referred to in two ways both of which are acceptable by the ASM-51. One way is to refer to their addresses, ie. 0 to 7FH. The other way is with reference to bytes 20H to 2FH. Thus, bits 0–7 can also be referred to as bits 20.0–20.7, and bits 8-FH are the same as 21.0–21.7 and so on.

Each of the 16 bytes in this segment can also be addressed as a byte.

3. Scratch Pad Area: Bytes 30H through 7FH are available to the user as data RAM. However, if the stack pointer has been initialized to this area, enough number of bytes should be left aside to prevent SP data destruction.

Figure 4 shows the different segments of the on-chip RAM.

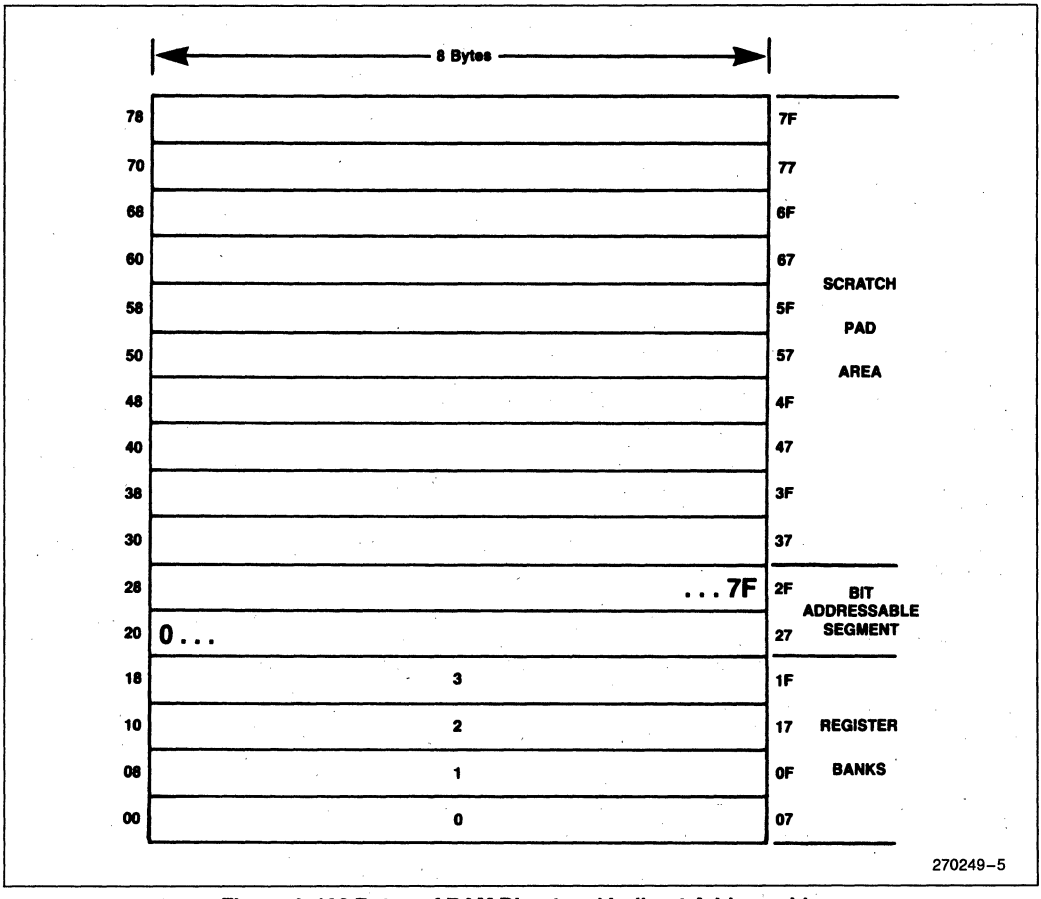


Figure 4. 128 Bytes of RAM Direct and Indirect Addressable

SPECIAL FUNCTION REGISTERS:

Table 1 contains a list of all the SFRs and their addresses.

Comparing Table 1 and Figure 5 shows that all of the SFRs that are byte and bit addressable are located on the first column of the diagram in Figure 5.

Table 1

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

* = Bit addressable
+ = 8052 only

WHAT DO THE SFRs CONTAIN JUST AFTER POWER-ON OR A RESET?

Table 2 lists the contents of each SFR after power-on or a hardware reset.

Table 2. Contents of the SFRs after reset

Register	Value in Binary
*ACC	00000000
*B	00000000
*PSW	00000000
SP	00000111
DPTR	
DPH	00000000
DPL	00000000
*P0	11111111
*P1	11111111
*P2	11111111
*P3	11111111
*IP	8051 XXX00000, 8052 XX000000
*IE	8051 0XX00000, 8052 0X000000
TMOD	00000000
*TCON	00000000
*+T2CON	00000000
TH0	00000000
TL0	00000000
TH1	00000000
TL1	00000000
+TH2	00000000
+TL2	00000000
+RCAP2H	00000000
+RCAP2L	00000000
*SCON	00000000
SBUF	Indeterminate
PCON	HMOS 0XXXXXXX CHMOS 0XXX0000

X = Undefined
* = Bit Addressable
\+ = 8052 only

SFR MEMORY MAP

	8 Bytes								
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

↑ Bit Addressable

Figure 5

Those SFRs that have their bits assigned for various functions are listed in this section. A brief description of each bit is provided for quick reference. For more detailed information refer to the Architecture Chapter of this book.

PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	F0	RS1	RS0	OV	—	P

CY PSW.7 Carry Flag.

AC PSW.6 Auxiliary Carry Flag.

F0 PSW.5 Flag 0 available to the user for general purpose.

RS1 PSW.4 Register Bank selector bit 1 (SEE NOTE 1).

RS0 PSW.3 Register Bank selector bit 0 (SEE NOTE 1).

OV PSW.2 Overflow Flag.

— PSW.1 User definable flag.

P PSW.0 Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bits in the accumulator.

NOTE:
1. The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.

SMOD	—	—	—	GF1	GF0	PD	IDL

SMOD Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.

— Not implemented, reserved for future use.*

— Not implemented, reserved for future use.*

— Not implemented, reserved for future use.*

GF1 General purpose flag bit.

GF0 General purpose flag bit.

PD Power Down bit. Setting this bit activates Power Down operation in the 80C51BH. (Available only in CHMOS).

IDL Idle Mode bit. Setting this bit activates Idle Mode operation in the 80C51BH. (Available only in CHMOS).

If 1s are written to PD and IDL at the same time, PD takes precedence.

*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

INTERRUPTS:

In order to use any of the interrupts in the MCS-51, the following three steps must be taken.

1. Set the EA (enable all) bit in the IE register to 1.
2. Set the corresponding individual interrupt enable bit in the IE register to 1.
3. Begin the interrupt service routine at the corresponding Vector Address of that interrupt. See Table below.

Interrupt Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI & TI	0023H
TF2 & EXF2	002BH

In addition, for external interrupts, pins $\overline{INT0}$ and $\overline{INT1}$ (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 1.

ITx = 0 level activated

ITx = 1 transition activated

IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	—	ET2	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Not implemented, reserved for future use.*
ET2	IE.5	Enable or disable the Timer 2 overflow or capture interrupt (8052 only).
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External Interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

ASSIGNING HIGHER PRIORITY TO ONE OR MORE INTERRUPTS:

In order to assign higher priority to an interrupt the corresponding bit in the IP register must be set to 1.

Remember that while an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.

PRIORITY WITHIN LEVEL:

Priority within level is only to resolve simultaneous requests of the same priority level.

From high to low, interrupt sources are listed below:

IE0
TF0
IE1
TF1
RI or TI
TF2 or EXF2

IP: INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority.

—	—	PT2	PS	PT1	PX1	PT0	PX0

— IP. 7 Not implemented, reserved for future use.*
— IP. 6 Not implemented, reserved for future use.*
PT2 IP. 5 Defines the Timer 2 interrupt priority level (8052 only).
PS IP. 4 Defines the Serial Port interrupt priority level.
PT1 IP. 3 Defines the Timer 1 interrupt priority level.
PX1 IP. 2 Defines External Interrupt 1 priority level.
PT0 IP. 1 Defines the Timer 0 interrupt priority level.
PX0 IP. 0 Defines the External Interrupt 0 priority level.

*User software should not write 1s to reserved bits. These bits may be used in future MCS-51 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 TCON. 7 Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.

TR1 TCON. 6 Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.

TF0 TCON. 5 Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.

TR0 TCON. 4 Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.

IE1 TCON. 3 External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.

IT1 TCON. 2 Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

IE0 TCON. 1 External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.

IT0 TCON. 0 Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
TIMER 1				TIMER 0			

GATE When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).

C/$\overline{T}$ Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

NOTE 1:

M1	M0		Operating Mode
0	0	0	13-bit Timer (MCS-48 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

TIMER SET-UP

Tables 3 through 6 give some values for TMOD which can be used to set up Timer 0 in different modes.

It is assumed that only one timer is being used at a time. If it is desired to run Timers 0 and 1 simultaneously, in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if it is desired to run Timer 0 in mode 1 GATE (external control), and Timer 1 in mode 2 COUNTER, then the value that must be loaded into TMOD is 69H (09H from Table 3 ORed with 60H from Table 6).

Moreover, it is assumed that the user, at this point, is not ready to turn the timers on and will do that at a different point in the program by setting bit TRx (in TCON) to 1.

TIMER/COUNTER 0

As a Timer:

Table 3

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	08H
1	16-bit Timer	01H	09H
2	8-bit Auto-Reload	02H	0AH
3	two 8-bit Timers	03H	0BH

As a Counter:

Table 4

MODE	COUNTER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	04H	0CH
1	16-bit Timer	05H	0DH
2	8-bit Auto-Reload	06H	0EH
3	one 8-bit Counter	07H	0FH

NOTES:
1. The Timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1 to 0 transition on $\overline{INT0}$ (P3.2) when TR0 = 1 (hardware control).

TIMER/COUNTER 1

As a Timer:

Table 5

MODE	TIMER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	80H
1	16-bit Timer	10H	90H
2	8-bit Auto-Reload	20H	A0H
3	does not run	30H	B0H

As a Counter:

Table 6

MODE	COUNTER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	not available	—	—

NOTES:
1. The Timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1 to 0 transition on $\overline{INT1}$ (P3.3) when TR1 = 1 (hardware control).

T2CON: TIMER/COUNTER 2 CONTROL REGISTER. BIT ADDRESSABLE

8052 Only

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

TF2 T2CON. 7 Timer 2 overflow flag set by hardware and cleared by software. TF2 cannot be set when either RCLK = 1 or CLK = 1

EXF2 T2CON. 6 Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX, and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.

RCLK T2CON. 5 Receive clock flag. When set, causes the Serial Port to use Timer 2 overflow pulses for its receive clock in modes 1 & 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.

TLCK T2CON. 4 Transmit clock flag. When set, causes the Serial Port to use Timer 2 overflow pulses for its transmit clock in modes 1 & 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.

EXEN2 T2CON. 3 Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of negative transition on T2EX if Timer 2 is not being used to clock the Serial Port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.

TR2 T2CON. 2 Software START/STOP control for Timer 2. A logic 1 starts the Timer.

C/$\overline{T2}$ T2CON. 1 Timer or Counter select.

0 = Internal Timer. 1 = External Event Counter (falling edge triggered).

CP/$\overline{RL2}$ T2CON. 0 Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, Auto-Reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the Timer is forced to Auto-Reload on Timer 2 overflow.

TIMER/COUNTER 2 SET-UP

Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set, separately, to turn the Timer on.

As a Timer:

Table 7

MODE	T2CON	
	INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
16-bit Auto-Reload	00H	08H
16-bit Capture	01H	09H
BAUD rate generator receive & transmit same baud rate	34H	36H
receive only	24H	26H
transmit only	14H	16H

As a Counter:

Table 8

MODE	TMOD	
	INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
16-bit Auto-Reload	02H	0AH
16-bit Capture	03H	0BH

NOTES:
1. Capture/Reload occurs only on Timer/Counter overflow.
2. Capture/Reload occurs on Timer/Counter overflow and a 1 to 0 transition on T2EX (P1.1) pin except when Timer 2 is used in the baud rate generating mode.

SCON: SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 SCON. 7 Serial Port mode specifier. (NOTE 1).

SM1 SCON. 6 Serial Port mode specifier. (NOTE 1).

SM2 SCON. 5 Enables the multiprocessor communication feature in modes 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 9).

REN SCON. 4 Set/Cleared by software to Enable/Disable reception.

TB8 SCON. 3 The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.

RB8 SCON. 2 In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.

TI SCON. 1 Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.

RI SCON. 0 Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc./64 OR Fosc./32
1	1	3	9-Bit UART	Variable

SERIAL PORT SET-UP:

Table 9

MODE	SCON	SM2 VARIATION
0	10H	Single Processor Environment (SM2 = 0)
1	50H	
2	90H	
3	D0H	
0	NA	Multiprocessor Environment (SM2 = 1)
1	70H	
2	B0H	
3	F0H	

GENERATING BAUD RATES

Serial Port in Mode 0:

Mode 0 has a fixed baud rate which is 1/12 of the oscillator frequency. To run the serial port in this mode none of the Timer/Counters need to be set up. Only the SCON register needs to be defined.

$$\text{Baud Rate} = \frac{\text{Osc Freq}}{12}$$

Serial Port in Mode 1:

Mode 1 has a variable baud rate. The baud rate can be generated by either Timer 1 or Timer 2 (8052 only).

USING TIMER/COUNTER 1 TO GENERATE BAUD RATES:

For this purpose, Timer 1 is used in mode 2 (Auto-Reload). Refer to Timer Setup section of this chapter.

$$\text{Baud Rate} = \frac{\text{K x Oscillator Freq.}}{\text{32 x 12 x [256 - (TH1)]}}$$

If SMOD = 0, then K = 1.
If SMOD = 1, then K = 2. (SMOD is the PCON register).

Most of the time the user knows the baud rate and needs to know the reload value for TH1.
Therefore, the equation to calculate TH1 can be written as:

$$\text{TH1} = 256 - \frac{\text{K x Osc Freq.}}{\text{384 x baud rate}}$$

TH1 must be an integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. In this case, the user may have to choose another crystal frequency.

Since the PCON register is not bit addressable, one way to set the bit is logical ORing the PCON register. (ie, ORL PCON,#80H). The address of PCON is 87H.

USING TIMER/COUNTER 2 TO GENERATE BAUD RATES:

For this purpose, Timer 2 must be used in the baud rate generating mode. Refer to Timer 2 Setup Table in this chapter. If Timer 2 is being clocked through pin T2 (P1.0) the baud rate is:

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

And if it is being clocked internally the baud rate is:

$$\text{Baud Rate} = \frac{\text{Osc Freq}}{\text{32 x [65536 - (RCAP2H, RCAP2L)]}}$$

To obtain the reload value for RCAP2H and RCAP2L the above equation can be rewritten as:

$$\text{RCAP2H, RCAP2L} = 65536 - \frac{\text{Osc Freq}}{\text{32 x Baud Rate}}$$

SERIAL PORT IN MODE 2:

The baud rate is fixed in this mode and is $1/32$ or $1/64$ of the oscillator frequency depending on the value of the SMOD bit in the PCON register.

In this mode none of the Timers are used and the clock comes from the internal phase 2 clock.

SMOD = 1, Baud Rate = $1/32$ Osc Freq.

SMOD = 0, Baud Rate = $1/64$ Osc Freq.

To set the SMOD bit: ORL PCON,#80H. The address of PCON is 87H.

SERIAL PORT IN MODE 3:

The baud rate in mode 3 is variable and sets up exactly the same as in mode 1.

MCS®-51 INSTRUCTION SET

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings(1)

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C,bit	X		
DA	X			ORL C,bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

(1)Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Note on instruction set and addressing modes:

Rn — Register R7–R0 of the currently selected Register Bank.

direct — 8-bit internal data location's address. This could be an Internal Data RAM location (0–127) or a SFR [i.e., I/O port, control register, status register, etc. (128–255)].

@Ri — 8-bit internal data RAM location (0–255) addressed indirectly through register R1 or R0.

#data — 8-bit constant included in instruction.

#data 16 — 16-bit constant included in instruction.

addr 16 — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.

addr 11 — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.

rel — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to +127 bytes relative to first byte of the following instruction.

bit — Direct Addressed bit in Internal Data RAM or Special Function Register.

Table 10. 8051 Instruction Set Summary

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,Rn	Subtract Register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment direct RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (Continued)				
INC	DPTR	Increment Data-Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12

Mnemonic		Description	Byte	Oscillator Period
LOGICAL OPERATIONS (Continued)				
RL	A	Rotate Accumulator Left	1	12
RLC	A	Rotate Accumulator Left through the Carry	1	12
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	Rn,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
DATA TRANSFER (Continued)				
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX	@Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order Digit indirect RAM with Acc	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to CARRY	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)				
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24
CJNE	Rn,#data,rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	Rn,rel	Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12

Table 11. Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,#data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr, code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr,A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

Table 11. Instruction Opcodes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

Table 11. Instruction Opcodes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

INSTRUCTION DEFINITIONS

ACALL addr11

Function: Absolute Call

Description: ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the Stack Pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, opcode bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2K block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

Example: Initially SP equals 07H. The label "SUBRTN" is at program memory location 0345 H. After executing the instruction,

ACALL SUBRTN

at location 0123H, SP will contain 09H, internal RAM locations 08H and 09H will contain 25H and 01H, respectively, and the PC will contain 0345H.

Bytes: 2

Cycles: 2

Encoding:

a10 a9 a8 1	0 0 0 1	a7 a6 a5 a4	a3 a2 a1 a0

Operation: ACALL
(PC) ← (PC) + 2
(SP) ← (SP) + 1
((SP)) ← (PC_{7-0})
(SP) ← (SP) + 1
((SP)) ← (PC_{15-8})
(PC_{10-0}) ← page address

ADD A,<src-byte>

Function: Add

Description: ADD adds the byte variable indicated to the Accumulator, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occured.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B). The instruction,

ADD A,R0

will leave 6DH (01101101B) in the Accumulator with the AC flag cleared and both the carry flag and OV set to 1.

ADD A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 0 1 0	1 r r r

Operation: ADD
(A) ← (A) + (Rn)

ADD A,direct

Bytes: 2

Cycles: 1

Encoding:

0 0 1 0	0 1 0 1	direct address

Operation: ADD
(A) ← (A) + (direct)

ADD A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 0 1 0	0 1 1 i

Operation: ADD
(A) ← (A) + ((Ri))

ADD A,#data

Bytes: 2

Cycles: 1

Encoding:

0 0 1 0	0 1 0 0	immediate data

Operation: ADD
(A) ← (A) + #data

ADDC A,<src-byte>

Function: Add with Carry

Description: ADDC simultaneously adds the byte variable indicated, the carry flag and the Accumulator contents, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occured.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) with the carry flag set. The instruction,

ADDC A,R0

will leave 6EH (01101110B) in the Accumulator with AC cleared and both the Carry flag and OV set to 1.

ADDC A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 0 1 1	1 r r r

Operation: ADDC
(A) ← (A) + (C) + (R_n)

ADDC A,direct

Bytes: 2

Cycles: 1

Encoding:

0 0 1 1	0 1 0 1	direct address

Operation: ADDC
(A) ← (A) + (C) + (direct)

ADDC A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 0 1 1	0 1 1 i

Operation: ADDC
(A) ← (A) + (C) + ((R_i))

ADDC A,#data

Bytes: 2

Cycles: 1

Encoding:

0 0 1 1	0 1 0 0	immediate data

Operation: ADDC
(A) ← (A) + (C) + #data

AJMP addr11

Function: Absolute Jump

Description: AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (*after* incrementing the PC twice), opcode bits 7-5, and the second byte of the instruction. The destination must therefore be within the same 2K block of program memory as the first byte of the instruction following AJMP.

Example: The label "JMPADR" is at program memory location 0123H. The instruction,

```
AJMP   JMPADR
```

is at location 0345H and will load the PC with 0123H.

Bytes: 2

Cycles: 2

Encoding:

a10 a9 a8 0	0 0 0 1

a7 a6 a5 a4	a3 a2 a1 a0

Operation: AJMP
(PC) ← (PC) + 2
($PC_{10\text{-}0}$) ← page address

ANL <dest-byte>,<src-byte>

Function: Logical-AND for byte variables

Description: ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: If the Accumulator holds 0C3H (11000011B) and register 0 holds 55H (01010101B) then the instruction,

```
ANL   A,R0
```

will leave 41H (01000001B) in the Accumulator.

When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the Accumulator at run-time. The instruction,

```
ANL   P1,#01110011B
```

will clear bits 7, 3, and 2 of output port 1.

ANL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 1 0 1	1 r r r

Operation: ANL
(A) ← (A) ∧ (Rn)

ANL A,direct

Bytes: 2

Cycles: 1

Encoding:

0 1 0 1	0 1 0 1	direct address

Operation: ANL

(A) ← (A) ∧ (direct)

ANL A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 1 0 1	0 1 1 i

Operation: ANL

(A) ← (A) ∧ ((Ri))

ANL A,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 0 1	0 1 0 0	immediate data

Operation: ANL
(A) ← (A) ∧ #data

ANL direct,A

Bytes: 2

Cycles: 1

Encoding:

0 1 0 1	0 0 1 0	direct address

Operation: ANL
(direct) ← (direct) ∧ (A)

ANL direct,#data

Bytes: 3

Cycles: 2

Encoding:

0 1 0 1	0 0 1 1	direct address	immediate data

Operation: ANL
(direct) ← (direct) ∧ #data

ANL C,<src-bit>

Function: Logical-AND for bit variables

Description: If the Boolean value of the source bit is a logical 0 then clear the carry flag; otherwise leave the carry flag in its current state. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, *but the source bit itself is not affected.* No other flags are affected.

Only direct addressing is allowed for the source operand.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC. 7 = 1, and OV = 0:

```
MOV   C,P1.0     ;LOAD CARRY WITH INPUT PIN STATE
ANL   C,ACC.7    ;AND CARRY WITH ACCUM. BIT 7
ANL   C,/OV      ;AND WITH INVERSE OF OVERFLOW FLAG
```

ANL C,bit

Bytes: 2

Cycles: 2

Encoding:

1 0 0 0	0 0 1 0	bit address

Operation: ANL
(C) ← (C) ∧ (bit)

ANL C,/bit

Bytes: 2

Cycles: 2

Encoding:

1 0 1 1	0 0 0 0	bit address

Operation: ANL
(C) ← (C) ∧ ¬(bit)

CJNE <dest-byte>,<src-byte>, rel

Function: Compare and Jump if Not Equal.

Description: CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The branch destination is computed by adding the signed relative-displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations: the Accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

Example: The Accumulator contains 34H. Register 7 contains 56H. The first instruction in the sequence,

```
                CJNE   R7,#60H, NOT_EQ
;               ...    .....          ;  R7 = 60H.
NOT_EQ:         JC     REQ_LOW        ;  IF R7 < 60H.
;               ...    .....          ;  R7 > 60H.
```

sets the carry flag and branches to the instruction at label NOT_EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60H.

If the data being presented to Port 1 is also 34H, then the instruction,

```
WAIT:  CJNE  A,P1,WAIT
```

clears the carry flag and continues with the next instruction in sequence, since the Accumulator does equal the data read from P1. (If some other value was being input on P1, the program will loop at this point until the P1 data changes to 34H.)

CJNE A,direct,rel

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	0 1 0 1	direct address	rel. address

Operation:

```
(PC) ← (PC) + 3
IF (A) <> (direct)
THEN
        (PC) ← (PC) + relative offset

IF (A) < (direct)
THEN
        (C) ← 1
ELSE
        (C) ← 0
```

CJNE A,#data,rel

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	0 1 0 0	immediate data	rel. address

Operation:

(PC) ← (PC) + 3
IF (A) <> *data*
THEN
(PC) ← (PC) + *relative offset*

IF (A) < *data*
THEN
(C) ← 1
ELSE
(C) ← 0

CJNE Rn,#data,rel

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	1 r r r	immediate data	rel. address

Operation:

(PC) ← (PC) + 3
IF (Rn) <> *data*
THEN
(PC) ← (PC) + *relative offset*

IF (Rn) < *data*
THEN
(C) ← 1
ELSE
(C) ← 0

CJNE @Ri,#data,rel

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	0 1 1 i	immediate data	rel. address

Operation:

(PC) ← (PC) + 3
IF ((Ri)) <> *data*
THEN
(PC) ← (PC) + *relative offset*

IF ((Ri)) < *data*
THEN
(C) ← 1
ELSE
(C) ← 0

CLR A

Function: Clear Accumulator

Description: The Accumulator is cleared (all bits set on zero). No flags are affected.

Example: The Accumulator contains 5CH (01011100B). The instruction,

CLR A

will leave the Accumulator set to 00H (00000000B).

Bytes: 1

Cycles: 1

Encoding:

1 1 1 0	0 1 0 0

Operation: CLR
(A) ← 0

CLR bit

Function: Clear bit

Description: The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.

Example: Port 1 has previously been written with 5DH (01011101B). The instruction,

CLR P1.2

will leave the port set to 59H (01011001B).

CLR C

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	0 0 1 1

Operation: CLR
(C) ← 0

CLR bit

Bytes: 2

Cycles: 1

Encoding:

1 1 0 0	0 0 1 0	bit address

Operation: CLR
(bit) ← 0

CPL A

Function: Complement Accumulator

Description: Each bit of the Accumulator is logically complemented (one's complement). Bits which previously contained a one are changed to a zero and vice-versa. No flags are affected.

Example: The Accumulator contains 5CH (01011100B). The instruction,

```
CPL   A
```

will leave the Accumulator set to 0A3H (10100011B).

Bytes: 1

Cycles: 1

Encoding:

1 1 1 1	0 1 0 0

Operation:

```
CPL
(A) ← ¬ (A)
```

CPL bit

Function: Complement bit

Description: The bit variable specified is complemented. A bit which had been a one is changed to zero and vice-versa. No other flags are affected. CLR can operate on the carry or any directly addressable bit.

Note: When this instruction is used to modify an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

Example: Port 1 has previously been written with 5BH (01011101B). The instruction sequence,

```
CPL   P1.1
CPL   P1.2
```

will leave the port set to 5BH (01011011B).

CPL C

Bytes: 1

Cycles: 1

Encoding:

1 0 1 1	0 0 1 1

Operation:

```
CPL
(C) ← ¬ (C)
```

CPL bit

Bytes: 2

Cycles: 1

Encoding:

1 0 1 1	0 0 1 0	bit address

Operation: CPL
(bit) ← ¬ (bit)

DA A

Function: Decimal-adjust Accumulator for Addition

Description: DA A adjusts the eight-bit value in the Accumulator resulting from the earlier addition of two variables (each in packed-BCD format), producing two four-bit digits. Any ADD or ADDC instruction may have been used to perform the addition.

If Accumulator bits 3-0 are greater than nine (xxxx1010-xxxx1111), or if the AC flag is one, six is added to the Accumulator producing the proper BCD digit in the low-order nibble. This internal addition would set the carry flag if a carry-out of the low-order four-bit field propagated through all high-order bits, but it would not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxxx-111xxxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this would set the carry flag if there was a carry-out of the high-order bits, but wouldn't clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal addition. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00H, 06H, 60H, or 66H to the Accumulator, depending on initial Accumulator and PSW conditions.

Note: DA A *cannot* simply convert a hexadecimal number in the Accumulator to BCD notation, nor does DA A apply to decimal subtraction.

Example: The Accumulator holds the value 56H (01010110B) representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67H (01100111B) representing the packed BCD digits of the decimal number 67. The carry flag is set. The instruction sequence.

```
ADDC  A,R3
DA    A
```

will first perform a standard twos-complement binary addition, resulting in the value 0BEH (10111110) in the Accumulator. The carry and auxiliary carry flags will be cleared.

The Decimal Adjust instruction will then alter the Accumulator to the value 24H (00100100B), indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56, 67, and the carry-in. The carry flag will be set by the Decimal Adjust instruction, indicating that a decimal overflow occurred. The true sum 56, 67, and 1 is 124.

BCD variables can be incremented or decremented by adding 01H or 99H. If the Accumulator initially holds 30H (representing the digits of 30 decimal), then the instruction sequence,

```
ADD   A,#99H

DA    A
```

will leave the carry set and 29H in the Accumulator, since 30 + 99 = 129. The low-order byte of the sum can be interpreted to mean 30 − 1 = 29.

Bytes: 1

Cycles: 1

Encoding:

1 1 0 1	0 1 0 0

Operation:

```
DA
-contents of Accumulator are BCD
IF    [[(A3-0) > 9] ∨ [(AC) = 1]]
      THEN(A3-0) ← (A3-0) + 6
           AND

IF    [[(A7-4) > 9] ∨ [(C) = 1]]
      THEN (A7-4) ← (A7-4) + 6
```

DEC byte

Function: Decrement

Description: The variable indicated is decremented by 1. An original value of 00H will underflow to 0FFH. No flags are affected. Four operand addressing modes are allowed: accumulator, register, direct, or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Register 0 contains 7FH (01111111B). Internal RAM locations 7EH and 7FH contain 00H and 40H, respectively. The instruction sequence,

```
DEC  @R0
DEC  R0
DEC  @R0
```

will leave register 0 set to 7EH and internal RAM locations 7EH and 7FH set to 0FFH and 3FH.

DEC A

Bytes: 1

Cycles: 1

Encoding:

0 0 0 1	0 1 0 0

Operation: DEC
(A) ← (A) − 1

DEC Rn

Bytes: 1

Cycles: 1

Encoding:

0 0 0 1	1 r r r

Operation: DEC
(Rn) ← (Rn) − 1

DEC direct

Bytes: 2

Cycles: 1

Encoding:

0 0 0 1	0 1 0 1	direct address

Operation: DEC
(direct) ← (direct) − 1

DEC @Ri

Bytes: 1

Cycles: 1

Encoding:

0 0 0 1	0 1 1 i

Operation: DEC
((Ri)) ← ((Ri)) − 1

DIV AB

Function: Divide

Description: DIV AB divides the unsigned eight-bit integer in the Accumulator by the unsigned eight-bit integer in register B. The Accumulator receives the integer part of the quotient; register B receives the integer remainder. The carry and OV flags will be cleared.

Exception: if B had originally contained 00H, the values returned in the Accumulator and B-register will be undefined and the overflow flag will be set. The carry flag is cleared in any case.

Example: The Accumulator contains 251 (0FBH or 11111011B) and B contains 18 (12H or 00010010B). The instruction,

DIV AB

will leave 13 in the Accumulator (0DH or 00001101B) and the value 17 (11H or 00010001B) in B, since 251 = (13 X 18) + 17. Carry and OV will both be cleared.

Bytes: 1

Cycles: 4

Encoding:

1 0 0 0	0 1 0 0

Operation: DIV
$(A)_{15\text{-}8}$
$(B)_{7\text{-}0}$ ← (A)/(B)

DJNZ <byte>,<rel-addr>

Function: Decrement and Jump if Not Zero

Description: DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00H will underflow to 0FFH. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Internal RAM locations 40H, 50H, and 60H contain the values 01H, 70H, and 15H, respectively. The instruction sequence,

```
DJNZ  40H,LABEL__1
DJNZ  50H,LABEL__2
DJNZ  60H,LABEL__3
```

will cause a jump to the instruction at label LABEL__2 with the values 00H, 6FH, and 15H in the three RAM locations. The first jump was *not* taken because the result was zero.

This instruction provides a simple way of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence,

```
          MOV     R2,#8
TOGGLE:   CPL     P1.7
          DJNZ    R2,TOGGLE
```

will toggle P1.7 eight times, causing four output pulses to appear at bit 7 of output Port 1. Each pulse will last three machine cycles; two for DJNZ and one to alter the pin.

DJNZ Rn,rel

Bytes: 2

Cycles: 2

Encoding:

1 1 0 1	1 r r r	rel. address

Operation:

```
DJNZ
(PC) ← (PC) + 2
(Rn) ← (Rn) − 1
IF  (Rn) > 0 or (Rn) < 0
    THEN
          (PC) ← (PC) + rel
```

DJNZ direct,rel

Bytes: 3

Cycles: 2

Encoding:

1 1 0 1	0 1 0 1	direct address	rel. address

Operation:

```
DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) − 1
IF (direct) > 0 or (direct) < 0
   THEN
         (PC) ← (PC) + rel
```

INC <byte>

Function: Increment

Description: INC increments the indicated variable by 1. An original value of 0FFH will overflow to 00H. No flags are affected. Three addressing modes are allowed: register, direct, or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Register 0 contains 7EH (011111110B). Internal RAM locations 7EH and 7FH contain 0FFH and 40H, respectively. The instruction sequence,

```
INC  @R0
INC  R0
INC  @R0
```

will leave register 0 set to 7FH and internal RAM locations 7EH and 7FH holding (respectively) 00H and 41H.

INC A

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	0 1 0 0

Operation:

```
INC
(A) ← (A) + 1
```

INC Rn

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	1 r r r

Operation: INC
(Rn) ← (Rn) + 1

INC direct

Bytes: 2

Cycles: 1

Encoding:

0 0 0 0	0 1 0 1	direct address

Operation: INC
(direct) ← (direct) + 1

INC @Ri

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	0 1 1 i

Operation: INC
((Ri)) ← ((Ri)) + 1

INC DPTR

Function: Increment Data Pointer

Description: Increment the 16-bit data pointer by 1. A 16-bit increment (modulo 2^{16}) is performed; an overflow of the low-order byte of the data pointer (DPL) from 0FFH to 00H will increment the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be incremented.

Example: Registers DPH and DPL contain 12H and 0FEH, respectively. The instruction sequence,

```
INC  DPTR
INC  DPTR
INC  DPTR
```

will change DPH and DPL to 13H and 01H.

Bytes: 1

Cycles: 2

Encoding:

1 0 1 0	0 0 1 1

Operation: INC
(DPTR) ← (DPTR) + 1

JB bit,rel

Function: Jump if Bit set

Description: If the indicated bit is a one, jump to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56 (01010110B). The instruction sequence,

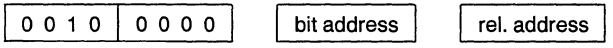
```
JB   P1.2,LABEL1

JB   ACC.2,LABEL2
```

will cause program execution to branch to the instruction at label LABEL2.

Bytes: 3

Cycles: 2

Encoding:



Operation:

```
JB
(PC) ← (PC) + 3
IF  (bit) = 1
    THEN
            (PC) ← (PC) + rel
```

JBC bit,rel

Function: Jump if Bit is set and Clear bit

Description: If the indicated bit is one, branch to the address indicated; otherwise proceed with the next instruction. *The bit will not be cleared if it is already a zero.* The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note: When this instruction is used to test an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

Example: The Accumulator holds 56H (01010110B). The instruction sequence,

```
JBC  ACC.3,LABEL1
JBC  ACC.2,LABEL2
```

will cause program execution to continue at the instruction identified by the label LABEL2, with the Accumulator modified to 52H (01010010B).

Bytes: 3

Cycles: 2

Encoding:

0 0 0 1	0 0 0 0	bit address	rel. address

Operation:

```
JBC
(PC) ← (PC) + 3
IF  (bit) = 1
    THEN
            (bit) ← 0
            (PC) ← (PC) + rel
```

JC rel

Function: Jump if Carry is set

Description: If the carry flag is set, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

Example: The carry flag is cleared. The instruction sequence,

```
JC    LABEL1
CPL   C
JC    LABEL 2
```

will set the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 0 0	0 0 0 0	rel. address

Operation:

```
JC
(PC) ← (PC) + 2
IF  (C) = 1
    THEN
            (PC) ← (PC) + rel
```

JMP @A+DPTR

Function: Jump indirect

Description: Add the eight-bit unsigned contents of the Accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo 2^{16}): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the Accumulator nor the Data Pointer is altered. No flags are affected.

Example: An even number from 0 to 6 is in the Accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP_TBL:

```
        MOV   DPTR,#JMP_TBL
        JMP   @A+DPTR
JMP_TBL: AJMP  LABEL0
        AJMP  LABEL1
        AJMP  LABEL2
        AJMP  LABEL3
```

If the Accumulator equals 04H when starting this sequence, execution will jump to label LABEL2. Remember that AJMP is a two-byte instruction, so the jump instructions start at every other address.

Bytes: 1

Cycles: 2

Encoding:

0 1 1 1	0 0 1 1

Operation: JMP
(PC) ← (A) + (DPTR)

JNB bit,rel

Function: Jump if Bit Not set

Description: If the indicated bit is a zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56H (01010110B). The instruction sequence,

```
JNB  P1.3,LABEL1
JNB  ACC.3,LABEL2
```

will cause program execution to continue at the instruction at label LABEL2.

Bytes: 3

Cycles: 2

Encoding:

0 0 1 1	0 0 0 0	bit address	rel. address

Operation:

```
JNB
(PC) ← (PC) + 3
IF  (bit) = 0
    THEN (PC) ← (PC) + rel.
```

JNC rel

Function: Jump if Carry not set

Description: If the carry flag is a zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

Example: The carry flag is set. The instruction sequence,

```
JNC  LABEL1
CPL  C
JNC  LABEL2
```

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 0 1	0 0 0 0	rel. address

Operation:

```
JNC
(PC) ← (PC) + 2
IF  (C) = 0
    THEN  (PC) ← (PC) + rel
```

JNZ rel

Function: Jump if Accumulator Not Zero

Description: If any bit of the Accumulator is a one, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally holds 00H. The instruction sequence,

```
JNZ  LABEL1
INC  A
JNZ  LABEL2
```

will set the Accumulator to 01H and continue at label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 1 1	0 0 0 0	rel. address

Operation:

```
JNZ
(PC) ← (PC) + 2
IF   (A) ≠ 0
     THEN  (PC) ← (PC) + rel
```

JZ rel

Function: Jump if Accumulator Zero

Description: If all bits of the Accumulator are zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally contains 01H. The instruction sequence,

```
JZ   LABEL1
DEC  A
JZ   LABEL2
```

will change the Accumulator to 00H and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 1 0	0 0 0 0	rel. address

Operation:

```
JZ
(PC) ← (PC) + 2
IF   (A) = 0
     THEN  (PC) ← (PC) + rel
```

LCALL addr16

Function: Long call

Description: LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the Stack Pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64K-byte program memory address space. No flags are affected.

Example: Initially the Stack Pointer equals 07H. The label "SUBRTN" is assigned to program memory location 1234H. After executing the instruction,

```
LCALL   SUBRTN
```

at location 0123H, the Stack Pointer will contain 09H, internal RAM locations 08H and 09H will contain 26H and 01H, and the PC will contain 1234H.

Bytes: 3

Cycles: 2

Encoding:

0 0 0 1	0 0 1 0	addr15-addr8	addr7-addr0

Operation:

```
LCALL
(PC) ← (PC) + 3
(SP) ← (SP) + 1
((SP)) ← (PC7-0)
(SP) ← (SP) + 1
((SP)) ← (PC15-8)
(PC) ← addr15-0
```

LJMP addr16

Function: Long Jump

Description: LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64K program memory address space. No flags are affected.

Example: The label "JMPADR" is assigned to the instruction at program memory location 1234H. The instruction,

```
LJMP   JMPADR
```

at location 0123H will load the program counter with 1234H.

Bytes: 3

Cycles: 2

Encoding:

0 0 0 0	0 0 1 0	addr15-addr8	addr7-addr0

Operation:

```
LJMP
(PC) ← addr15-0
```

MOV <dest-byte>,<src-byte>

Function: Move byte variable

Description: The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.

Example: Internal RAM location 30H holds 40H. The value of RAM location 40H is 10H. The data present at input port 1 is 11001010B (0CAH).

```
MOV   R0,#30H   ;R0 <= 30H
MOV   A,@R0     ;A <= 40H
MOV   R1,A      ;R1 <= 40H
MOV   B,@R1     ;B <= 10H
MOV   @R1,P1    ;RAM (40H) <= 0CAH
MOV   P2,P1     ;P2 #0CAH
```

leaves the value 30H in register 0, 40H in both the Accumulator and register 1, 10H in register B, and 0CAH (11001010B) both in RAM location 40H and output on port 2.

MOV A,Rn

Bytes: 1

Cycles: 1

Encoding:

1 1 1 0	1 r r r

Operation: MOV
(A) ← (Rn)

*MOV A,direct

Bytes: 2

Cycles: 1

Encoding:

1 1 1 0	0 1 0 1	direct address

Operation: MOV
(A) ← (direct)

MOV A,ACC is not a valid instruction.

MOV A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1 1 1 0	0 1 1 i

Operation: MOV
(A) ← ((Ri))

MOV A,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 1 1	0 1 0 0	immediate data

Operation: MOV
(A) ← #data

MOV Rn,A

Bytes: 1

Cycles: 1

Encoding:

1 1 1 1	1 r r r

Operation: MOV
(Rn) ← (A)

MOV Rn,direct

Bytes: 2

Cycles: 2

Encoding:

1 0 1 0	1 r r r	direct addr.

Operation: MOV
(Rn) ← (direct)

MOV Rn,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 1 1	1 r r r	immediate data

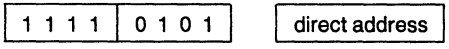
Operation: MOV
(Rn) ← #data

MOV direct,A

Bytes: 2

Cycles: 1

Encoding:



Operation: MOV
(direct) ← (A)

MOV direct,Rn

Bytes: 2

Cycles: 2

Encoding:

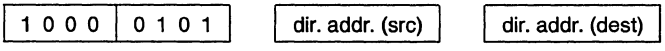
Operation: MOV
(direct) ← (Rn)

MOV direct,direct

Bytes: 3

Cycles: 2

Encoding:



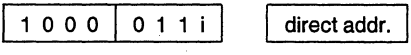
Operation: MOV
(direct) ← (direct)

MOV direct,@Ri

Bytes: 2

Cycles: 2

Encoding:



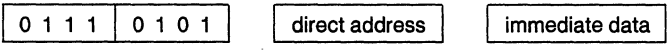
Operation: MOV
(direct) ← ((Ri))

MOV direct,#data

Bytes: 3

Cycles: 2

Encoding:



Operation: MOV
(direct) ← #data

MOV @Ri,A

Bytes: 1

Cycles: 1

Encoding:

1 1 1 1	0 1 1 i

Operation: MOV
((Ri)) ← (A)

MOV @Ri,direct

Bytes: 2

Cycles: 2

Encoding:

1 0 1 0	0 1 1 i	direct addr.

Operation: MOV
((Ri)) ← (direct)

MOV @Ri,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 1 1	0 1 1 i	immediate data

Operation: MOV
((RI)) ← #data

MOV <dest-bit>,<src-bit>

Function: Move bit data

Description: The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.

Example: The carry flag is originally set. The data present at input Port 3 is 11000101B. The data previously written to output Port 1 is 35H (00110101B).

```
MOV  P1.3,C
MOV  C,P3.3
MOV  P1.2,C
```

will leave the carry cleared and change Port 1 to 39H (00111001B).

MOV C,bit

Bytes: 2

Cycles: 1

Encoding:

1 0 1 0	0 0 1 0	bit address

Operation: MOV
(C) ← (bit)

MOV bit,C

Bytes: 2

Cycles: 2

Encoding:

1 0 0 1	0 0 1 0	bit address

Operation: MOV
(bit) ← (C)

MOV DPTR,#data16

Function: Load Data Pointer with a 16-bit constant

Description: The Data Pointer is loaded with the 16-bit constant indicated. The 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction which moves 16 bits of data at once.

Example: The instruction,

MOV DPTR,#1234H

will load the value 1234H into the Data Pointer: DPH will hold 12H and DPL will hold 34H.

Bytes: 3

Cycles: 2

Encoding:

1 0 0 1	0 0 0 0	immed. data15-8	immed. data7-0

Operation: MOV
(DPTR) ← #data$_{15\text{-}0}$
DPH □ DPL ← #data$_{15\text{-}8}$ □ #data$_{7\text{-}0}$

MOVC A,@A+<base-reg>

Function: Move Code byte

Description: The MOVC instructions load the Accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit Accumulator contents and the contents of a sixteen-bit base register, which may be either the Data Pointer or the PC. In the latter case, the PC is incremented to the address of the following instruction before being added with the Accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the Accumulator. The following instructions will translate the value in the Accumulator to one of four values defined by the DB (define byte) directive.

```
REL_PC: INC   A
        MOVC  A,@A+PC
        RET
        DB    66H
        DB    77H
        DB    88H
        DB    99H
```

If the subroutine is called with the Accumulator equal to 01H, it will return with 77H in the Accumulator. The INC A before the MOVC instruction is needed to "get around" the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the Accumulator instead.

MOVC A,@A+DPTR

Bytes: 1

Cycles: 2

Encoding:

1 0 0 1	0 0 1 1

Operation: MOVC
(A) ← ((A) + (DPTR))

MOVC A,@A + PC

Bytes: 1

Cycles: 2

Encoding:

1 0 0 0	0 0 1 1

Operation: MOVC
(PC) ← (PC) + 1
(A) ← ((A) + (PC))

MOVX <dest-byte>,<src-byte>

Function: Move External

Description: The MOVX instructions transfer data between the Accumulator and a byte of external data memory, hence the "X" appended to MOV. There are two types of instructions, differing in whether they provide an eight-bit or sixteen-bit indirect address to the external data RAM.

In the first type, the contents of R0 or R1 in the current register bank provide an eight-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, any output port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVX.

In the second type of MOVX instruction, the Data Pointer generates a sixteen-bit address. P2 outputs the high-order eight address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 Special Function Register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64K bytes), since no additional instructions are needed to set up the output ports.

It is possible in some situations to mix the two MOVX types. A large RAM array with its high-order address lines driven by P2 can be addressed via the Data Pointer, or with code to output high-order address bits to P2 followed by a MOVX instruction using R0 or R1.

Example: An external 256 byte RAM using multiplexed address/data lines (e.g., an Intel 8155 RAM/I/O/Timer) is connected to the 8051 Port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12H and 34H. Location 34H of the external RAM holds the value 56H. The instruction sequence,

```
MOVX  A,@R1

MOVX  @R0,A
```

copies the value 56H into both the Accumulator and external RAM location 12H.

MOVX A,@Ri

Bytes: 1

Cycles: 2

Encoding:

1 1 1 0	0 0 1 i

Operation: MOVX
(A) ← ((Ri))

MOVX A,@DPTR

Bytes: 1

Cycles: 2

Encoding:

1 1 1 0	0 0 0 0

Operation: MOVX
(A) ← ((DPTR))

MOVX @Ri,A

Bytes: 1

Cycles: 2

Encoding:

1 1 1 1	0 0 1 i

Operation: MOVX
((Ri)) ← (A)

MOVX @DPTR,A

Bytes: 1

Cycles: 2

Encoding:

1 1 1 1	0 0 0 0

Operation: MOVX
(DPTR) ← (A)

MUL AB

Function: Multiply

Description: MUL AB multiplies the unsigned eight-bit integers in the Accumulator and register B. The low-order byte of the sixteen-bit product is left in the Accumulator, and the high-order byte in B. If the product is greater than 255 (0FFH) the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.

Example: Originally the Accumulator holds the value 80 (50H). Register B holds the value 160 (0A0H). The instruction,

```
MUL   AB
```

will give the product 12,800 (3200H), so B is changed to 32H (00110010B) and the Accumulator is cleared. The overflow flag is set, carry is cleared.

Bytes: 1

Cycles: 4

Encoding:

1 0 1 0	0 1 0 0

Operation:
MUL
$(A)_{7\text{-}0} \leftarrow (A) \times (B)_{15\text{-}8}$

NOP

Function: No Operation

Description: Execution continues at the following instruction. Other than the PC, no registers or flags are affected.

Example: It is desired to produce a low-going output pulse on bit 7 of Port 2 lasting exactly 5 cycles. A simple SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming no interrupts are enabled) with the instruction sequence,

```
CLR     P2.7
NOP
NOP
NOP
NOP
SETB    P2.7
```

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	0 0 0 0

Operation:
NOP
$(PC) \leftarrow (PC) + 1$

ORL <dest-byte> <src-byte>

Function: Logical-OR for byte variables

Description: ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the destination byte. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: If the Accumulator holds 0C3H (11000011B) and R0 holds 55H (01010101B) then the instruction,

```
ORL   A,R0
```

will leave the Accumulator holding the value 0D7H (11010111B).

When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be either a constant data value in the instruction or a variable computed in the Accumulator at run-time. The instruction,

```
ORL   P1,#00110010B
```

will set bits 5, 4, and 1 of output Port 1.

ORL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 1 0 0	1 r r r

Operation: ORL
(A) ← (A) ∨ (Rn)

ORL A,direct

Bytes: 2

Cycles: 1

Encoding:

0 1 0 0	0 1 0 1	direct address

Operation: ORL
(A) ← (A) ∨ (direct)

ORL A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 1 0 0	0 1 1 i

Operation: ORL
(A) ← (A) ∨ ((Ri))

ORL A,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 0 0	0 1 0 0	immediate data

Operation: ORL
(A) ← (A) ∨ #data

ORL direct,A

Bytes: 2

Cycles: 1

Encoding:

0 1 0 0	0 0 1 0	direct address

Operation: ORL
(direct) ← (direct) ∨ (A)

ORL direct,#data

Bytes: 3

Cycles: 2

Encoding:

0 1 0 0	0 0 1 1	direct addr.	immediate data

Operation: ORL
(direct) ← (direct) ∨ #data

ORL C,<src-bit>

Function: Logical-OR for bit variables

Description: Set the carry flag if the Boolean value is a logical 1; leave the carry in its current state otherwise . A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Example: Set the carry flag if and only if P1.0 = 1, ACC. 7 = 1, or OV = 0:

```
MOV  C,P1.0     ;LOAD CARRY WITH INPUT PIN P10
ORL  C,ACC.7    ;OR CARRY WITH THE ACC. BIT 7
ORL  C,/OV      ;OR CARRY WITH THE INVERSE OF OV.
```

ORL C,bit

Bytes: 2

Cycles: 2

Encoding:

0 1 1 1	0 0 1 0	bit address

Operation: ORL
(C) ← (C) ∨ (bit)

ORL C,/bit

Bytes: 2

Cycles: 2

Encoding:

1 0 1 0	0 0 0 0	bit address

Operation: ORL
(C) ← (C) ∨ ($\overline{\text{bit}}$)

POP direct

Function: Pop from stack.

Description: The contents of the internal RAM location addressed by the Stack Pointer is read, and the Stack Pointer is decremented by one. The value read is then transferred to the directly addressed byte indicated. No flags are affected.

Example: The Stack Pointer originally contains the value 32H, and internal RAM locations 30H through 32H contain the values 20H, 23H, and 01H, respectively. The instruction sequence,

```
POP   DPH
POP   DPL
```

will leave the Stack Pointer equal to the value 30H and the Data Pointer set to 0123H. At this point the instruction,

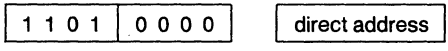
```
POP   SP
```

will leave the Stack Pointer set to 20H. Note that in this special case the Stack Pointer was decremented to 2FH before being loaded with the value popped (20H).

Bytes: 2

Cycles: 2

Encoding:



Operation:
POP
(direct) ← ((SP))
(SP) ← (SP) − 1

PUSH direct

Function: Push onto stack

Description: The Stack Pointer is incremented by one. The contents of the indicated variable is then copied into the internal RAM location addressed by the Stack Pointer. Otherwise no flags are affected.

Example: On entering an interrupt routine the Stack Pointer contains 09H. The Data Pointer holds the value 0123H. The instruction sequence,

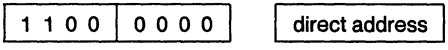
```
PUSH   DPL
PUSH   DPH
```

will leave the Stack Pointer set to 0BH and store 23H and 01H in internal RAM locations 0AH and 0BH, respectively.

Bytes: 2

Cycles: 2

Encoding:



Operation:
PUSH
(SP) ← (SP) + 1
((SP)) ← (direct)

RET

Function: Return from subroutine

Description: RET pops the high- and low-order bytes of the PC successively from the stack, decrementing the Stack Pointer by two. Program execution continues at the resulting address, generally the instruction immediately following an ACALL or LCALL. No flags are affected.

Example: The Stack Pointer originally contains the value 0BH. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RET

will leave the Stack Pointer equal to the value 09H. Program execution will continue at location 0123H.

Bytes: 1

Cycles: 2

Encoding:

0 0 1 0	0 0 1 0

Operation: RET
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RETI

Function: Return from interrupt

Description: RETI pops the high- and low-order bytes of the PC successively from the stack, and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The Stack Pointer is left decremented by two. No other registers are affected; the PSW is *not* automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is generally the instruction immediately after the point at which the interrupt request was detected. If a lower- or same-level interrupt had been pending when the RETI instruction is executed, that one instruction will be executed before the pending interrupt is processed.

Example: The Stack Pointer originally contains the value 0BH. An interrupt was detected during the instruction ending at location 0122H. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RETI

will leave the Stack Pointer equal to 09H and return program execution to location 0123H.

Bytes: 1

Cycles: 2

Encoding:

0 0 1 1	0 0 1 0

Operation: RETI
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RL A

Function: Rotate Accumulator Left

Description: The eight bits in the Accumulator are rotated one bit to the left. Bit 7 is rotated into the bit 0 position. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The instruction,

```
RL  A
```

leaves the Accumulator holding the value 8BH (10001011B) with the carry unaffected.

Bytes: 1

Cycles: 1

Encoding:

0 0 1 0	0 0 1 1

Operation:
RL
$(A_n + 1) \leftarrow (An)$ n = 0 – 6
(A0) ← (A7)

RLC A

Function: Rotate Accumulator Left through the Carry flag

Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag; the original state of the carry flag moves into the bit 0 position. No other flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction,

```
RLC  A
```

leaves the Accumulator holding the value 8BH (10001010B) with the carry set.

Bytes: 1

Cycles: 1

Encoding:

0 0 1 1	0 0 1 1

Operation:
RLC
(An + 1) ← (An) n = 0 – 6
(A0) ← (C)
(C) ← (A7)

RR A

Function: Rotate Accumulator Right

Description: The eight bits in the Accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The instruction,

```
RR  A
```

leaves the Accumulator holding the value 0E2H (11100010B) with the carry unaffected.

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	0 0 1 1

Operation:
RR
(An) ← (A_n + 1) n = 0 − 6
(A7) ← (A0)

RRC A

Function: Rotate Accumulator Right through Carry flag

Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag; the original value of the carry flag moves into the bit 7 position. No other flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B), the carry is zero. The instruction,

```
RRC  A
```

leaves the Accumulator holding the value 62 (01100010B) with the carry set.

Bytes: 1

Cycles: 1

Encoding:

0 0 0 1	0 0 1 1

Operation:
RRC
(An) ← (An + 1) n = 0 − 6
(A7) ← (C)
(C) ← (A0)

SETB <bit>

Function: Set Bit

Description: SETB sets the indicated bit to one. SETB can operate on the carry flag or any directly addressable bit. No other flags are affected.

Example: The carry flag is cleared. Output Port 1 has been written with the value 34H (00110100B). The instructions,

SETB C

SETB P1.0

will leave the carry flag set to 1 and change the data output on Port 1 to 35H (00110101B).

SETB C

Bytes: 1

Cycles: 1

Encoding:

1 1 0 1	0 0 1 1

Operation: SETB
(C) ← 1

SETB bit

Bytes: 2

Cycles: 1

Encoding:

1 1 0 1	0 0 1 0	bit address

Operation: SETB
(bit) ← 1

SJMP rel

Function: Short Jump

Description: Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.

Example: The label "RELADR" is assigned to an instruction at program memory location 0123H. The instruction,

```
SJMP   RELADR
```

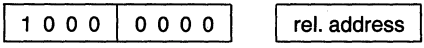
will assemble into location 0100H. After the instruction is executed, the PC will contain the value 0123H.

(*Note:* Under the above conditions the instruction following SJMP will be at 102H. Therefore, the displacement byte of the instruction will be the relative offset (0123H-0102H) = 21H. Put another way, an SJMP with a displacement of 0FEH would be a one-instruction infinite loop.)

Bytes: 2

Cycles: 2

Encoding:



Operation:

```
SJMP
(PC) ← (PC) + 2
(PC) ← (PC) + rel
```

SUBB A,<src-byte>

Function: Subtract with borrow

Description: SUBB subtracts the indicated variable and the carry flag together from the Accumulator, leaving the result in the Accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit 7, and clears C otherwise. (If C was set *before* executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple precision subtraction, so the carry is subtracted from the Accumulator along with the source operand.) AC is set if a borrow is needed for bit 3, and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

When subtracting signed integers OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.

The source operand allows four addressing modes: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C9H (11001001B), register 2 holds 54H (01010100B), and the carry flag is set. The instruction,

SUBB A,R2

will leave the value 74H (01110100B) in the accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9H minus 54H is 75H. The difference between this and the above result is due to the carry (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision subtraction, it should be explicitly cleared by a CLR C instruction.

SUBB A,Rn

Bytes: 1

Cycles: 1

Encoding:

1 0 0 1	1 r r r

Operation: SUBB

(A) ← (A) − (C) − (Rn)

SUBB A,direct

Bytes: 2

Cycles: 1

Encoding:

1 0 0 1	0 1 0 1	direct address

Operation: SUBB
(A) ← (A) − (C) − (direct)

SUBB A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1 0 0 1	0 1 1 i

Operation: SUBB
(A) ← (A) − (C) − ((Ri))

SUBB A,#data

Bytes: 2

Cycles: 1

Encoding:

1 0 0 1	0 1 0 0	immediate data

Operation: SUBB
(A) ← (A) − (C) − #data

SWAP A

Function: Swap nibbles within the Accumulator

Description: SWAP A interchanges the low- and high-order nibbles (four-bit fields) of the Accumulator (bits 3-0 and bits 7-4). The operation can also be thought of as a four-bit rotate instruction. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The instruction,

SWAP A

leaves the Accumulator holding the value 5CH (01011100B).

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	0 1 0 0

Operation: SWAP
$(A_{3-0}) \rightleftarrows (A_{7-4})$

XCH A,<byte>

Function: Exchange Accumulator with byte variable

Description: XCH loads the Accumulator with the contents of the indicated variable, at the same time writing the original Accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.

Example: R0 contains the address 20H. The Accumulator holds the value 3FH (00111111B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCH A,@R0

will leave RAM location 20H holding the values 3FH (00111111B) and 75H (01110101B) in the accumulator.

XCH A,Rn

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	1 r r r

Operation: XCH
(A) $\rightleftarrows$ (Rn)

XCH A,direct

Bytes: 2

Cycles: 1

Encoding:

1 1 0 0	0 1 0 1	direct address

Operation: XCH
(A) $\rightleftarrows$ (direct)

XCH A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	0 1 1 i

Operation: XCH
(A) $\rightleftarrows$ ((Ri))

XCHD A,@Ri

Function: Exchange Digit

Description: XCHD exchanges the low-order nibble of the Accumulator (bits 3-0), generally representing a hexadecimal or BCD digit, with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (bits 7-4) of each register are not affected. No flags are affected.

Example: R0 contains the address 20H. The Accumulator holds the value 36H (00110110B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCHD A,@R0

will leave RAM location 20H holding the value 76H (01110110B) and 35H (00110101B) in the Accumulator.

Bytes: 1

Cycles: 1

Encoding:

1 1 0 1	0 1 1 i

Operation: XCHD
$(A_{3\text{-}0}) \rightleftarrows ((Ri_{3\text{-}0}))$

XRL <dest-byte>,<src-byte>

Function: Logical Exclusive-OR for byte variables

Description: XRL performs the bitwise logical Exclusive-OR operation between the indicated variables, storing the results in the destination. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

(*Note:* When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.)

Example: If the Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction,

XRL A,R0

will leave the Accumulator holding the value 69H (01101001B).

When the destination is a directly addressed byte, this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte, either a constant contained in the instruction or a variable computed in the Accumulator at run-time. The instruction,

XRL P1,#00110001B

will complement bits 5, 4, and 0 of output Port 1.

XRL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 1 1 0	1 r r r

Operation: XRL
(A) ← (A) ∀ (Rn)

XRL A,direct

Bytes: 2

Cycles: 1

Encoding:

0 1 1 0	0 1 0 1	direct address

Operation: XRL
(A) ← (A) ∀ (direct)

XRL A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 1 1 0	0 1 1 i

Operation: XRL
(A) ← (A) ∀ ((Ri))

XRL A,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 1 0	0 1 0 0	immediate data

Operation: XRL
(A) ← (A) ∀ #data

XRL direct,A

Bytes: 2

Cycles: 1

Encoding:

0 1 1 0	0 0 1 0	direct address

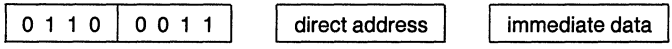
Operation: XRL
(direct) ← (direct) ∀ (A)

XRL direct, #data

Bytes: 3

Cycles: 2

Encoding:



Operation: XRL
(direct) ← (direct) ∀ #data

Using the Intel MCS®-51 Boolean Processing Capabilities

7

intel

USING THE INTEL MCS®-51 BOOLEAN PROCESSING CAPABILITIES

1.0 INTRODUCTION

The Intel microcontroller family now has three new members: the Intel® 8031, 8051, and 8751 single-chip microcomputers. These devices, shown in Figure 1, will allow whole new classes of products to benefit from recent advances in Integrated Electronics. Thanks to Intel's new HMOS technology, they provide larger program and data memory spaces, more flexible I/O and peripheral capabilities, greater speed, and lower system cost than any previous-generation single-chip microcomputer.

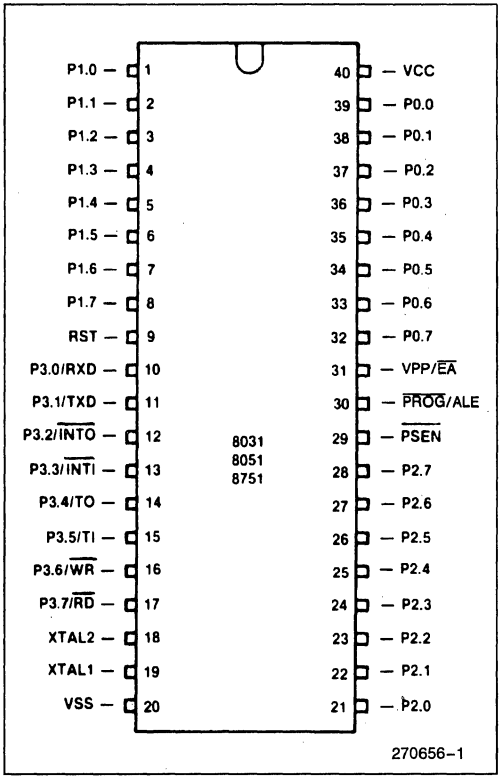


Figure 1. 8051 Family Pinout Diagram

Table 1 summarizes the quantitative differences between the members of the MCS®-48 and 8051 families. The 8751 contains 4K bytes of EPROM program memory fabricated on-chip, while the 8051 replaces the EPROM with 4K bytes of lower-cost mask-programmed ROM. The 8031 has no program memory on-chip; instead, it accesses up to 64K bytes of program memory from external memory. Otherwise, the three new family members are identical. Throughout this Note, the term "8051" will represent all members of the 8051 Family, unless specifically stated otherwise.

The CPU in each microcomputer is one of the industry's fastest and most efficient for numerical calculations on byte operands. But controllers often deal with bits, not bytes: in the real world, switch contacts can only be open or closed, indicators should be either lit or dark, motors are either turned on or off, and so forth. For such control situations the most significant aspect of the MCS®-51 architecture is its complete hardware support for one-bit, or *Boolean* variables (named in honor of Mathematician George Boole) as a separate data type.

The 8051 incorporates a number of special features which support the direct manipulation and testing of individual bits and allow the use of single-bit variables in performing logical operations. Taken together, these features are referred to as the MCS-51 *Boolean Processor.* While the bit-processing capabilities alone would be adequate to solve many control applications, their true power comes when they are used in conjunction with the microcomputer's byte-processing and numerical capabilities.

Many concepts embodied by the Boolean Processor will certainly be new even to experienced microcomputer system designers. The purpose of this Application Note is to explain these concepts and show how they are used.

For detailed information on these parts refer to the **Intel Microcontroller Handbook,** order number 210918. The instruction set, assembly language, and use of the 8051 assembler (ASM51) are further described in the **MCS®-51 Macro Assembler User's Guide for DOS Systems,** order number 122753.

Table 1. Features of Intel's Single-Chip Microcomputers

EPROM Program Memory	ROM Program Memory	External Program Memory	Program Memory (Int/Max)	Data Memory (Bytes)	Instr. Cycle Time	Input/ Output Pins	Interrupt Sources	Reg. Banks
8748	8048	8035	1K 4K	64	2.5 μs	27	2	2
—	8049	8039	2K 4K	128	1.36 μs	27	2	2
8751	8051	8031	4K 64K	128	1.0 μs	32	5	4

2.0 BOOLEAN PROCESSOR OPERATION

The Boolean Processing capabilities of the 8051 are based on concepts which have been around for some time. Digital computer systems of widely varying designs all have four functional elements in common (Figure 2):

- a central processor (CPU) with the control, timing, and logic circuits needed to execute stored instructions:
- a memory to store the sequence of instructions making up a program or algorithm:
- data memory to store variables used by the program:
  and
- some means of communicating with the outside world.

The CPU usually includes one or more accumulators or special registers for computing or storing values during program execution. The instruction set of such a processor generally includes, at a minimum, operation classes to perform arithmetic or logical functions on program variables, move variables from one place to another, cause program execution to jump or conditionally branch based on register or variable states, and instructions to call and return from subroutines. The program and data memory functions sometimes share a single memory space, but this is not always the case. When the address spaces are separated, program and data memory need not even have the same basic word width.

A digital computer's flexibility comes in part from combining simple fast operations to produce more complex (albeit slower) ones, which in turn link together eventually solving the problem at hand. A four-bit CPU executing multiple precision subroutines can, for example, perform 64-bit addition and subtraction. The subroutines could in turn be building blocks for floating-point multiplication and division routines. Eventually, the four-bit CPU can simulate a far more complex "virtual" machine.

In fact, *any* digital computer with the above four functional elements can (given time) complete *any* algorithm (though the proverbial room full of chimpanzees at word processors might first re-create Shakespeare's classics and this Application Note)! This fact offers little consolation to product designers who want programs to run as quickly as possible. By definition, a real-time control algorithm *must* proceed quickly enough to meet the preordained speed constraints of other equipment.

One of the factors determining how long it will take a microcomputer to complete a given chore is the number of instructions it must execute. What makes a given computer architecture particularly well- or poorly-suited for a class of problems is how well its instruction set matches the tasks to be performed. The better the "primitive" operations correspond to the steps taken by the control algorithm, the lower the number of instructions needed, and the quicker the program will run. All else being equal, a CPU supporting 64-bit arithmetic directly could clearly perform floating-point math faster than a machine bogged-down by multiple-precision subroutines. In the same way, direct support for bit manipulation naturally leads to more efficient programs handling the binary input and output conditions inherent in digital control problems.

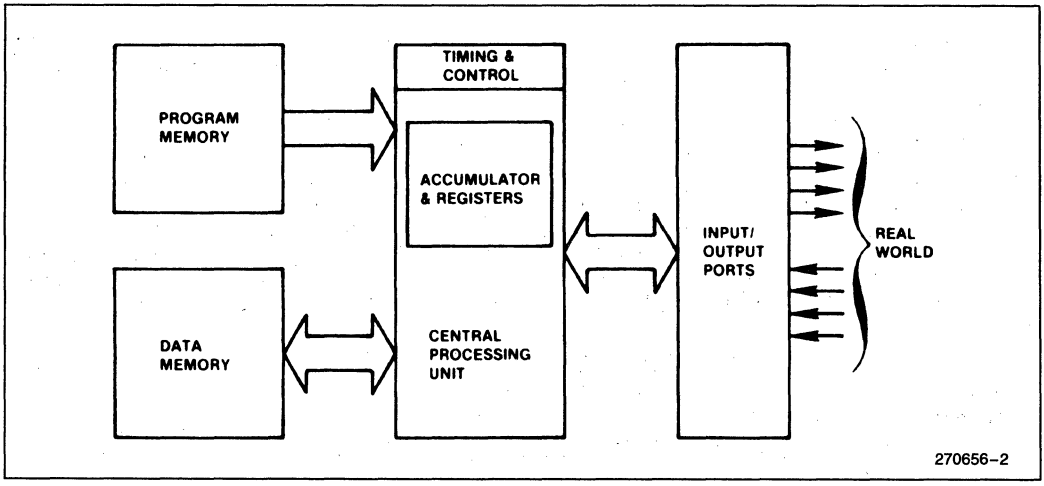


Figure 2. Block Diagram for Abstract Digital Computer

Processing Elements

The introduction stated that the 8051's bit-handling capabilities alone would be sufficient to solve some control applications. Let's see how the four basic elements of a digital computer—a CPU with associated registers, program memory, addressable data RAM, and I/O capability—relate to Boolean variables.

CPU. The 8051 CPU incorporates special logic devoted to executing several bit-wide operations. All told, there are 17 such instructions, all listed in Table 2. Not shown are 94 other (mostly byte-oriented) 8051 instructions.

Program Memory. Bit-processing instructions are fetched from the same program memory as other arithmetic and logical operations. In addition to the instructions of Table 2, several sophisticated program control features like multiple addressing modes, subroutine nesting, and a two-level interrupt structure are useful in structuring Boolean Processor-based programs.

Table 2. MCS-51™ Boolean Processing Instruction Subset

Mnemonic		Description	Byte	Cyc
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
MOV	C.bit	Move direct bit to Carry flag	2	1
MOV	bit.C	Move Carry flag to direct bit	2	2
ANL	C.bit	AND direct bit to Carry flag	2	2
ANL	C.bit	AND complement of direct bit to Carry flag	2	2
ORL	C.bit	OR direct bit to Carry flag	2	2
ORL	C.bit	OR complement of direct bit to Carry flag	2	2
JC	rel	Jump if Carry is flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2
JB	bit.rel	Jump if direct Bit set	3	2
JNB	bit.rel	Jump if direct Bit Not set	3	2
JBC	bit.rel	Jump if direct Bit is set & Clear bit	3	2

Address mode abbreviations

C—Carry flag.
bit—128 software flags, any I/O pin, control or status bit.
rel—All conditional jumps include an 8-bit offset byte. Range is +127 −128 bytes relative to first byte of the following instruction.

Boolean instructions are one, two, or three bytes long, depending on what function they perform. Those involving only the carry flag have either a single-byte opcode or an opcode followed by a conditional-branch destination byte (Figure 3a). The more general instructions add a "direct address" byte after the opcode to specify the bit affected, yielding two or three byte encodings (Figure 3b). Though this format allows potentially 256 directly addressable bit locations, not all of them are implemented in the 8051 family.

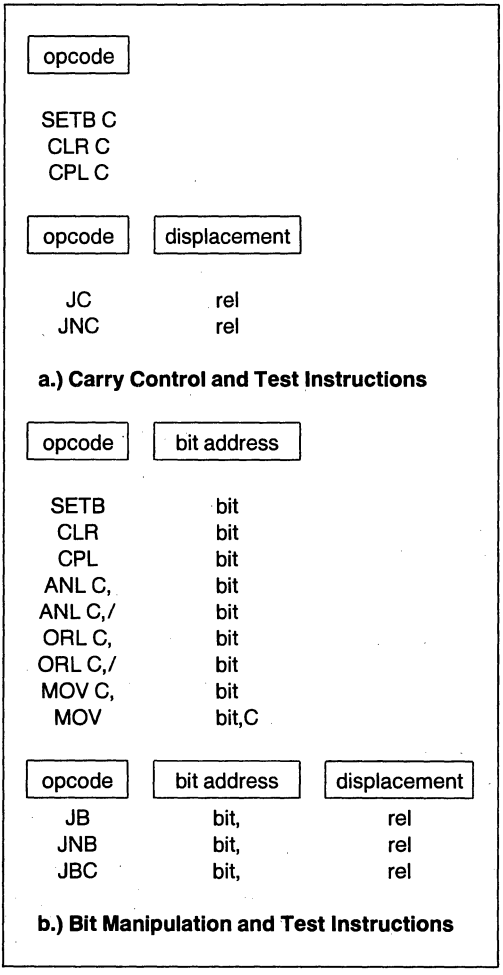


Figure 3. Bit Addressing Instruction Formats

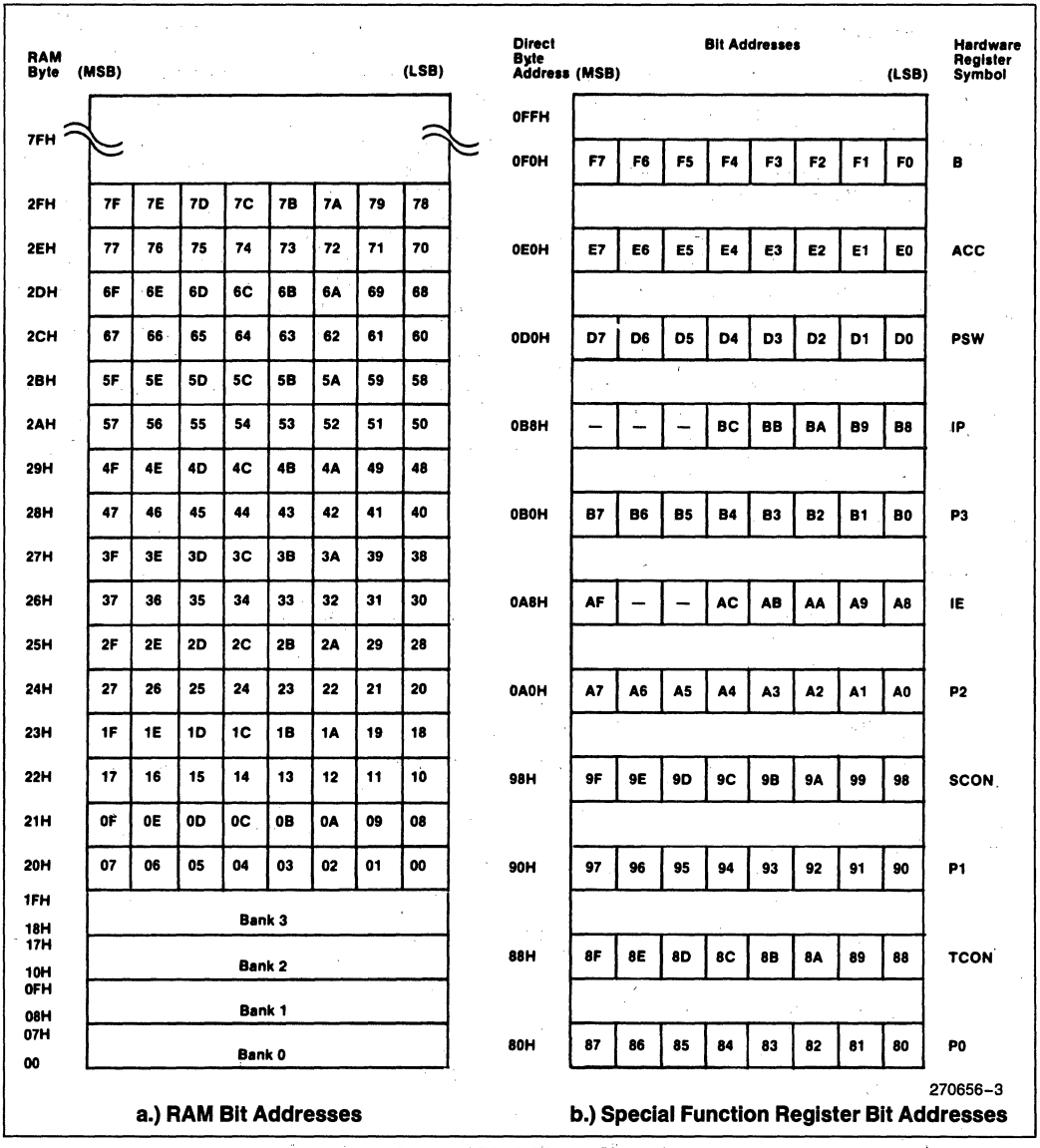


Figure 4. Bit Address Maps

Data Memory. The instructions in Figure 3b can operate directly upon 144 general purpose bits forming the Boolean processor "RAM." These bits can be used as software flags or to store program variables. Two operand instructions use the CPU's carry flag ("C") as a special one-bit register: in a sense, the carry is a "Boolean accumulator" for logical operations and data transfers.

Input/Output. All 32 I/O pins can be addressed as individual inputs, outputs, or both, in any combination. Any pin can be a control strobe output, status (Test) input, or serial I/O link implemented via software. An additional 33 individually addressable bits reconfigure, control, and monitor the status of the CPU and all on-chip peripheral functions (timer counters, serial port modes, interrupt logic, and so forth).

(MSB)							(LSB)
CY	AC	F0	RS1	RS0	OV	—	P

Symbol	Position	Name and Significance
CY	PSW.7	Carry flag. Set/cleared by hardware or software during certain arithmetic and logical instructions.
AC	PSW.6	Auxiliary Carry flag. Set/cleared by hardware during addition or subtraction instructions to indicate carry or borrow out of bit 3.
F0	PSW.5	Flag 0. Set/cleared/tested by software as a user-defined status flag.
RS1	PSW.4	Register bank Select control bits.
RS0	PSW.3	1 & 0. Set/cleared by software to determine working register bank (see Note).
OV	PSW.2	Overflow flag. Set/cleared by hardware during arithmetic instructions to indicate overflow conditions.
—	PSW.1	(reserved)
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e., even parity.

Note- the contents of (RS1, RS0) enable the working register banks as follows:

(0,0) - Bank 0 (00H–07H)
(0,1) - Bank 1 (08H–0FH)
(1,0) - Bank 2 (10H–17H)
(1,1) - Bank 3 (18H–1FH)

Figure 5. PSW—Program Status Word Organization

(MSB)							(LSB)
RD	WR	T1	T0	INT1	INT0	TXD	RXD

Symbol	Position	Name and Significance
RD	P3.7	Read data control output. Active low pulse generated by hardware when external data memory is read.
WR	P3.6	Write data control output. Active low pulse generated by hardware when external data memory is written.
T1	P3.5	Timer/counter 1 external input or test pin.
T0	P3.4	Timer/counter 0 external input or test pin.
INT1	P3.3	Interrupt 1 input pin. Low-level or falling-edge triggered.
INT0	P3.2	Interrupt 0 input pin. Low-level or falling-edge triggered.
TXD	P3.1	Transmit Data pin for serial port in UART mode. Clock output in shift register mode.
RXD	P3.0	Receive Data pin for serial port in UART mode. Data I/O pin in shift register mode.

Figure 6. P3—Alternate I/O Functions of Port 3

Direct Bit Addressing

The most significant bit of the direct address byte selects one of two groups of bits. Values between 0 and 127 (00H and 7FH) define bits in a block of 32 bytes of on-chip RAM, between RAM addresses 20H and 2FH (Figure 4a). They are numbered consecutively from the lowest-order byte's lowest-order bit through the highest-order byte's highest-order bit.

Bit addresses between 128 and 255 (80H and 0FFH) correspond to bits in a number of special registers, mostly used for I/O or peripheral control. These positions are numbered with a different scheme than RAM: the five high-order address bits match those of the register's own address, while the three low-order bits identify the bit position within that register (Figure 4b).

Notice the column labeled "Symbol" in Figure 5. Bits with special meanings in the PSW and other registers have corresponding symbolic names. General-purpose (as opposed to carry-specific) instructions may access the carry like any other bit by using the mnemonic CY in place of C, P0, P1, P2, and P3 are the 8051's four I/O ports: secondary functions assigned to each of the eight pins of P3 are shown in Figure 6.

Figure 7 shows the last four bit addressable registers. TCON (Timer Control) and SCON (Serial port Control) control and monitor the corresponding peripherals, while IE (Interrupt Enable) and IP (Interrupt Priority) enable and prioritize the five hardware interrupt sources. Like the reserved hardware register addresses, the five bits not implemented in IE and IP should not be accessed: they can *not* be used as software flags.

Addressable Register Set. There are 20 special function registers in the 8051, but the advantages of bit addressing only relate to the 11 described below. Five potentially bit-addressable register addresses (0C0H, 0C8H, 0D8H, 0E8H, & 0F8H) are being reserved for possible future expansion in microcomputers based on the MCS-51 architecture. Reading or writing non-existent registers in the 8051 series is pointless, and may cause unpredictable results. Byte-wide logical operations can be used to manipulate bits in all *non*-bit addressable registers and RAM.

(MSB)							(LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol Position Name and Significance

TF1 TCON.7 Timer 1 overflow Flag. Set by hardware on timer/counter overflow. Cleared when interrupt processed.

TR1 TCON.6 Timer 1 Run control bit. Set/cleared by software to turn timer/counter on/off.

TF0 TCON.5 Timer 0 overflow Flag. Set by hardware on timer/counter overflow. Cleared when interrupt processed.

TR0 TCON.4 Timer 0 Run control bit. Set/cleared by software to turn timer/counter on/off.

IE1 TCON.3 Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.

IT1 TCON.2 Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

IE0 TCON.1 Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.

IT0 TCON.0 Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

a.) TCON—Timer/Counter Control/Status Register

(MSB)							(LSB)
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Symbol Position Name and Significance

SM0 SCON.7 Serial port Mode control bit 0. Set/cleared by software (see note).

SM1 SCON.6 Serial port Mode control bit 1. Set/cleared by software (see note).

SM2 SCON.5 Serial port Mode control bit 2. Set by software to disable reception of frames for which bit 8 is zero.

REN SCON.4 Receiver Enable control bit. Set/cleared by software to enable/disable serial data reception.

TB8 SCON.3 Transmit Bit 8. Set/cleared by hardware to determine state of ninth data bit transmitted in 9-bit UART mode.

RB8 SCON.2 Receive Bit 8. Set/cleared by hardware to indicate state of ninth data bit received.

TI SCON.1 Transmit Interrupt flag. Set by hardware when byte transmitted. Cleared by software after servicing.

RI SCON.0 Receive Interrupt flag. Set by hardware when byte received. Cleared by software after servicing.

Note- the state of (SM0, SM1) selects:
(0,0)—Shift register I/O expansion.
(0,1)—8-bit UART, variable data rate.
(1,0)—9-bit UART, fixed data rate.
(1,1)—9-bit UART, variable data rate.

b.) SCON—Serial Port Control/Status Register

Figure 7. Peripheral Configuration Registers

(MSB) (LSB)

EA	—	—	ES	ET1	EX1	ET1	EX0

Symbol	Position	Name and Significance
EA	IE.7	Enable All control bit. Cleared by software to disable all interrupts, independent of the state of IE.4–IE.0.
—	IE.6	(reserved)
—	IE.5	
ES	IE.4	Enable Serial port control bit. Set/cleared by software to enable/disable interrupts from TI or RI flags.
ET1	IE.3	Enable Timer 1 control bit. Set/cleared by software to enable/disable interrupts from timer/counter 1.
EX1	IE.2	Enable External interrupt 1 control bit. Set/cleared by software to enable/disable interrupts from INT1.
ET0	IE.1	Enable Timer 0 control bit. Set/cleared by software to enable/disable interrupts from timer/counter 0.
EX0	IE.0	Enable External interrupt 0 control bit. Set/cleared by software to enable/disable interrupts from INT0.

c.) IE—Interrupt Enable Register

(MSB) (LSB)

—	—	—	PS	PT1	PX1	PT0	PX0

Symbol	Position	Name and Significance
—	IP.7	(reserved)
—	IP.6	(reserved)
—	IP.5	(reserved)
PS	IP.4	Serial port Priority control bit. Set/cleared by software to specify high/low priority interrupts for Serial port.
PT1	IP.3	Timer 1 Priority control bit. Set/cleared by software to specify high/low priority interrupts for timer/counter 1.
PX1	IP.2	External interrupt 1 Priority control bit. Set/cleared by software to specify high/low priority interrupts for INT1.
PT0	IP.1	Timer 0 Priority control bit. Set/cleared by software to specify high/low priority interrupts for timer/counter 0.
PX0	IP.0	External interrupt 0 Priority control bit. Set/cleared by software to specify high/low priority interrupts for INT0.

d.) IP—Interrupt Priority Control Register

Figure 7. Peripheral Configuration Registers (Continued)

The accumulator and B registers (A and B) are normally involved in byte-wide arithmetic, but their individual bits can also be used as 16 general software flags. Added with the 128 flags in RAM, this gives 144 general purpose variables for bit-intensive programs. The program status word (PSW) in Figure 5 is a collection of flags and machine status bits including the carry flag itself. Byte operations acting on the PSW can therefore affect the carry.

Instruction Set

Having looked at the bit variables available to the Boolean Processor, we will now look at the four classes of instructions that manipulate these bits. It may be helpful to refer back to Table 2 while reading this section.

State Control. Addressable bits or flags may be set, cleared, or logically complemented in one instruction cycle with the two-byte instructions SETB, CLR, and CPL. (The "B" affixed to SETB distinguishes it from the assembler "SET" directive used for symbol definition.) SETB and CLR are analogous to loading a bit with a constant: 1 or 0. Single byte versions perform the same three operations on the carry.

The MCS-51 assembly language specifies a bit address in any of three ways:

- by a number or expression corresponding to the direct bit address (0–255):

- by the name or address of the register containing the bit, the *dot operator* symbol (a period: "."), and the bit's position in the register (7–0):
- in the case of control and status registers, by the predefined assembler symbols listed in the first columns of Figures 5–7.

Bits may also be given user-defined names with the assembler "BIT" directive and any of the above techniques. For example, bit 5 of the PSW may be cleared by any of the four instructions.

```
USR_FLG BIT  PSW.5    ; User Symbol Definition
        ...  ...
        CLR  0D5H     ; Absolute Addressing
        CLR  PSW.5    ; Use of Dot Operator
        CLR  F0       ; Pre-Defined Assembler
                      ; Symbol
        CLR  USR_FLG  ; User-Defined Symbol
```

Data Transfers. The two-byte MOV instructions can transport any addressable bit to the carry in one cycle, or copy the carry to the bit in two cycles. A bit can be moved between two arbitrary locations via the carry by combining the two instructions. (If necessary, push and pop the PSW to preserve the previous contents of the carry.) These instructions can replace the multi-instruction sequence of Figure 8, a program structure appearing in controller applications whenever flags or outputs are conditionally switched on or off.

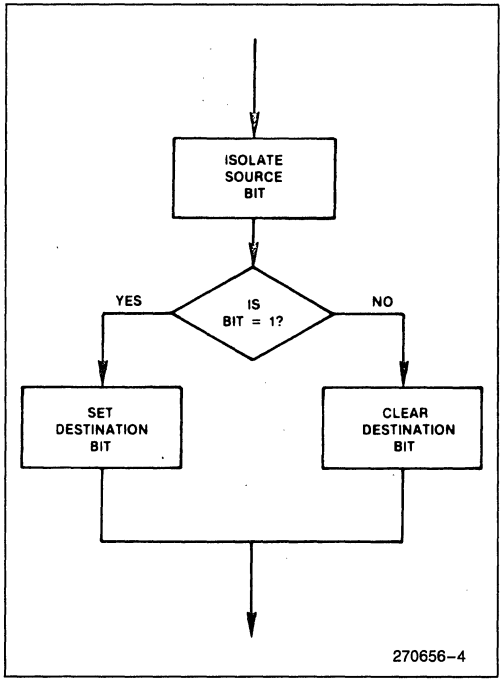


Figure 8. Bit Transfer Instruction Operation

Logical Operations. Four instructions perform the logical-AND and logical-OR operations between the carry and another bit, and leave the results in the carry. The instruction mnemonics are ANL and ORL; the absence or presence of a slash mark ("/") before the source operand indicates whether to use the positive-logic value or the logical complement of the addressed bit. (The source operand itself is never affected.)

Bit-test Instructions. The conditional jump instructions "JC rel" (Jump on Carry) and "JNC rel" (Jump on Not Carry) test the state of the carry flag, branching if it is a one or zero, respectively. (The letters "rel" denote relative code addressing.) The three-byte instructions "JB bit.rel" and "JNB bit.rel" (Jump on Bit and Jump on Not Bit) test the state of *any* addressable bit in a similar manner. A fifth instruction combines the Jump on Bit and Clear operations. "JBC bit.rel" conditionally branches to the indicated address, then clears the bit in the same two cycle instruction. This operation is the same as the MCS-48 "JTF" instructions.

All 8051 conditional jump instructions use program counter-relative addressing, and all execute in two cycles. The last instruction byte encodes a signed displacement ranging from −128 to +127. During execution, the CPU adds this value to the incremented program counter to produce the jump destination. Put another way, a conditional jump to the immediately following instruction would encode 00H in the offset byte.

A section of program or subroutine written using only relative jumps to nearby addresses will have the same machine code independent of the code's location. An assembled routine may be repositioned anywhere in memory, even crossing memory page boundaries, without having to modify the program or recompute destination addresses. To facilitate this flexibility, there is an unconditional "Short Jump" (SJMP) which uses relative addressing as well. Since a programmer would have quite a chore trying to compute relative offset values from one instruction to another, ASM51 automatically computes the displacement needed given only the destination address or label. An error message will alert the programmer if the destination is "out of range."

The so-called "Bit Test" instructions implemented on many other microprocessors simply perform the logical-AND operation between a byte variable and a constant mask, and set or clear a zero flag depending on the result. This is essentially equivalent to the 8051 "MOV C.bit" instruction. A second instruction is then needed to conditionally branch based on the state of the zero flag. This does *not* constitute abstract bit-addressing in the MCS-51 sense. A flag exists only as a field

within a register: to reference a bit the programmer must know and specify both the encompassing register and the bit's position therein. This constraint severely limits the flexibility of symbolic bit addressing and reduces the machine's code-efficiency and speed.

Interaction with Other Instructions. The carry flag is also affected by the instructions listed in Table 3. It can be rotated through the accumulator, and altered as a side effect of arithmetic instructions. Refer to the User's Manual for details on how these instructions operate.

Simple Instruction Combinations

By combining general purpose bit operations with certain addressable bits, one can "custom build" several hundred useful instructions. All eight bits of the PSW can be tested directly with conditional jump instructions to monitor (among other things) parity and overflow status. Programmers can take advantage of 128 software flags to keep track of operating modes, resource usage, and so forth.

The Boolean instructions are also the most efficient way to control or reconfigure peripheral and I/O registers. All 32 I/O lines become "test pins," for example, tested by conditional jump instructions. Any output pin can be toggled (complemented) in a single instruction cycle. Setting or clearing the Timer Run flags (TR0 and TR1) turn the timer/counters on or off; polling the same flags elsewhere lets the program determine if a timer is running. The respective overflow flags (TF0 and TF1) can be tested to determine when the desired period or count has elapsed, then cleared in preparation for the next repetition. (For the record, these bits are all part of the TCON register, Figure 7a. Thanks to symbolic bit addressing, the programmer only needs to remember the mnemonic associated with each function. In other words, don't bother memorizing control word layouts.)

In the MCS-48 family, instructions corresponding to some of the above functions require specific opcodes. Ten different opcodes serve to clear complement the software flags F0 and F1, enable/disable each interrupt, and start/stop the timer. In the 8051 instruction set, just three opcodes (SETB, CLR, CPL) with a direct bit address appended perform the same functions. Two test instructions (JB and JNB) can be combined with bit addresses to test the software flags, the 8048 I/O pins T0, T1, and INT, and the eight accumulator bits, replacing 15 more different instructions.

Table 4a shows how 8051 programs implement software flag and machine control functions associated with special opcodes in the 8048. In every case the MCS-51 solution requires the same number of machine cycles, and executes 2.5 times faster.

Table 3. Other Instructions Affecting the Carry Flag

Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry flag	1	1
ADDC	A,direct	Add direct byte to Accumulator with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry flag	1	1
ADDC	A,#data	Add immediate data to Acc with Carry flag	2	1
SUBB	A,Rn	Subtract register from Accumulator with borrow	1	1
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from Acc with borrow	1	1
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1
RLC	A	Rotate Accumulator Left through the Carry flag	1	1
RRC	A	Rotate Accumulator Right through Carry flag	1	1
CJNE	A,direct.rel	Compare direct byte to Acc & Jump if Not Equal	3	2
CJNE	A,#data.rel	Compare immediate to Acc & Jump if Not Equal	3	2
CJNE	Rn,#data.rel	Compare immed to register & Jump if Not Equal	3	2
CJNE	@Ri,#data.rel	Compare immed to indirect & Jump if Not Equal	3	2

Table 4a. Contrasting 8048 and 8051 Bit Control and Testing Instructions

8048 Instruction		Bytes	Cycles	μSec	8x51 Instruction		Bytes	Cycles & μSec
Flag Control								
CLR	C	1	1	2.5	CLR	C	1	1
CPL	F0	1	1	2.5	CPL	F0	2	1
Flag Testing								
JNC	offset	2	2	5.0	JNC	rel	2	2
JF0	offset	2	2	5.0	JB	F0.rel	3	2
JB7	offset	2	2	5.0	JB	ACC.7.rel	3	2
Peripheral Polling								
JT0	offset	2	2	5.0	JB	T0.rel	3	2
JN1	offset	2	2	5.0	JNB	INT0.rel	3	2
JTF	offset	2	2	5.0	JBC	TF0.rel	3	2
Machine and Peripheral Control								
STRT	T	1	1	2.5	SETB	TR0	2	1
EN	1	1	1	2.5	SETB	EX0	2	1
DIS	TCNT1	1	1	2.5	CLR	ET0	2	1

Table 4b. Replacing 8048 Instruction Sequences with Single 8x51 Instructions

8048 Instruction		Bytes	Cycles	μSec	8051 Instruction		Bytes	Cycles & μSec
Flag Control								
Set carry								
CLR	C							
CPL	C	= 2	2	5.0	SETB	C	1	1
Set Software Flag								
CLR	F0							
CPL	F0	= 2	2	5.0	SETB	F0	2	1
Turn Off Output Pin								
ANL	P1.#0FBH	= 2	2	5.0	CLR	P1.2	2	1
Complement Output Pin								
IN	A.P1							
XRL	A.#04H							
OUTL	P1.A	= 4	6	15.0	CPL	P1.2	2	1
Clear Flag in RAM								
MOV	R0.#FLGADR							
MOV	A.@R0							
ANL	A.#FLGMASK							
MOV	@R0.A	= 6	6	15.0	CLR	USER_FLG	2	1

Table 4b. Replacing 8048 Instruction Sequences with Single 8x51 Instructions (Continued)

8048 Instruction			Bytes	Cycles	μSec	8x51 Instruction		Bytes	Cycles & μSec
Flag Testing:									
Jump if Software Flag is 0									
JF0	$+4								
JMP	offset	=	4	4	10.0	JNB	F0.rel	3	2
Jump if Accumulator bit is 0									
CPL	A								
JB7	offset								
CPL	A	=	4	4	10.0	JNB	ACC.7.rel	3	2
Peripheral Polling									
Test if Input Pin is Grounded									
IN	A.P1								
CPL	A								
JB3	offset	=	4	5	12.5	JNB	P1.3.rel	3	2
Test if Interrupt Pin is High									
JN1	$+4								
JMP	offset	=	4	4	10.0	JB	INT0.rel	3	2

3.0 BOOLEAN PROCESSOR APPLICATIONS

So what? Then what does all this buy you?

Qualitatively, nothing. All the same capabilities *could* be (and often have been) implemented on other machines using awkward sequences of other basic operations. As mentioned earlier, any CPU can solve any problem given enough time.

Quantitatively, the differences between a solution allowed by the 8051 and those required by previous architectures are numerous. What the 8051 Family buys you is a faster, cleaner, lower-cost solution to microcontroller applications.

The opcode space freed by condensing many specific 8048 instructions into a few general operations has been used to add new functionality to the MCS-51 architecture—both for byte and bit operations. 144 software flags replace the 8048's two. These flags (and the carry) may be directly set, not just cleared and complemented, and all can be tested for either state, not just one. Operating mode bits previously inaccessible may be read, tested, or saved. Situations where the 8051 instruction set provides new capabilities are contrasted with 8048 instruction sequences in Table 4b. Here the 8051 speed advantage ranges from 5x to 15x!

Combining Boolean and byte-wide instructions can produce great synergy. An MCS-51 based application will prove to be:

- simpler to write since the architecture correlates more closely with the problems being solved:
- easier to debug because more individual instructions have no unexpected or undesirable side-effects:
- more byte efficient due to direct bit addressing and program counter relative branching:
- faster running because fewer bytes of instruction need to be fetched and fewer conditional jumps are processed:
- lower cost because of the high level of system-integration within one component.

These rather unabashed claims of excellence shall not go unsubstantiated. The rest of this chapter examines less trivial tasks simplified by the Boolean processor. The first three compare the 8051 with other microprocessors; the last two go into 8051-based system designs in much greater depth.

Design Example # 1—Bit Permutation

First off, we'll use the bit-transfer instructions to permute a lengthy pattern of bits.

A steadily increasing number of data communication products use encoding methods to protect the security of sensitive information. By law, interstate financial transactions involving the Federal banking system must be transmitted using the Federal Information Processing *Data Encryption Standard* (DES).

Basically, the DES combines eight bytes of "plaintext" data (in binary, ASCII, or any other format) with a 56-bit "key", producing a 64-bit encrypted value for transmission. At the receiving end the same algorithm is applied to the incoming data using the same key, reproducing the original eight byte message. The algorithm used for these permutations is fixed; different user-defined keys ensure data privacy.

It is not the purpose of this note to describe the DES in any detail. Suffice it to say that encryption/decryption is a long, iterative process consisting of rotations, exclusive -OR operations, function table look-ups, and an extensive (and quite bizarre) sequence of bit permutation, packing, and unpacking steps. (For further details refer to the June 21, 1979 issue of **Electronics** magazine.) The bit manipulation steps are included, it is rumored, to impede a general purpose digital supercomputer trying to "break" the code. Any algorithm implementing the DES with previous generation microprocessors would spend virtually all of its time diddling bits.

The bit manipulation performed is typified by the Key Schedule Calculation represented in Figure 9. This step is repeated 16 times for each key used in the course of a transmission. In essence, a seven-byte, 56-bit "Shifted Key Buffer" is transformed into an eight-byte, "Permutation Buffer" without altering the shifted Key. The arrows in Figure 9 indicate a few of the translation steps. Only six bits of each byte of the Permutation Buffer are used; the two high-order bits of each byte are cleared. This means only 48 of the 56 Shifted Key Buffer bits are used in any one iteration.

Different microprocessor architectures would best implement this type of permutation in different ways. Most approaches would share the steps of Figure 10a:

- Initialize the Permutation Buffer to default state (ones or zeroes):
- Isolate the state of a bit of a byte from the Key Buffer. Depending on the CPU, this might be accomplished by rotating a word of the Key Buffer through a carry flag or testing a bit in memory or an accumulator against a mask byte:
- Perform a conditional jump based on the carry or zero flag if the Permutation Buffer default state is correct:
- Otherwise reverse the corresponding bit in the permutation buffer with logical operations and mask bytes.

Each step above may require several instructions. The last three steps must be repeated for all 48 bits. Most microprocessors would spend 300 to 3,000 microseconds on each of the 16 iterations.

Notice, though, that this flow chart looks a lot like Figure 8. The Boolean Processor can permute bits by simply moving them from the source to the carry to the destination—a total of two instructions taking four bytes and three microseconds per bit. Assume the Shifted Key Buffer and Permutation Buffer both reside in bit-addressable RAM, with the bits of the former assigned symbolic names SKB__1, SKB__2, . . . SKB__56, and that the bytes of the latter are named PB__1, . . . PB__8. Then working from Figure 9, the software for the permutation algorithm would be that of Example 1a. The total routine length would be 192 bytes, requiring 144 microseconds.

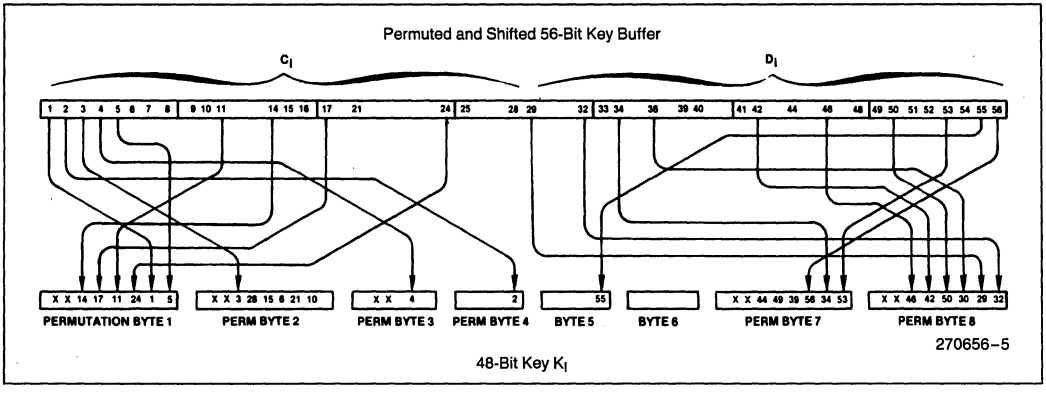


Figure 9. DES Key Schedule Transformation

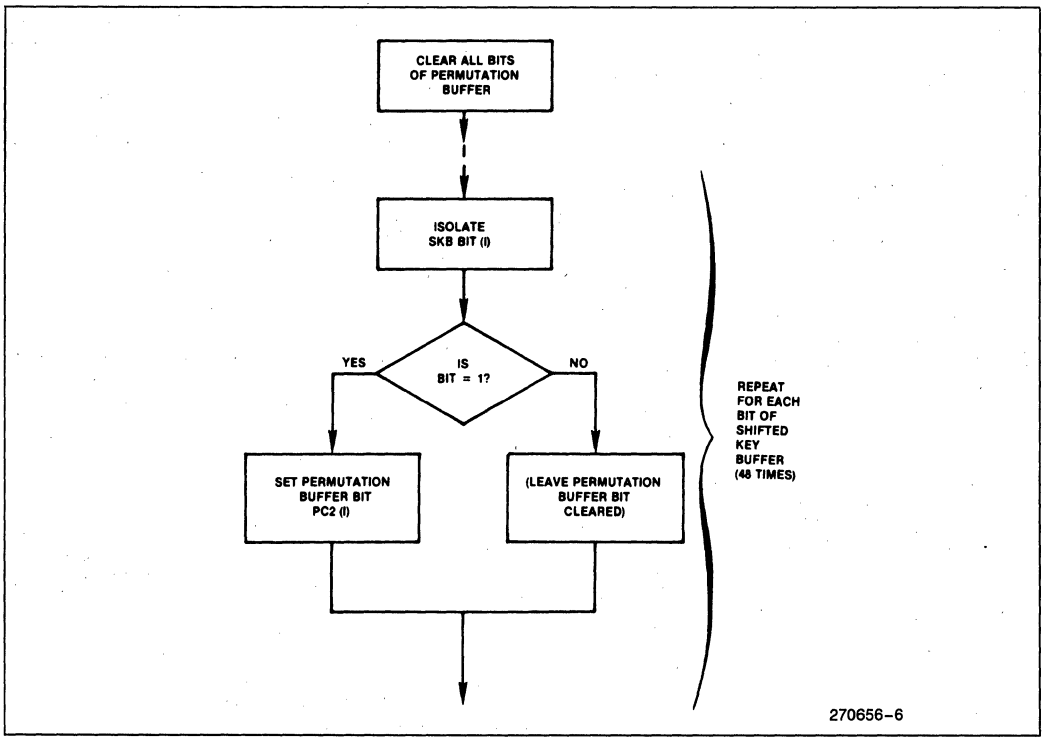


Figure 10a. Flowchart for Key Permutation Attempted with a Byte Processor

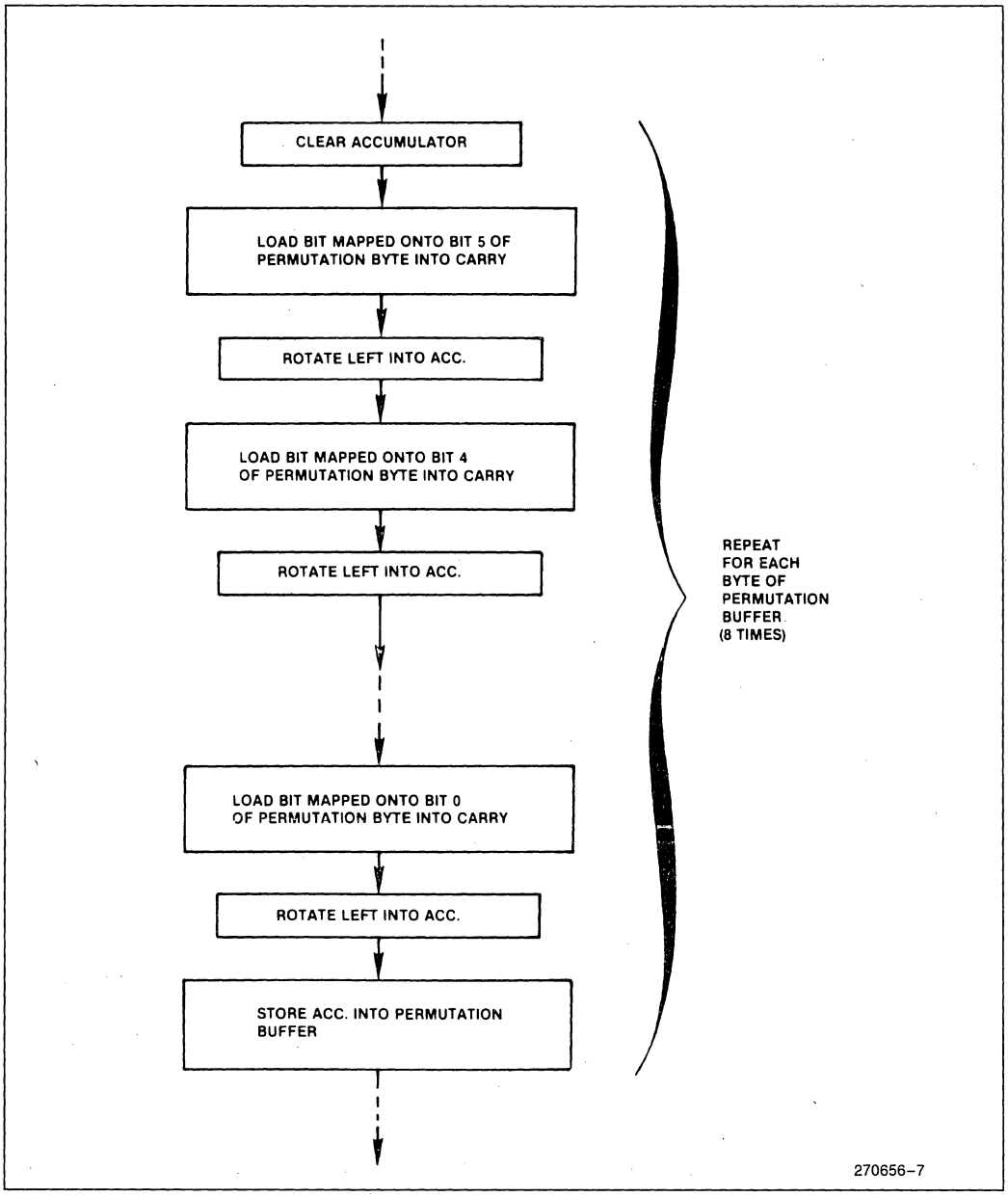


Figure 10b. DES Key Permutation with Boolean Processor

The algorithm of Figure 10b is just slightly more efficient in this time-critical application and illustrates the synergy of an integrated byte and bit processor. The bits needed for each byte of the Permutation Buffer are assimilated by loading each bit into the carry (1 μs.) and shifting it into the accumulator (1 μs.). Each byte is stored in RAM when completed. Forty-eight bits thus need a total of 112 instructions, some of which are listed in Example 1b.

Worst-case execution time would be 112 microseconds, since each instruction takes a single cycle. Routine length would also decrease, to 168 bytes. (Actually, in the context of the complete encryption algorithm, each permuted byte would be processed as soon as it is assimilated—saving memory and cutting execution time by another 8 μs.)

To date, most banking terminals and other systems using the DES have needed special boards or peripheral controller chips just for the encryption/decryption process, and still more hardware to form a serial bit stream for transmission (Figure 11a). An 8051 solution could pack most of the entire system onto the one chip (Figure 11b). The whole DES algorithm would require less than one-fourth of the on-chip program memory, with the remaining bytes free for operating the banking terminal (or whatever) itself.

Moreover, since transmission and reception of data is performed through the on-board UART, the unencrypted data (plaintext) never even exists outside the microcomputer! Naturally, this would afford a high degree of security from data interception.

Example 1. DES Key Permutation Software.

a.) "Brute Force" technique

```
MOV     C,SKB_1
MOV     PB_1.1,C
MOV     C,SKB_2
MOV     PB_4.0,C
MOV     C,SKB_3
MOV     PB_2.5,C
MOV     C,SKB_4
MOV     PB_1.0,C
...     .....
...     .....
MOV     C,SKB_55
MOV     PB_5.0,C
MOV     C,SKB_56
MOV     PB_7.2,C
```

b.) Using Accumulator to Collect Bits

```
CLR     A
MOV     C,SKB_14
RLC     A
MOV     C,SKB_17
RLC     A
MOV     C,SKB_11
RLC     A
MOV     C,SKB_24
RLC     A
MOV     C,SKB_1
RLC     A
MOV     C,SKB_5
RLC     A
MOV     PB_1,A
...     .....
...     .....
MOV     C,SKB_29
RLC     A
MOV     C,SKB_32
RLC     A
MOV     PB_8,A
```

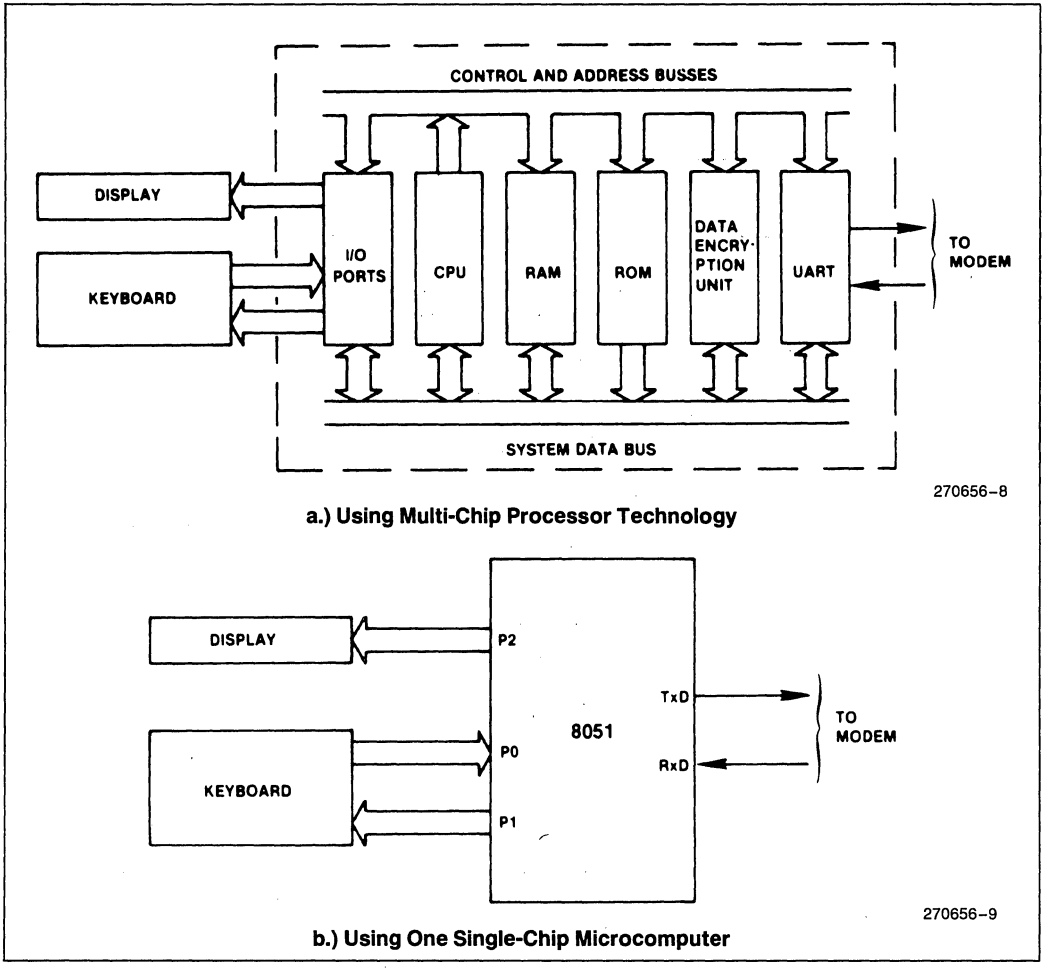


Figure 11. Secure Banking Terminal Block Diagram

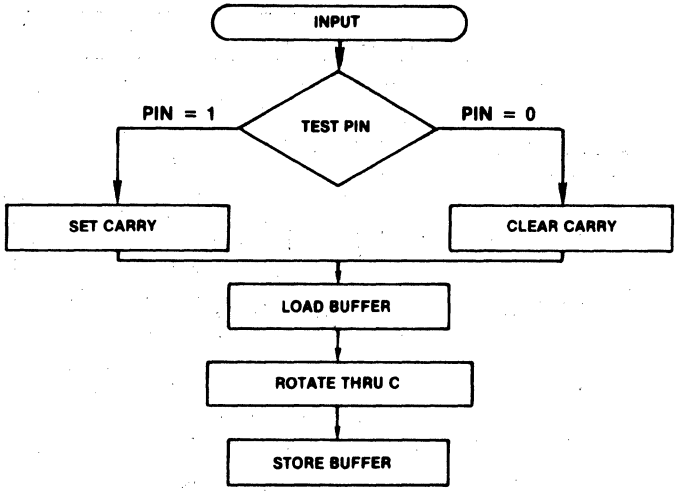
Design Example #2—Software Serial I/O

An exercise often imposed on beginning microcomputer students is to write a program simulating a UART. Though doing this with the 8051 Family may appear to be a moot point (given that the hardware for a full UART is on-chip), it is still instructive to see how it would be done, and maintains a product line tradition.

As it turns out, the 8051 microcomputers can receive or transmit serial data via software very efficiently using the Boolean instruction set. Since any I/O pin may be a serial input or output, several serial links could be maintained at once.

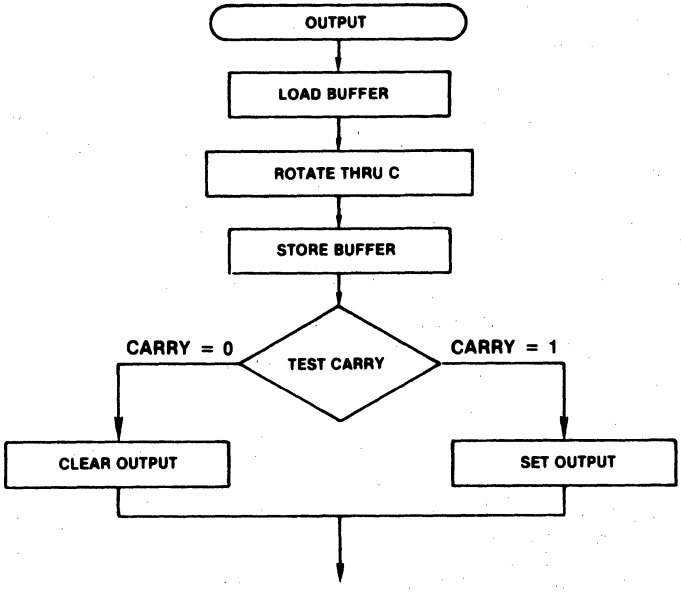
Figures 12a and 12b show algorithms for receiving or transmitting a byte of data. (Another section of program would invoke this algorithm eight times, synchronizing it with a start bit, clock signal, software delay, or timer interrupt.) Data is received by testing an input pin, setting the carry to the same state, shifting the carry into a data buffer, and saving the partial frame in internal RAM. Data is transmitted by shifting an output buffer through the carry, and generating each bit on an output pin.

A side-by-side comparison of the software for this common "bit-banging" application with three different microprocessor architectures is shown in Table 5a and 5b. The 8051 solution is more efficient than the others on every count!



270656-10

a.) Reception



270656-11

b.) Transmission

Figure 12. Serial I/O Algorithms

Table 5. Serial I/O Programs for Various Microprocessors

a.) Input Routine.

```
        8085                    8048                    8051
        IN    SERPORT                                   MOV   C,SERPIN
        ANI   MASK              CLR   C
        JZ    LO                JNT0  LO
        CMC                     CPL   C
LO:     LXI   HL,SERBUF         MOV   R0,#SERBUF
        MOV   A,M               MOV   A,@R0             MOV   A,SERBUF
        RR                      RRC   A                 RRC   A
        MOV   M,A               MOV   @R0,A             MOV   SERBUF,A

RESULTS:
        8 INSTRUCTIONS          7 INSTRUCTIONS          4 INSTRUCTIONS
        14 BYTES                9 BYTES                 7 BYTES
        56 STATES               9 CYCLES                4 CYCLES
        19 uSEC.                22.5 uSEC.              4 uSEC.
```

b.) Output Routine.

```
        8085                    8048                    8051
        LXI   HL,SERBUF         MOV   R0,#SERBUF
        MOV   A,M               MOV   A,@R0             MOV   A,SERBUF
        RR                      RRC   A                 RRC   A
        MOV   M,A               MOV   @R0,A             MOV   SERBUF,A
        IN    SERPORT
        JC    HI                JC    HI
LO:     ANI   NOT MASK          ANL   SERPRT,#NOT MASK  MOV   SERPIN,C
        JMP   CNT               JMP   CNT
HI:     ORI   MASK        HI:   ORL   SERPRT,#MASK
CNT:    OUT   SERPORT     CNT:

RESULTS:
        10 INSTRUCTIONS         8 INSTRUCTIONS          4 INSTRUCTIONS
        20 BYTES                13 BYTES                7 BYTES
        72 STATES               11 CYCLES               5 CYCLES
        24 uSEC.                27.5 uSEC.              5 uSEC.
```

270656-30

Design Example #3—Combinatorial Logic Equations

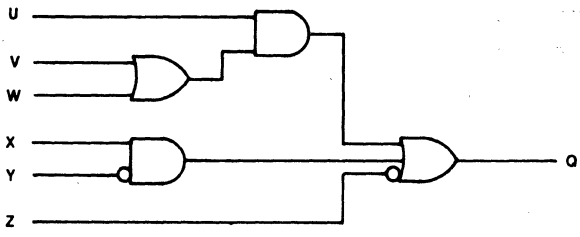
Next we'll look at some simple uses for bit-test instructions and logical operations. (This example is also presented in Application Note AP-69.)

Virtually all hardware designers have solved complex functions using combinatorial logic. While the hardware involved may vary from relay logic, vacuum tubes, or TTL or to more esoteric technologies like fluidics, in each case the goal is the same: to solve a problem represented by a logical function of several Boolean variables.

Figure 13 shows TTL and relay logic diagrams for a function of the six variables U through Z. Each is a solution of the equation.

$$Q = (U \bullet (V + W)) + (X \bullet \overline{Y}) + \overline{Z}$$

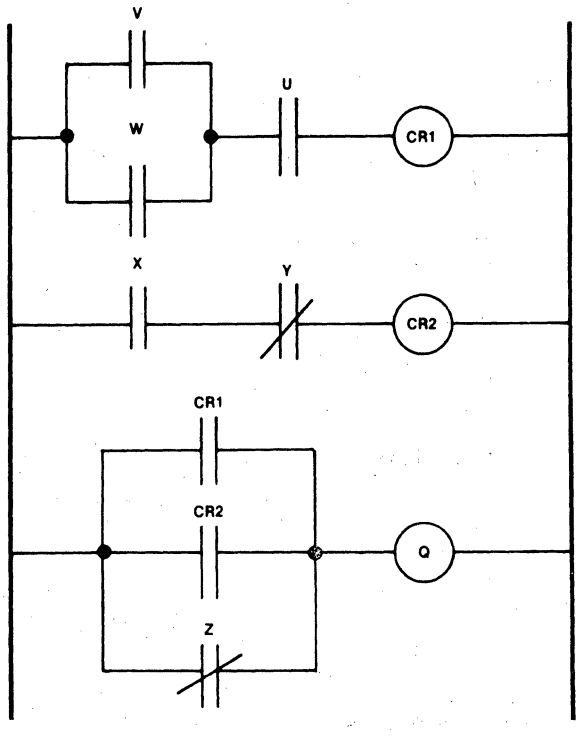
Equations of this sort might be reduced using Karnaugh Maps or algebraic techniques, but that is not the purpose of this example. As the logic complexity increases, so does the difficulty of the reduction process. Even a minor change to the function equations as the design evolves would require tedious re-reduction from scratch.



270656-12

$Q = (U \bullet (V + W)) + (X \bullet \overline{Y}) + \overline{Z}$

a.) Using TTL



270656-13

b.) Using Relay Logic

Figure 13. Hardware Implementations of Boolean Functions

For the sake of comparison we will implement this function three ways, restricting the software to three proper subsets of the MCS-51 instruction set. We will also assume that U and V are input pins from different input ports, W and X are status bits for two peripheral controllers, and Y and Z are software flags set up earlier in the program. The end result must be written to an output pin on some third port. The first two implementations follow the flow-chart shown in Figure 14. Program flow would embark on a route down a test-and-branch tree and leaves either the "True" or "Not True" exit ASAP—as soon as the proper result has been determined. These exits then rewrite the output port with the result bit respectively one or zero.

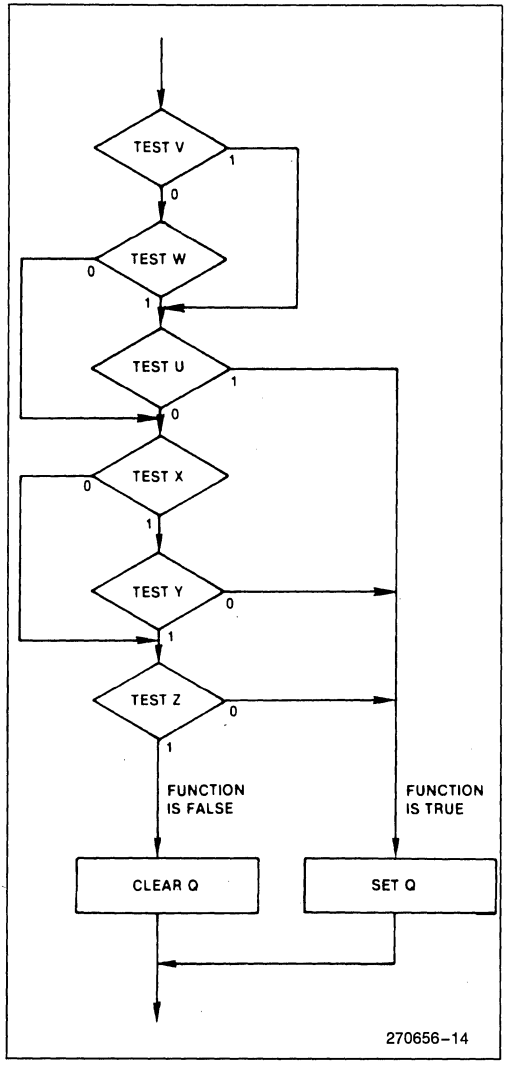


Figure 14. Flow Chart for Tree-Branching Algorithm

Other digital computers must solve equations of this type with standard word-wide logical instructions and conditional jumps. So for the first implementation, we won't use any generalized bit-addressing instructions. As we shall soon see, being constrained to such an instruction subset produces somewhat sloppy software solutions. MCS-51 mnemonics are used in Example 2a: other machines might further cloud the situation by requiring operation-specific mnemonics like INPUT, OUTPUT, LOAD, STORE, etc., instead of the MOV mnemonic used for all variable transfers in the 8051 instruction set.

The code which results is cumbersome and error prone. It would be difficult to prove whether the software worked for all input combinations in programs of this sort. Furthermore, execution time will vary widely with input data.

Thanks to the direct bit-test operations, a single instruction can replace each move mask conditional jump sequence in Example 2a, but the algorithm would be equally convoluted (see Example 2b). To lessen the confusion "a bit" each input variable is assigned a symbolic name.

A more elegant and efficient implementation (Example 2c) strings together the Boolean ANL and ORL functions to generate the output function with straight-line code. When finished, the carry flag contains the result, which is simply copied out to the destination pin. No flow chart is needed—code can be written directly from the logic diagrams in Figure 14. The result is simplicity itself: fast, flexible, reliable, easy to design, and easy to debug.

An 8051 program can simulate an N-input AND or OR gate with at most N+1 lines of source program—one for each input and one line to store the results. To simulate NAND and NOR gates, complement the carry after computing the function. When some inputs to the gate have "inversion bubbles", perform the ANL or ORL operation on inverted operands. When the first input is inverted, either load the operand into the carry and then complement it, or use DeMorgan's Theorem to convert the gate to a different form.

Example 2. Software Solutions to Logic Function of Figure 13.

a.) Using only byte-wide logical instructions

```
:BFUNCI  SOLVE RANDOM LOGIC
;        FUNCTION OF 6 VARIABLES
;        BY LOADING AND MASKING
;        THE APPROPRIATE BITS IN
;        THE ACCUMULATOR. THEN
;        EXECUTING CONDITIONAL
;        JUMPS BASED ON ZERO
;        CONDITION. (APPROACH USED
;        BY BYTE-ORIENTED
;        ARCHITECTURES.) BYTE AND
;        MASK VALUES CORRESPOND TO
;        RESPECTIVE BYTE ADDRESS
;        AND BIT POSITIONS.
;
OUTBUF   DATA 22H
;OUTPUT PIN STATE MAP
;
```

```
TESTV:  MOV   A,P2
        ANL   A,#00000100B
        JNZ   TESTU
        MOV   A,TCON
        ANL   A,#00100000B
        JZ    TESTX
TESTU:  MOV   A,P1
        ANL   A,#00000010B
        JNZ   SETQ
TESTX:  MOV   A,TCON
        ANL   A,#00001000B
        JZ    TESTZ
        MOV   A,20H
        ANL   A,#00000001B
        JZ    SETQ
TESTZ:  MOV   A,21H
        ANL   A,#00000010B
        JZ    SETQ
CLRQ:   MOV   A,OUTBUF
        ANL   A,#11110111B
        JMP   OUTQ
SETQ:   MOV   A,OUTBUF
        ORL   A,#00001000B
OUTQ:   MOV   OUTBUF,A
        MOV   P3,A
```

b.) Using only bit-test instructions

```
:BFUNC2  SOLVE A RANDOM LOGIC
;        FUNCTION OF 6 VARIABLES
;        BY DIRECTLY POLLING EACH
;        BIT. (APPROACH USING
;        MCS-51 UNIQUE BIT-TEST
;        INSTRUCTION CAPABILITY.)
;        SYMBOLS USED IN LOGIC
;        DIAGRAM ASSIGNED TO
;        CORRESPONDING 8x51 BIT
;        ADDRESSES.
;
```

```
U        BIT   P1.1
V        BIT   P2.2
W        BIT   TF0
X        BIT   IE1
Y        BIT   20H.0
Z        BIT   21H.1
Q        BIT   P3.3
;        ...   ....
TEST_V:  JB    V,TEST_U
         JNB   W,TEST_X
TEST_U:  JB    U,SET_Q
TEST_X:  JNB   X,TEST_Z
         JNB   Y,SET_Q
TEST_Z:  JNB   Z,SET_Q
CLR_Q:   CLR   Q
         JMP   NXTTST
SET_Q:   SETB  Q
NXTTST:(CONTINUATION OF
       :PROGRAM)
```

c.) Using logical operations on Boolean variables

```
:FUNC3  SOLVE A RANDOM LOGIC
;       FUNCTION OF 6 VARIABLES
;       USING STRAIGHT_LINE
;       LOGICAL INSTRUCTIONS ON
;       MCS-51 BOOLEAN VARIABLES.
;
MOV C,V
ORL C,W   ;OUTPUT OF OR GATE
ANL C,U   ;OUTPUT OF TOP AND GATE
MOV F0,C  ;SAVE INTERMEDIATE STATE
MOV C,X
ANL C,Y   ;OUTPUT OF BOTTOM AND GATE
ORL C,F0  ;INCLUDE VALUE SAVED ABOVE
ORL C,Z   ;INCLUDE LAST INPUT
          ;VARIABLE
MOV Q,C   ;OUTPUT COMPUTED RESULT
```

An upper-limit can be placed on the complexity of software to simulate a large number of gates by summing the total number of inputs and outputs. The *actual* total should be somewhat shorter, since calculations can be "chained," as shown. The output of one gate is often the first input to another, bypassing the intermediate variable to eliminate two lines of source.

Design Example #4—Automotive Dashboard Functions

Now let's apply these techniques to designing the software for a complete controller system. This application is patterned after a familiar real-world application which isn't nearly as trivial as it might first appear: automobile turn signals.

Imagine the three position turn lever on the steering column as a single-pole, triple-throw toggle switch. In its central position all contacts are open. In the up or down positions contacts close causing corresponding lights in the rear of the car to blink. So far very simple.

Two more turn signals blink in the front of the car, and two others in the dashboard. All six bulbs flash when an emergency switch is closed. A thermo-mechanical relay (accessible under the dashboard in case it wears out) causes the blinking.

Applying the brake pedal turns the tail light filaments on constantly . . . unless a turn is in progress, in which case the blinking tail light is not affected. (Of course, the front turn signals and dashboard indicators are not affected by the brake pedal.) Table 6 summarizes these operating modes.

Table 6. Truth Table for Turn-Signal Operation

Input Signals				Output Signals			
Brake Switch	Emerg. Switch	Left Turn Switch	Right Turn Switch	Left Front & Dash	Right Front & Dash	Left Rear	Right Rear
0	0	0	0	Off	Off	Off	Off
0	0	0	1	Off	Blink	Off	Blink
0	0	1	0	Blink	Off	Blink	Off
0	1	0	0	Blink	Blink	Blink	Blink
0	1	0	1	Blink	Blink	Blink	Blink
0	1	1	0	Blink	Blink	Blink	Blink
1	0	0	0	Off	Off	On	On
1	0	0	1	Off	Blink	On	Blink
1	0	1	0	Blink	Off	Blink	On
1	1	0	0	Blink	Blink	On	On
1	1	0	1	Blink	Blink	On	Blink
1	1	1	0	Blink	Blink	Blink	On

But we're not done yet. Each of the exterior turn signal (but not the dashboard) bulbs has a second, somewhat dimmer filament for the parking lights. Figure 15 shows TTL circuitry which could control all six bulbs. The signals labeled "High Freq." and "Low Freq." represent two square-wave inputs. Basically, when one of the turn switches is closed or the emergency switch is activated the low frequency signal (about 1 Hz) is gated through to the appropriate dashboard indicator(s) and turn signal(s). The rear signals are also activated when the brake pedal is depressed provided a turn is not being made in the same direction. When the parking light switch is closed the higher frequency oscillator is gated to each front and rear turn signal, sustaining a low-intensity background level. (This is to eliminate the need for additional parking light filaments.)

In most cars, the switching logic to generate these functions requires a number of multiple-throw contacts. As many as 18 conductors thread the steering column of some automobiles solely for turn-signal and emergency blinker functions. (The author discovered this recently to his astonishment and dismay when replacing the whole assembly because of one burned contact.)

A multiple-conductor wiring harness runs to each corner of the car, behind the dash, up the steering column, and down to the blinker relay below. Connectors at each termination for each filament lead to extra cost and labor during construction, lower reliability and safety, and more costly repairs. And considering the system's present complexity, increasing its reliability or detecting failures would be quite difficult.

There are two reasons for going into such painful detail describing this example. First, to show that the messiest part of many system designs is determining what the controller should do. Writing the software to solve these functions will be comparatively easy. Secondly, to show the many potential failure points in the system. Later we'll see how the peripheral functions and intelligence built into a microcomputer (with a little creativity) can greatly reduce external interconnections and mechanical part count.

The Single-Chip Solution

The circuit shown in Figure 16 indicates five input pins to the five input variables—left-turn select, right-turn select, brake pedal down, emergency switch on, and parking lights on. Six output pins turn on the front, rear, and dashboard indicators for each side. The microcomputer implements all logical functions through software, which periodically updates the output signals as time elapses and input conditions change.

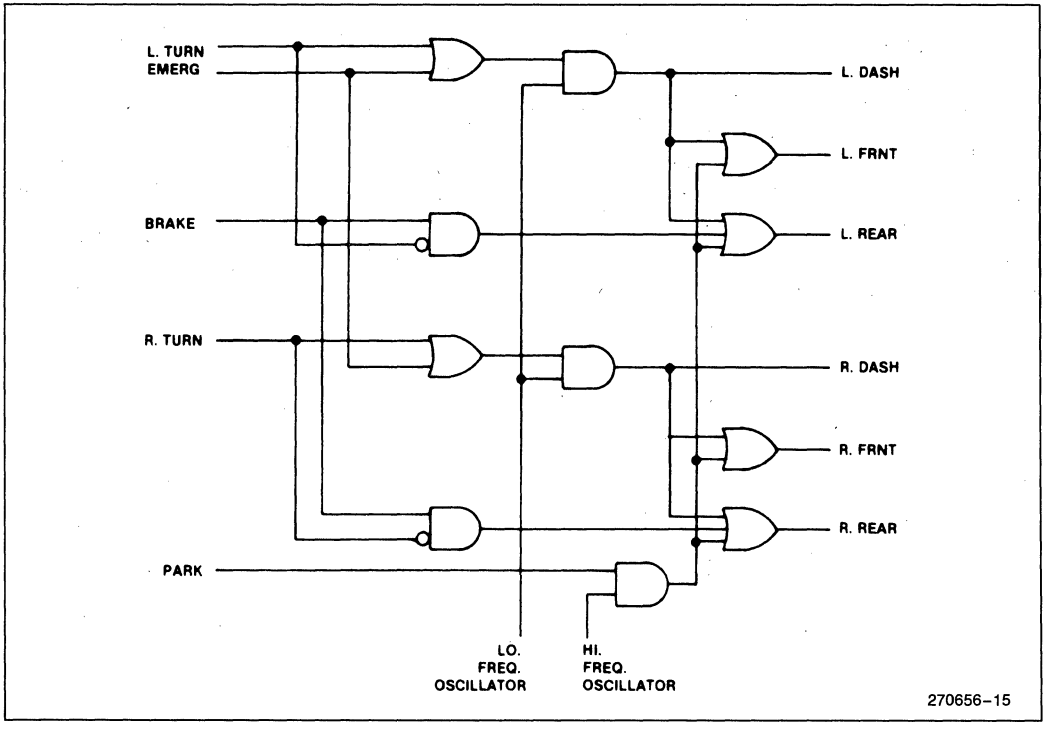


Figure 15. TTL Logic Implementation of Automotive Turn Signals

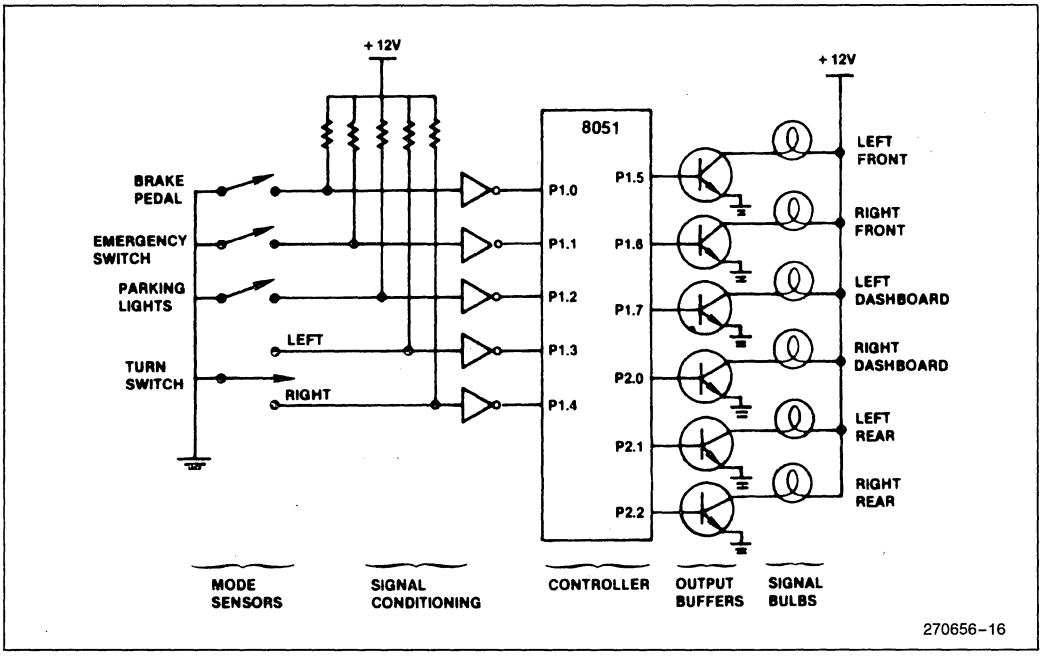


Figure 16. Microcomputer Turn-Signal Connections

Design Example #3 demonstrated that symbolic addressing with user-defined bit names makes code and documentation easier to write and maintain. Accordingly, we'll assign these I/O pins names for use throughout the program. (The format of this example will differ somewhat from the others. Segments of the overall program will be presented in sequence as each is described.)

```
;
;       INPUT PIN DECLARATIONS:
;(ALL INPUTS ARE POSITIVE-TRUE LOGIC)
;
BRAKE   BIT P1.0 ;BRAKE PEDAL
                 ;DEPRESSED
EMERG   BIT P1.1 ;EMERGENCY BLINKER
                 ;ACTIVATED
PARK    BIT P1.2 ;PARKING LIGHTS ON
I_TURN  BIT P1.3 ;TURN LEVER DOWN
R_TURN  BIT P1.4 ;TURN LEVER UP
;
;       OUTPUT PIN DECLARATIONS:
;
I_FRNT  BIT P1.5 ;FRONT LEFT-TURN
                 ;INDICATOR
R_FRNT  BIT P1.6 ;FRONT RIGHT-TURN
                 ;INDICATOR
I_DASH  BIT P1.7 ;DASHBOARD LEFT-TURN
                 ;INDICATOR
```

```
R_DASH  BIT P2.0 ;DASHBOARD RIGHT-
                 ;TURN INDICATOR
I_REAR  BIT P2.1 ;REAR LEFT-TURN
                 ;INDICATOR
R_REAR  BIT P2.2 ;REAR RIGHT-TURN
                 ;INDICATOR
;
```

Another key advantage of symbolic addressing will appear further on in the design cycle. The locations of cable connectors, signal conditioning circuitry, voltage regulators, heat sinks, and the like all affect P.C. board layout. It's quite likely that the somewhat arbitrary pin assignment defined early in the software design cycle will prove to be less than optimum; rearranging the I/O pin assignment could well allow a more compact module, or eliminate costly jumpers on a single-sided board. (These considerations apply especially to automotive and other cost-sensitive applications needing single-chip controllers.) Since other architectures mask bytes or use "clever" algorithms to isolate bits by rotating them into the carry, re-routing an input signal (from bit 1 of port 1, for example, to bit 4 of port 3) could require extensive modifications throughout the software.

The Boolean Processor's direct bit addressing makes such changes absolutely trivial. The number of the port containing the pin is irrelevent, and masks and complex

program structures are not needed. Only the initial Boolean variable declarations need to be changed; ASM51 automatically adjusts all addresses and symbolic references to the reassigned variables. The user is assured that no additional debugging or software verification will be required.

```
;           ...         .....
;INTERRUPT RATE SUBDIVIDER
SUB_DIV    DATA        20H
;HIGH-FREQUENCY OSCILLATOR BIT
HI_FREQ    BIT         SUB_DIV,0
;LOW-FREQUENCY OSCILLATOR BIT
LO_FREQ    BIT         SUB_DIV,7
;           ...
           ORG         0000H
JMP        INIT
;           ...         .....
           ORG         100H
;PUT TIMER 0 IN MODE 1
INIT;      MOV         TMOD,#00000001B
;INITIALIZE TIMER REGISTERS
           MOV         TL0,#0
           MOV         TH0,#-16
;SUBDIVIDE INTERRUPT RATE BY 244
           MOV         SUB_DIV,#244
;ENABLE TIMER INTERRUPTS
           SETB        ET0
;GLOBALLY ENABLE ALL INTERRUPTS
           SETB        EA
;START TIMER
           SETB        TR0
;
;(CONTINUE WITH BACKGROUND PROGRAM)
;
;PUT TIMER 0 IN MODE 1
;INITIALIZE TIMER REGISTERS

;SUBDIVIDE INTERRUPT RATE BY 244
;ENABLE TIMER INTERRUPTS
;GLOBALLY ENABLE ALL INTERRUPTS
;START TIMER
```

Timer 0 (one of the two on-chip timer counters) replaces the thermo-mechanical blinker relay in the dashboard controller. During system initialization it is configured as a timer in mode 1 by setting the least significant bit of the timer mode register (TMOD). In this configuration the low-order byte (TL0) is incremented every machine cycle, overflowing and incrementing the high-order byte (TH0) every 256 μs. Timer interrupt 0 is enabled so that a hardware interrupt will occur each time TH0 overflows.

An eight-bit variable in the bit-addressable RAM array will be needed to further subdivide the interrupts via software. The lowest-order bit of this counter toggles very fast to modulate the parking lights: bit 7 will be "tuned" to approximately 1 Hz for the turn- and emergency-indicator blinking rate.

Loading TH0 with -16 will cause an interrupt after 4.096 ms. The interrupt service routine reloads the high-order byte of timer 0 for the next interval, saves the CPU registers likely to be affected on the stack, and then decrements SUB_DIV. Loading SUB_DIV. with 244 initially and each time it decrements to zero will produce a 0.999 second period for the highest-order bit.

```
ORG  000BH   ;TIMER 0 SERVICE VECTOR
MOV  TH0,#-16
PUSH PSW
PUSH ACC
PUSH B
DJNZ SUB_DIV,T0SERV
MOV  SUB_DIV,#244
```

The code to sample inputs, perform calculations, and update outputs—the real "meat" of the signal controller algorithm—may be performed either as part of the interrupt service routine or as part of a background program loop. The only concern is that it must be executed at least serveral dozen times per second to prevent parking light flickering. We will assume the former case, and insert the code into the timer 0 service routine.

First, notice from the logic diagram (Figure 15) that the subterm (PARK • H_FREQ), asserted when the parking lights are to be on dimly, figures into four of the six output functions. Accordingly, we will first compute that term and save it in a temporary location named "DIM". The PSW contains two general purpose flags: F0, which corresponds to the 8048 flag of the same name, and PSW.1. Since the PSW has been saved and will be restored to its previous state after servicing the interrupt, we can use either bit for temporary storage.

```
  DIM BIT   PSW.1 ;DECLARE TEMP
                  ;STORAGE FLAG
; ... .....
  MOV C,PARK      ;GATE PARKING
                  ;LIGHT SWITCH
  ANL HI_FREQ     ;WITH HIGH
                  ;FREQUENCY
                  ;SIGNAL
  MOV DIM,C       ;AND SAVE IN
                  ;TEMP. VARIABLE
```

This simple three-line section of code illustrates a remarkable point. The software indicates in very abstract terms exactly what function is being performed, inde-

pendent of the hardware configuration. The fact that these three bits include an input pin, a bit within a program variable, and a software flag in the PSW is totally invisible to the programmer.

Now generate and output the dashboard left turn signal.

```
;
  MOV C,L_TURN      ;SET CARRY IF
                    ;TURN
  ORL C,EMERG       ;OR EMERGENCY
                    ;SELECTED
  ANL C,LO_FREQ     ;GATE IN 1 HZ
                    ;SIGNAL
  MOV I_DASH,C      ;AND OUTPUT TO
                    ;DASHBOARD
```

To generate the left front turn signal we only need to add the parking light function in F0. But notice that the function in the carry will also be needed for the rear signal. We can save effort later by saving its current state in F0.

```
;
  MOV F0,C          ;SAVE FUNCTION
                    ;SO FAR
  ORL C,DIM         ;ADD IN PARKING
                    ;LIGHT FUNCTION
  MOV L_FRNT,C      ;AND OUTPUT TO
                    ;TURN SIGNAL
```

Finally, the rear left turn signal should also be on when the brake pedal is depressed, provided a left turn is not in progress.

```
MOV C,BRAKE       ;GATE BRAKE
                  ;PEDAL SWITCH
ANL C,L_TURN      ;WITH TURN
                  ;LEVER
ORL C,F0          ;INCLUDE TEMP.
                  ;VARIABLE FROM DASH
ORL C,DIM         ;AND PARKING
                  ;LIGHT FUNCTION
MOV L_REAR,C      ;AND OUTPUT TO
                  ;TURN SIGNAL
```

Now we have to go through a similar sequence for the right-hand equivalents to all the left-turn lights. This also gives us a chance to see how the code segments above look when combined.

```
MOV C.R_TURN      ;SET CARRY H-
                  ;TURN
ORL C.EMERG       ;OR EMERGENCY
                  ;SELECTED
ANL C,LO_FREQ     ;IF SO. GATE IN 1
                  ;HZ SIGNAL
MOV R_DASH.C      ;AND OUTPUT TO
                  ;DASHBOARD
MOV F0.C          ;SAVE FUNCTION
                  ;SO FAR
ORL C.DIM         ;ADD IN PARKING
                  ;LIGHT FUNCTION
MOV R_FRNT.C      ;AND OUTPUT TO
                  ;TURN SIGNAL
MOV C.BRAKE       ;GATE BRAKE
                  ;PEDAL SWITCH
ANL C. R_TURN     ;WITH TURN
                  ;LEVER
ORL C.F0          ;INCLUDE TEMP.
                  ;VARIABLE FROM
                  ;DASH
ORL C.DIM         ;AND PARKING
                  ;LIGHT FUNCTION
MOV R_REAR.C      ;AND OUTPUT TO
                  ;TURN SIGNAL
```

(The perceptive reader may notice that simply rearranging the steps could eliminate one instruction from each sequence.)

Now that all six bulbs are in the proper states, we can return from the interrupt routine, and the program is finished. This code essentially needs to reverse the status saving steps at the beginning of the interrupt.

Table 7. Non-Trivial Duty Cycles

Sub_Div Bits								Duty Cycles						
7	6	5	4	3	2	1	0	12.5%	25.0%	37.5%	50.0%	62.5%	75.0%	87.5%
X	X	X	X	X	0	0	0	Off	Off	Off	Off	Off	Off	Off
X	X	X	X	X	0	0	1	Off	Off	Off	Off	Off	Off	On
X	X	X	X	X	0	1	0	Off	Off	Off	Off	Off	On	On
X	X	X	X	X	0	1	1	Off	Off	Off	Off	On	On	On
X	X	X	X	X	1	0	0	Off	Off	Off	On	On	On	On
X	X	X	X	X	1	0	1	Off	Off	On	On	On	On	On
X	X	X	X	X	1	1	0	Off	On	On	On	On	On	On
X	X	X	X	X	1	1	1	On	On	On	On	On	On	On

```
POP  B              ;RESTORE CPU
                    ;REGISTERS.
POP ACC
POP PSW
RETI
```

Program Refinements. The luminescence of an incandescent light bulb filament is generally non-linear: the 50% duty cycle of HI_FREQ may not produce the desired intensity. If the application requires, duty cycles of 25%, 75%, etc. are easily achieved by ANDing and ORing in additional low-order bits of SUB_DIV. For example, 30 H/ signals of seven different duty cycles could be produced by considering bits 2–0 as shown in Table 7. The only software change required would be to the code which sets-up variable DIM;

```
MOV C,SUB_DIV.1 ;START WITH 50
                ;PERCENT
ANL C,SUB_DIV.0 ;MASK DOWN TO 25
                ;PERCENT
ORL C,SUB_DIV.2 ;AND BUILD BACK TO
                ;62 PERCENT
MOV DIM,C       ;DUTY CYCLE FOR
                ;PARKING LIGHTS.
```

Interconnections increase cost and decrease reliability. The simple buffered pin-per-function circuit in Figure 16 is insufficient when many outputs require higher-than-TTL drive levels. A lower-cost solution uses the 8051 serial port in the shift-register mode to augment I/O. In mode 0, writing a byte to the serial port data buffer (SBUF) causes the data to be output sequentially through the "RXD" pin while a burst of eight clock pulses is generated on the "TXD" pin. A shift register connected to these pins (Figure 17) will load the data byte as it is shifted out. A number of special peripheral driver circuits combining shift-register inputs with high drive level outputs have been introduced recently.

Cascading multiple shift registers end-to-end will expand the number of outputs even further. The data rate in the I/O expansion mode is one megabaud, or 8 μs. per byte. This is the mode which the serial port defaults to following a reset, so no initialization is required.

The software for this technique uses the B register as a "map" corresponding to the different output functions. The program manipulates these bits instead of the output pins. After all functions have been calculated the B register is shifted by the serial port to the shift-register driver. (While some outputs may glitch as data is shifted through them, at 1 Megabaud most people wouldn't notice. Some shift registers provide an "enable" bit to hold the output states while new data is being shifted in.)

This is where the earlier decision to address bits symbolically throughout the program is going to pay off. This major I/O restructuring is nearly as simple to implement as rearranging the input pins. Again, only the bit declarations need to be changed.

```
I_FRNT BIT B.0  ;FRONT LEFT-TURN
                ;INDICATOR
R_FRNT BIT B.1  ;FRONT RIGHT-TURN
                ;INDICATOR
I_DASH BIT B.2  ;DASHBOARD LEFT-TURN
                ;INDICATOR
R_DASH BIT B.3  ;DASHBOARD RIGHT-TURN
                ;INDICATOR
I_REAR BIT B.4  ;REAR LEFT-TURN
                ;INDICATOR
R_REAR BIT B.5  ;REAR RIGHT-TURN
                ;INDICATOR
```

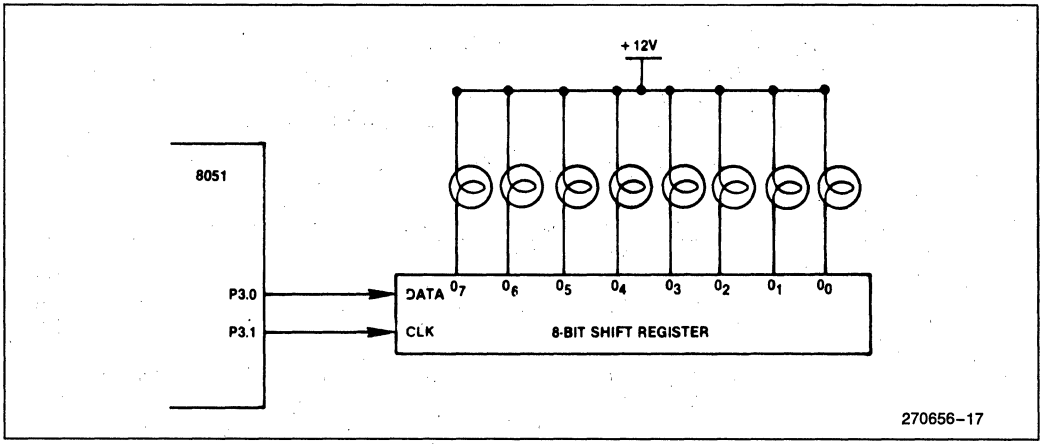


Figure 17. Output Expansion Using Serial Port

The original program to compute the functions need not change. After computing the output variables, the control map is transmitted to the buffered shift register through the serial port.

```
MOV SBUF,B ;LOAD BUFFER AND TRANSMIT
```

The Boolean Processor solution holds a number of advantages over older methods. Fewer switches are required. Each is simpler, requiring fewer poles and lower current contacts. The flasher relay is eliminated entirely. Only six filaments are driven, rather than 10. The wiring harness is therefore simpler and less expensive—one conductor for each of the six lamps and each of the five sensor switches. The fewer conductors use far fewer connectors. The whole system is more reliable.

And since the system is much simpler it would be feasible to implement redundancy and or fault detection on the four main turn indicators. Each could still be a standard double filament bulb, but with the filaments driven in parallel to tolerate single-element failures.

Even with redundancy, the lights will eventually fail. To handle this inescapable fact current or voltage sensing circuits on each main drive wire can verify that each bulb and its high-current driver is functioning properly. Figure 18 shows one such circuit.

Assume all of the lights are turned on except one: i.e., all but one of the collectors are grounded. For the bulb which is turned off, if there is continuity from +12V through the bulb base and filament, the control wire, all connectors, and the P.C. board traces, and if the transistor is indeed not shorted to ground, then the collector will be pulled to +12V. This turns on the base of Q8 through the corresponding resistor, and grounds the input pin, verifying that the bulb circuit is operational. The continuity of each circuit can be checked by software in this way.

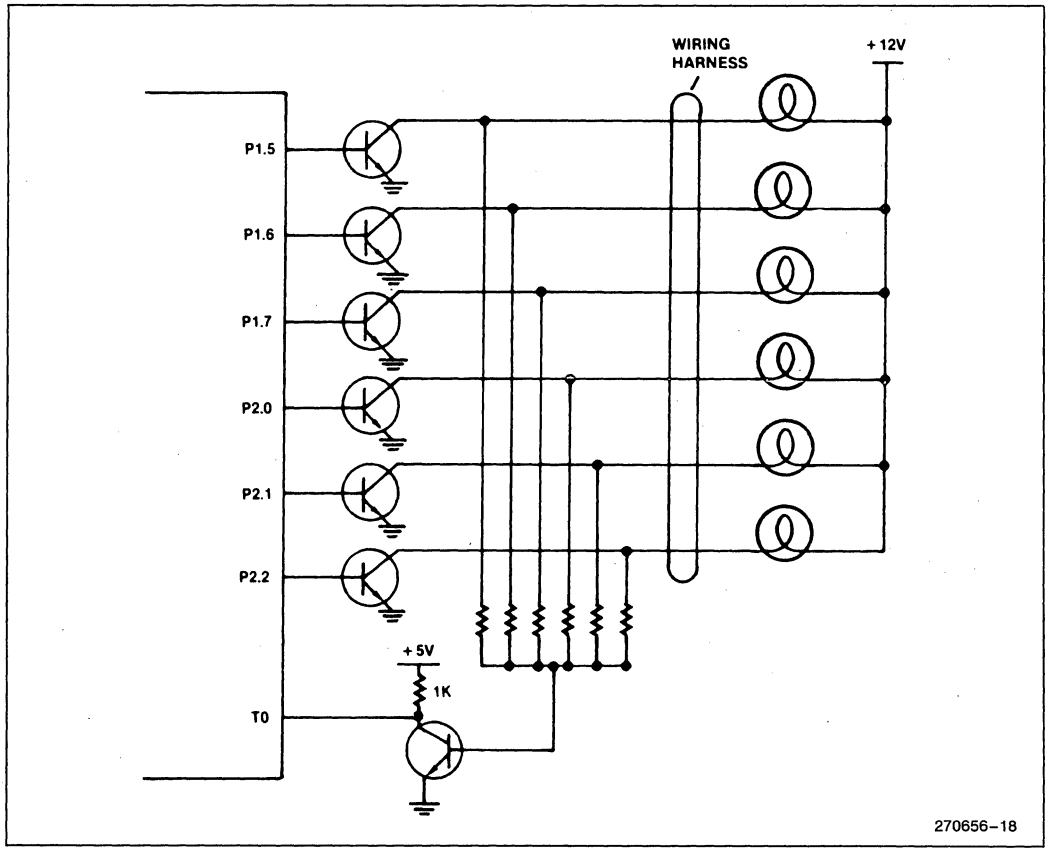


Figure 18

Now turn *all* the bulbs on, grounding all the collectors. Q7 should be turned off, and the Test pin should be high. However, a control wire shorted to + 12V or an open-circuited drive transistor would leave one of the collectors at the higher voltage even now. This too would turn on Q7, indicating a different type of failure. Software could perform these checks once per second by executing the routine every time the software counter SUB_DIV is reloaded by the interrupt routine.

```
DJNZ SUB_DIV,TOSERV
MOV SUB_DIV,#244      ;RELOAD COUNTER
ORL P1,#11100000B     ;SET CONTROL
                      ;OUTPUTS HIGH
ORL P2,#00000111B
CLR I_FRNT            ;FLOAT DRIVE
                      ;COLLECTOR
JB  TO,FAULT          ;TO SHOULD BE
                      ;PULLED LOW
SETB L_FRNT           ;PULL COLLECTOR
                      ;BACK DOWN
    CLR  L_DASH
    JB   TO,FAULT
    SETB L_DASH
    CLR  L_REAR
    JB   TO,FAULT
    SETB L_REAR
    CLR  R_FRNT
    JB   TO,FAULT
    SETB R_FRNT
    CLR  R_DASH
    JB   TO,FAULT
    SETB R_DASH
    CLR  R_REAR
    JB   TO,FAULT
    SETB R_REAR
;
;WITH ALL COLLECTORS GROUNDED. TO
 SHOULD BE HIGH
;IF SO. CONTINUE WITH INTERRUPT
 ROUTINE.
  JB   TO,TOSERV
FAULT:                ;ELECTRICAL
                      ;FAILURE
                      ;PROCESSING
                      ;ROUTINE
                      ;(LEFT TO
                      ;READER'S
                      ;IMAGINATION)
TOSERV:               ;CONTINUE WITH
                      ;INTERRUPT
                      ;PROCESSING
;
;
```

The complete assembled program listing is printed in Appendix A. The resulting code consists of 67 program statements, not counting declarations and comments, which assemble into 150 bytes of object code. Each pass through the service routine requires (coincidently) 67 μs plus 32 μs once per second for the electrical test. If executed every 4 ms as suggested this software would typically reduce the throughput of the background program by less than 2%.

Once a microcomputer has been designed into a system, new features suddenly become virtually free. Software could make the emergency blinkers flash alternately or at a rate faster than the turn signals. Turn signals could override the emergency blinkers. Adding more bulbs would allow multiple tail light sequencing and syncopation—true flash factor, so to speak.

Design Example #5—Complex Control Functions

Finally, we'll mix byte and bit operations to extend the use of 8051 into extremely complex applications.

Programmers can arbitrarily assign I/O pins to input and output functions only if the total does not exceed 32, which is insufficient for applications with a very large number of input variables. One way to expand the number of inputs is with a technique similar to multiplexed-keyboard scanning.

Figure 19 shows a block diagram for a moderately complex programmable industrial controller with the following characteristics:

- 64 input variable sensors:
- 12 output signals:
- Combinational and sequential logic computations:
- Remote operation with communications to a host processor via a high-speed full-duplex serial link:
- Two prioritized external interrupts:
- Internal real-time and time-of-day clocks.

While many microprocessors could be programmed to provide these capabilities with assorted peripheral support chips, an 8051 microcomputer needs no other integrated circuits!

The 64 input sensors are logically arranged as an 8x8 matrix. The pins of Port 1 sequentially enable each column of the sensor matrix: as each is enabled Port 0 reads in the state of each sensor in that column. An eight-byte block in bit-addressable RAM remembers the data as it is read in so that after each complete scan cycle there is an internal map of the current state of all sensors. Logic functions can then directly address the elements of the bit map.

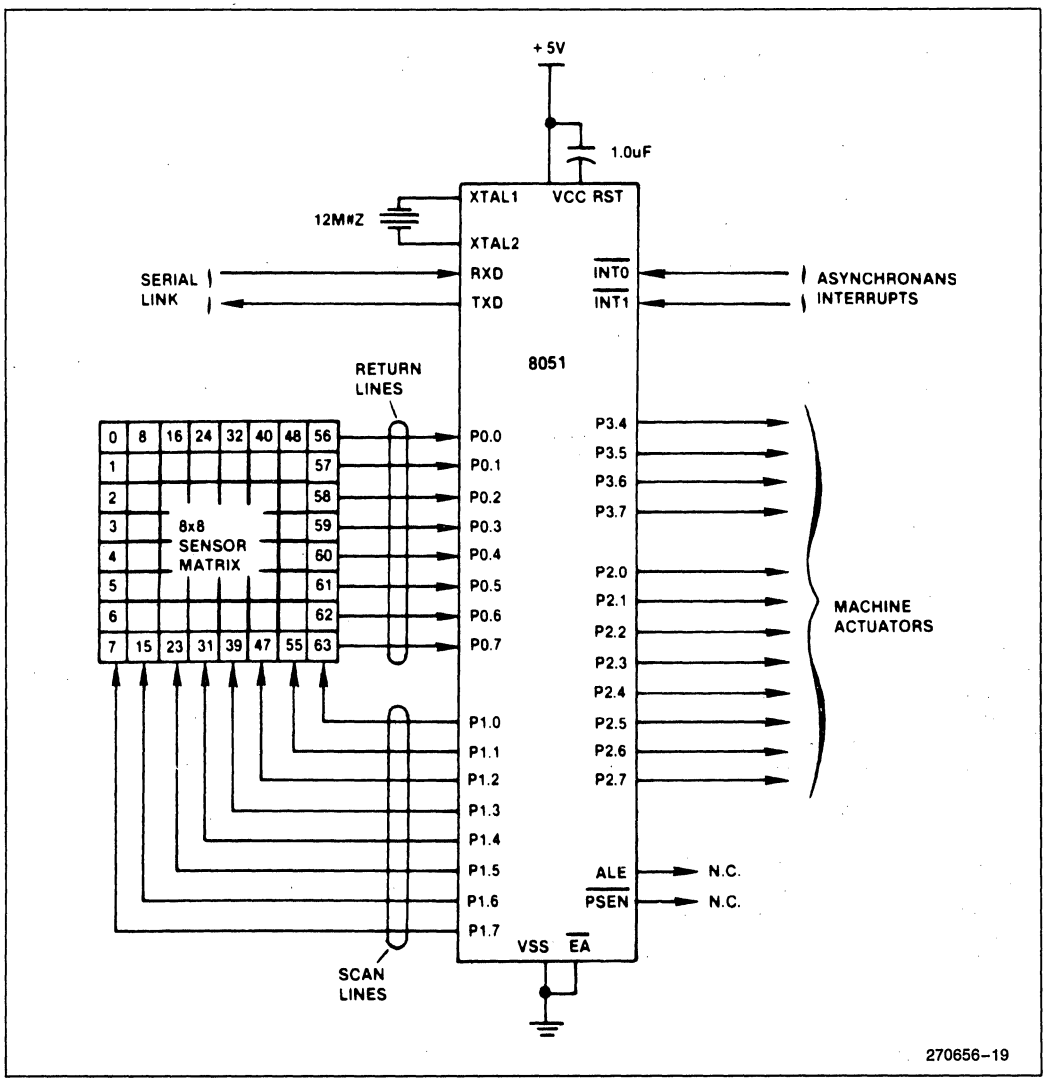


Figure 19. Block Diagram of 64-Input Machine Controller

The computer's serial port is configured as a nine-bit UART, transferring data at 17,000 bytes-per-second. The ninth bit may distinguish between address and data bytes.

The 8051 serial port can be configured to detect bytes with the address bit set, automatically ignoring all others. Pins INT0 and INT1 are interrupts configured respectively as high-priority, falling-edge triggered and low-priority, low-level triggered. The remaining 12 I/O pins output TTL-level control signals to 12 actuators.

There are several ways to implement the sensor matrix circuitry, all logically similar. Figure 20a shows one possibility. Each of the 64 sensors consists of a pair of simple switch contacts in series with a diode to permit multiple contact closures throughout the matrix.

The scan lines from Port 1 provide eight un-encoded active-high scan signals for enabling columns of the matrix. The return lines on rows where a contact is closed are pulled high and read as logic ones. Open return lines are pulled to ground by one of the 40 kΩ resistors and are read as zeroes. (The resistor values must be chosen to ensure all return lines are pulled above the 2.0V logic threshold, even in the worst-case,

where all contacts in an enabled column are closed.) Since P0 is provided open-collector outputs and high-impedance MOS inputs its input loading may be considered negligible.

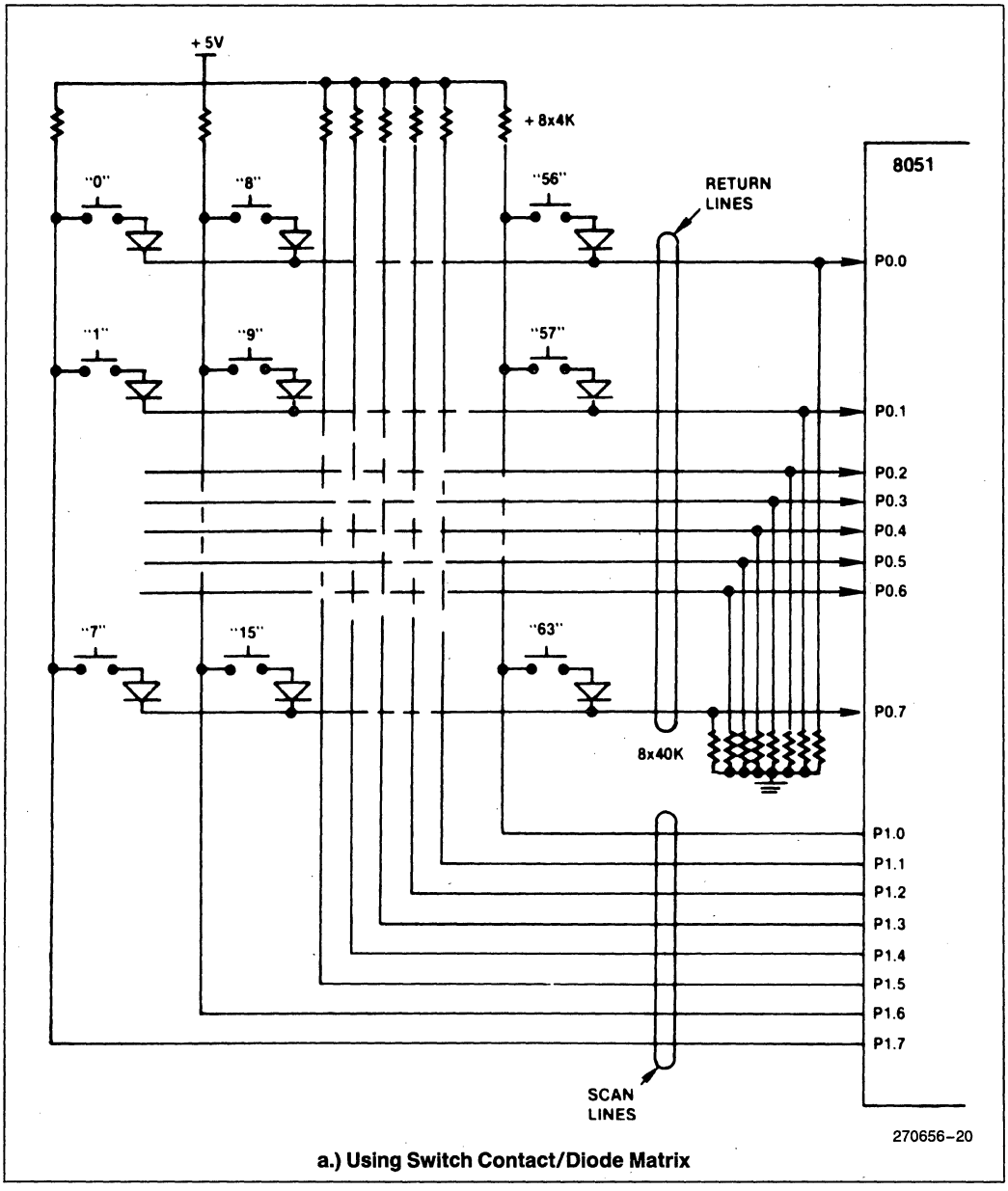
The circuits in Figures 20b–20d are variations on this theme. When input signals must be electrically isolated from the computer circuitry as in noisy industrial environments, phototransistors can replace the switch diode pairs and provide optical isolation as in Figure 20b. Additional opto-isolators could also be used on the control output and special signal lines.

The other circuits assume that input signals are already at TTL levels. Figure 20c uses octal three-state buffers enabled by active-low scan signals to gate eight signals onto Port 0. Port 0 is available for memory expansion or peripheral chip interfacing between sensor matrix scans. Eight-to-one multiplexers in Figure 20d select one of eight inputs for each return line as determined by encoded address bits output on three pins of Port 1. (Five more output pins are thus freed for more control functions.) Each output can drive at least one standard TTL or up to 10 low-power TTL loads without additional buffering.

Going back to the original matrix circuit, Figure 21 shows the method used to scan the sensor matrix. Two complete bit maps are maintained in the bit-addressable region of the RAM: one for the current state and one for the previous state read for each sensor. If the need arises, the program could then sense input transitions and or debounce contact closures by comparing each bit with its earlier value.

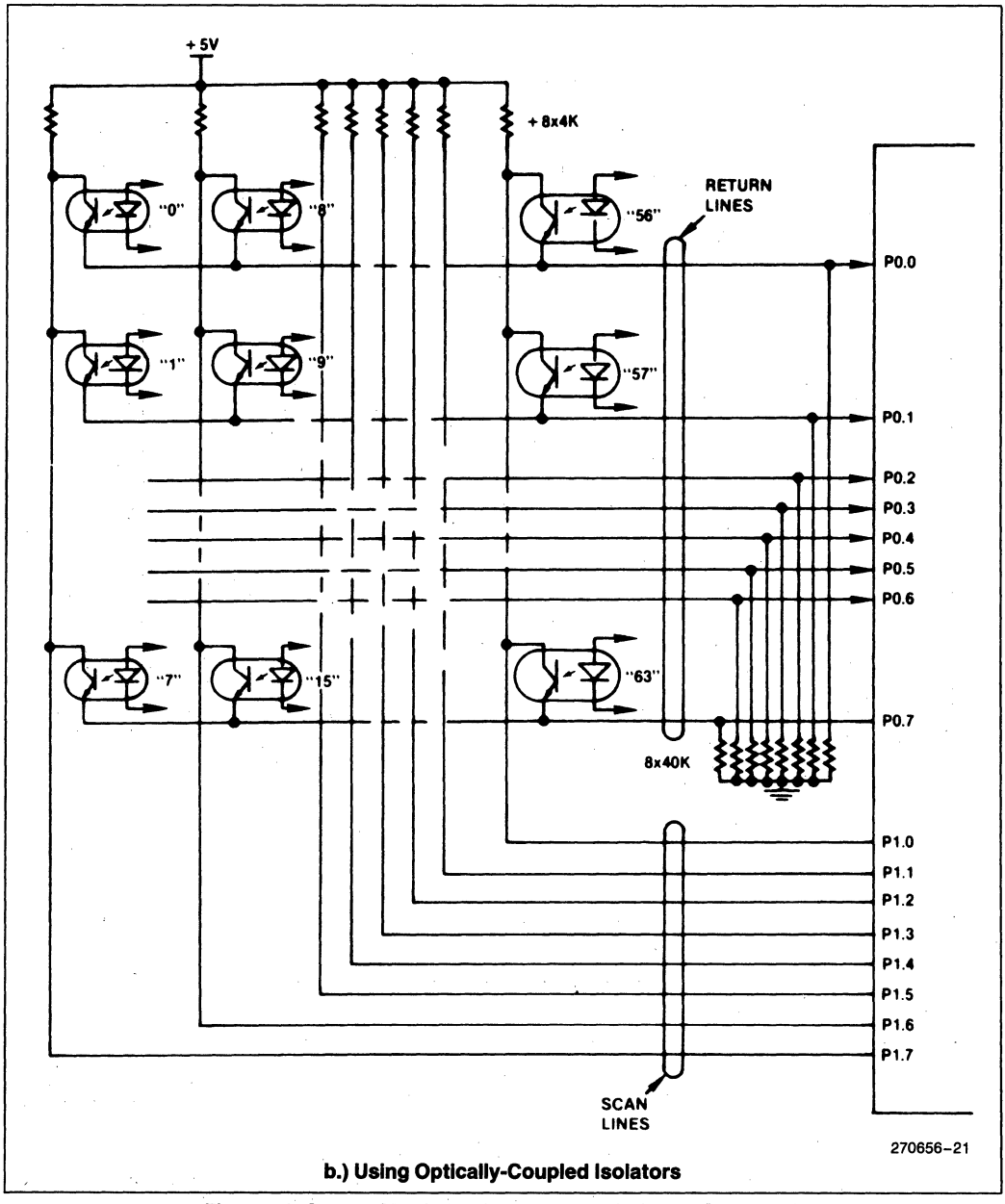
The code in Example 3 implements the scanning algorithm for the circuits in Figure 20a. Each column is enabled by setting a single bit in a field of zeroes. The bit maps are positive logic: ones represent contacts that are closed or isolators turned on.

```
Example 3.
INPUT_SCAN:     ;SUBROUTINE TO READ
                ;CURRENT STATE
                ;OF 64 SENSORS AND
                ;SAVE IN RAM 20H-27H
      MOV R0,#20H    ;INITIALIZE
                     ;POINTERS
      MOV R1,#28H    ;FOR BIT MAP
                     ;BASES
      MOV A,#80H     ;SET FIRST BIT
                     ;IN ACC
SCAN; MOV P1,A       ;OUTPUT TO SCAN
                     ;LINES
      RR  A          ;SHIFT TO ENABLE
                     ;NEXT COLUMN
                     ;NEXT
      MOV R2,A       ;REMEMBER CUR-
                     ;RENT SCAN
                     ;POSITION
      MOV A,P0       ;READ RETURN
                     ;LINES
      XCH A,@R0      ;SWITCH WITH
                     ;PREVIOUS MAP
                     ;BITS
      MOV @R1,A      ;SAVE PREVIOUS
                     ;STATE AS WELL
      INC R0         ;BUMP POINTERS
      INC R1
      MOV A,R2       ;RELOAD SCAN
                     ;LINE MASK
      JNB ACC,7;SCAN;LOOP UNTIL ALL
                     ;EIGHT COLUMNS
                     ;READ
      RET
```



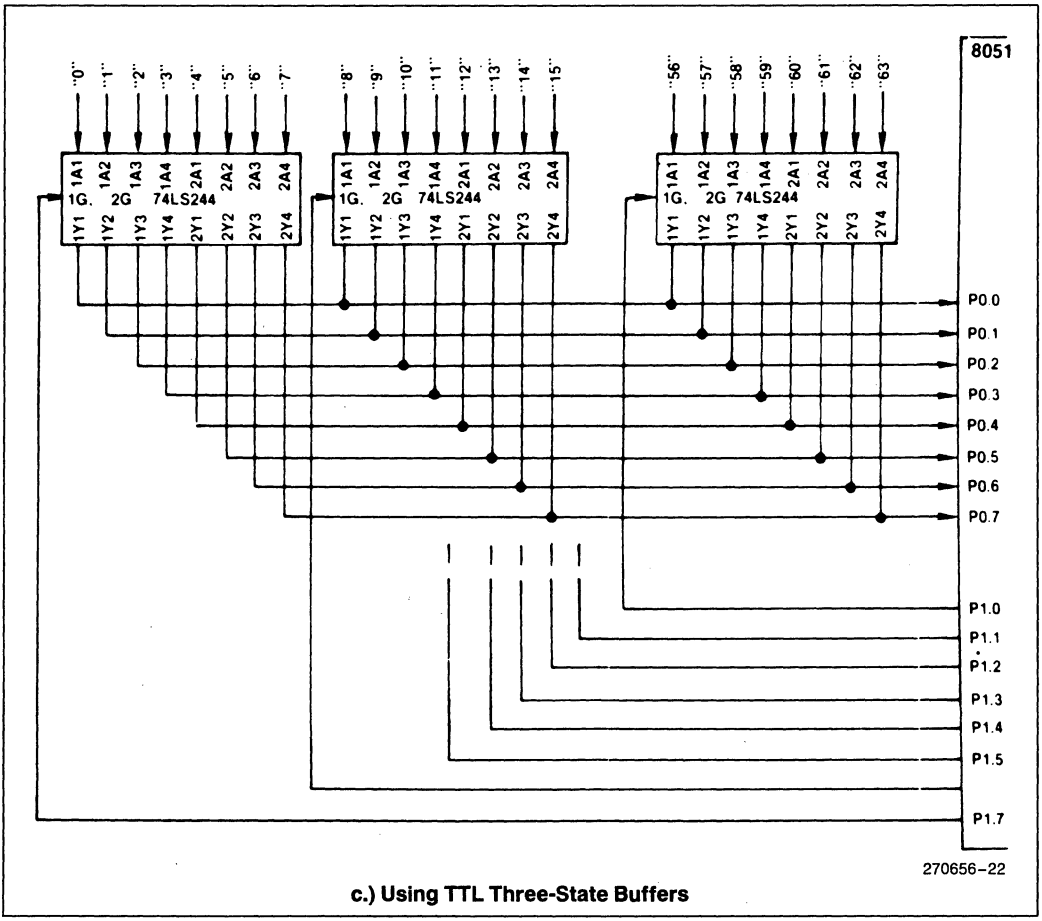
a.) Using Switch Contact/Diode Matrix

Figure 20. Sensor Matrix Implementation Methods



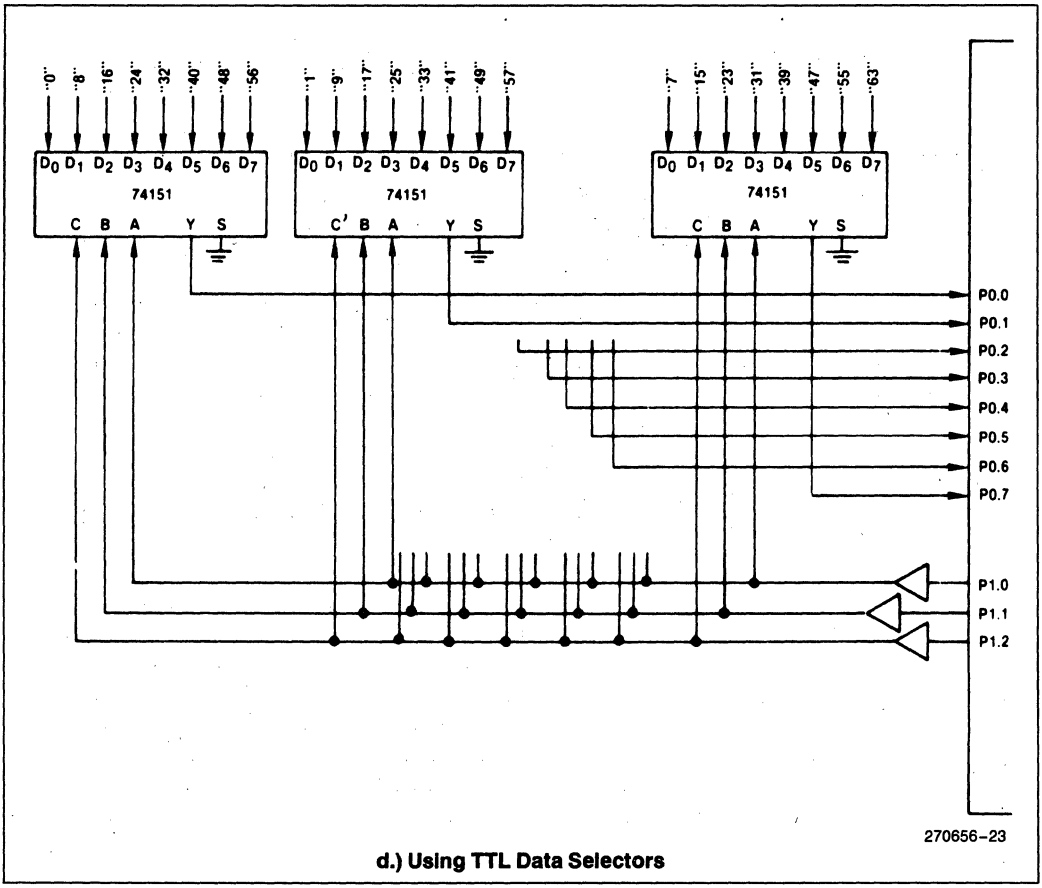
b.) Using Optically-Coupled Isolators

Figure 20. Sensor Matrix Implementation Methods (Continued)



c.) Using TTL Three-State Buffers

Figure 20. Sensor Matrix Implementation Methods (Continued)



d.) Using TTL Data Selectors

Figure 20. Sensor Matrix Implementation Methods (Continued)

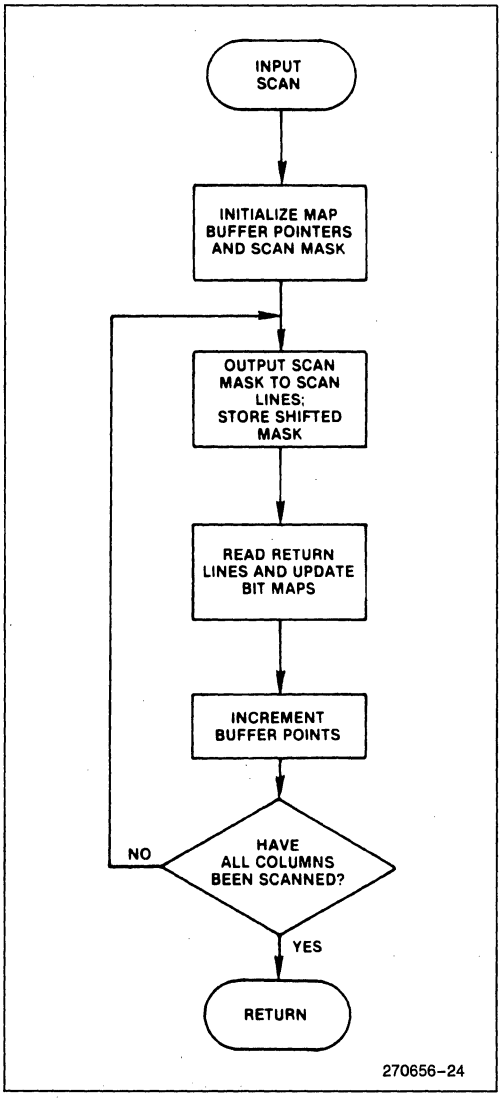


Figure 21. Flowchart for Reading in Sensor Matrix

What happens after the sensors have been scanned depends on the individual application. Rather than inventing some artificial design problem, software corresponding to commonplace logic elements will be discussed.

Combinatorial Output Variables. An output variable which is a simple (or not so simple) combinational function of several input variables is computed in the spirit of Design Example 3. All 64 inputs are represented in the bit maps: in fact, the sensor numbers in Figure 20 correspond to the absolute bit addresses in RAM! The code in Example 4 activates an actuator connected to P2.2 when sensors 12, 23, and 34 are closed and sensors 45 and 56 are open.

Example 4.
Simple Combinatorial Output Variables.

```
;SET P2.2=(12)(23)(34)( 45)( 56)
      MOV C,12
      ANL C,23
      ANL C,34
      ANL C, 45
      ANL C, 56
      MOV P2.2,C
```

Intermediate Variables. The examination of a typical relay-logic ladder diagram will show that many of the rungs control *not* outputs but rather relays whose contacts figure into the computation of other functions. In effect, these relays indicate the state of intermediate variables of a computation.

The MCS-51 solution can use any directly addressable bit for the storage of such intermediate variables. Even when all 128 bits of the RAM array are dedicated (to input bit maps in this example), the accumulator, PSW, and B register provide 18 additional flags for intermediate variables.

For example, suppose switches 0 through 3 control a safety interlock system. Closing any of them should deactivate certain outputs. Figure 22 is a ladder diagram for this situation. The interlock function could be recomputed for every output affected, or it may be computed once and save (as implied by the diagram). As the program proceeds this bit can qualify each output.

Example 5. Incorporating Override signal into actuator outputs.

```
;       CALL INPUT_SCAN
        MOV C,0
        ORL C,1
        ORL C,2
        ORL C,3
        MOV F0,C
;     .....  .....
;     COMPUTE FUNCTION 0
;
        ANL C, F0
        MOV PL0,C
;     .....  .....
;     COMPUTE FUNCTION 1
;
        ANL C, F0
        MOV P1,1,C
;     .....  .....
;     COMPUTE FUNCTION 2
;
        ANL C, F0
        MOV P1,2,C
;     .....  .....
```

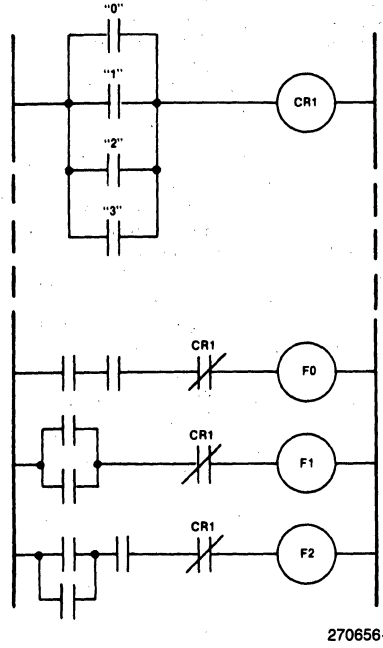


Figure 22. Ladder Diagram for Output Override Circuitry

Latching Relays. A latching relay can be forced into either the ON or OFF state by two corresponding input signals, where it will remain until forced onto the opposite state—analogous to a TTL Set/Reset flip-flop. The relay is used as an intermediate variable for other calculations. In the previous example, the emergency condition could be remembered and remain active until an "emergency cleared" button is pressed.

Any flag or addressable bit may represent a latching relay with a few lines of code (see Example 6).

Example 6. Simulating a latching relay.

```
;I_SET SET FLAG 0 IF C=1
I_SET:  ORL C,F0
        MOV F0,C
;

;I_RSET RESET FLAG 0 IF C=1
I_RSET: CPS C
        ANL C,F0
        MOV F0,C
;
```

Time Delay Relays. A time delay relay does not respond to an input signal until it has been present (or absent) for some predefined time. For example, a ballast or load resistor may be switched in series with a D.C. motor when it is first turned on, and shunted from the circuit after one second. This sort of time delay may be simulated by an interrupt routine driven by one of the two 8051 timer counters. The procedure followed by the routine depends heavily on the details of the exact function needed: time-outs or time delays with resettable or non-resettable inputs are possible. If the interrupt routine is executed every 10 milliseconds the code in Example 7 will clear an intermediate variable set by the background program after it has been active for two seconds.

Example 7. Code to clear USRFLG after a fixed time delay.

```
        JNB   USR_FLG,NXTTST
        DJNZ  DLAY_COUNT,NXTTST
        CLR   USR_FLG
        MOV   DLAY_COUNT,#200
NXTTST; ;...  .....
```

Serial Interface to Remote Processor. When it detects emergency conditions represented by certain input combinations (such as the earlier Emergency Override), the controller could shut down the machine immediately and/or alert the host processor via the serial port. Code bytes indicating the nature of the problem could be transmitted to a central computer. In fact, at 17,000 bytes-per-second, the entire contents of both bit maps could be sent to the host processor for further analysis in less than a millisecond! If the host decides that conditions warrant, it could alert other remote processors in the system that a problem exists and specify which shut-down sequence each should initiate. For more information on using the serial port, consult the MCS-51 User's Manual.

Response Timing

One difference between relay and programmed industrial controllers (when each is considered as a "black box") is their respective reaction times to input changes. As reflected by a ladder diagram, relay systems contain a large number of "rungs" operating in parallel. A change in input conditions will begin propagating through the system immediately, possibly affecting the output state within milliseconds.

Software, on the other hand, operates sequentially. A change in input states will not be detected until the next time an input scan is performed, and will not affect the outputs until that section of the program is reached. For that reason the raw speed of computing the logical functions is of extreme importance.

Here the Boolean processor pays off. *Every instruction mentioned in this Note* completes in one or two microseconds—the *minimum* instruction execution time for many other microcontrollers! A ladder diagram containing a hundred rungs, with an average of four contacts per rung can be replaced by approximately five hundred lines of software. A complete pass through the entire matrix scanning routine and all computations would require about a millisecond: less than the time it takes for most relays to change state.

A programmed controller which simulates each Boolean function with a subroutine would be less efficient by at least an order of magnitude. Extra software is needed for the simulation routines, and each step takes longer to execute for three reasons: several byte-wide logical instructions are executed per user program step (rather than one Boolean operation): most of those instructions take longer to execute with microprocessors performing multiple off-chip accesses: and calling and returning from the various subroutines requires overhead for stack operations.

In fact, the speed of the Boolean Processor solution is likely to be much faster than the system requires. The CPU might use the time left over to compute feedback parameters, collect and analyze execution statistics, perform system diagnostics, and so forth.

Additional Functions and Uses

With the building-block basics mentioned above many more operations may be synthesized by short instruction sequences.

Exclusive-OR. There are no common mechanical devices or relays analogous to the Exclusive-OR operation, so this instruction was omitted from the Boolean Processor. However, the Exclusive-OR or Exclusive-NOR operation may be performed in two instructions by conditionally complementing the carry or a Boolean variable based on the state of any other testable bit.

```
;EXCLUSIVE-;OR FUNCTION IMPOSED ON CARRY
;USING F0 IS INPUT VARIABLE.
;XOR_F0: JNB F0,XORCNT ;("JB" FOR X-NOR)
        CPL C
;XORCNT: ... .....
```

XCH. The contents of the carry and some other bit may be exchanged (switched) by using the accumulator as temporary storage. Bits can be moved into and out of the accumulator simultaneously using the Rotate-

through-carry instructions, though this would alter the accumulator data.

```
;EXCHANGE CARRY WITH USRFLG
XCHBIT: RLC   A
        MOV   C,USR_FLG
        RRC   A
        MOV   USR_FLG,C
        RLC   A
```

Extended Bit Addressing. The 8051 can directly address 144 general-purpose bits for all instructions in Figure 3b. Similar operations may be extended to any bit anywhere on the chip with some loss of efficiency.

The logical operations AND, OR, and Exclusive-OR are performed on byte variables using six different addressing modes, one of which lets the source be an immediate mask, and the destination any directly addressable byte. Any bit may thus be set, cleared, or complemented with a three-byte, two-cycle instruction if the mask has all bits but one set or cleared.

Byte variables, registers, and indirectly addressed RAM may be moved to a bit addressable register (usually the accumulator) in one instruction. Once transferred, the bits may be tested with a conditional jump, allowing any bit to be polled in 3 microseconds—still much faster than most architectures—or used for logical calculations. (This technique can also simulate additional bit addressing modes with byte operations.)

Parity of bytes or bits. The parity of the current accumulator contents is always available in the PSW, from whence it may be moved to the carry and further processed. Error-correcting Hamming codes and similar applications require computing parity on groups of isolated bits. This can be done by conditionally complementing the carry flag based on those bits or by gathering the bits into the accumulator (as shown in the DES example) and then testing the parallel parity flag.

Multiple byte shift and CRC codes

Though the 8051 serial port can accommodate eight- or nine-bit data transmissions, some protocols involve much longer bit streams. The algorithms presented in Design Example 2 can be extended quite readily to 16 or more bits by using multi-byte input and output buffers.

Many mass data storage peripherals and serial communications protocols include Cyclic Redundancy (CRC) codes to verify data integrity. The function is generally computed serially by hardware using shift registers and Exclusive-OR gates, but it can be done with software. As each bit is received into the carry, appropriate bits in the multi-byte data buffer are conditionally complemented based on the incoming data bit. When finished, the CRC register contents may be checked for zero by ORing the two bytes in the accumulator.

4.0 SUMMARY

A truly unique facet of the Intel MCS-51 microcomputer family design is the collection of features optimized for the one-bit operations so often desired in real-world, real-time control applications. Included are 17 special instructions, a Boolean accumulator, implicit and direct addressing modes, program and mass data storage, and many I/O options. These are the world's first single-chip microcomputers able to efficiently manipulate, operate on, and transfer either bytes or individual bits as data.

This Application Note has detailed the information needed by a microcomputer system designer to make full use of these capabilities. Five design examples were used to contrast the solutions allowed by the 8051 and those required by previous architectures. Depending on the individual application, the 8051 solution will be easier to design, more reliable to implement, debug, and verify, use less program memory, and run up to an order of magnitude faster than the same function implemented on previous digital computer architectures.

Combining byte- and bit-handling capabilities in a single microcomputer has a strong synergistic effect: the power of the result exceeds the power of byte- and bit-processors laboring individually. Virtually all user applications will benefit in some way from this duality. Data intensive applications will use bit addressing for test pin monitoring or program control flags: control applications will use byte manipulation for parallel I/O expansion or arithmetic calculations.

It is hoped that these design examples give the reader an appreciation of these unique features and suggest ways to exploit them in his or her own application.

Automobile Turn-Indicator Controller Program Listing

```
ISIS-II MCS-51 MACRO ASSEMBLER V1.0
OBJECT MODULE PLACED IN :F0:AP70.HEX
ASSEMBLER INVOKED BY:  :F1:asm51 ap70 src date(328)
LOC  OBJ             LINE     SOURCE

                        1     $XREF TITLE(AP-70 APPENDIX)
                        2     ;*******************************************************
                        3     ;
                        4     ;       THE FOLLOWING PROGRAM USES THE BOOLEAN INSTRUCTION SET
                        5     ;       OF THE INTEL 8051 MICROCOMPUTER TO PERFORM A NUMBER OF
                        6     ;       AUTOMOTIVE DASHBOARD CONTROL FUNCTIONS RELATING TO
                        7     ;       TURN SIGNAL CONTROL, EMERGENCY BLINKERS, BRAKE LIGHT
                        8     ;       CONTROL, AND PARKING LIGHT OPERATION.
                        9     ;       THE ALGORITHMS AND HARDWARE ARE DESCRIBED IN DESIGN
                       10     ;       EXAMPLE #4 OF INTEL APPLICATION NOTE AP-70,
                       11     ;               "USING THE INTEL MCS-51(TM)
                       12     ;             BOOLEAN PROCESSING CAPABILITIES"
                       13     ;
                       14     ;*******************************************************
                       15     ;
                       16     ;       INPUT PIN DECLARATIONS:
                       17     ;        (ALL INPUTS ARE POSITIVE-TRUE LOGIC.
                       18     ;       INPUTS ARE HIGH WHEN RESPECTIVE SWITCH CONTACT IS CLOSED.)
                       19     ;
0090                   20     BRAKE   BIT     P1.0    ; BRAKE PEDAL DEPRESSED
0091                   21     EMERG   BIT     P1.1    ; EMERGENCY BLINKER ACTIVATED
0092                   22     PARK    BIT     P1.2    ; PARKING LIGHTS ON
0093                   23     L_TURN  BIT     P1.3    ; TURN LEVER DOWN
0094                   24     R_TURN  BIT     P1.4    ; TURN LEVER UP
                       25     ;
                       26     ;       OUTPUT PIN DECLARATIONS:
                       27     ;       (ALL OUTPUTS ARE POSITIVE TRUE LOGIC.
                       28     ;       BULB IS TURNED ON WHEN OUTPUT PIN IS HIGH.)
                       29     ;
0095                   30     L_FRNT  BIT     P1.5    ; FRONT LEFT-TURN INDICATOR
0096                   31     R_FRNT  BIT     P1.6    ; FRONT RIGHT-TURN INDICATOR
0097                   32     L_DASH  BIT     P1.7    ; DASHBOARD LEFT-TURN INDICATOR
00A0                   33     R_DASH  BIT     P2.0    ; DASHBOARD RIGHT-TURN INDICATOR
00A1                   34     L_REAR  BIT     P2.1    ; REAR LEFT-TURN INDICATOR
00A2                   35     R_REAR  BIT     P2.2    ; REAR RIGHT-TURN INDICATOR
                       36     ;
00A3                   37     S_FAIL  BIT     P2.3    ; ELECTRICAL SYSTEM FAULT INDICATOR
                       38     ;
                       39     ;       INTERNAL VARIABLE DEFINITIONS:
                       40     ;
0020                   41     SUB_DIV DATA    20H             ; INTERRUPT RATE SUBDIVIDER
0000                   42     HI_FREQ BIT     SUB_DIV.0       ; HIGH-FREQUENCY OSCILLATOR BIT
0007                   43     LO_FREQ BIT     SUB_DIV.7       ; LOW-FREQUENCY OSCILLATOR BIT
                       44     ;       ...     .....
00D1                   45     DIM     BIT     PSW.1           ; PARKING LIGHTS ON FLAG
                       46     ;
                       47     ;=======================================================
                       48 +1  $EJECT
```

270656-26

```
LOC  OBJ              LINE     SOURCE

                        49             ORG     0000H           ; RESET VECTOR
0000 020040             50             LJMP    INIT
                        51     ;
0008                    52             ORG     000BH           ; TIMER 0 SERVICE VECTOR
0008 758CF0             53             MOV     TH0,#-16        ; HIGH TIMER BYTE ADJUSTED TO CONTROL INT. RATE
000E C0D0               54             PUSH    PSW             ; EXECUTE CODE TO SAVE ANY REGISTERS USED BELOW
0010 0154               55             AJMP    UPDATE          ;      (CONTINUE WITH REST OF ROUTINE)
                        56     ;
0040                    57             ORG     0040H
0040 758A00             58     INIT:   MOV     TL0,#0          ; ZERO LOADED INTO LOW-ORDER BYTE AND
0043 758CF0             59             MOV     TH0,#-16        ; -16 IN HIGH-ORDER BYTE GIVES 4 MSEC PERIOD
0046 758961             60             MOV     TMOD,#01100001B ; 8-BIT AUTO RELOAD COUNTER MODE FOR TIMER 1,
                        61                                     ;  16-BIT TIMER MODE FOR TIMER 0 SELECTED
0049 7520F4             62             MOV     SUB_DIV,#244    ; SUBDIVIDE INTERRUPT RATE BY 244 FOR 1 HZ
004C D2A9               63             SETB    ET0             ; USE TIMER 0 OVERFLOWS TO INTERRUPT PROGRAM
004E D2AF               64             SETB    EA              ; CONFIGURE IE TO GLOBALLY ENABLE INTERRUPTS
0050 D28C               65             SETB    TR0             ; KEEP INSTRUCTION CYCLE COUNT UNTIL OVERFLOW
0052 80FE               66             SJMP    $               ; START BACKGROUND PROGRAM EXECUTION
                        67     ,
                        68     ;
0054 D52038             69     UPDATE: DJNZ    SUB_DIV,TOSERV  ; EXECUTE SYSTEM TEST ONLY ONCE PER SECOND
0057 7520F4             70             MOV     SUB_DIV,#244    ; GET VALUE FOR NEXT ONE SECOND DELAY AND
                        71     ;                               ; GO THROUGH ELECTRICAL SYSTEM TEST CODE:
005A 4390E0             72             ORL     P1,#11100000B   ; SET CONTROL OUTPUTS HIGH
005D 43A007             73             ORL     P2,#00000111B
0060 C295               74             CLR     L_FRNT          ; FLOAT DRIVE COLLECTOR
0062 20B428             75             JB      T0,FAULT        ; T0 SHOULD BE PULLED LOW
0065 D295               76             SETB    L_FRNT          ; PULL COLLECTOR BACK DOWN
0067 C297               77             CLR     L_DASH          ; REPEAT SEQUENCE FOR L_DASH,
0069 20B421             78             JB      T0,FAULT
006C D297               79             SETB    L_DASH
006E C2A1               80             CLR     L_REAR          ;                 L_REAR,
0070 20B41A             81             JB      T0,FAULT
0073 D2A1               82             SETB    L_REAR
0075 C296               83             CLR     R_FRNT          ;                 R_FRNT,
0077 20B413             84             JB      T0,FAULT
007A D296               85             SETB    R_FRNT
007C C2A0               86             CLR     R_DASH          ;                 R_DASH,
007E 20B40C             87             JB      T0,FAULT
0081 D2A0               88             SETB    R_DASH
0083 C2A2               89             CLR     R_REAR          ;                 AND R_REAR
0085 20B405             90             JB      T0,FAULT
0088 D2A2               91             SETB    R_REAR
                        92     ;
                        93     ;       WITH ALL COLLECTORS GROUNDED, T0 SHOULD BE HIGH
                        94     ;       IF SO, CONTINUE WITH INTERRUPT ROUTINE.
                        95     ;
008A 20B402             96             JB      T0,TOSERV
008D B2A3               97     FAULT:  CPL     S_FAIL          ; ELECTRICAL FAILURE PROCESSING ROUTINE
                        98                                     ; (TOGGLE INDICATOR ONCE PER SECOND)
                        99 +1  $EJECT
```

270656-27

```
LOC  OBJ            LINE     SOURCE

                     100      ;       CONTINUE WITH INTERRUPT PROCESSING:
                     101      ;
                     102      ;   1)  COMPUTE LOW BULB INTENSITY WHEN PARKING LIGHTS ARE ON.
                     103      ;
008F A201            104      TOSERV: MOV     C,SUB_DIV.1     ; START WITH 50 PERCENT,
0091 8200            105              ANL     C,SUB_DIV.0     ; MASK DOWN TO 25 PERCENT,
0093 7202            106              ORL     C,SUB_DIV.2     ; BUILD BACK TO 62.5 PERCENT,
0095 8292            107              ANL     C,PARK          ; GATE WITH PARKING LIGHT SWITCH,
0097 92D1            108              MOV     DIM,C           ; AND SAVE IN TEMP. VARIABLE.
                     109      ;
                     110      ;   2)  COMPUTE AND OUTPUT LEFT-HAND DASHBOARD INDICATOR.
                     111      ;
0099 A293            112              MOV     C,L_TURN        ; SET CARRY IF TURN
009B 7291            113              ORL     C,EMERG         ; OR EMERGENCY SELECTED.
009D 8207            114              ANL     C,LO_FREQ       ; IF SO, GATE IN 1 HZ SIGNAL
009F 9297            115              MOV     L_DASH,C        ; AND OUTPUT TO DASHBOARD.
                     116      ;
                     117      ;   3)  COMPUTE AND OUTPUT LEFT-HAND FRONT TURN SIGNAL.
                     118      ;
00A1 92D5            119              MOV     F0,C            ; SAVE FUNCTION SO FAR.
00A3 72D1            120              ORL     C,DIM           ; ADD IN PARKING LIGHT FUNCTION
00A5 9295            121              MOV     L_FRNT,C        ; AND OUTPUT TO TURN SIGNAL.
                     122      ;
                     123      ;   4)  COMPUTE AND OUTPUT LEFT-HAND REAR TURN SIGNAL.
                     124      ;
00A7 A290            125              MOV     C,BRAKE         ; GATE BRAKE PEDAL SWITCH
00A9 B093            126              ANL     C,/L_TURN       ; WITH TURN LEVER.
00AB 72D5            127              ORL     C,F0            ; INCLUDE TEMP. VARIABLE FROM DASH
00AD 72D1            128              ORL     C,DIM           ; AND PARKING LIGHT FUNCTION
00AF 92A1            129              MOV     L_REAR,C        ; AND OUTPUT TO TURN SIGNAL.
                     130      ;
                     131      ;   5)  REPEAT ALL OF ABOVE FOR RIGHT-HAND COUNTERPARTS.
                     132      ;
00B1 A294            133              MOV     C,R_TURN        ; SET CARRY IF TURN
00B3 7291            134              ORL     C,EMERG         ; OR EMERGENCY SELECTED.
00B5 8207            135              ANL     C,LO_FREQ       ; IF SO, GATE IN 1 HZ SIGNAL
00B7 92A0            136              MOV     R_DASH,C        ; AND OUTPUT TO DASHBOARD.
00B9 92D5            137              MOV     F0,C            ; SAVE FUNCTION SO FAR.
00BB 72D1            138              ORL     C,DIM           ; ADD IN PARKING LIGHT FUNCTION
00BD 9296            139              MOV     R_FRNT,C        ; AND OUTPUT TO TURN SIGNAL.
00BF A290            140              MOV     C,BRAKE         ; GATE BRAKE PEDAL SWITCH
00C1 B094            141              ANL     C,/R_TURN       ; WITH TURN LEVER.
00C3 72D5            142              ORL     C,F0            ; INCLUDE TEMP. VARIABLE FROM DASH
00C5 72D1            143              ORL     C,DIM           ; AND PARKING LIGHT FUNCTION
00C7 92A2            144              MOV     R_REAR,C        ; AND OUTPUT TO TURN SIGNAL.
                     145      ;
                     146      ;       RESTORE STATUS REGISTER AND RETURN.
                     147      ;
00C9 D0D0            148              POP     PSW             ; RESTORE PSW
00CB 32              149              RETI                    ; AND RETURN FROM INTERRUPT ROUTINE
                     150      ;
                     151              END
```

270656-28

```
XREF SYMBOL TABLE LISTING
---- ------ ----- -------


NAME      TYPE     VALUE AND REFERENCES

BRAKE     N BSEG   0090H  20# 125 140
DIM       N BSEG   00D1H  45# 108 120 128 138 143
EA        N BSEG   00AFH  64
EMERG     N BSEG   0091H  21# 113 134
ETO       N BSEG   00A9H  63
FO        N BSEG   00D5H  119 127 137 142
FAULT     L CSEG   008DH  75 78 81 84 87 90 97#
HI_FREQ   N BSEG   0000H  42#
INIT      L CSEG   0040H  50 58#
L_DASH    N BSEG   0097H  32# 77 79 115
L_FRNT    N BSEG   0095H  30# 74 76 121
L_REAR    N BSEG   00A1H  34# 80 82 129
L_TURN    N BSEG   0093H  23# 112 126
LO_FREQ   N BSEG   0007H  43# 114 135
P1        N DSEG   0090H  20 21 22 23 24 30 31 32 72
P2        N DSEG   00A0H  33 34 35 37 73
PARK      N BSEG   0092H  22# 107
PSW       N DSEG   00D0H  45 54 148
R_DASH    N BSEG   00A0H  33# 86 88 136
R_FRNT    N BSEG   0096H  31# 83 85 139
R_REAR    N BSEG   00A2H  35# 89 91 144
R_TURN    N BSEG   0094H  24# 133 141
S_FAIL    N BSEG   00A3H  37# 97
SUB_DIV   N DSEG   0020H  41# 42 43 62 69 70 104 105 106
TO        N BSEG   00B4H  75 78 81 84 87 90 96
TOSERV    L CSEG   008FH  69 96 104#
THO       N DSEG   008CH  53 59
TLO       N DSEG   008AH  58
TMOD      N DSEG   0089H  60
TRO       N BSEG   008CH  65
UPDATE    L CSEG   0054H  55 69#


ASSEMBLY COMPLETE, NO ERRORS FOUND
```

270656-29

8051, 8052 and 80C51

8

HARDWARE DESCRIPTION OF THE 8051, 8052 AND 80C51

INTRODUCTION

This chapter presents a comprehensive description of the on-chip hardware features of the MCS®-51 microcontrollers. Included in this description are

- The port drivers and how they function both as ports and, for Ports 0 and 2, in bus operations
- The Timer/Counters
- The Serial Interface
- The Interrupt System
- Reset
- The Reduced Power Modes in the CHMOS devices
- The EPROM versions of the 8051AH, 8052AH, and 80C51BH

The devices under consideration are listed in Table 1. As it becomes unwieldy to be constantly referring to each of these devices by their individual names, we will adopt a convention of referring to them generically as 8051s and 8052s, unless a specific member of the group is being referred to, in which case it will be specifically named. The "8051s" include the 8051, 8051AH, and 80C51BH, and their ROMless and EPROM versions. The "8052s" are the 8052AH, 8032AH, and 8752BH.

Figure 1 shows a functional block diagram of the 8051s and 8052s.

Table 1. The MCS-51 Family of Microcontrollers

Device Name	ROMless Version	EPROM Version	ROM Bytes	RAM Bytes	16-bit Timers	Ckt Type
8051	8031	(8751)	4K	128	2	HMOS
8051AH	8031AH	8751H	4K	128	2	HMOS
8052AH	8032AH	8752BH	8K	256	3	HMOS
80C51BH	80C31BH	87C51	4K	128	2	CHMOS

Special Function Registers

A map of the on-chip memory area called SFR (Special Function Register) space is shown in Figure 2. SFRs marked by parentheses are resident in the 8052s but not in the 8051s.

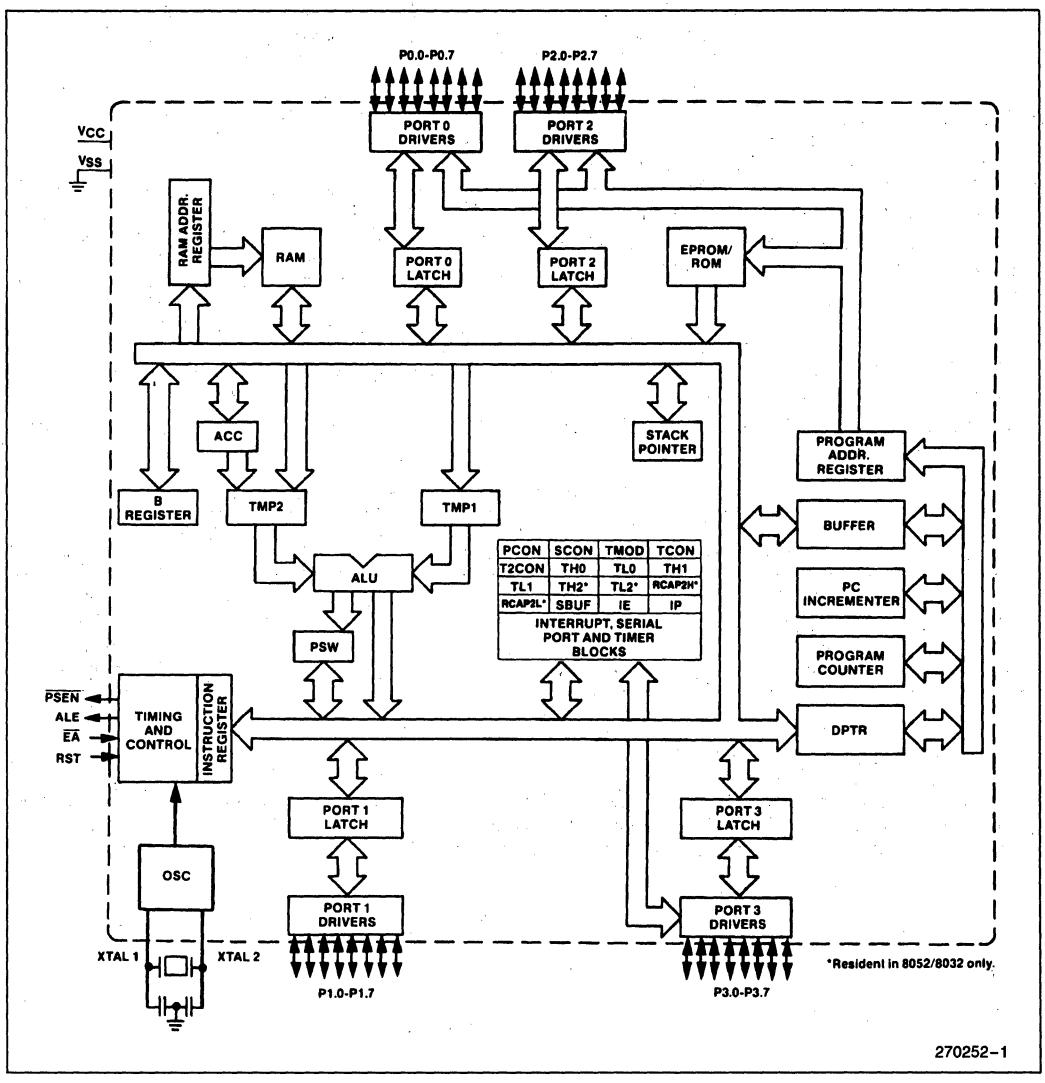


Figure 1. MCS-51 Architectural Block Diagram

	8 Bytes								
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	(T2CON)		(RCAP2L)	(RCAP2H)	(TL2)	(TH2)			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

Figure 2. SFR Map. (. . .) Indicates Resident in 8052s, not in 8051s

Note that not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have no effect.

User software should not write 1s to these unimplemented locations, since they may be used in future MCS-51 products to invoke new features. In that case the reset or inactive values of the new bits will always be 0, and their active values will be 1.

The functions of the SFRs are outlined below.

ACCUMULATOR

ACC is the Accumulator register. The mnemonics for Accumulator-Specific instructions, however, refer to the Accumulator simply as A.

B REGISTER

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

PROGRAM STATUS WORD

The PSW register contains program status information as detailed in Figure 3.

STACK POINTER

The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

DATA POINTER

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

PORTS 0 TO 3

P0, P1, P2 and P3 are the SFR latches of Ports 0, 1, 2 and 3, respectively.

SERIAL DATA BUFFER

The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

TIMER REGISTERS

Register pairs (TH0, TL0), (TH1, TL1), and (TH2, TL2) are the 16-bit Counting registers for Timer/Counters 0, 1, and 2, respectively.

CAPTURE REGISTERS

The register pair (RCAP2H, RCAP2L) are the Capture registers for the Timer 2 "Capture Mode." In this mode, in response to a transition at the 8052's T2EX pin, TH2 and TL2 are copied into RCAP2H and RCAP2L. Timer 2 also has a 16-bit auto-reload mode, and RCAP2H and RCAP2L hold the reload value for this mode. More about Timer 2's features in a later section.

CONTROL REGISTERS

Special Function Registers IP, IE, TMOD, TCON, T2CON, SCON, and PCON contain control and status bits for the interrupt system, the Timer/Counters, and the serial port. They are described in later sections.

(MSB)							(LSB)
CY	AC	F0	RS1	RS0	OV	—	P

Symbol	Position	Name and Significance
CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary Carry flag. (For BCD operations.)
F0	PSW.5	Flag 0 (Available to the user for general purposes.)
RS1	PSW.4	Register bank select control bits 1 &
RS0	PSW.3	0. Set/cleared by software to determine working register bank (see Note).
OV	PSW.2	Overflow flag.
—	PSW.1	User definable flag.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/ even number of "one" bits in the Accumulator, i.e., even parity.

NOTE:
The contents of (RS1, RS0) enable the working register banks as follows:

(0.0)—Bank 0 (00H–07H)
(0.1)—Bank 1 (08H–0FH)
(1.0)—Bank 2 (10H–17H)
(1.1)—Bank 3 (18H–1FH)

Figure 3. PSW: Program Status Word Register

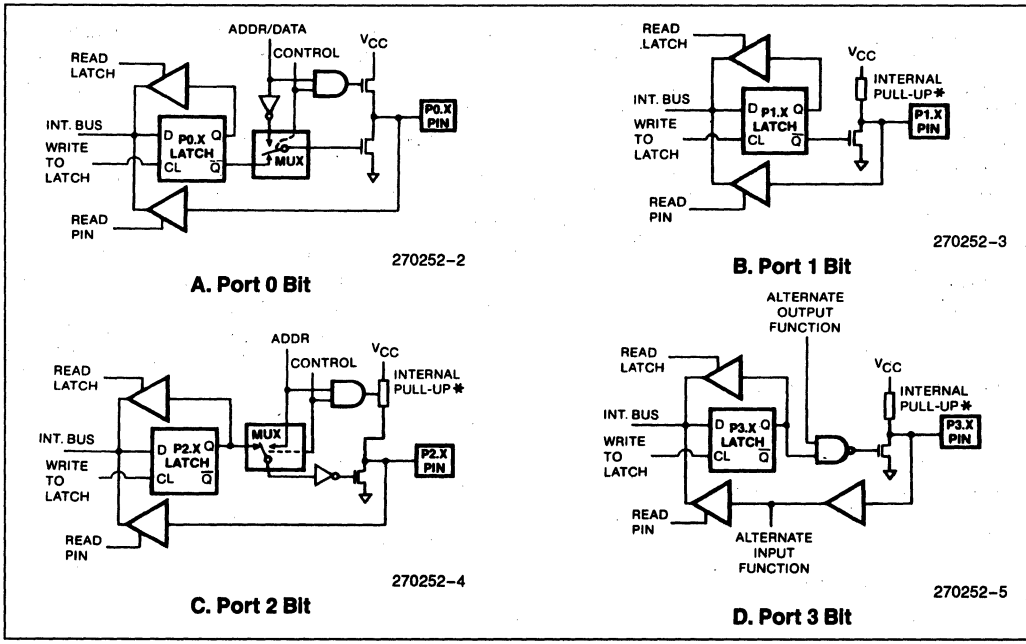


Figure 4. 8051 Port Bit Latches and I/O Buffers
*See Figure 5 for details of the internal pullup.

PORT STRUCTURES AND OPERATION

All four ports in the 8051 are bidirectional. Each consists of a latch (Special Function Registers P0 through P3), an output driver, and an input buffer.

The output drivers of Ports 0 and 2, and the input buffers of Port 0, are used in accesses to external memory. In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content.

All the Port 3 pins, and (in the 8052) two Port 1 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed on the following page.

Port Pin	Alternate Function
*P1.0	T2 (Timer/Counter 2 external input)
*P1.1	T2EX (Timer/Counter 2 Capture/Reload trigger)
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt)
P3.3	$\overline{INT1}$ (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	$\overline{WR}$ (external Data Memory write strobe)
P3.7	$\overline{RD}$ (external Data Memory read strobe)

*P1.0 and P1.1 serve these alternate functions only on the 8052.

The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise the port pin is stuck at 0.

I/O Configurations

Figure 4 shows a functional diagram of a typical bit latch and I/O buffer in each of the four ports. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal. More about that later.

As shown in Figure 4, the output drivers of Ports 0 and 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the P0 SFR gets 1s written to it.

Also shown in Figure 4, is that if a P3 bit latch contains a 1, then the output level is controlled by the signal labeled "alternate output function." The actual P3.X pin level is always available to the pin's alternate input function, if any.

Ports 1, 2, and 3 have internal pullups. Port 0 has open drain outputs. Each I/O line can be independently used as an input or an output. (Ports 0 and 2 may not be used as general purpose I/O when being used as the ADDR/DATA BUS). To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for Ports 1, 2, and 3, the pin is pulled high by the internal pullup, but can be pulled low by an external source.

Port 0 differs in not having internal pullups. The pullup FET in the P0 output driver (see Figure 4) is used only when the Port is emitting 1s during external memory accesses. Otherwise the pullup FET is off. Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, so the pin floats. In that condition it can be used a high-impedance input.

Because Ports 1, 2, and 3 have fixed internal pullups they are sometimes called "quasi-bidirectional" ports. When configured as inputs they pull high and will source current (IIL, in the data sheets) when externally pulled low. Port 0, on the other hand, is considered "true" bidirectional, because when configured as an input it floats.

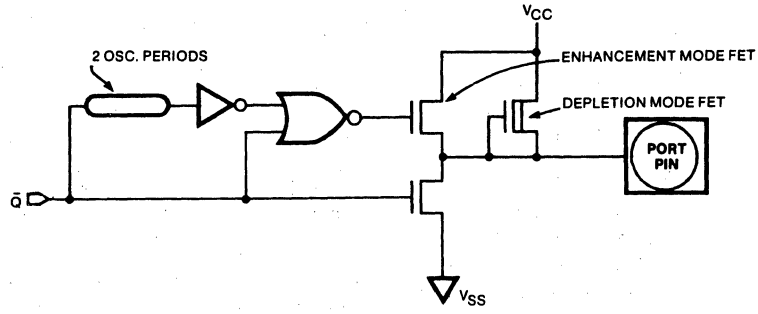
All the port latches in the 8051 have 1s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writing a 1 to it.

Writing to a Port

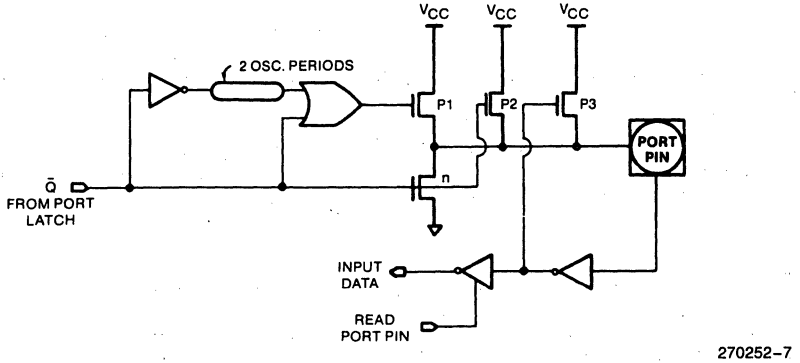
In the execution of an instruction that changes the value in a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during Phase 1 of any clock period. (During Phase 2 the output buffer holds the value it saw during the previous Phase 1). Consequently, the new value in the port latch won't actually appear at the output pin until the next Phase 1, which will be at S1P1 of the next machine cycle. See Figure 39 in the Internal Timing section.

If the change requires a 0-to-1 transition in Port 1, 2, or 3, an additional pullup is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pullup can source about 100 times the current that the normal pullup can. It should be noted that the internal pullups are field-effect transistors, not linear resistors. The pullup arrangements are shown in Figure 5.

In HMOS versions of the 8051, the fixed part of the pullup is a depletion-mode transistor with the gate wired to the source. This transistor will allow the pin to source about 0.25 mA when shorted to ground. In parallel with the fixed pullup is an enhancement-mode transistor, which is activated during S1 whenever the port bit does a 0-to-1 transition. During this interval, if the port pin is shorted to ground, this extra transistor will allow the pin to source an additional 30 mA.



A. HMOS Configuration. The enhancement mode transistor is turned on for 2 osc. periods after $\overline{Q}$ makes a 0-to-1 transition.



B. CHMOS Configuration. pFET 1 is turned on for 2 osc. periods after $\overline{Q}$ makes a 0-to-1 transition. During this time, pFET 1 also turns on pFET 3 through the inverter to form a latch which holds the 1. pFET 2 is also on.

Figure 5. Ports 1 And 3 HMOS And CHMOS Internal Pullup Configurations. Port 2 is Similar Except That It Holds The Strong Pullup On While Emitting 1s That Are Address Bits. (See Text, "Accessing External Memory".)

In the CHMOS versions, the pullup consists of three pFETs. It should be noted that an n-channel FET (nFET) is turned on when a logical 1 is applied to its gate, and is turned off when a logical 0 is applied to its gate. A p-channel FET (pFET) is the opposite: it is on when its gate sees a 0, and off when its gate sees a 1.

pFET1 in Figure 5 is the transistor that is turned on for 2 oscillator periods after a 0-to-1 transition in the port latch. While it's on, it turns on pFET3 (a weak pullup), through the inverter. This inverter and pFET form a latch which hold the 1.

Note that if the pin is emitting a 1, a negative glitch on the pin from some external source can turn off pFET3, causing the pin to go into a float state. pFET2 is a very weak pullup which is on whenever the nFET is off, in traditional CMOS style. It's only about $1/10$ the strength of pFET3. Its function is to restore a 1 to the pin in the event the pin *had* a 1 and lost it to a glitch.

Port Loading and Interfacing

The output buffers of Ports 1, 2, and 3 can each drive 4 LS TTL inputs. These ports on HMOS versions can be driven in a normal manner by any TTL or NMOS circuit. Both HMOS and CHMOS pins can be driven by open-collector and open-drain outputs, but note that 0-to-1 transitions will not be fast. In the HMOS device, if the pin is driven by an open-collector output, a 0-to-1 transition will have to be driven by the relatively weak depletion mode FET in Figure 5(A). In the CHMOS device, an input 0 turns off pullup pFET3, leaving only the very weak pullup pFET2 to drive the transition.

In external bus mode, Port 0 output buffers can each drive 8 LS TTL inputs. As port pins, they require external pullups to drive any inputs.

Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Which ones do which? The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions. The instructions listed below are read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin:

ANL	(logical AND, e.g., ANL P1, A)
ORL	(logical OR, e.g., ORL P2, A)
XRL	(logical EX-OR, e.g., XRL P3, A)
JBC	(jump if bit = 1 and clear bit, e.g., JBC P1.1, LABEL)
CPL	(complement bit, e.g., CPL P3.0)
INC	(increment, e.g., INC P2)
DEC	(decrement, e.g., DEC P2)
DJNZ	(decrement and jump if not zero, e.g., DJNZ P3, LABEL)
MOV, PX.Y, C	(move carry bit to bit Y of Port X)
CLR PX.Y	(clear bit Y of Port X)
SETB PX.Y	(set bit Y of Port X)

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as a 0. Reading the latch rather than the pin will return the correct value of 1.

ACCESSING EXTERNAL MEMORY

Accesses to external memory are of two types: accesses to external Program Memory and accesses to external Data Memory. Accesses to external Program Memory use signal $\overline{PSEN}$ (program store enable) as the read strobe. Accesses to external Data Memory use $\overline{RD}$ or $\overline{WR}$ (alternate functions of P3.7 and P3.6) to strobe the memory. Refer to Figures 36 through 38 in the Internal Timing section.

Fetches from external Program Memory always use a 16-bit address. Accesses to external Data Memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri).

Whenever a 16-bit address is used, the high byte of the address comes out on Port 2, where it is held for the duration of the read or write cycle. Note that the Port 2 drivers use the strong pullups during the entire time that they are emitting address bits that are 1s. This is during the execution of a MOVX @DPTR instruction. During this time the Port 2 latch (the Special Function Register) does not have to contain 1s, and the contents of the Port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port 2 SFR will reappear in the next cycle.

If an 8-bit address is being used (MOVX @Ri), the contents of the Port 2 SFR remain at the Port 2 pins throughout the external memory cycle. This will facilitate paging.

In any case, the low byte of the address is time-multiplexed with the data byte on Port 0. The ADDR/DATA signal drives both FETs in the Port 0 output buffers. Thus, in this application the Port 0 pins are not open-drain outputs, and do not require external pullups. Signal ALE (Address Latch Enable) should be used to capture the address byte into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on Port 0 just before $\overline{WR}$ is activated, and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe is deactivated.

During any access to external memory, the CPU writes 0FFH to the Port 0 latch (the Special Function Register), thus obliterating whatever information the Port 0 SFR may have been holding. If the user writes to Port 0 during an external memory fetch, the incoming code byte is corrupted. Therefore, do not write to Port 0 if external program memory is used.

External Program Memory is accessed under two conditions:

1) Whenever signal $\overline{EA}$ is active; or
2) Whenever the program counter (PC) contains a number that is larger than 0FFFH (1FFFH for the 8052).

This requires that the ROMless versions have $\overline{EA}$ wired low to enable the lower 4K (8K for the 8032) program bytes to be fetched from external memory.

When the CPU is executing out of external Program Memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches they output the high byte of the PC. During this time the Port 2 drivers use the strong pullups to emit PC bits that are 1s.

TIMER/COUNTERS

The 8051 has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. The 8052 has these two plus one

more: Timer 2. All three can be configured to operate either as timers or event counters.

In the "Timer" function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is $1/12$ of the oscillator frequency.

In the "Counter" function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0, T1 or (in the 8052) T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is $1/24$ of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

In addition to the "Timer" or "Counter" selection, Timer 0 and Timer 1 have four operating modes from which to select. Timer 2, in the 8052, has three modes of operation: "Capture," "Auto-Reload" and "baud rate generator."

Timer 0 and Timer 1

These Timer/Counters are present in both the 8051 and the 8052. The "Timer" or "Counter" function is selected by control bits C/$\overline{T}$ in the Special Function Register TMOD (Figure 6). These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timer/Counters. Mode 3 is different. The four operating modes are described in the following text.

MODE 0

Either Timer in Mode 0 is an 8-bit Counter with a divide-by-32 prescaler. This 13-bit timer is MCS-48 compatible. Figure 7 shows the Mode 0 operation as it applies to Timer 1.

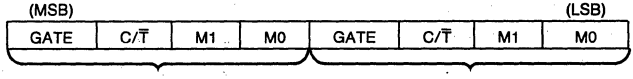
In this mode, the Timer register is configured as a 13-Bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or $\overline{INT1}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements.) TR1 is a control bit in the Special Function Register TCON (Figure 8). GATE is in TMOD.

The 13-Bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1. Substitute TR0, TF0 and $\overline{INT0}$ for the corresponding Timer 1 signals in Figure 7. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

MODE 1

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.



GATE — Gating control when set. Timer/Counter "x" is enabled only while "$\overline{INTx}$" pin is high and "TRx" control pin is set. When cleared Timer "x" is enabled whenever "TRx" control bit is set.

C/$\overline{T}$ — Timer or Counter Selector cleared for Timer operation (input from internal system clock). Set for Counter operation (input from "Tx" input pin).

M1	M0	Operating Mode
0	0	8-bit Timer/Counter "THx" with "TLx" as 5-bit prescaler.
0	1	16-bit Timer/Counter "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-bit auto-reload Timer/Counter "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

Figure 6. TMOD: Timer/Counter Mode Control Register

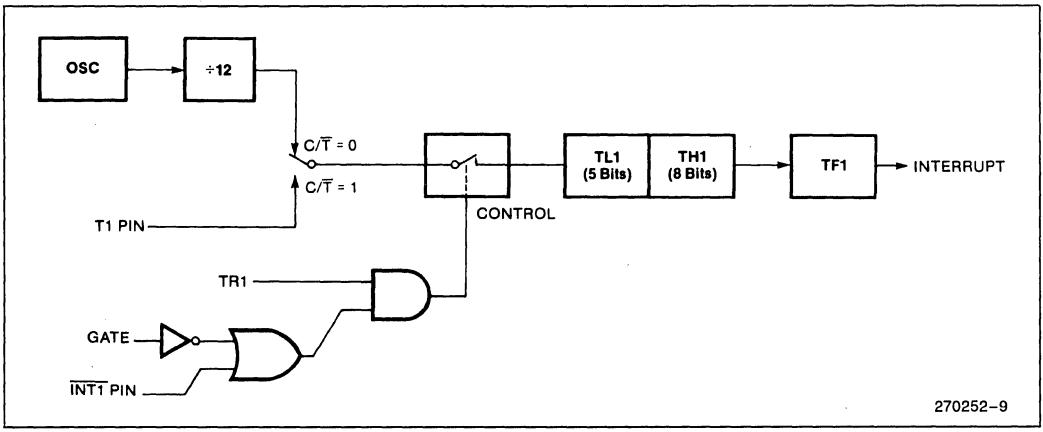


Figure 7. Timer/Counter 1 Mode 0: 13-Bit Counter

(MSB)							(LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 overflow Flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TF0	TCON.5	Timer 0 overflow Flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Figure 8.TCON: Timer/Counter Control Register

MODE 2

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 9. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

Mode 2 operation is the same for Timer/Counter 0.

MODE 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 10. TL0 uses the Timer 0 control bits: C/$\overline{T}$, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the "Timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, an 8051 can look like it has three Timer/Counters, and an 8052, like it has four. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

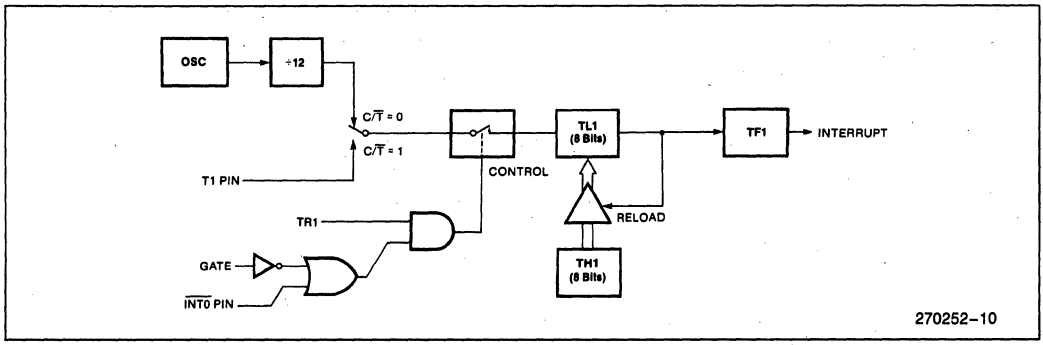


Figure 9. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

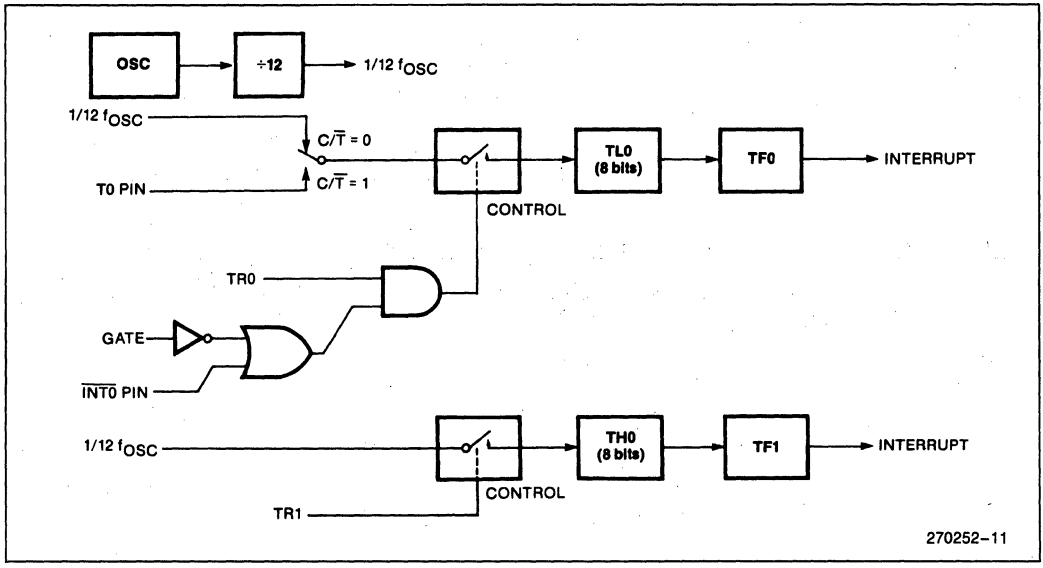


Figure 10. Timer/Counter 0 Mode 3: Two 8-Bit Counters

Timer 2

Timer 2 is a 16-bit Timer/Counter which is present only in the 8052. Like Timers 0 and 1, it can operate either as a timer or as an event counter. This is selected by bit C/$\overline{T2}$ in the Special Function Register T2CON (Figure 11). It has three operating modes: "capture," "auto-load" and "baud rate generator," which are selected by bits in T2CON as shown in Table 2.

Table 2. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	Mode
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(off)

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/$\overline{T2}$	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered).
CP/$\overline{RL2}$	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Figure 11. T2CON: Timer/Counter 2 Control Register

In the Capture Mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which upon overflowing sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. (RCAP2L and RCAP2H are new Special Function Registers in the 8052.) In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt.

The Capture Mode is illustrated in Figure 12.

In the auto-reload mode there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2.

The auto-reload mode is illustrated in Figure 13.

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1. It will be described in conjunction with the serial port.

SERIAL INTERFACE

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. (However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

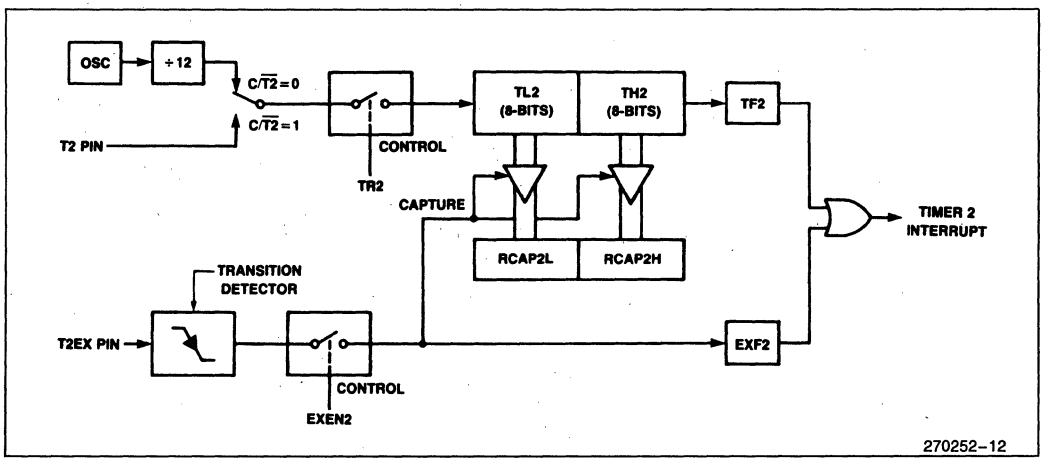


Figure 12. Timer 2 in Capture Mode

The serial port can operate in 4 modes:

Mode 0: Serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Mode 1: 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Functon Register SCON, while the stop bit is ignored. The baud rate is programmable to either $1/32$ or $1/64$ the oscillator frequency.

Mode 3: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

Serial Port Control Register

The serial port control and status register is the Special Function Register SCON, shown in Figure 14. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

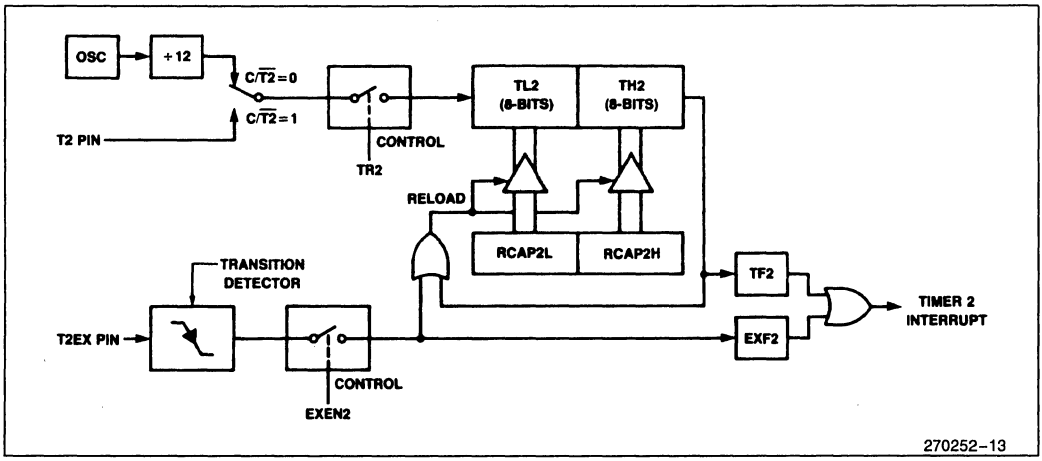


Figure 13. Timer 2 in Auto-Reload Mode

(MSB)							(LSB)
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Where SM0, SM1 specify the serial port mode, as follows:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	shift register	$f_{osc.}/12$
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	$f_{osc.}/64$ or $f_{osc.}/32$
1	1	3	9-bit UART	variable

- SM2 enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be 0.
- REN enables serial reception. Set by software to enable reception. Clear by software to disable reception.
- TB8 is the 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
- RB8 in Modes 2 and 3, is the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
- TI is transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
- RI is receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

Figure 14. SCON: Serial Port Control Register

Baud Rates

The baud rate in Mode 0 is fixed:

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{12}$$

The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is the value on reset), the baud rate $^1/_{64}$ the oscillator frequency. If SMOD = 1, the baud rate is $^1/_{32}$ the oscillator frequency.

$$\text{Mode 2 Baud Rate} = \frac{2^{SMOD}}{64} \times (\text{Oscillator Frequency})$$

In the 8051, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate. In the 8052, these baud rates can be determined by Timer 1, or by Timer 2, or by both (one for transmit and the other for receive).

Using Timer 1 to Generate Baud Rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{SMOD}}{32} \times (\text{Timer 1 Overflow Rate})$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case, the baud rate is given by the formula

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator Frequency}}{12x\,[256 - (TH1)]}$$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload.

Figure 15 lists various commonly used baud rates and how they can be obtained from Timer 1.

Baud Rate	fosc	SMOD	Timer 1		
			C/T	Mode	Reload Value
Mode 0 Max: 1 MHZ	12 MHZ	X	X	X	X
Mode 2 Max: 375K	12 MHZ	1	X	X	X
Modes 1, 3: 62.5K	12 MHZ	1	0	2	FFH
19.2K	11.059 MHZ	1	0	2	FDH
9.6K	11.059 MHZ	0	0	2	FDH
4.8K	11.059 MHZ	0	0	2	FAH
2.4K	11.059 MHZ	0	0	2	F4H
1.2K	11.059 MHZ	0	0	2	E8H
137.5	11.986 MHZ	0	0	2	1DH
110	6 MHZ	0	0	2	72H
110	12 MHZ	0	0	1	FEEBH

Figure 15. Timer 1 Generated Commonly Used Baud Rates

Using Timer 2 to Generate Baud Rates

In the 8052, Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Figure 11). Note then the baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 16.

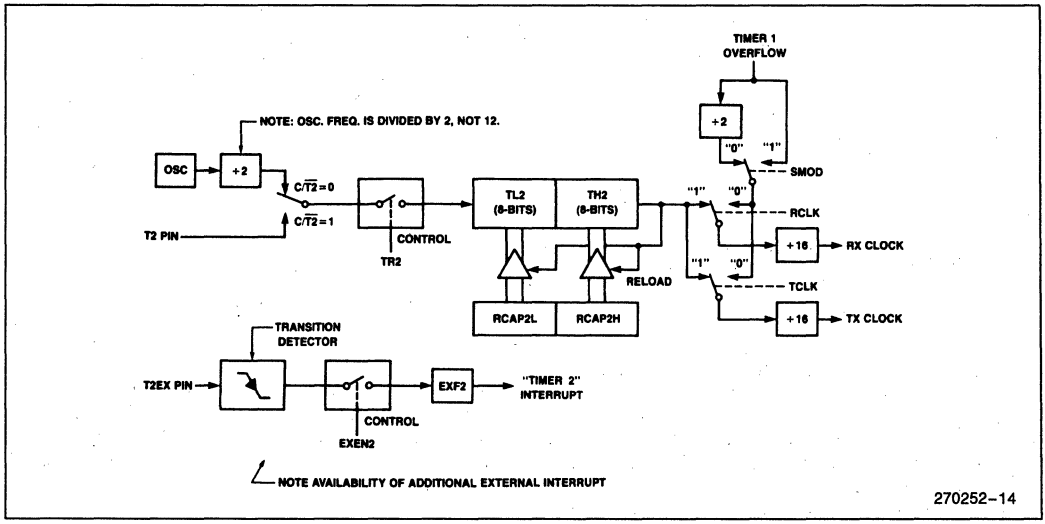


Figure 16. Timer 2 in Baud Rate Generator Mode

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

Now, the baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate as follows:

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either "timer" or "counter" operation. In the most typical applications, it is configured for "timer" operation (C/T2 = 0). "Timer" operation is a little different for Timer 2 when it's being used as a baud rate generator. Normally, as a timer it would increment every machine cycle (thus at $1/12$ the oscillator frequency). As a baud rate generator, however, it increments every state time (thus at $1/2$ the oscillator frequency). In that case the baud rate is given by the formula

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32\text{x}\,[65536 - (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 16. This Figure is valid only if RCLK + TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the Timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but shouldn't be written to, because a write might overlap a reload and cause write and/or reload errors. Turn the Timer off (clear TR2) before accessing the Timer 2 or RCAP registers, in this case.

More About Mode 0

Serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at $1/12$ the oscillator frequency.

Figure 17 shows a simplified functional diagram of the serial port in Mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal at S6P2 also loads a 1 into the 9th position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "write to SBUF," and activation of SEND.

SEND enables the output of the shift register to the alternate output function line of P3.0, and also enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1 and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift register are shifted to the right one position.

As data bits shift out to the right, zeroes come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeroes. This condition flags the TX Control block to do one last shift and then deactivate SEND and set TI. Both of these actions occur at S1P1 of the 10th machine cycle after "write to SBUF."

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

More About Mode 1

Ten bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the 8051 the baud rate is determined by the Timer 1 overflow rate. In the 8052 it is determined either by the Timer 1 overflow rate, or the Timer 2 overflow rate, or both (one for transmit and the other for receive).

Figure 18 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit receive.

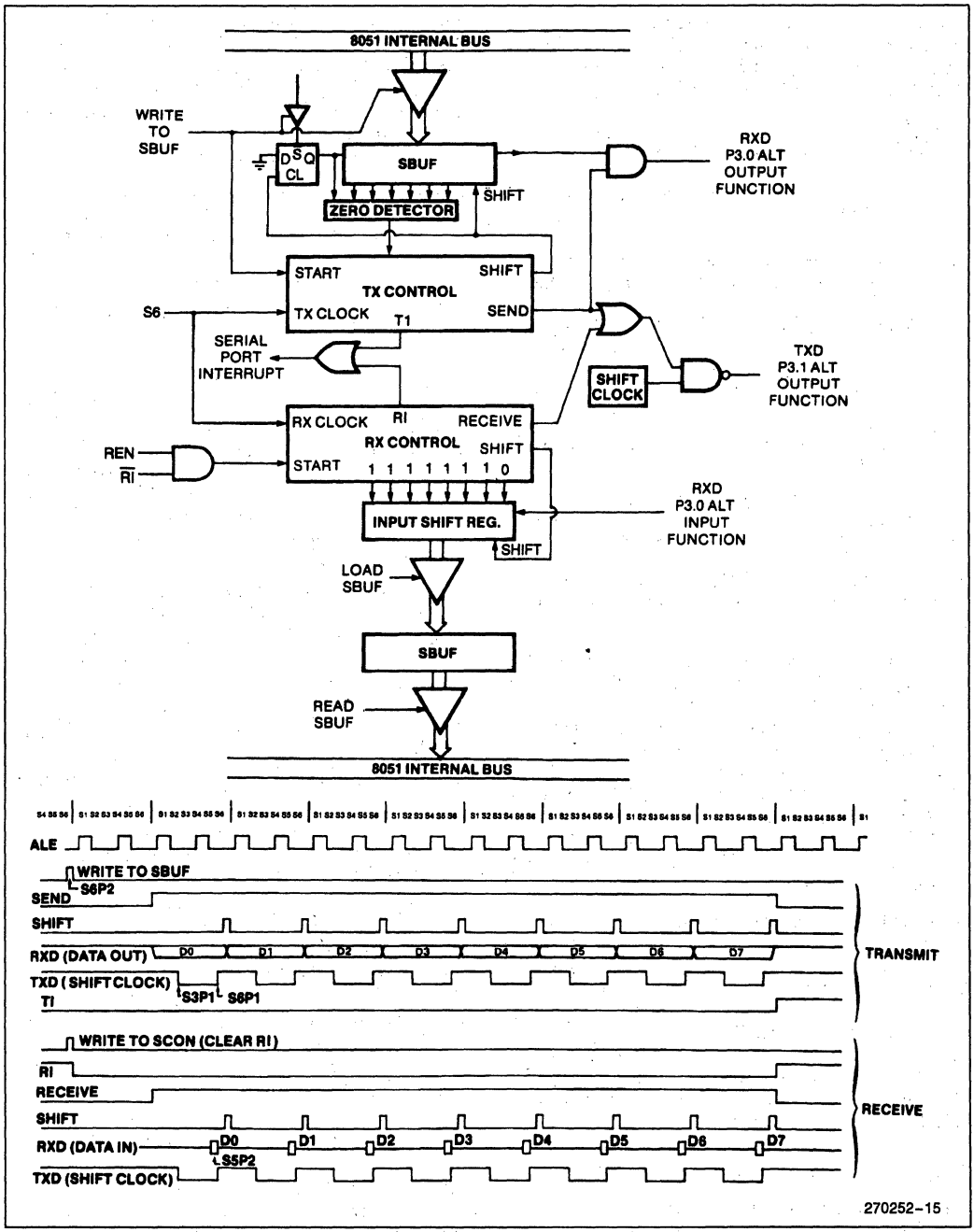


Figure 17. Serial Port Mode 0

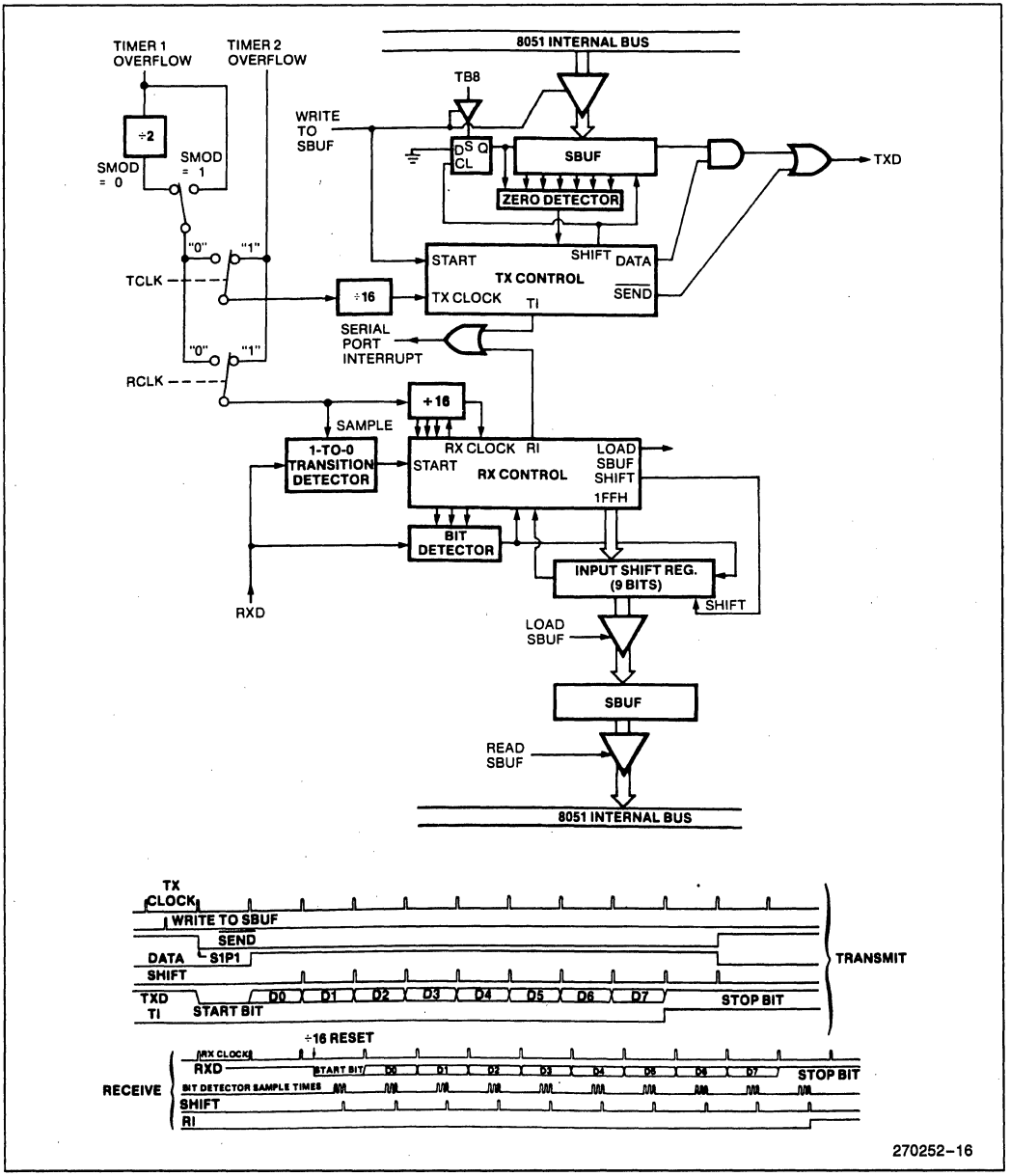


Figure 18. Serial Port Mode 1. TCLK, RCLK and Timer 2 are Present in the 8052/8032 Only.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal).

The transmission begins with activation of $\overline{\text{SEND}}$, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeroes are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeroes. This condition flags the TX Control unit to do one last shift and then deactivate $\overline{\text{SEND}}$ and set TI. This occurs at the 10th divide-by-16 rollover after "write to SBUF."

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register, (which in mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1) RI = 0, and
2) Either SM2 = 0, or the received stop bit = 1

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RXD.

More About Modes 2 and 3

Eleven bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either $1/_{32}$ or $1/_{64}$ the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from either Timer 1 or 2 depending on the state of TCLK and RCLK.

Figures 19 and 20 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.)

The transmission begins with activation of SEND, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeroes are clocked in. Thus, as data bits shift out to the right, zeroes are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeroes. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "write to SBUF."

Reception is initiated by a detected 1-to-0 transition at RXD. For this purpose RXD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

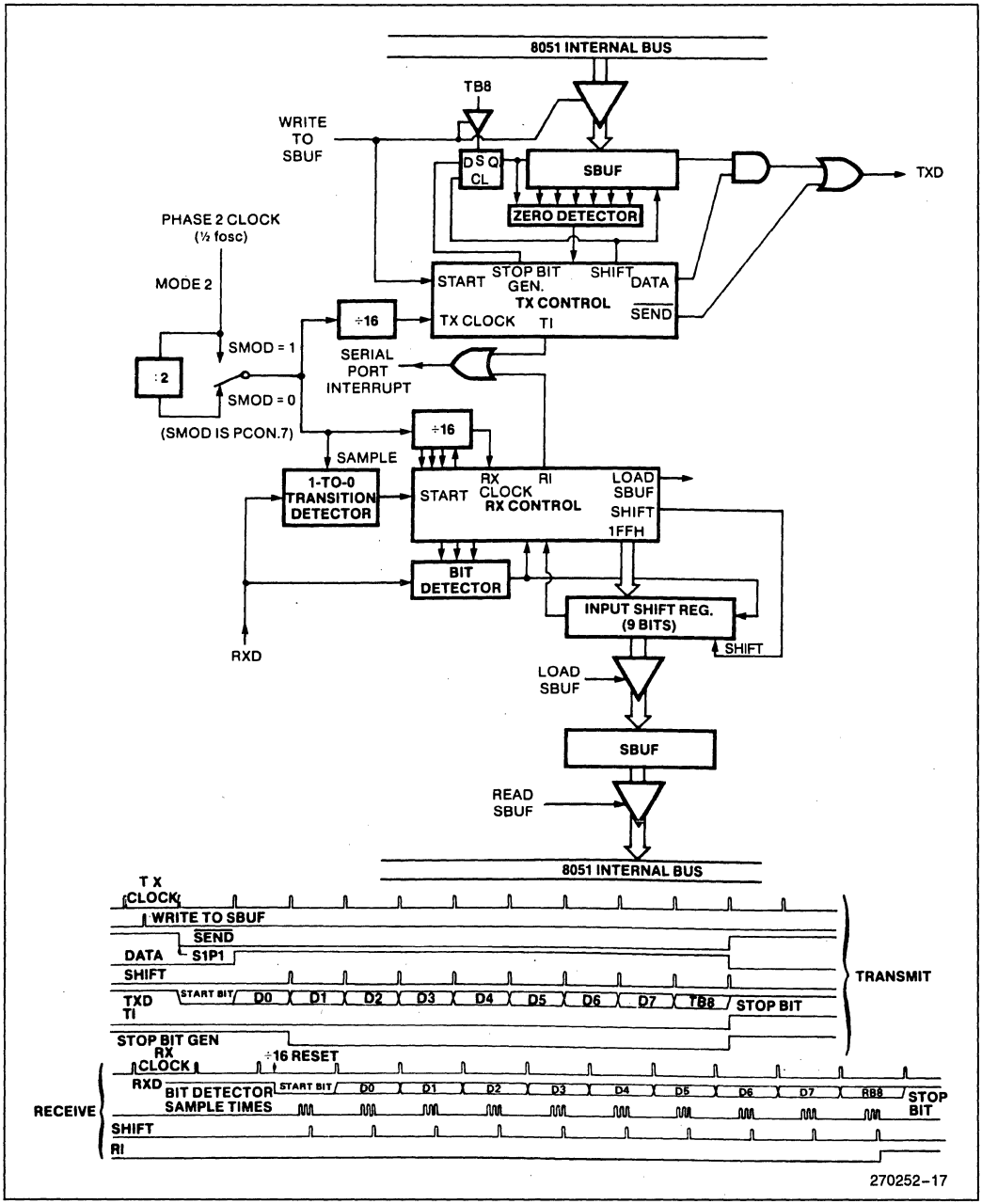
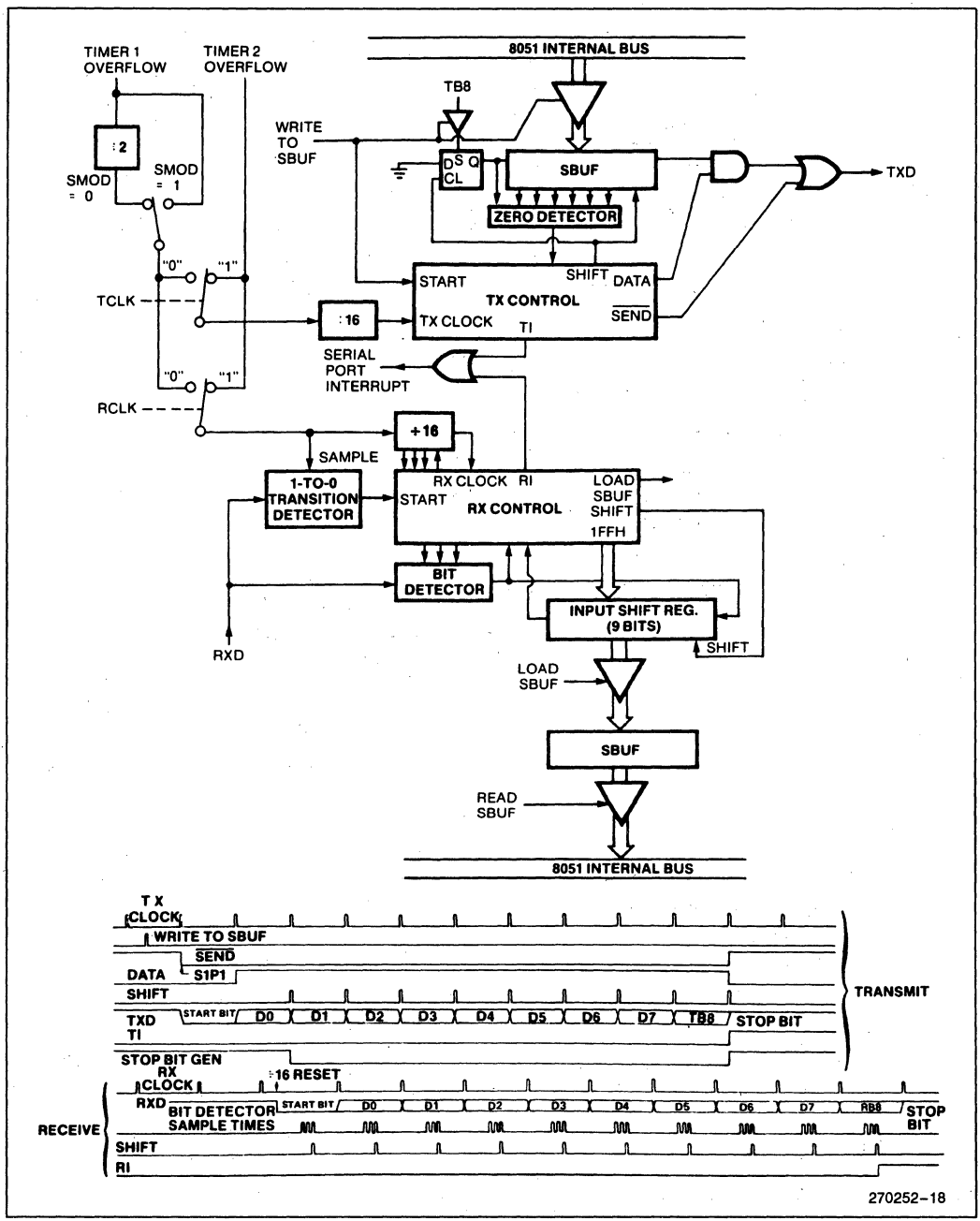


Figure 19. Serial Port Mode 2



270252-18

Figure 20. Serial Port Mode 3. TCLK, RCLK, and Timer 2 are Present in the 8052/8032 Only.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in Modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI = 0, and
2) Either SM2 = 0 or the received 9th data bit = 1

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a 1-to-0 transition at the RXD input.

Note that the value of the received stop bit is irrelevant to SBUF, RB8, or RI.

INTERRUPTS

The 8051 provides 5 interrupt sources. The 8052 provides 6. These are shown in Figure 21.

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

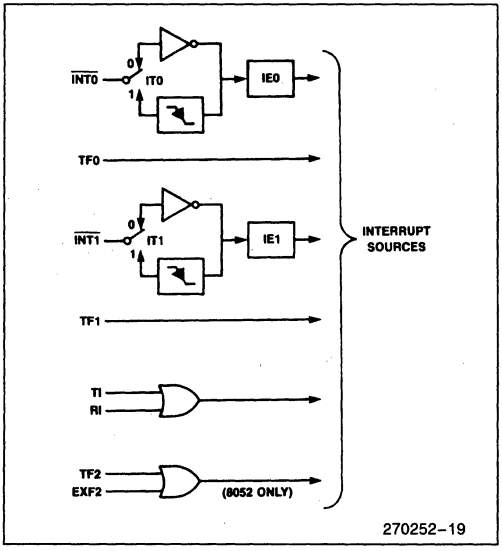


Figure 21. MCS®-51 Interrupt Sources

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except see Timer 0 in Mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

In the 8052, the Timer 2 Interrupt is generated by the logical OR of TF2 and EXF2. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in software.

(MSB) (LSB)

EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.
Enable Bit = 0 disables it.

Symbol	Position	Function
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, since they may be used in future MCS-51 products.

Figure 22. IE: Interrupt Enable Register

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (Figure 22). IE contains also a global disable bit, EA, which disables all interrupts at once.

Note in Figure 22 that bit position IE.6 is unimplemented. In the 8051s, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future MCS-51 products.

Priority Level Structure

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in Special Function Register IP (Figure 23). A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

(MSB) (LSB)

—	—	PT2	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority.
Priority bit = 0 assigns low priority.

Symbol	Position	Function
—	IP.7	reserved
—	IP.6	reserved
PT2	IP.5	Timer 2 interrupt priority bit.
PS	IP.4	Serial Port interrupt priority bit.
PT1	IP.3	Timer 1 interrupt priority bit.
PX1	IP.2	External interrupt 1 priority bit.
PT0	IP.1	Timer 0 interrupt priority bit.
PX0	IP.0	External interrupt 0 priority bit.

User software should never write 1s to unimplemented bits, since they may be used in future MCS-51 products.

Figure 23. IP: Interrupt Priority Register

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the **same** priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence, as follows:

	Source	Priority Within Level
1.	IE0	(highest)
2.	TF0	
3.	IE1	
4.	TF1	
5.	RI + TI	
6.	TF2 + EXF2	(lowest)

Note that the "priority within level" structure is only used to resolve *simultaneous requests of the same priority level.*

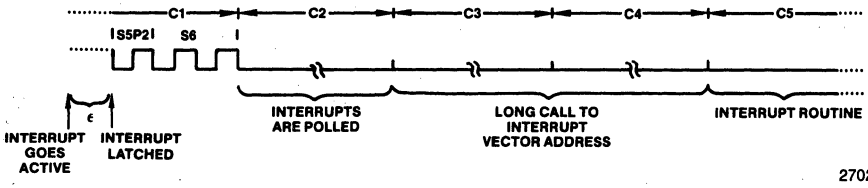
The IP register contains a number of unimplemented bits. IP.7 and IP.6 are vacant in the 8052s, and in the 8051s these and IP.5 are vacant. User software should not write 1s to these bit positions, since they may be used in future MCS-51 products.

How Interrupts Are Handled

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. The 8052's Timer 2 interrupt cycle is different, as described in the Response Time Section. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be



This is the fastest possible response when C2 is the final cycle of an instruction other than RETI or an access to IE or IP.

Figure 24. Interrupt Response Timing Diagram

completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least *one more* instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note then that if an interrupt flag is active but not being responded to for one of the above conditions, and is not *still* active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in Figure 24.

Note that if an interrupt of higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 24, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine having been executed.

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it doesn't. It never clears the Serial Port or Timer 2 flags. This has to be done in the user's software. It clears an external interrupt flag (IE0 or IE1) only if it was transition-activated. The hardware-generated LCALL pushes the contents of the Program Counter onto the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to, as shown below.

Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI + TI	0023H
TF2 + EXF2	002BH

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress.

External Interrupts

The external sources can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the $\overline{\text{INTx}}$ pin. If ITx = 1, external interrupt x is edge-triggered. In this mode if successive samples of the $\overline{\text{INTx}}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called.

If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

Response Time

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ levels are inverted and latched into the interrupt flags IE0 and IE1 at S5P2 of every machine cycle. Similarly, the Timer 2 flag EXF2 and the Serial Port flags RI and TI are set at S5P2. The values are not actually polled by the circuitry until the next machine cycle.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag TF2 is set at S2P2 and is polled in the same cycle in which the timer overflows.

If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine. Figure 24 shows interrupt response timings.

A longer response time would result if the request is blocked by one of the 3 previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles, since the longest instructions (MUL and DIV) are only 4

cycles long, and if the instruction in progress is RETI or an access to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single-interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

SINGLE-STEP OPERATION

The 8051 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be responded to while an interrupt of equal priority level is still in progress, nor will it be responded to after RETI until at least one other instruction has been executed. Thus, once an interrupt routine has been entered, it cannot be re-entered until at least one instruction of the interrupted program is executed. One way to use this feature for single-stop operation is to program one of the external interrupts (say, $\overline{INT0}$) to be level-activated. The service routine for the interrupt will terminate with the following code:

```
JNB  P3.2,$  ;Wait Here Till INT0 Goes High
JB   P3.2,$  ;Now Wait Here Till it Goes Low
RETI         :Go Back and Execute One Instruction
```

Now if the $\overline{INT0}$ pin, which is also the P3.2 pin, is held normally low, the CPU will go right into the External Interrupt 0 routine and stay there until $\overline{INT0}$ is pulsed (from low to high to low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately re-enter the External Interrupt 0 routine to await the next pulsing of P3.2. One step of the task program is executed each time P3.2 is pulsed.

RESET

The reset input is the RST pin, which is the input to a Schmitt Trigger.

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), *while the oscillator is running.* The CPU responds by generating an internal reset, with the timing shown in Figure 25.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins will maintain their current activities for 19 oscillator periods after a logic 1 has been sampled at the RST pin; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

While the RST pin is high, ALE and PSEN are weakly pulled high. After RST is pulled low, it will take 1 to 2 machine cycles for ALE and PSEN to start clocking. For this reason, other devices can not be synchronized to the internal timings of the 8051.

Driving the ALE and $\overline{PSEN}$ pins to 0 while reset is active could cause the device to go into an indeterminate state.

The internal reset algorithm writes 0s to all the SFRs except the port latches, the Stack Pointer, and SBUF. The port latches are initialized to FFH, the Stack Pointer to 07H, and SBUF is indeterminate. Table 3 lists the SFRs and their reset values.

The internal RAM is not affected by reset. On power up the RAM content is indeterminate.

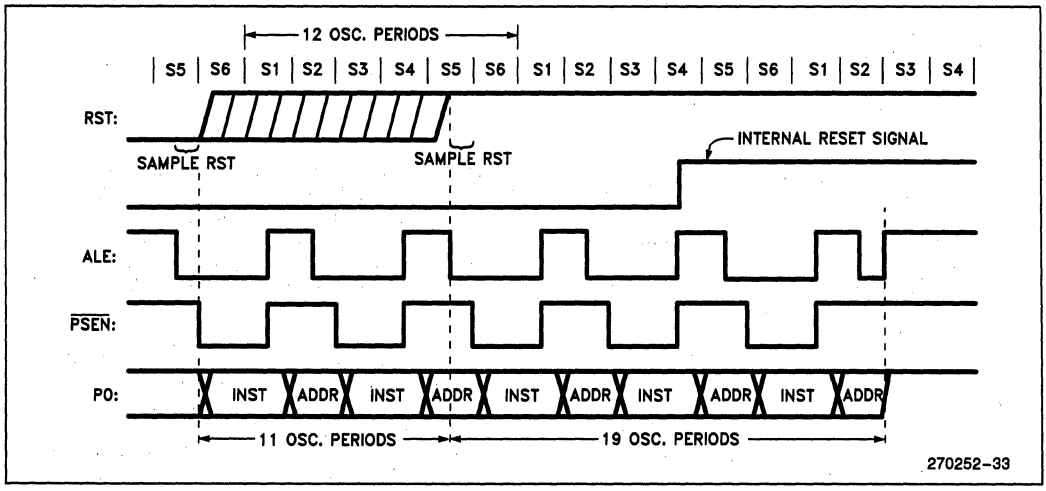


Figure 25. Reset Timing

Table 3. Reset Values of the SFRs

SFR Name	Reset Value
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0–P3	FFH
IP (8051)	XXX00000B
IP (8052)	XX000000B
IE (8051)	0XX00000B
IE (8052)	0X000000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2 (8052)	00H
TL2 (8052)	00H
RCAP2H (8052)	00H
RCAP2L (8052)	00H
SCON	00H
SBUF	Indeterminate
PCON (HMOS)	0XXXXXXXB
PCON (CHMOS)	0XXX0000B

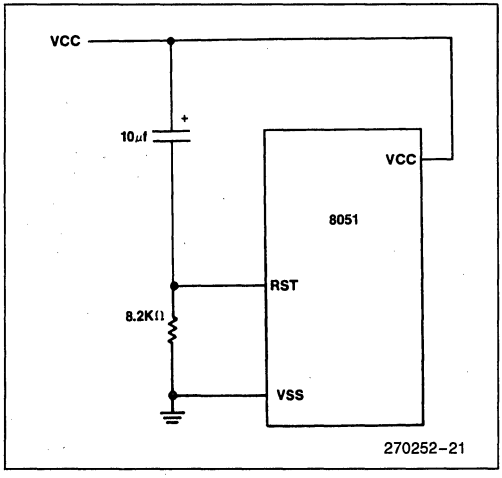


Figure 26. Power on Reset Circuit

POWER-ON RESET

For HMOS devices when V_{CC} is turned on an automatic reset can be obtained by connecting the RST pin to V_{CC} through a 10 μF capacitor and to V_{SS} through an 8.2 KΩ resistor (Figure 26). The CHMOS devices do not require this resistor although its presence does no harm. In fact, for CHMOS devices the external resistor can be removed because they have an internal pulldown on the RST pin. The capacitor value could then be reduced to 1 μF.

When power is turned on, the circuit holds the RST pin high for an amount of time that depends on the capacitor value and the rate at which it charges. To ensure a valid reset the RST pin must be held high long enough to allow the oscillator to start up plus two machine cycles.

On power up, V_{CC} should rise within approximately ten milliseconds. The oscillator start-up time will depend on the oscillator frequency. For a 10 MHz crystal, the start-up time is typically 1 ms. For a 1 MHz crystal, the start-up time is typically 10 ms.

With the given circuit, reducing V_{CC} quickly to 0 causes the RST pin voltage to momentarily fall below 0V. However, this voltage is internally limited and will not harm the device.

NOTE:
The port pins will be in a random state until the oscillator has started and the internal reset algorithm has written 1s to them.

Powering up the device without a valid reset could cause the CPU to start executing instructions from an indeterminate location. This is because the SFRs, specifically the Program Counter, may not get properly initialized.

POWER-SAVING MODES OF OPERATION

For applications where power consumption is critical the CHMOS version provides power reduced modes of operation as a standard feature. The power down mode in HMOS devices is no longer a standard feature and is being phased out.

CHMOS Power Reduction Modes

CHMOS versions have two power-reducing modes, Idle and Power Down. The input through which back-up power is supplied during these operations is VCC. Figure 27 shows the internal circuitry which implements these features. In the Idle mode (IDL = 1), the oscillator continues to run and the Interrupt, Serial Port, and Timer blocks continue to be clocked, but the

clock signal is gated off to the CPU. In Power Down (PD = 1), the oscillator is frozen. The Idle and Power Down modes are activated by setting bits in Special Function Register PCON. The address of this register is 87H. Figure 26 details its contents.

In the HMOS devices the PCON register only contains SMOD. The other four bits are implemented only in the CHMOS devices. User software should never write 1s to unimplemented bits, since they may be used in future MCS-51 products.

IDLE MODE

An instruction that sets PCON.0 causes that to be the last instruction executed before going into the Idle mode. In the Idle mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical states they had at the time Idle was activated. ALE and $\overline{PSEN}$ hold at logic high levels.

There are two ways to terminate the Idle. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into Idle.

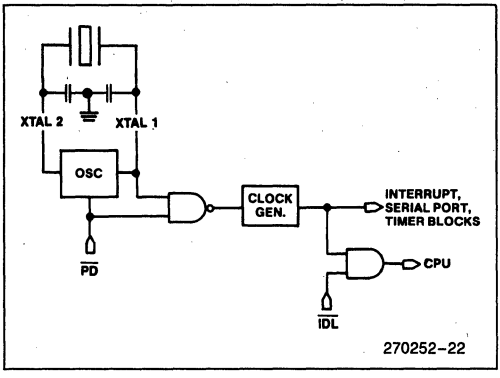


Figure 27. Idle and Power Down Hardware

(MSB)							(LSB)
SMOD	-	-	-	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1 and Timer 1 is used to generate baud rate, and the Serial Port is used in modes 1, 2, or 3.
—	PCON.6	(Reserved)
—	PCON.5	(Reserved)
—	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX0000).
In the HMOS devices the PCON register only contains SMOD. The other four bits are implemented only in the CHMOS devices. User software should never write 1s to unimplemented bits, since they may be used in future MCS-51 products.

Figure 28. PCON: Power Control Register

The flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during an Idle. For example, an instruction that activates Idle can also set one or both flag bits. When Idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset needs to be held active for only two machine cycles (24 oscillator periods) to complete the reset.

The signal at the RST pin clears the IDL bit directly and asynchronously. At this time the CPU resumes program execution from where it left off; that is, at the instruction following the one that invoked the Idle Mode. As shown in Figure 25, two or three machine cycles of program execution may take place before the internal reset algorithm takes control. On-chip hardware inhibits access to the internal RAM during this time, but access to the port pins is not inhibited. To eliminate the possibility of unexpected outputs at the port pins, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external Data RAM.

POWER DOWN MODE

An instruction that sets PCON.1 causes that to be the last instruction executed before going into the Power Down mode. In the Power Down mode, the on-chip oscillator is stopped. With the clock frozen, all func-

Table 4. EPROM Versions of the 8051 and 8052

Device Name	EPROM Version	EPROM Bytes	Ckt Type	VPP	Time Required to Program Entire Array
8051	(8751)	4K	HMOS	21.0V	4 minutes
8051AH	8751H	4K	HMOS	21.0V	4 minutes
80C51BH	87C51	4K	CHMOS	12.75V	13 seconds
8052AH	8752BH	8K	HMOS	12.75V	26 seconds

tions are stopped, but the on-chip RAM and Special Function Registers are held. The port pins output the values held by their respective SFRs. ALE and $\overline{PSEN}$ output lows.

The only exit from Power Down for the 80C51 is a hardware reset. Reset redefines all the SFRs, but does not change the on-chip RAM.

In the Power Down mode of operation, VCC can be reduced to as low as 2V. Care must be taken, however, to ensure that VCC is not reduced before the Power Down mode is invoked, and that VCC is restored to its normal operating level, before the Power Down mode is terminated. The reset that terminates Power Down also frees the oscillator. The reset should not be activated before VCC is restored to its normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10 msec).

EPROM VERSIONS

The EPROM versions of these devices are listed in Table 4. The 8751H programs at VPP = 21V using one 50 msec $\overline{PROG}$ pulse per byte programmed. This results in a total programming time (4K bytes) of approximately 4 minutes.

The 8752BH and 87C51 use the faster "Quick-Pulse" programming™ algorithm. These devices program at VPP = 12.75V using a series of twenty-five 100 μs $\overline{PROG}$ pulses per byte programmed. This results in a total programming time of approximately 26 seconds for the 8752BH (8K bytes) and 13 seconds for the 87C51 (4K bytes).

Detailed procedures for programming and verifying each device are given in the data sheets.

EXPOSURE TO LIGHT

It is good practice to cover the EPROM window with an opaque label when the device is in operation. This is not so much to protect the EPROM array from inadvertent erasure, but to protect the RAM and other on-chip logic. Allowing light to impinge on the silicon die while the device is operating can cause logical malfunction.

Program Memory Locks

In some microcontroller applications it is desirable that the Program Memory be secure from software piracy. Intel has responded to this need by implementing a Program Memory locking scheme in some of the MCS-51 devices. While it is impossible for anyone to guarantee absolute security against all levels of technological sophistication, the Program Memory locks in the MCS-51 devices will present a formidable barrier against illegal readout of protected software.

One Lock Bit Scheme on 8751H

The 8751H contains a lock bit which, once programmed, denies electrical access by any external means to the on-chip Program Memory. The effect of this lock bit is that while it is programmed the internal Program Memory can not be read out, the device can not be further programmed, and it *can not execute external Program Memory.* Erasing the EPROM array deactivates the lock bit and restores the device's full functionality. It can then be re-programmed.

The procedure for programming the lock bit is detailed in the 8751H data sheet.

Two-Level Program Memory Lock Scheme

The 87C51 and 8752BH contain two Program Memory locking schemes: Encrypted Verify and Lock Bits.

Encrypted Verify: These devices implement a 32-byte EPROM array that can be programmed by the customer, and which can then be used to encrypt the program code bytes during EPROM verification. The EPROM verification procedure is performed as usual, except that each code byte comes out X-NORed with one of the 32 key bytes. The key bytes are gone through in sequence. Therefore, to read the ROM code, one has to know the 32 key bytes in their proper sequence.

Unprogrammed bytes have the value FFH. Therefore, if the Encryption Array is left unprogrammed all the key bytes have the value FFH. Since any code byte X-NORed with FFH leaves the code byte unchanged, leaving the Encryption Array unprogrammed in effect bypasses the encryption feature.

Lock Bits: Also on the chip are two Lock Bits which can be left unprogrammed (U) or programmed (P) to obtain the following features:

Bit 2	Bit 1	Additional Features
U	U	None
U	P	• Externally fetched code can not access internal Program Memory. • Further programming disabled.
P	U	(Reserved for Future definition.)
P	P	• Externally fetched code can not access internal Program Memory. • Further programming disabled. • Program verification is disabled.

When Lock Bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

ROM Protection

The 8051AHP and 80C51BHP are ROM Protected versions of the 8051AH and 80C51BH, respectively. To incorporate this Protection Feature, program verification has been disabled and external memory accesses have been limited to 4K. Refer to the data sheets on these parts for more information.

ONCE Mode

The ONCE ("on-circuit emulation") mode facilitates testing and debugging of systems using the device without the device having to be removed from the circuit. The ONCE mode is invoked by:

1. Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored after a normal reset is applied.

THE ON-CHIP OSCILLATORS

HMOS Versions

The on-chip oscillator circuitry for the HMOS (HMOS-I and HMOS-II) members of the MCS-51 family is a single stage linear inverter (Figure 29), intended for use as a crystal-controlled, positive reactance oscillator (Figure 30). In this application the crystal is operated in its fundamental response mode as an inductive reactance in parallel resonance with capacitance external to the crystal.

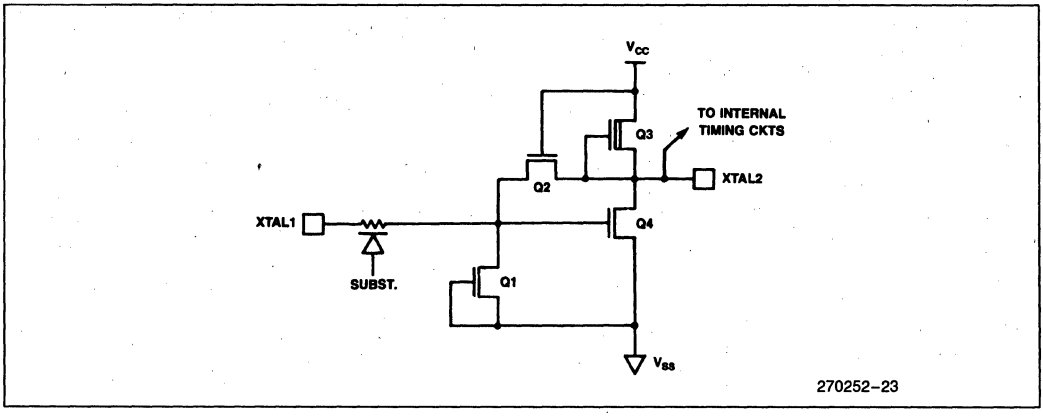


Figure 29. On-Chip Oscillator Circuitry in the HMOS Versions of the MCS®-51 Family

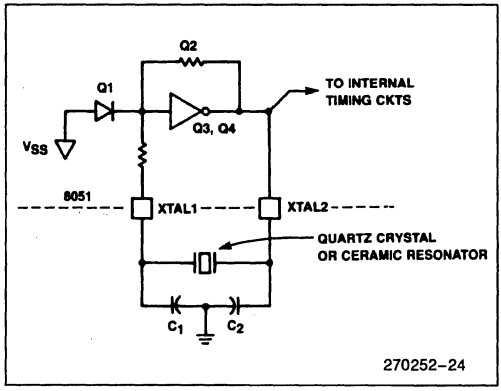


Figure 30. Using the HMOS On-Chip Oscillator

The crystal specifications and capacitance values (C1 and C2 in Figure 30) are not critical. 30 pF can be used in these positions at any frequency with good quality crystals. A ceramic resonator can be used in place of the crystal in cost-sensitive applications. When a ceramic resonator is used, C1 and C2 are normally selected to be of somewhat higher values, typically, 47 pF. The manufacturer of the ceramic resonator should be consulted for recommendations on the values of these capacitors.

In general, crystals used with these devices typically have the following specifications:

ESR (Equivalent Series Resistance)	see Figure 31
C_O (Shunt Capacitance)	7.0 pF max.
C_L (Load Capacitance)	30 pF ±3 pF
Drive Level	1 MW

Frequency, tolerance and temperature range are determined by the system requirements.

A more in-depth discussion of crystal specifications, ceramic resonators, and the selection of values for C1 and C2 can be found in Application Note AP-155, "Oscillators for Microcontrollers," which is included in the *Embedded Control Applications Handbook.*

To drive the HMOS parts with an external clock source, apply the external clock signal to XTAL2, and ground XTAL1, as shown in Figure 32. A pullup resistor may be used (to increase noise margin), but is optional if VOH of the driving gate exceeds the VIH MIN specification of XTAL2.

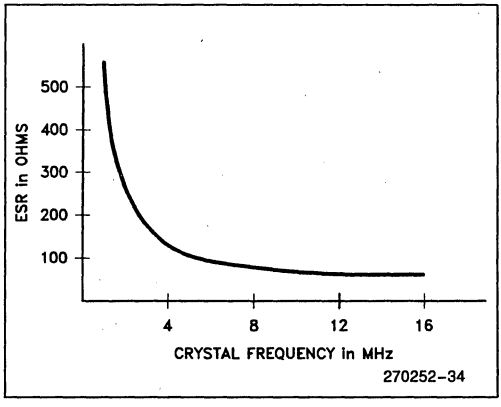


Figure 31. ESR vs Frequency

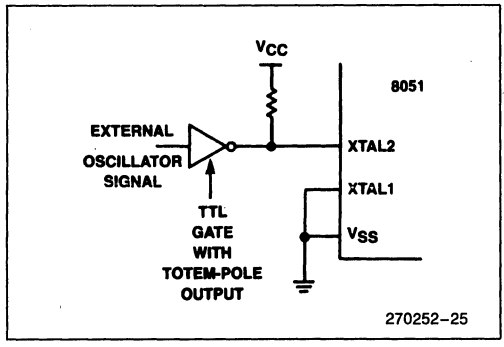


Figure 32. Driving the HMOS MCS®-51 Parts with an External Clock Source

CHMOS VERSIONS

The on-chip oscillator circuitry for the 80C51BH, shown in Figure 33, consists of a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator in the same manner as the HMOS parts. However, there are some important differences.

One difference is that the 80C51BH is able to turn off its oscillator under software control (by writing a 1 to the PD bit in PCON). Another difference is that in the 80C51BH the internal clocking circuitry is driven by the signal at XTAL1, whereas in the HMOS versions it is by the signal at XTAL2.

The feedback resistor R_f in Figure 33 consists of paralleled n- and p- channel FETs controlled by the PD bit, such that R_f is opened when PD = 1. The diodes D1 and D2, which act as clamps to VCC and VSS, are parasitic to the R_f FETs.

The oscillator can be used with the same external components as the HMOS versions, as shown in Figure 34. Typically, C1 = C2 = 30 pF when the feedback element is a quartz crystal, and C1 = C2 = 47 pF when a ceramic resonator is used.

To drive the CHMOS parts with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 float, as shown in Figure 35.

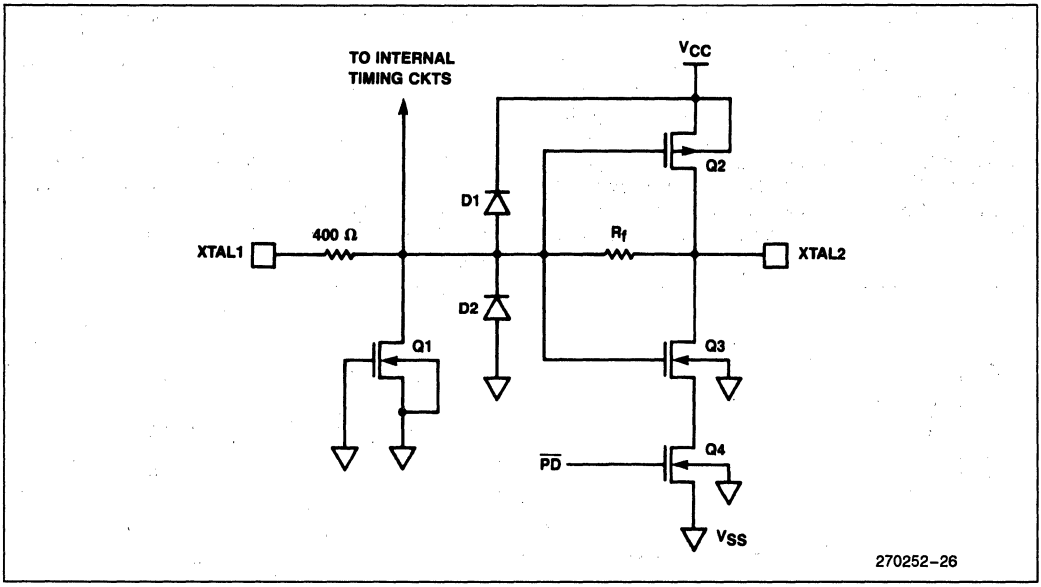


Figure 33. On-Chip Oscillator Circuitry in the CHMOS Versions of the MCS®-51 Family

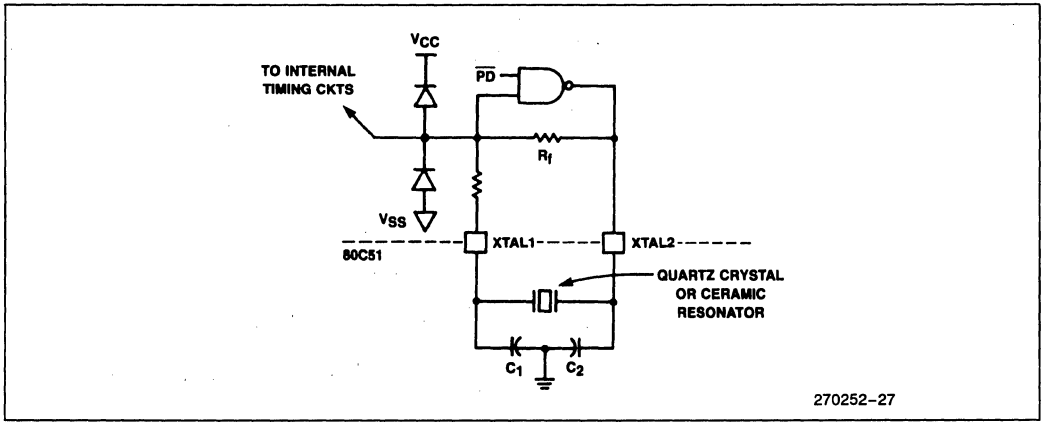


Figure 34. Using the CHMOS On-Chip Oscillator

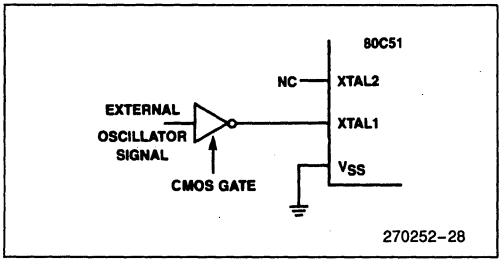


Figure 35. Driving the CHMOS MCS®-51 Parts with an External Clock Source

The reason for this change from the way the HMOS part is driven can be seen by comparing Figures 29 and 33. In the HMOS devices the internal timing circuits are driven by the signal at XTAL2. In the CHMOS devices the internal timing circuits are driven by the signal at XTAL1.

INTERNAL TIMING

Figures 36 through 39 show when the various strobe and port signals are clocked internally. The figures do not show rise and fall times of the signals, nor do they show propagation delays between the XTAL signal and events at other pins.

Rise and fall times are dependent on the external loading that each pin must drive. They are often taken to be something in the neighborhood of 10 nsec, measured between 0.8V and 2.0V.

Propagation delays are different for different pins. For a given pin they vary with pin loading, temperature, VCC, and manufacturing lot. If the XTAL waveform is taken as the timing reference, prop delays may vary from 25 to 125 nsec.

The AC Timings section of the data sheets do not reference any timing to the XTAL waveform. Rather, they relate the critical edges of control and input signals to each other. The timings published in the data sheets include the effects of propagation delays under the specified test conditions.

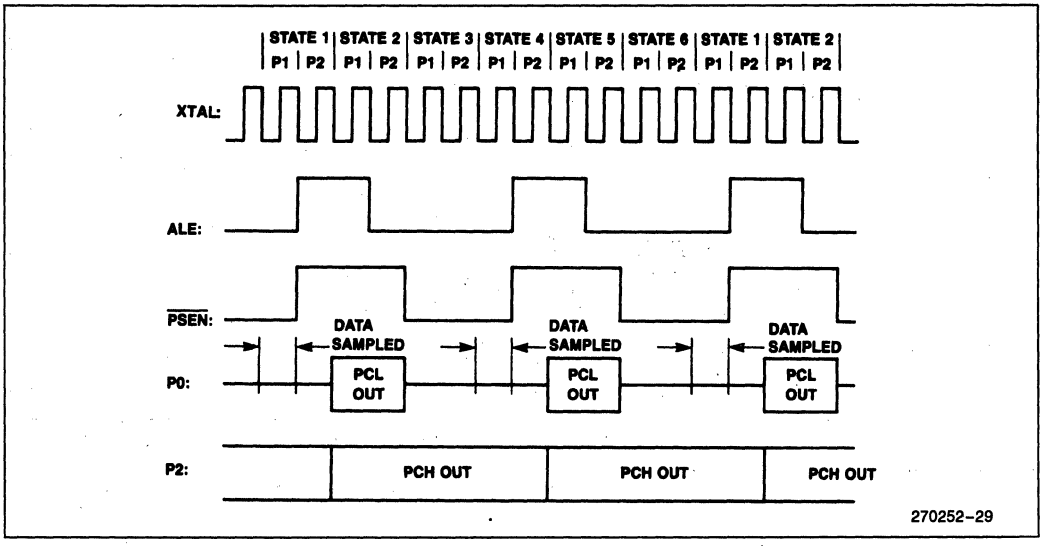


Figure 36. External Program Memory Fetches

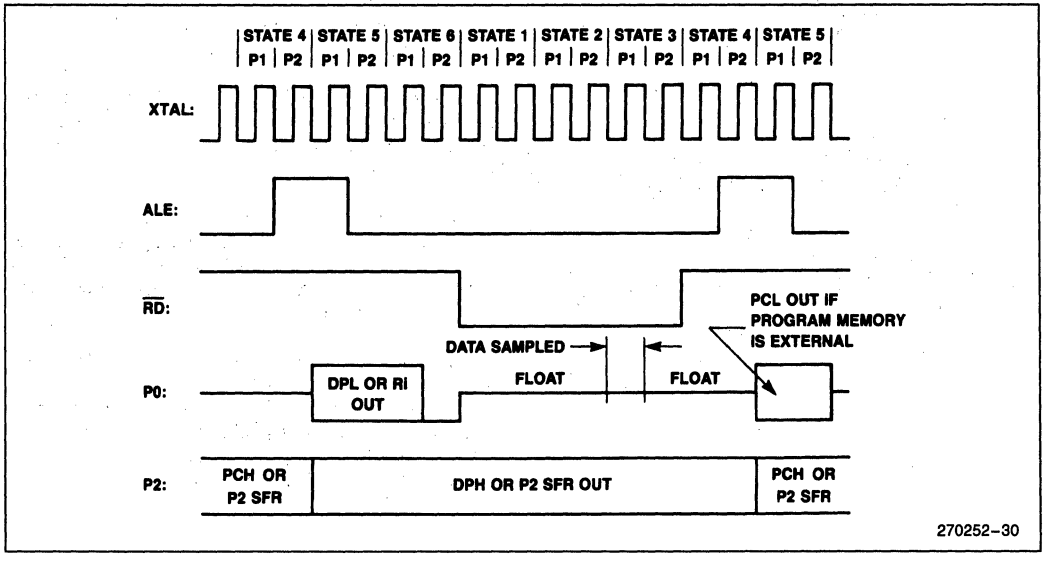


Figure 37. External Data Memory Read Cycle

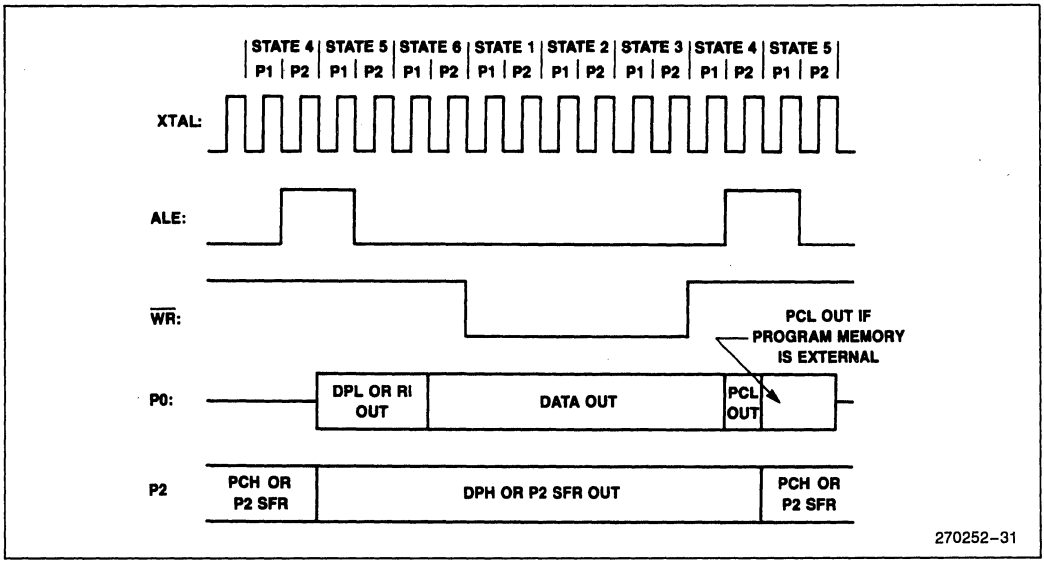


Figure 38. External Data Memory Write Cycle

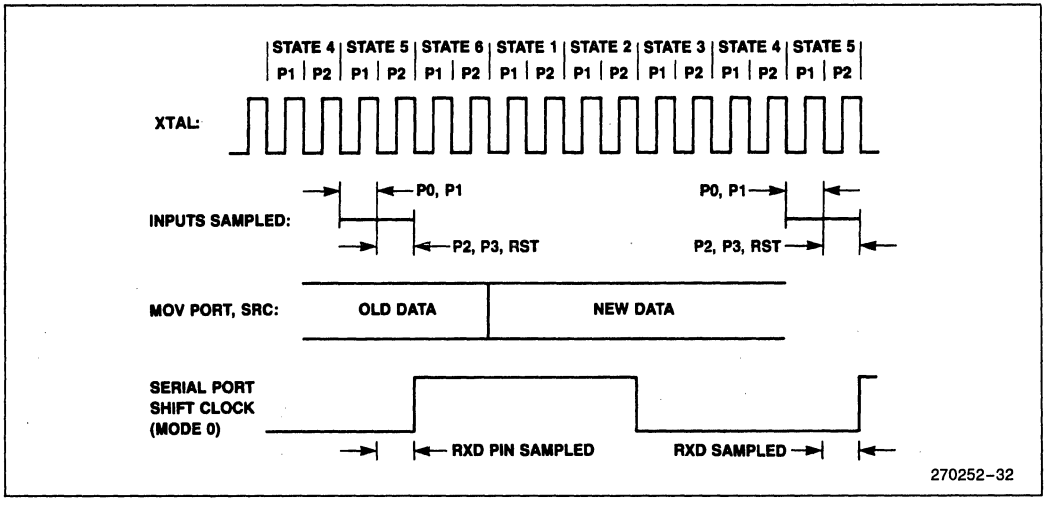


Figure 39. Port Operation

ADDITIONAL REFERENCES

The following application notes and articles are found in the *Embedded Control Applications* handbook. (Order Number: 270535)

1. AP-125 "Designing Microcontroller Systems for Electrically Noisy Environments".
2. AP-155 "Oscillators for Microcontrollers".
3. AP-252 "Designing with the 80C51BH".
4. AR-409 "Increased Functions in Chip Result in Lighter, Less Costly Portable Computer".
5. AR-517 "Using the 8051 Microcontroller with Resonant Transducers".

MCS®-51
8-BIT CONTROL-ORIENTED MICROCOMPUTERS
8031/8051
8031AH/8051AH
8032AH/8052AH
8751H/8751H-8

- **High Performance HMOS Process**
- **Internal Timers/Event Counters**
- **2-Level Interrupt Priority Structure**
- **32 I/O Lines (Four 8-Bit Ports)**
- **64K Program Memory Space**
- **Security Feature Protects EPROM Parts Against Software Piracy**
- **Boolean Processor**
- **Bit-Addressable RAM**
- **Programmable Full Duplex Serial Channel**
- **111 Instructions (64 Single-Cycle)**
- **64K Data Memory Space**

The MCS®-51 products are optimized for control applications. Byte-processing and numerical operations on small data structures are facilitated by a variety of fast addressing modes for accessing the internal RAM. The instruction set provides a convenient menu of 8-bit arithmetic instructions, including multiply and divide instructions. Extensive on-chip support is provided for one-bit variables as a separate data type, allowing direct bit manipulation and testing in control and logic systems that require Boolean processing.

The 8051 is the original member of the MCS-51 family. The 8051AH is identical to the 8051, but it is fabricated with HMOS II technology.

The 8751H is an EPROM version of the 8051AH; that is, the on-chip Program Memory can be electrically programmed, and can be erased by exposure to ultraviolet light. It is fully compatible with its predecessor, the 8751-8, but incorporates two new features: a Program Memory Security bit that can be used to protect the EPROM against unauthorized read-out, and a programmable baud rate modification bit (SMOD). The 8751H-8 is identical to the 8751H but only operates up to 8 MHz.

The 8052AH is an enhanced version of the 8051AH. It is backwards compatible with the 8051AH and is fabricated with HMOS II technology. The 8052AH enhancements are listed in the table below. Also refer to this table for the ROM, ROMless, and EPROM versions of each product.

Device	Internal Memory		Timers/ Event Counters	Interrupts
	Program	Data		
8052AH	8K x 8 ROM	256 x 8 RAM	3 x 16-Bit	6
8051AH	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8051	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8032AH	none	256 x 8 RAM	3 x 16-Bit	6
8031AH	none	128 x 8 RAM	2 x 16-Bit	5
8031	none	128 x 8 RAM	2 x 16-Bit	5
8751H	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5
8751H-8	4K x 8 EPROM	128 x 8 RAM	2 x 16-Bit	5

October 1988
Order Number: 270048-004

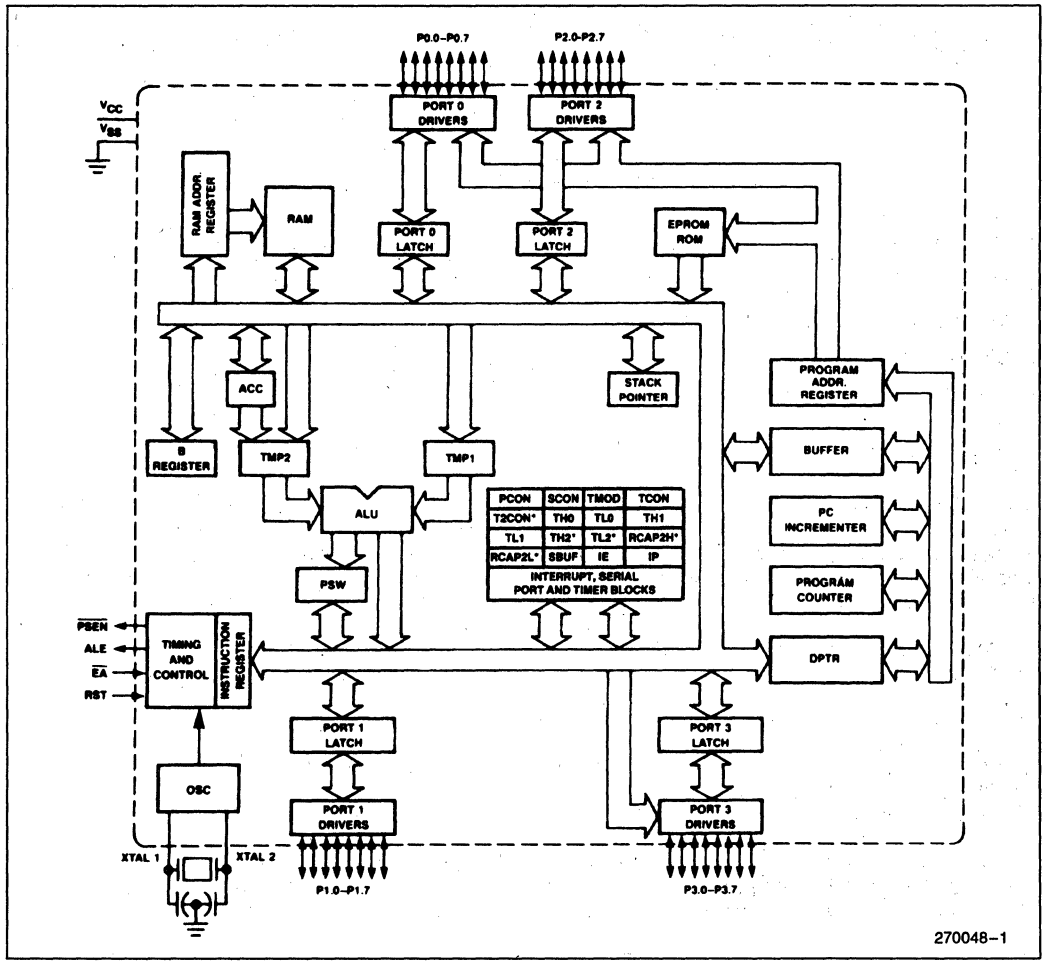


Figure 1. MCS®-51 Block Diagram

PACKAGES

Part	Prefix	Package Type
8051AH/ 8031AH	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-Pin PLCC
8052AH/ 8032AH	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-Pin PLCC
8751H/ 8751H-8	D	40-Pin CERDIP
	R	44-Pin LCC

PIN DESCRIPTIONS

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs.

Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s and can source and sink 8 LS TTL inputs.

Port 0 also receives the code bytes during programming of the EPROM parts, and outputs the code bytes during program verification of the ROM and EPROM parts. External pullups are required during program verification.

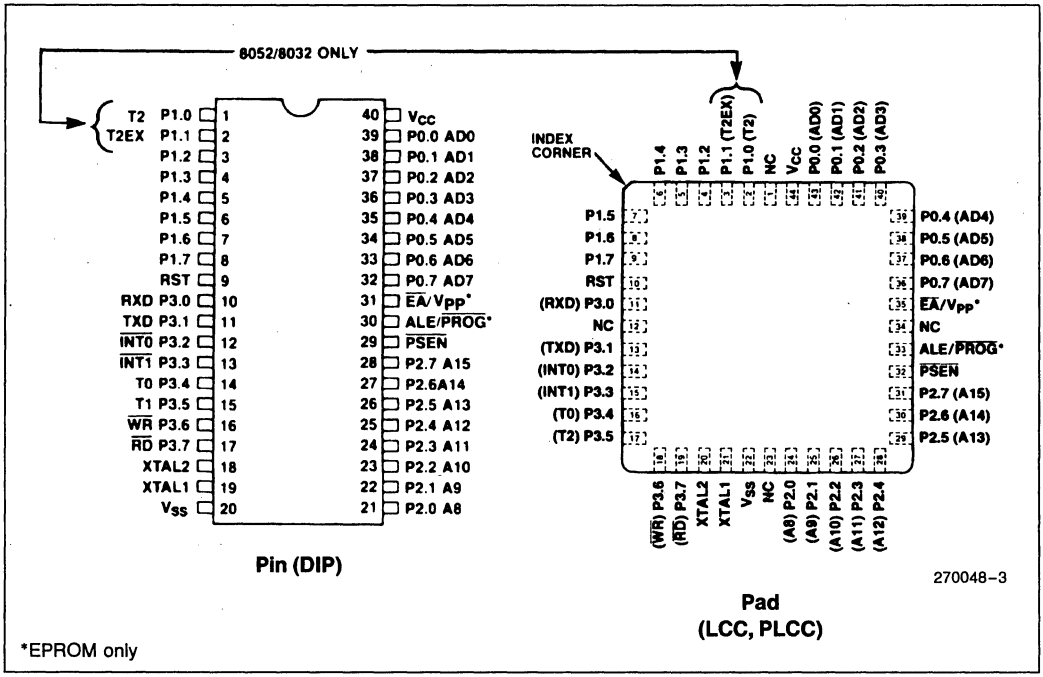


Figure 2. MCS®-51 Connections

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source 4 LS TTL inputs. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

In the 8032AH and 8052AH, Port 1 pins P1.0 and P1.1 also serve the T2 and T2EX functions, respectively.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL inputs. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL} on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternative Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during programming of the EPROM parts.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$: Program Store Enable is the read strobe to external Program Memory.

When the device is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory.

$\overline{EA}$/V_{PP}: External Access enable $\overline{EA}$ must be strapped to V_{SS} in order to enable any MCS-51 device to fetch code from external Program memory locations starting at 0000H up to FFFFH. $\overline{EA}$ must be strapped to V_{CC} for internal program execution.

Note, however, that if the Security Bit in the EPROM devices is programmed, the device will not fetch code from any location in external Program Memory.

This pin also receives the 21V programming supply voltage (VPP) during programming of the EPROM parts.

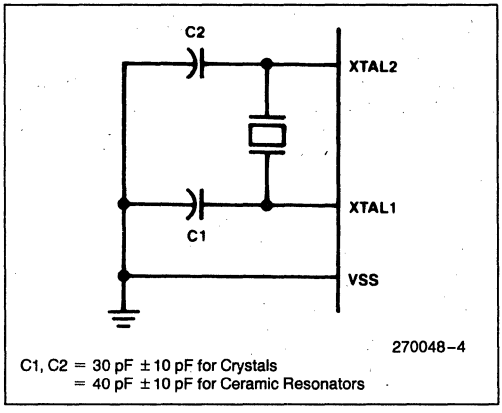


Figure 3. Oscillator Connections

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be grounded, while XTAL2 is driven, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

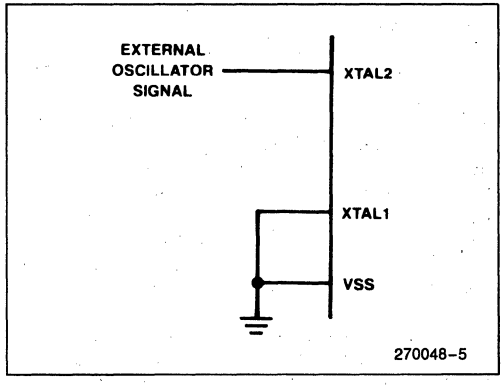


Figure 4. External Drive Configuration

DESIGN CONSIDERATIONS

If an 8751BH or 8752BH may replace an 8751H in a future design, the user should carefully compare both data sheets for DC or AC Characteristic differences. Note that the V_{IH} and I_{IH} specifications for the $\overline{EA}$ pin differ significantly between the devices.

Exposure to light when the EPROM device is in operation may cause logic errors. For this reason, it is suggested that an opaque label be placed over the window when the die is exposed to ambient light.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on $\overline{EA}$/V_{PP} Pin to V_{SS} ...−0.5V to +21.5V
Voltage on Any Other Pin to V_{SS} ....−0.5V to +7V
Power Dissipation.....1.5W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter		Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$ Pin of 8751H & 8751H-8)		−0.5	0.8	V	
V_{IL1}	Input Low Voltage to $\overline{EA}$ Pin of 8751H & 8751H-8		0	0.7	V	
V_{IH}	Input High Voltage (Except XTAL2, RST)		2.0	V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage to XTAL2, RST		2.5	V_{CC} + 0.5	V	XTAL1 = V_{SS}
V_{OL}	Output Low Voltage (Ports 1, 2, 3)*			0.45	V	I_{OL} = 1.6 mA
V_{OL1}	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$)*					
		8751H, 8751H-8		0.60 0.45	V V	I_{OL} = 3.2 mA I_{OL} = 2.4 mA
		All Others		0.45	V	I_{OL} = 3.2 mA
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$)		2.4		V	I_{OH} = −80 µA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)		2.4		V	I_{OH} = −400 µA
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3, RST) 8032AH, 8052AH All Others			−800 −500	µA µA	V_{IN} = 0.45V V_{IN} = 0.45V
I_{IL1}	Logical 0 Input Current to $\overline{EA}$ Pin of 8751H & 8751H-8 Only			−15	mA	V_{IN} = 0.45V
I_{IL2}	Logical 0 Input Current (XTAL2)			−3.2	mA	V_{IN} = 0.45V
I_{LI}	Input Leakage Current (Port 0) 8751H & 8751H-8 All Others			 ±100 ±10	 µA µA	 0.45 ≤ V_{IN} ≤ V_{CC} 0.45 ≤ V_{IN} ≤ V_{CC}
I_{IH}	Logical 1 Input Current to $\overline{EA}$ Pin of 8751H & 8751H-8			500	µA	V_{IN} = 2.4V
I_{IH1}	Input Current to RST to Activate Reset			500	µA	V_{IN} < (V_{CC} − 1.5V)
I_{CC}	Power Supply Current: 8031/8051 8031AH/8051AH 8032AH/8052AH 8751H/8751H-8			 160 125 175 250	 mA mA mA mA	 All Outputs Disconnected; $\overline{EA}$ = V_{CC}
C_{IO}	Pin Capacitance			10	pF	Test freq = 1 MHz

***NOTE:**
Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.

A.C. CHARACTERISTICS

T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V;
Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF;
Load Capacitance for All Other Outputs = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency			3.5	12.0	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL−40		ns
TLLAX	Address Hold after ALE Low	48		TCLCL−35		ns
TLLIV	ALE Low to Valid Instr In 8751H All Others		 183 233		 4TCLCL−150 4TCLCL−100	 ns ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	58		TCLCL−25		ns
TPLPH	$\overline{PSEN}$ Pulse Width 8751H All Others	 190 215		 3TCLCL−60 3TCLCL−35		 ns ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr In 8751H All Others		 100 125		 3TCLCL−150 3TCLCL−125	 ns ns
TPXIX	Input Instr Hold after $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float after $\overline{PSEN}$		63		TCLCL−20	ns
TPXAV	$\overline{PSEN}$ to Address Valid	75		TCLCL−8		ns
TAVIV	Address to Valid Instr In 8751H All Others		 267 302		 5TCLCL−150 5TCLCL−115	 ns ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		20		20	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold after $\overline{RD}$	0		0		ns
TRHDZ	Data Float after $\overline{RD}$		97		2TCLCL−70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition 8751H All Others	 13 23		 TCLCL−70 TCLCL−60		 ns ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL−150		ns
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL−50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		20		20	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High 8751H All Others	 33 43	 133 123	 TCLCL−50 TCLCL−40	 TCLCL+50 TCLCL+40	 ns ns

NOTE:
*This table does not include the 8751-8 A.C. characteristics (see next page).

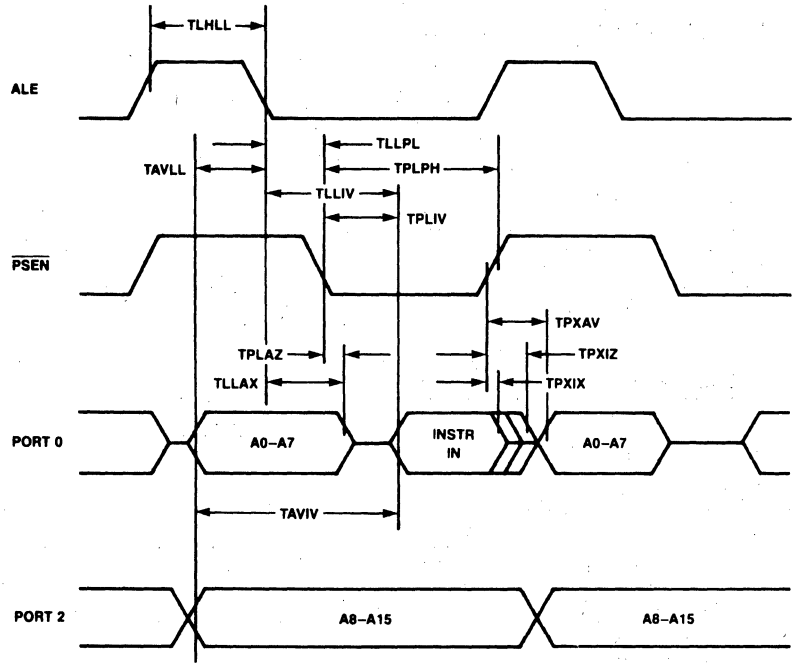
This Table is only for the 8751H-8

A.C. CHARACTERISTICS

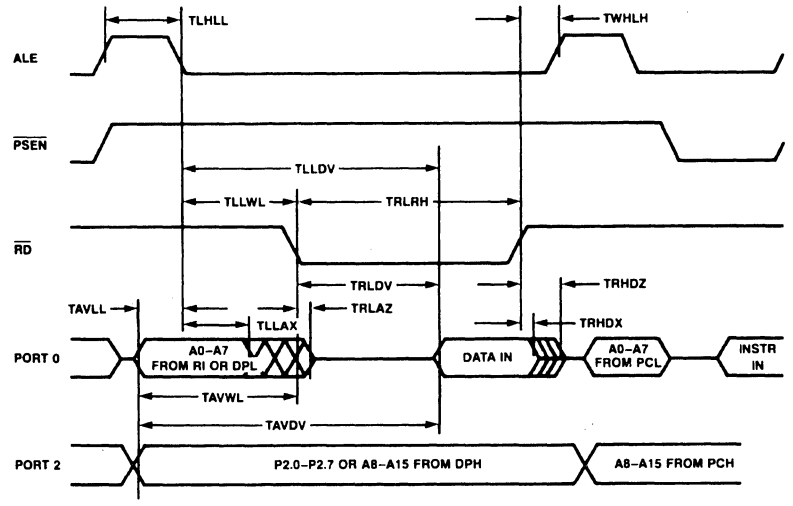
T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V;
Load Capacitance for Port 0, ALE, and PSEN = 100 pF;
Load Capacitance for All Other Outputs = 80 pF

Symbol	Parameter	8 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency			3.5	8.0	MHz
TLHLL	ALE Pulse Width	210		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	85		TCLCL−40		ns
TLLAX	Address Hold after ALE Low	90		TCLCL−35		ns
TLLIV	ALE Low to Valid Instr In		350		4TCLCL−150	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	100		TCLCL−25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	315		3TCLCL−60		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr In		225		3TCLCL−150	ns
TPXIX	Input Instr Hold after $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float after $\overline{PSEN}$		105		TCLCL−20	ns
TPXAV	$\overline{PSEN}$ to Address Valid	117		TCLCL−8		ns
TAVIV	Address to Valid Instr In		475		5TCLCL−150	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		20		20	ns
TRLRH	$\overline{RD}$ Pulse Width	650		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	650		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		460		5TCLCL−165	ns
TRHDX	Data Hold after $\overline{RD}$	0		0		ns
TRHDZ	Data Float after $\overline{RD}$		180		2TCLCL−70	ns
TLLDV	ALE Low to Valid Data In		850		8TCLCL−150	ns
TAVDV	Address to Valid Data In		960		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	325	425	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	370		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	55		TCLCL−70		ns
TQVWH	Data Valid to $\overline{WR}$ High	725		7TCLCL−150		ns
TWHQX	Data Hold after $\overline{WR}$	75		TCLCL−50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		20		20	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	75	175	TCLCL−50	TCLCL+50	ns

EXTERNAL PROGRAM MEMORY READ CYCLE

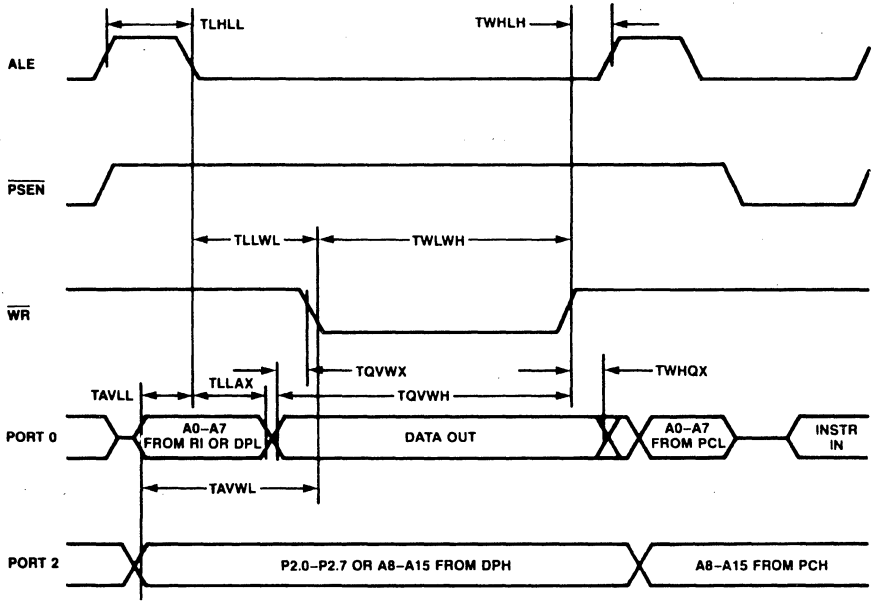


EXTERNAL DATA MEMORY READ CYCLE



270048-7

EXTERNAL DATA MEMORY WRITE CYCLE



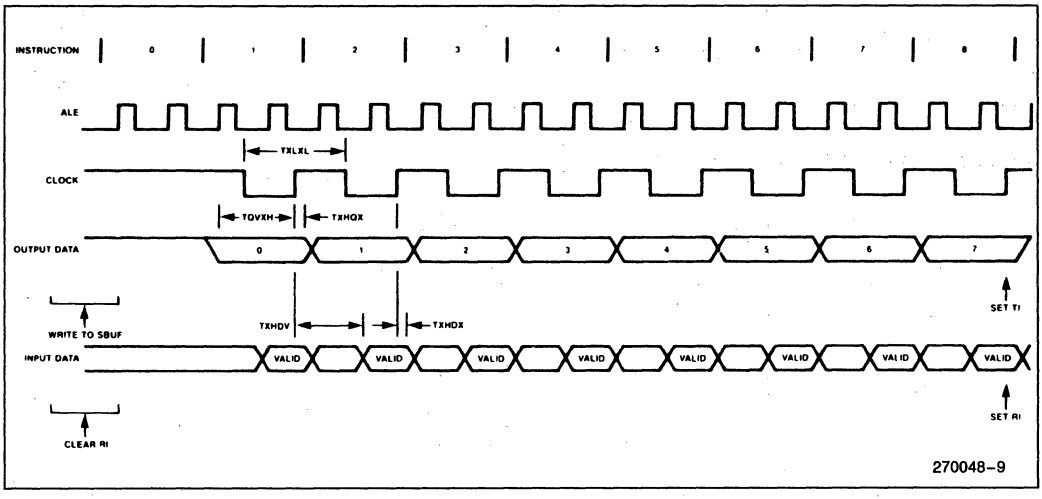
270048-8

SERIAL PORT TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to 70°C; VCC = 5V ±10%; VSS = 0V; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold after Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns

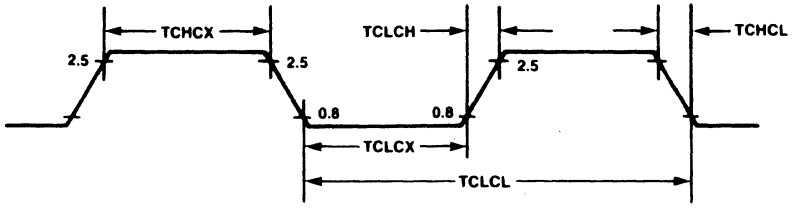
SHIFT REGISTER TIMING WAVEFORMS



EXTERNAL CLOCK DRIVE

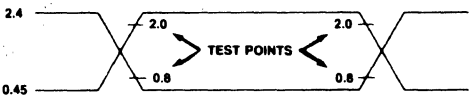
Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency (except 8751H-8) 8751H-8	3.5 3.5	12 8	MHz MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

EXTERNAL CLOCK DRIVE WAVEFORM



270048-10

A.C. TESTING INPUT, OUTPUT WAVEFORM



270048-11

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

EPROM CHARACTERISTICS

Table 3. EPROM Programming Modes

Mode	RST	$\overline{PSEN}$	ALE	$\overline{EA}$	P2.7	P2.6	P2.5	P2.4
Program	1	0	0*	VPP	1	0	X	X
Inhibit	1	0	1	X	1	0	X	X
Verify	1	0	1	1	0	0	X	X
Security Set	1	0	0*	VPP	1	1	X	X

NOTE:
"1" = logic high for that pin
"0" = logic low for that pin
"X" = "don't care"
"VPP" = +21V ±0.5V
*ALE is pulsed low for 50 ms.

Programming the EPROM

To be programmed, the part must be running with a 4 to 6 MHz oscillator. (The reason the oscillator needs to be running is that the internal bus is being used to transfer address and program data to appropriate internal registers.) The address of an EPROM location to be programmed is applied to Port 1 and pins P2.0–P2.3 of Port 2, while the code byte to be programmed into that location is applied to Port 0. The other Port 2 pins, and RST, $\overline{PSEN}$, and $\overline{EA}$ should be held at the "Program" levels indicated in Table 3. ALE is **pulsed** low for 50 ms to program the code byte into the addressed EPROM location. The setup is shown in Figure 5.

Normally $\overline{EA}$ is held at a logic high until just before ALE is to be pulsed. Then $\overline{EA}$ is raised to +21V, ALE is pulsed, and then $\overline{EA}$ is returned to a logic high. Waveforms and detailed timing specifications are shown in later sections of this data sheet.

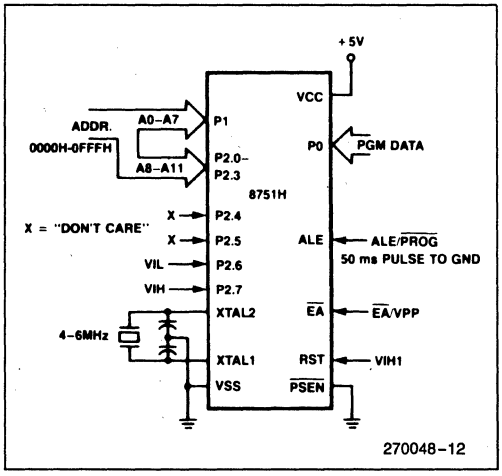


Figure 5. Programming Configuration

Note that the $\overline{EA}$/VPP pin must not be allowed to go above the maximum specified VPP level of 21.5V for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The VPP source should be well regulated and free of glitches.

Program Verification

If the Security Bit has not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired, either during or after the programming operation. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0–P2.3. The other pins should be held at the "Verify" levels indicated in Table 3. The contents of the addressed location will come out on Port 0. External pullups are required on Port 0 for this operation.

The setup, which is shown in Figure 6, is the same as for programming the EPROM except that pin P2.7 is held at a logic low, or may be used as an active-low read strobe.

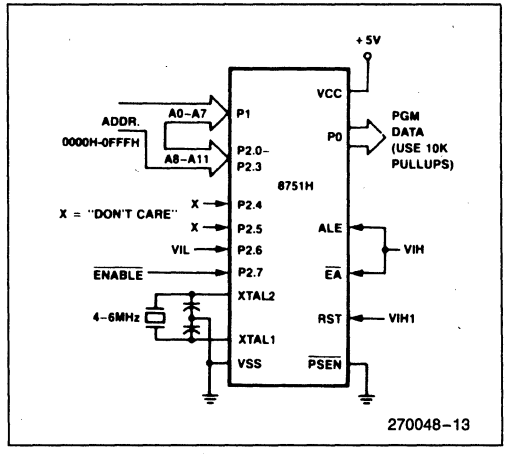


Figure 6. Program Verification

EPROM Security

The security feature consists of a "locking" bit which when programmed denies electrical access by any external means to the on-chip Program Memory. The bit is programmed as shown in Figure 7. The setup and procedure are the same as for normal EPROM programming, except that P2.6 is held at a logic high. Port 0, Port 1, and pins P2.0–P2.3 may be in any state. The other pins should be held at the "Security" levels indicated in Table 3.

Once the Security Bit has been programmed, it can be cleared only by full erasure of the Program Memory. While it is programmed, the internal Program Memory can not be read out, the device can not be further programmed, and it **can not execute out of external program memory**. Erasing the EPROM, thus clearing the Security Bit, restores the device's full functionality. It can then be reprogrammed.

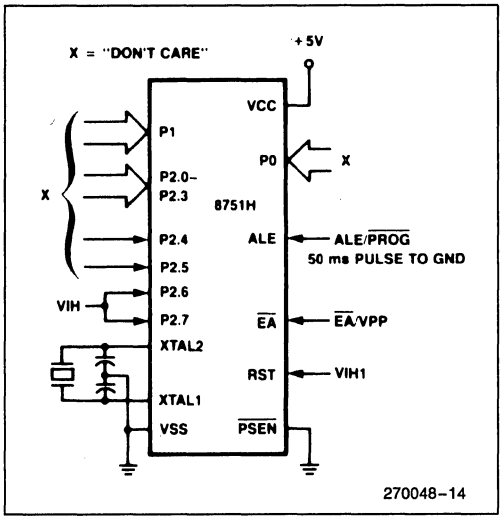


Figure 7. Programming the Security Bit

Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 Angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room-level fluorescent lighting) could cause inadvertent erasure. If an application subjects the device to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 Angstroms) to an integrated dose of at least 15 W-sec/cm^2. Exposing the EPROM to an ultraviolet lamp of 12,000 μW/cm^2 rating for 20 to 30 minutes, at a distance of about 1 inch, should be sufficient.

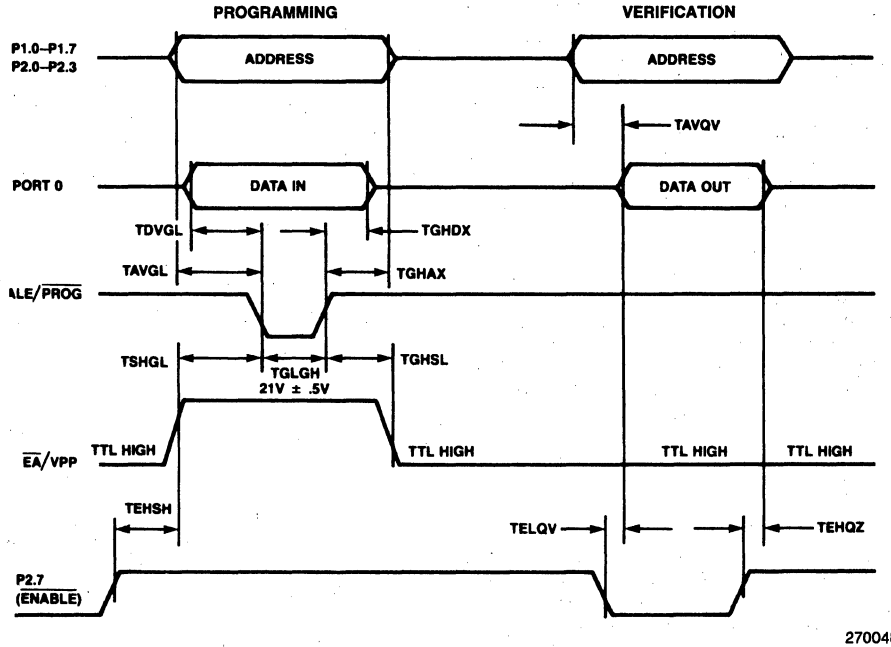
Erasure leaves the array in an all 1s state.

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T_A = 21°C to 27°C; VCC = 5V ±10%; VSS = 0V

Symbol	Parameter	Min	Max	Units
VPP	Programming Supply Voltage	20.5	21.5	V
IPP	Programming Supply Current		30	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{\text{PROG}}$ Low	48TCLCL		
TGHAX	Address Hold after $\overline{\text{PROG}}$	48TCLCL		
TDVGL	Data Setup to $\overline{\text{PROG}}$ Low	48TCLCL		
TGHDX	Data Hold after $\overline{\text{PROG}}$	48TCLCL		
TEHSH	P2.7 ($\overline{\text{ENABLE}}$) High to VPP	48TCLCL		
TSHGL	VPP Setup to $\overline{\text{PROG}}$ Low	10		μs
TGHSL	VPP Hold after $\overline{\text{PROG}}$	10		μs
TGLGH	$\overline{\text{PROG}}$ Width	45	55	ms
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{\text{ENABLE}}$ Low to Data Valid		48TCLCL	
TEHQZ	Data Float after $\overline{\text{ENABLE}}$	0	48TCLCL	

EPROM PROGRAMMING AND VERIFICATION WAVEFORMS



For programming conditions see Figure 5.

For verification conditions see Figure 6.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -003 version of this data sheet:

1. Introduction was expanded to include product descriptions.
2. Package table was added.
3. Design Considerations added.
4. Test Conditions for I_{IL1} and I_{IH} specifications added to the DC Characteristics.
5. Data Sheet Revision Summary added.

8051AHP
MCS®-51 FAMILY
8-BIT CONTROL-ORIENTED MICROCONTROLLER WITH PROTECTED ROM

- **High Performance HMOS Process**
- **Internal Timers/Event Counters**
- **2-Level Interrupt Priority Structure**
- **32 I/O Lines (Four 8-Bit Ports)**
- **4K Program Memory Space**
- **Protection Feature Protects ROM Parts Against Software Piracy**
- **Boolean Processor**
- **Bit-Addressable RAM**
- **Programmable Full Duplex Serial Channel**
- **111 Instructions (64 Single-Cycle)**
- **4K Data Memory Space***
  *Expandable to 64K
- **Available in 40 Pin Plastic and CERDIP Packages**
  (See Packaging Outlines and Dimensions Order #231369)

The MCS®-51 products are optimized for control applications. Byte-processing and numerical operations on small data structures are facilitated by a variety of fast addressing modes for accessing the internal RAM. The instruction set provides a convenient menu of 8-bit arithmetic instructions, including multiply and divide instructions. Extensive on-chip support is provided for one-bit variables as a separate data type, allowing direct bit manipulation and testing in control and logic systems that require Boolean processing.

MCS-51 HMOS Family Device	Internal Memory		Timers/ Event Counters	Interrupts
	Program	Data		
8051AH	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5
8051AHP	4K x 8 ROM	128 x 8 RAM	2 x 16-Bit	5

The 8051AHP is identical to the 8051AH with the exception of the Protection Feature. To incorporate this Protection Feature, program verification has been disabled and external memory accesses have been limited to 4K.

September 1988
Order Number: 270279-003

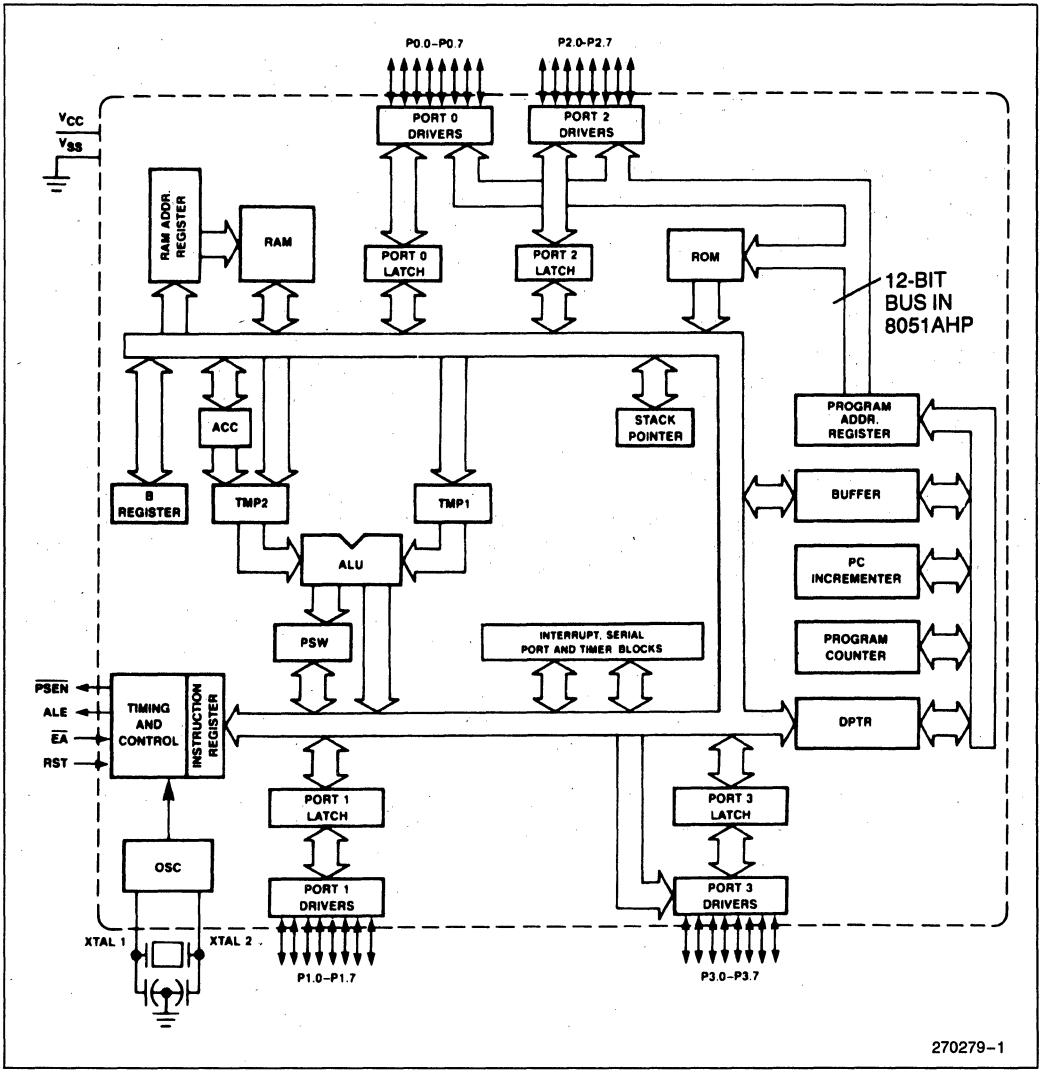


Figure 1. MCS®-51 Block Diagram

PACKAGES

Part	Prefix	Package Type
8051AHP	P	40-Pin Plastic DIP
	D	40-Pin CERDIP

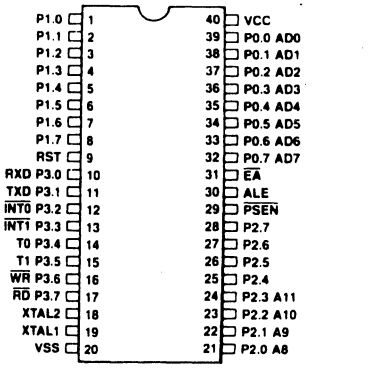


Figure 2. MCS®-51 Connections

PIN DESCRIPTIONS

V_{CC}

Supply voltage.

V_{SS}

Circuit ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs.

Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s and can source and sink 8 LS TTL inputs.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink source 4 LS TTL inputs. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL inputs. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL} on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. Bits P2.4 through P2.7 are forced to 0, effectively limiting external Data and Code space to 4K each in the 8051AHP during external accesses*. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL} on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternative Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

*Protection feature

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external Program Memory.

When the device is executing code from external Program Memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external Data Memory.

$\overline{\text{EA}}$

External Access enable $\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions. $\overline{\text{EA}}$ must be strapped to V_{SS} in order to enable any MCS-51 device to fetch code from external Program memory locations starting at 0000H up to FFFFH.

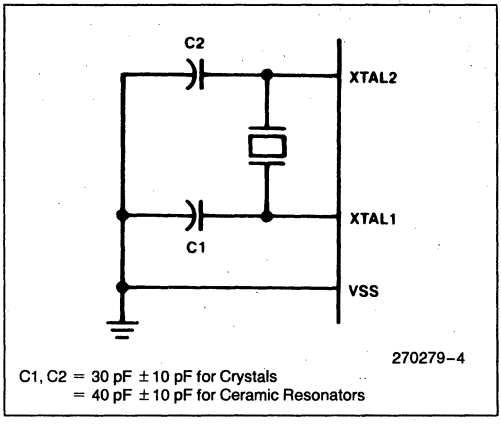


Figure 3. Oscillator Connections

XTAL1

Input to the inverting oscillator amplifier.

XTAL2

Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be grounded, while XTAL2 is driven, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

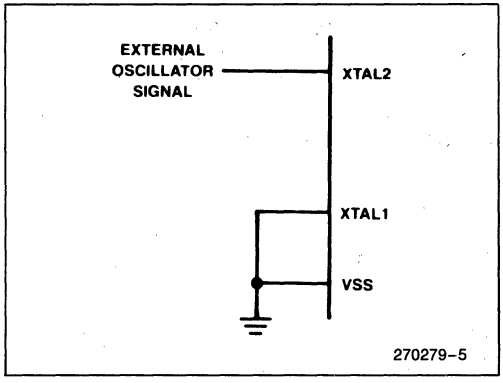


Figure 4. External Drive Configuration

DESIGN CONSIDERATION

The 8051AHP cannot access external Program or Data memory above 4K. This means that the following instructions that use the Data Pointer only read/write data at address locations below 0FFFH:

```
MOVX A, @DPTR
MOVX @DPTR, A
```

When the Data Pointer contains an address above the 4K limit, those locations will not be accessed.

To access Data Memory above 4K, the MOVX, @Ri, A or MOVX A, @Ri instructions must be used.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on $\overline{EA}/V_{PP}$ Pin to V_{SS} ...−0.5V to +21.5V
Voltage on Any Other Pin to V_{SS} ....−0.5V to +7V
Power Dissipation.....1.5W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage (Except XTAL2, RST)	2.0	V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage to XTAL2, RST	2.5	V_{CC} + 0.5	V	XTAL1 = V_{SS}
V_{OL}	Output Low Voltage (Ports 1, 2, 3)*		0.45	V	I_{OL} = 1.6 mA
V_{OL1}	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$)*		0.45	V	I_{OL} = 3.2 mA
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$)	2.4		V	I_{OH} = −80 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4		V	I_{OH} = −400 μA
I_{IL}	Logical 0 Input Current		−500	μA	V_{IN} = 0.45V
I_{IL2}	Logical 0 Input Current (XTAL2)		−3.2	mA	V_{IN} = 0.45V
I_{LI}	Input Leakage Current (Port 0)		±10	μA	0.45 ≤ V_{IN} ≤ V_{CC}
I_{IH}	Input Current to RST to Activate Reset		500	μA	V_{IN} < (V_{CC} − 1.5V)
I_{CC}	Power Supply Current		125	mA	All Outputs Disconnected; $\overline{EA}$ = VCC
CIO	Pin Capacitance		10	pF	Test freq = 1 MHz

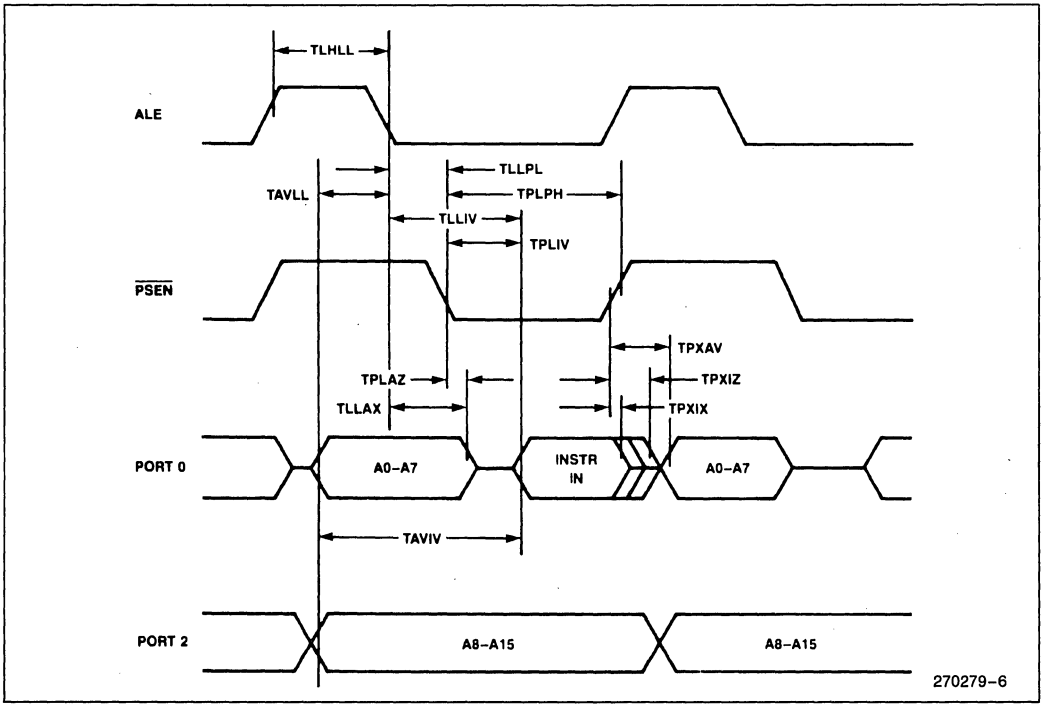
***NOTE:**
Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.

A.C. CHARACTERISTICS

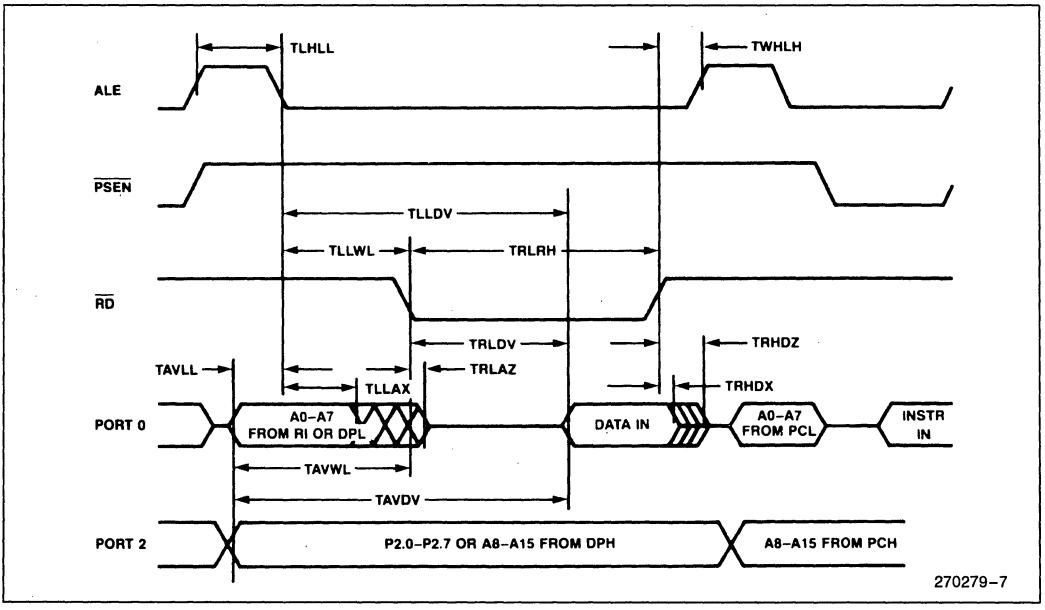
T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V;
Load Capacitance for Port 0, ALE, and PSEN = 100 pF;
Load Capacitance for All Other Outputs = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency			3.5	12.0	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL−40		ns
TLLAX	Address Hold after ALE Low	48		TCLCL−35		ns
TLLIV	ALE Low to Valid Instr In		233		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	58		TCLCL−25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	215		3TCLCL−35		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr In		125		3TCLCL−125	ns
TPXIX	Input Instr Hold after $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float after $\overline{PSEN}$		63		TCLCL−20	ns
TPXAV	$\overline{PSEN}$ to Address Valid	75		TCLCL−8		ns
TAVIV	Address to Valid Instr In		302		5TCLCL−115	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		20		20	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold after $\overline{RD}$	0		0		ns
TRHDZ	Data Float after $\overline{RD}$		97		2TCLCL−70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		TCLCL−60		ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL−150		ns
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL−50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		20		20	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL−40	TCLCL+40	ns

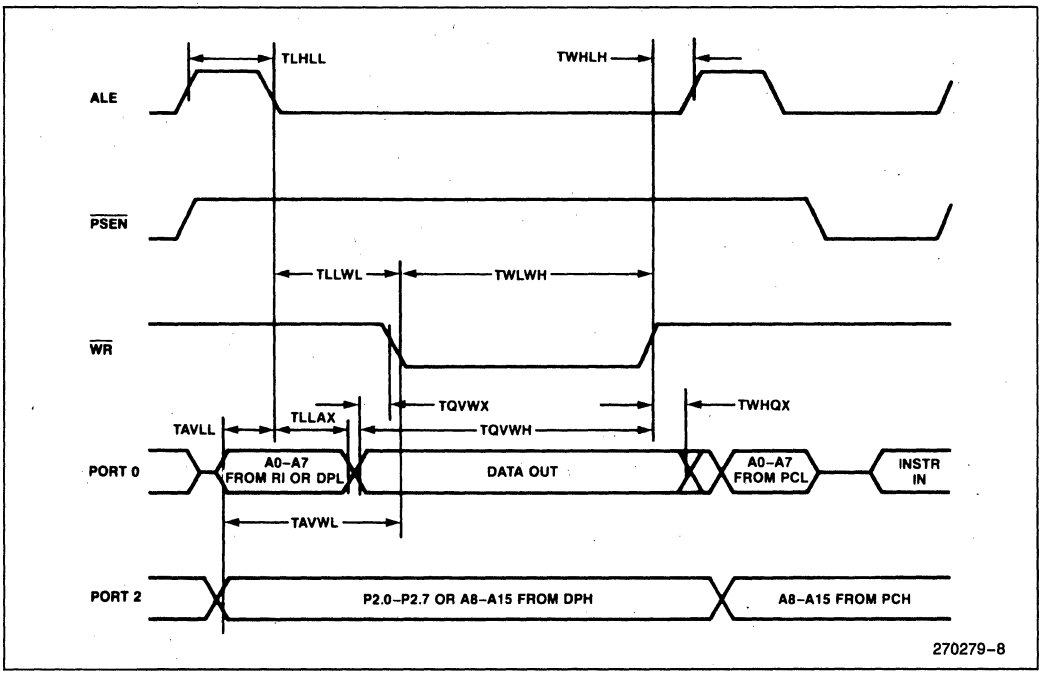
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE

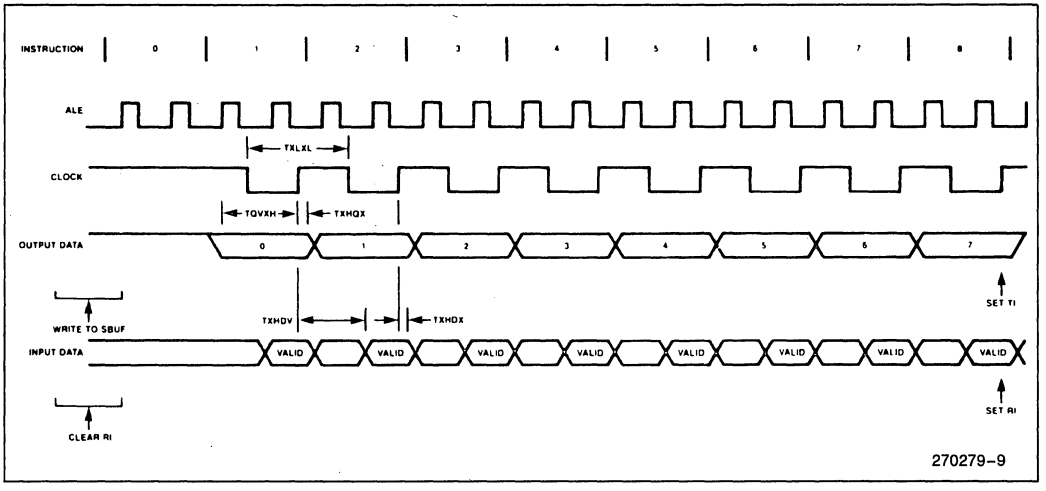


SERIAL PORT TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold after Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns

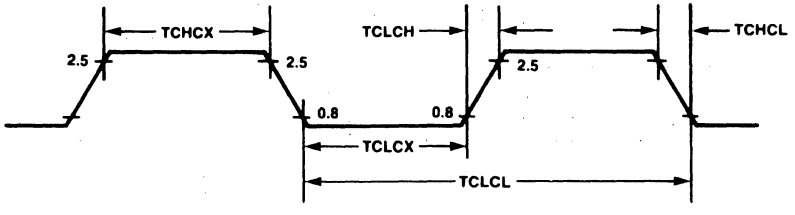
SHIFT REGISTER TIMING WAVEFORMS



EXTERNAL CLOCK DRIVE

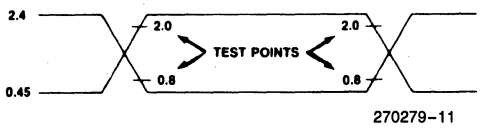
Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	3.5	12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

EXTERNAL CLOCK DRIVE WAVEFORM



270279-10

A.C. TESTING INPUT, OUTPUT WAVEFORM



270279-11

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

Program Verification

The program verification test mode has been eliminated on the 8051AHP. It is not possible to verify the ROM contents using this mode, the way EPROM programmers typically do. Also, the ROM contents cannot be verified by a program executing out of external program memory due to the restricted addressing on the 8051AHP.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -002 version of 8051AHP data sheet:

1. Package Table was added.
2. Added clearer explanation to DESIGN CONSIDERATION.
3. Data Sheet Revision Summary was added.

8031AH/8051AH
8032AH/8052AH
8751H/8751H-8

EXPRESS

■ **Extended Temperature Range** ■ **Burn-In**

The Intel EXPRESS system offers enhancements to the operational specifications of the MCS®-51 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in, and an extended temperature range with or without burn-in.

With the commercial standard temperature range operational characteristics are guaranteed over the temperature range of 0°C to 70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

The optional burn-in is dynamic, for a minimum time of 160 hours at 125°C with V_{CC} = 5.5V ±0.25V, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets.

D.C. CHARACTERISTICS T_A = −40°C to +85°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.75	V	
V_{IH}	Input High Voltage (Except XTAL2, RST)	2.1	V_{CC} + 0.5	V	
I_{CC}	Power Supply Current: 8051AH, 8031AH 8052AH, 8032AH 8751H, 8751H-8		 135 175 265	 mA mA mA	 All Outputs Disconnected; $\overline{EA}$ = V_{CC}
I_{IL2}	Logic 0 Input Current (XTAL2)		−4.0	mA	V_{in} = 0.45V

October 1987
Order Number: 270007-002

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range	Burn-In
P	plastic	commercial	no
D	cerdip	commercial	no
C	ceramic	commercial	no
N	PLCC	commercial	no
R	LCC	commercial	no
TP	plastic	extended	no
TD	cerdip	extended	no
TC	ceramic	extended	no
QP	plastic	commercial	yes
QD	cerdip	commercial	yes
QC	ceramic	commercial	yes
LP	plastic	extended	yes
LD	cerdip	extended	yes
LC	ceramic	extended	yes

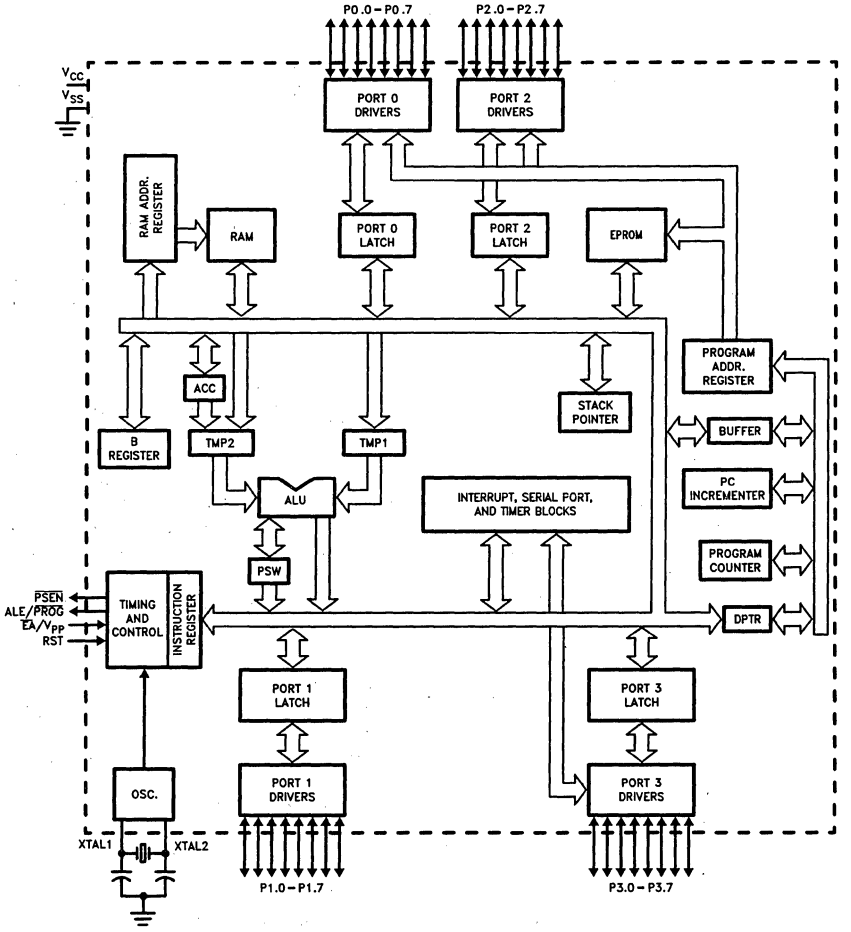
Please note:

- Commercial temperature range is 0°C to 70°C. Extended temperature range is −40°C to +85°C.
- Burn-in is dynamic, for a minimum time of 160 hours at 125°C, V_{CC} = 5.5V ±0.25V, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).
- The following devices are not available in ceramic packages:
  8051AH, 8031AH
  8052AH, 8032AH
- The following devices are not available in extended temperature range:
  8751H, 8751H-8

Examples: P8031AH indicates 8031AH in a plastic package and specified for commercial temperature range, without burn-in. LD8051AH indicates 8051AH in a cerdip package and specified for extended temperature range with burn-in.

8751BH
SINGLE-CHIP 8-BIT MICROCOMPUTER WITH 4K BYTES OF EPROM PROGRAM MEMORY

- Program Memory Lock
- 128 Bytes Data Ram
- Quick Pulse Programming™ Algorithm
- 12.75 Volt Programming Voltage
- Boolean Processor
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- 5 Interrupt Sources
- Programmable Serial Channel
- 64K External Program Memory Space
- 64K External Data Memory Space



270248-1

Figure 1. 8751BH Block Diagram

September 1988
Order Number: 270248-003

PACKAGES

Part	Prefix	Package Type
8751BH	P	40-Pin Plastic DIP
	N	44-Pin PLCC

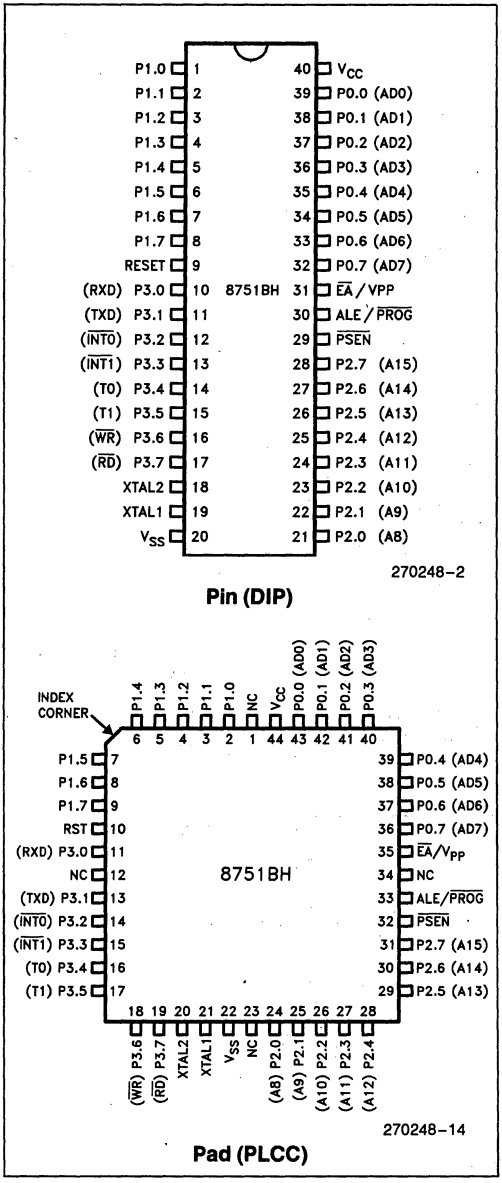


Figure 2. Pin Connections

PIN DESCRIPTIONS

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s, and can source and sink 8 LS TTL inputs.

Port 0 also receives the code bytes during EPROM programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source 4 LS TTL inputs. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during EPROM programming and program verification.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL inputs. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits during EPROM programming and program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS®-51 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during EPROM programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$: Program Store Enable is the Read strobe to External Program Memory.

When the 8751BH is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to External Data Memory.

$\overline{EA}$/V_{PP}: External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the device to fetch code from External Program Memory locations starting at 0000H up to FFFFH. Note, however, that if either of the Lock Bits are programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12.75V programming supply voltage (V_{PP}) during EPROM programming.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Applications Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be grounded, while XTAL2 is driven, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

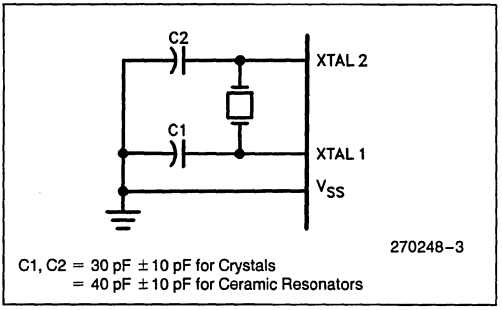


Figure 3. Oscillator Connections

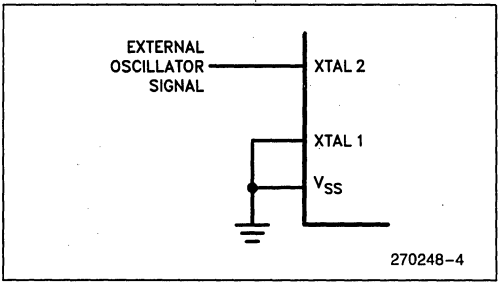


Figure 4. External Clock Drive Configuration

DESIGN CONSIDERATIONS

If an 8751BH is replacing an 8751H in an existing design, the user should carefully compare both data sheets for DC or AC Characteristic differences. Note that the V_{IH} and I_{IH} specifications for the EA pin differ significantly between the 8751H and 8751BH.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on $\overline{EA}$/V_{PP} Pin to V_{SS} ...−0.5V to +13.0V
Voltage on Any Other Pin to V_{SS} ....−0.5V to +7V
Maximum I_{OL} Per I/O Pin .....15 mA
Power Dissipation.....1.5W
(based on PACKAGE heat transfer limitations, not device power consumption)

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V)

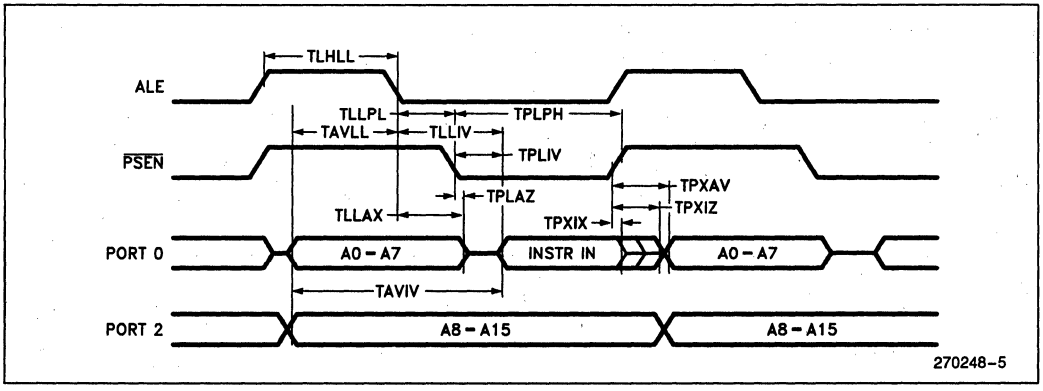
Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage $\overline{EA}$	V_{SS}	0.7	V	
V_{IH}	Input High Voltage (Except XTAL2, RST, $\overline{EA}$)	2.0	V_{CC}+0.5	V	
V_{IH1}	Input High Voltage XTAL2, RST	2.5	V_{CC}+0.5	V	XTAL1 = VSS
V_{IH2}	Input High Voltage to $\overline{EA}$	4.5	5.5	V	
V_{OL}	Output Low Voltage (Note 3) (Ports 1, 2 and 3)		0.45	V	I_{OL} = 1.6 mA (Note 1)
V_{OL1}	Output Low Voltage (Note 3) (Port 0, ALE/$\overline{PROG}$, $\overline{PSEN}$)		0.45	V	I_{OL} = 3.2 mA (Notes 1, 2)
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE/$\overline{PROG}$ and $\overline{PSEN}$)	2.4		V	I_{OH} = −80 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4		V	I_{OH} = −400 μA
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3 and RST)		−1	mA	V_{IN} = 0.45V
I_{IL1}	Logical 0 Input Current ($\overline{EA}$)		−10	mA	V_{IN} = VSS
I_{IL2}	Logical 0 Input Current (XTAL2)		−3.2	mA	V_{IN} = 0.45 V XTAL1 = V_{SS}
I_{LI}	Input Leakage Current (Port 0)		±10	μA	0.45 < V_{IN} < V_{CC}
I_{IH}	Logical 1 Input Current ($\overline{EA}$)		1	mA	4.5V < V_{IN} < 5.5V
I_{IH1}	Input Current to RST to Activate Reset		500	μA	V_{IN} < (V_{CC} − 1.5V)
I_{CC}	Power Supply Current		175	mA	All Outputs Disconnected
C_{IO}	Pin Capacitance		10	pF	Test Freq = 1MHz

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE/$\overline{PROG}$ and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE/$\overline{PROG}$ pin may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. ALE/$\overline{PROG}$ refers to a pin on the 8751BH. ALE refers to a timing signal that is output on the ALE/$\overline{PROG}$ pin.
3. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port -
   Port 0: 26 mA
   Ports 1, 2, and 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

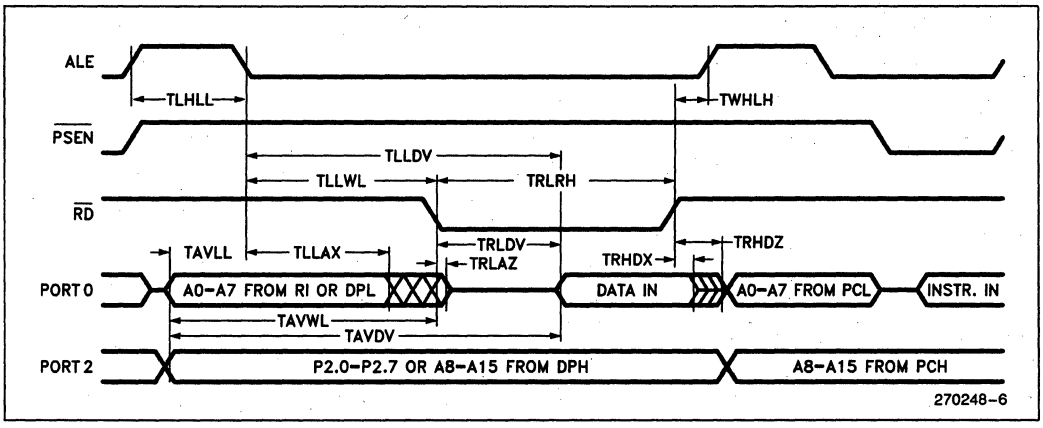
A.C. CHARACTERISTICS (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V); Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for All Other Outputs = 80 pF)

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

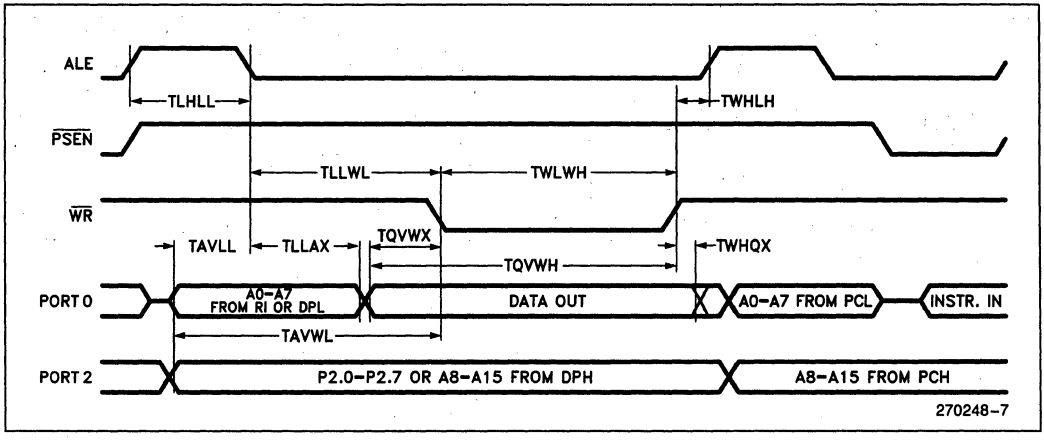
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency			3.5	12.0	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL−40		ns
TLLAX	Address Hold After ALE Low	48		TCLCL−35		ns
TLLIV	ALE Low to Valid Instruction In		233		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	58		TCLCL−25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	215		3TCLCL−35		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instruction In		125		3TCLCL−125	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		63		TCLCL−20	ns
TPXAV	$\overline{PSEN}$ to Address Valid	75		TCLCL−8		ns
TAVIV	Address to Valid Instruction In		302		5TCLCL−115	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		20		20	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL−70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		TCLCL−60		ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL−150		ns
TWHQX	Data Held After $\overline{WR}$	33		TCLCL−50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL−40	TCLCL+40	ns



External Program Memory Read Cycle



External Data Memory Read Cycle

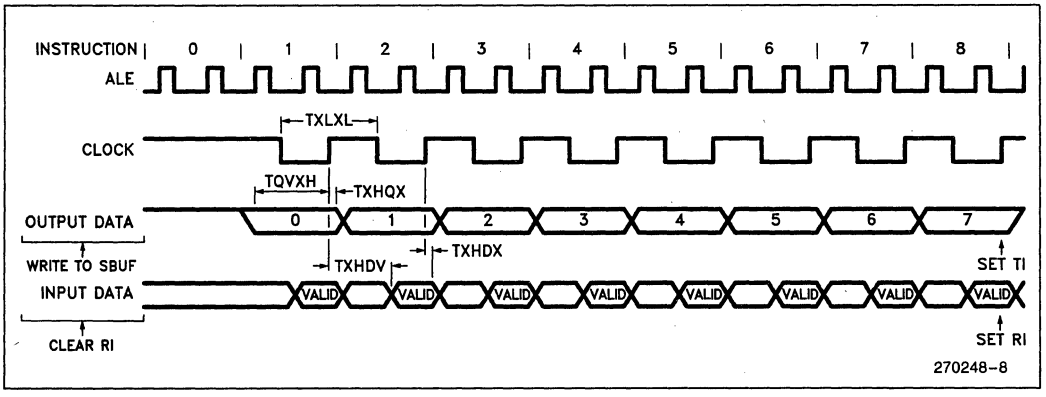


External Data Memory Write Cycle

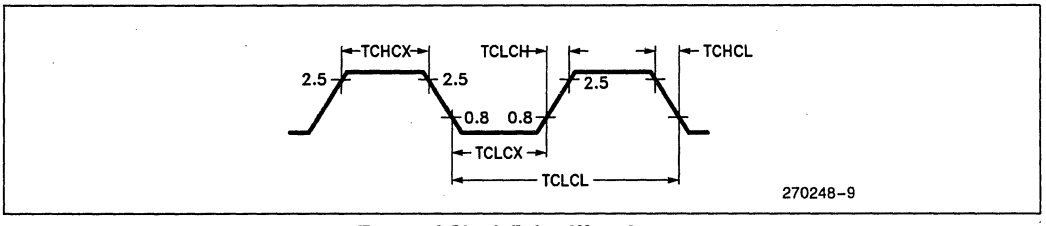
SERIAL PORT TIMING — SHIFT REGISTER MODE

TEST CONDITIONS (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF)

Symbol	Parameter	12MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns



Shift Register Mode Timing Waveforms

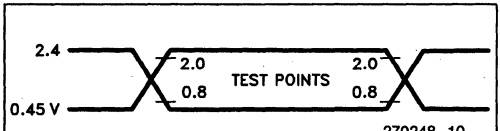


External Clock Drive Waveforms

EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	3.5	12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

AC TESTING INPUT/OUTPUT WAVEFORMS



AC inputs during testing are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

EPROM CHARACTERISTICS

Programming the EPROM

To be programmed, the part must be running with a 4 to 6 MHz oscillator. (The reason the oscillator needs to be running is that the internal bus is being used to transfer address and program data to appropriate internal registers.) The address of an EPROM location to be programmed is applied to Port 1 and pins P2.0 - P2.3 of Port 2, while the code byte to be programmed into that location is applied to Port 0. The other Port 2 and 3 pins, and RST, $\overline{PSEN}$, and $\overline{EA}/V_{PP}$ should be held at the "Program" levels indicated in Table 1. ALE/$\overline{PROG}$ is pulsed low to program the code byte into the addressed EPROM location. The setup is shown in Figure 5.

Normally $\overline{EA}/V_{PP}$ is held at a logic high until just before ALE/$\overline{PROG}$ is to be pulsed. Then $\overline{EA}/V_{PP}$ is raised to V_{PP}, ALE/$\overline{PROG}$ is pulsed low, and then $\overline{EA}/V_{PP}$ is returned to a valid high voltage. The voltage on the $\overline{EA}/V_{PP}$ pin must be at the valid $\overline{EA}/V_{PP}$ high level before a verify is attempted. Waveforms and detailed timing specifications are shown in later sections of this data sheet.

Note that the $\overline{EA}/V_{PP}$ pin must not be allowed to go above the maximum specified V_{PP} level for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The V_{PP} source should be well regulated and free of glitches.

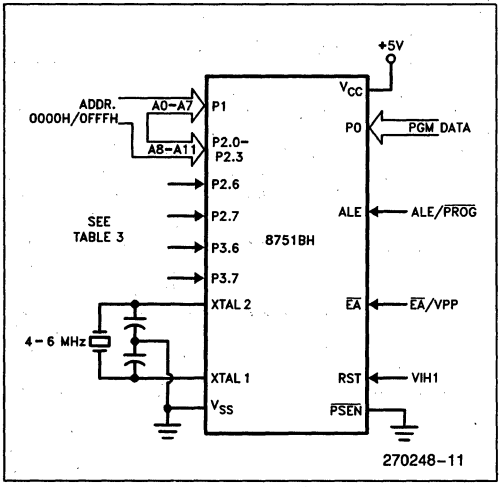


Figure 5. Programming the EPROM

Table 1. EPROM Programming Modes

MODE	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.7	P2.6	P3.6	P3.7
Program Code Data	1	0	0*	V_{PP}	1	0	1	1
Verify Code Data	1	0	1	1	0	0	1	1
Program Encryption Table Use Addresses 0-1FH	1	0	0*	V_{PP}	1	0	0	1
Program Lock x=1	1	0	0*	V_{PP}	1	1	1	1
Bits (LBx) x=2	1	0	0*	V_{PP}	1	1	0	0
Read Signature	1	0	1	1	0	0	0	0

NOTES:
"1" = Valid high for that pin
"0" = Valid low for that pin
"V_{PP}" = +12.75V ±0.25V
* ALE/$\overline{PROG}$ is pulsed low for 100 uS for programming. (Quick-Pulse Programming™)

QUICK-PULSE PROGRAMMING™ ALGORITHM

The 8751BH can be programmed using the Quick-Pulse Programming Algorithm for microcontrollers. The features of the new programming method are a lower V_{PP} (12.75 volts as compared to 21 volts) and a shorter programming pulse. It is possible to program the entire 4K Bytes of EPROM memory in less than 13 seconds with this algorithm

To program the part using the new algorithm, V_{PP} must be 12.75 ±0.25 Volts. ALE/$\overline{PROG}$ is pulsed low for 100 μseconds, 25 times. Then, the byte just programmed may be verified. After programming, the entire array should be verified. The Program Lock features are programmed using the same method, but with the setup as shown in Table 1. The only difference in programming Lock features is that the Lock features cannot be directly verified. Instead, verification of programming is by observing that their features are enabled.

PROGRAM VERIFICATION

If the Lock Bits have not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired, either during or after the programming operation. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0 - P2.3. The other pins should be held at the "Verify" levels indicated in Table 1. The contents of the addressed location will come out on Port 0. External pullups are required on Port 0 for this operation. (If the Encryption Array in the EPROM has been programmed, the data present at Port 0 will be Code Data XNOR Encryption Data. The user must know the Encryption Array contents to manually "unencrypt" the data during verify.)

The setup, which is shown in Figure 6, is the same as for programming the EPROM except that pin P2.7 is held at a logic low, or may be used as an active low read strobe.

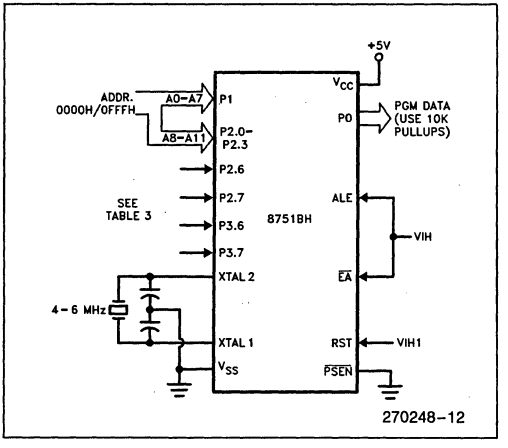


Figure 6. Verifying the EPROM

PROGRAM MEMORY LOCK

The two-level Program Lock system consists of 2 Lock bits and a 32-byte Encryption Array which are used to protect the program memory against software piracy.

Encryption Array

Within the EPROM array are 32 bytes of Encryption Array that are initially unprogrammed (all 1s). Every time that a byte is addressed during a verify, 5 address lines are used to select a byte of the Encryption Array. This byte is then exclusive-NORed (XNOR) with the code byte, creating an Encrypted Verify byte. The algorithm, with the array in the unprogrammed state (all 1s), will return the code in its original, unmodified form.

It is recommended that whenever the Encryption Array is used, at least one of the Lock Bits be programmed as well.

Lock Bits

Also included in the EPROM Program Lock scheme are two Lock Bits which function as shown in Table 2.

Table 2. Lock Bits and their Features

Lock Bits		Logic Enabled
LB1	LB2	
U	U	Minimum Program Lock features enabled. (Code Verify will still be encrypted by the Encryption Array)
P	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the EPROM is disabled
P	P	Same as above, but Verify is also disabled
U	P	Reserved for Future Definition

P = Programmed
U = Unprogrammed

To ensure proper functionality of the chip, the internally latched value of the $\overline{EA}$ pin must agree with its external state.

ERASURE CHARACTERISTICS

This device is in a plastic package without a window and, therefore, cannot be erased.

Reading the Signature Bytes

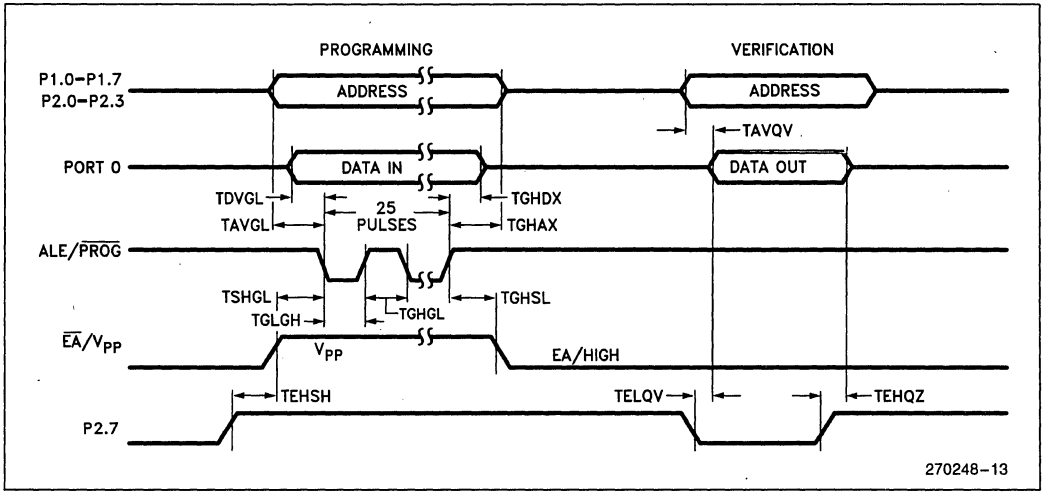
The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. the values returned are:

(030H) = 89H indicates manufactured by Intel
(031H) = 51H indicates 8751BH

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

(T_A = 21°C to 27°C, V_{CC} = 5.0V ±10%, V_{SS} = 0V)

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	12.5	13.0	V
IPP	Programming Supply Current		50	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$ Low	48TCLCL		
TGHAX	Address Hold After $\overline{PROG}$	48TCLCL		
TDVGL	Data Setup to $\overline{PROG}$ Low	48TCLCL		
TGHDX	Data Hold After $\overline{PROG}$	48TCLCL		
TEHSH	P2.7 ($\overline{ENABLE}$) High to V_{PP}	48TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$ Low	10		µsec
TGHSL	V_{PP} Hold After $\overline{PROG}$	10		µsec
TGLGH	$\overline{PROG}$ Width	90	110	µsec
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{ENABLE}$ Low to Data Valid		48TCLCL	
TEHQZ	Data Float After $\overline{ENABLE}$	0	48TCLCL	
TGHGL	$\overline{PROG}$ High to $\overline{PROG}$ Low	10		µsec



EPROM Programming and Verification Waveforms

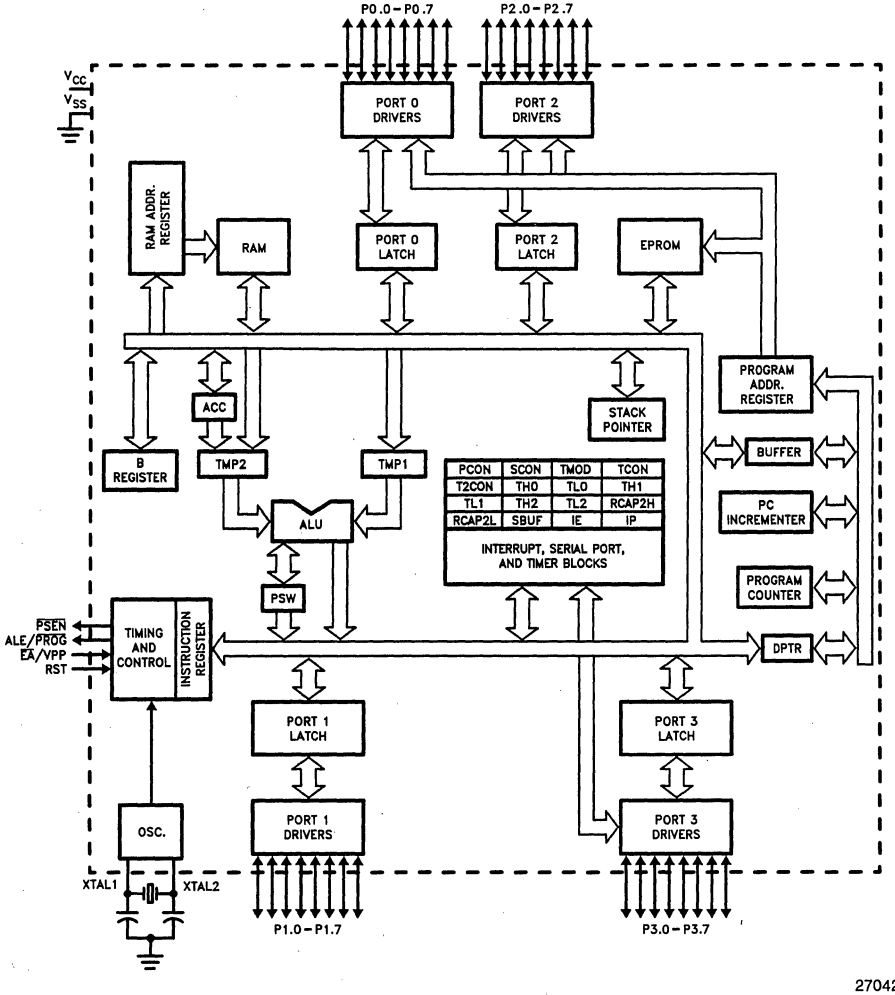
DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -002 version of 8751BH data sheet:

1. Status went from ADVANCE INFORMATION to PRELIMINARY.
2. Package Table was added.
3. PLCC pin connections shown.
4. Design Considerations section replaced with reference to previous designs using the 8751H.
5. Note 3 on maximum current specification was added to DC Characteristics.
6. Table 1 updated to show Read Signature Mode.
7. ERASING THE EPROM paragraph deleted.
8. ERASURE CHARACTERISTICS section changed to indicate plastic packages only.
9. Signature Bytes added.
10. Data Sheet Revision Summary was added.

8752BH
SINGLE-CHIP 8-BIT MICROCOMPUTER WITH 8K BYTES OF EPROM PROGRAM MEMORY

- 2-Bit Program Memory Lock
- 256 Bytes Data Ram
- Quick Pulse Programming™ Algorithm
- 12.75 Volt Programming Voltage
- Boolean Processor
- 32 Programmable I/O Lines
- Three 16-Bit Timer/Counters
- 6 Interrupt Sources
- Programmable Serial Channel
- Separate Transmit/Receive Baud Rate Capability
- 64K External Program Memory Space
- 64K External Data Memory Space



270429–1

Figure 1. Block Diagram

September 1988
Order Number: 270429-002

PACKAGES

Part	Prefix	Package Type
8752BH	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-Pin PLCC
	R	44-Pin LCC

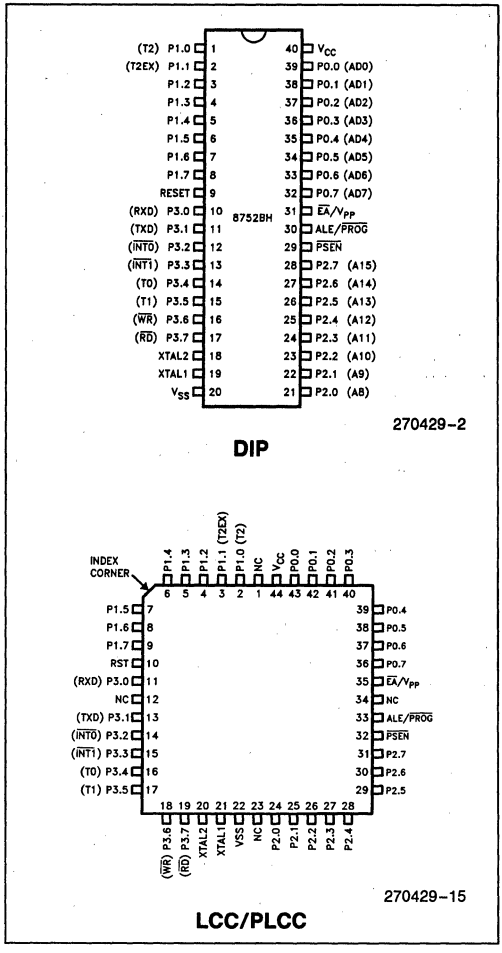


Figure 2. Pin Connections

PIN DESCRIPTION

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s, and can source and sink 8 LS TTL inputs.

Port 0 also receives the code bytes during EPROM programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source 4 LS TTL inputs. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during EPROM programming and program verification.

In addition, P1.0 and P1.1 serve the functions of the following special features of the MCS®-51 Family:

Port Pin	Alternate Function
P1.0	T2 (Timer/Counter 2 External Input)
P1.1	T2EX (Timer/Counter 2 Capture/Reload Trigger)

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL inputs. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits during EPROM programming and program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS®-51 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during EPROM programming on the 8752BH.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$: Program Store Enable is the Read strobe to External Program Memory.

When the device is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to External Data Memory.

$\overline{EA}$/V_{PP}: External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the device to fetch code from External Program Memory locations starting at 0000H up to FFFFH. Note, however, that if either of the Lock Bits are programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12.75V programming supply voltage (V_{PP}) during EPROM programming.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Applications Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be grounded, while XTAL2 is driven, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

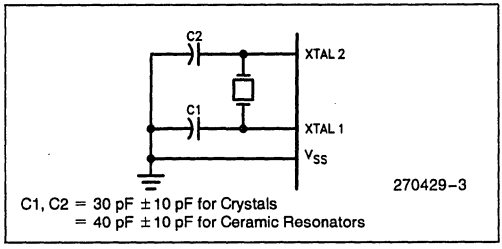


Figure 3. Oscillator Connections

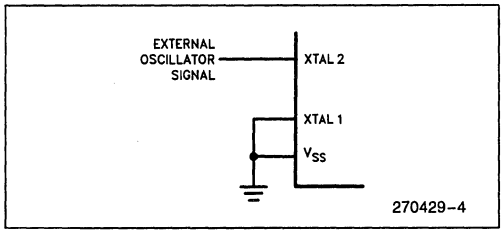


Figure 4. External Clock Drive Configuration

DESIGN CONSIDERATIONS

Exposure to light when the 8752BH is in operation may cause logic errors. For this reason, it is suggested that an opaque label be placed over the window of the 8752BH when the die is exposed to ambient light.

Due to a timing problem in the Timer/Counter 2 interrupt circuitry, the device may vector to location 03H (External Interrupt 0 vector address). It happens when a low priority interrupt has been in progress for either 1 or 2 machine cycles and Timer/Counter 2 generates a priority 1 interrupt. Therefore, Timer/Counter 2 should only be assigned priority level 0.

If an 8752BH is replacing an 8751H in an existing design, the user should carefully compare both data sheets for DC or AC characteristic differences. Note that the V_{IH} and I_{IH} specifications for the EA pin differ significantly between the 8751H and 8752BH.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on $\overline{EA}/V_{PP}$ Pin to V_{SS} ...−0.5V to +13.0V
Voltage on Any Other Pin to V_{SS} ....−0.5V to +7V
Maximum I_{OL} Per I/O Pin .....15 mA
Power Dissipation.....1.5W
(based on PACKAGE heat transfer limitations, not device power consumption)

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V)

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage $\overline{EA}$	V_{SS}	0.7	V	
V_{IH}	Input High Voltage (Except XTAL2, RST, $\overline{EA}$)	2.0	V_{CC}+0.5	V	
V_{IH1}	Input High Voltage XTAL2, RST	2.5	V_{CC}+0.5	V	XTAL1 = V_{SS}
V_{IH2}	Input High Voltage to $\overline{EA}$	4.5	5.5	V	
V_{OL}	Output Low Voltage (Note 3) (Ports 1, 2 and 3)		0.45	V	I_{OL} = 1.6 mA (Note 1)
V_{OL1}	Output Low Voltage (Note 3) (Port 0, ALE/$\overline{PROG}$, $\overline{PSEN}$)		0.45	V	I_{OL} = 3.2 mA (Note 1, 2)
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE/$\overline{PROG}$ and $\overline{PSEN}$)	2.4		V	I_{OH} = −80 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4		V	I_{OH} = −400 μA
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3 and RST)		−500	μA	V_{IN} = 0.45V
I_{IL1}	Logical 0 Input Current ($\overline{EA}$)	−10 500		mA μA	V_{IN} = V_{SS}
I_{IL2}	Logical 0 Input Current (XTAL2)		−3.2	mA	V_{IN} = 0.45V XTAL1 = V_{SS}
I_{LI}	Input Leakage Current (Port 0)		±10	μA	0.45 < V_{IN} < V_{CC}
I_{IH}	Logical 1 Input Current ($\overline{EA}$)		1	mA	4.5V < V_{IN} < 5.5V
I_{IH1}	Input Current to RST to activate Reset		500	μA	V_{IN} < (V_{CC} − 1.5V)
I_{CC}	Power Supply Current		175	mA	All Outputs Disconnected
C_{IO}	Pin Capacitance		10	pF	Test freq = 1 MHz

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE/$\overline{PROG}$ and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE/$\overline{PROG}$ pin may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. ALE/$\overline{PROG}$ refers to a pin on the device. ALE refers to a timing signal that is output on the ALE/$\overline{PROG}$ pin.
3. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port-
   Port 0: 26 mA
   Ports 1, 2, and 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A:Address
C:Clock
D:Input Data
H:Logic level HIGH
I:Instruction (program memory contents)
L:Logic level LOW, or ALE
P:$\overline{PSEN}$
Q:Output data
R:$\overline{RD}$ signal
T:Time
V:Valid
W:$\overline{WR}$ signal
X:No longer a valid logic level
Z:Float

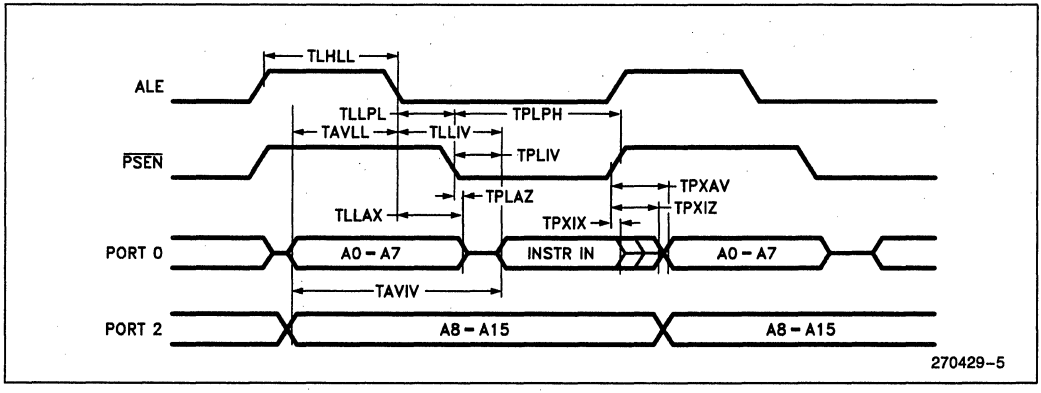
For example,

TAVLL = Time from Address Valid to ALE Low.
TLLPL = Time from ALE Low to PSEN Low.

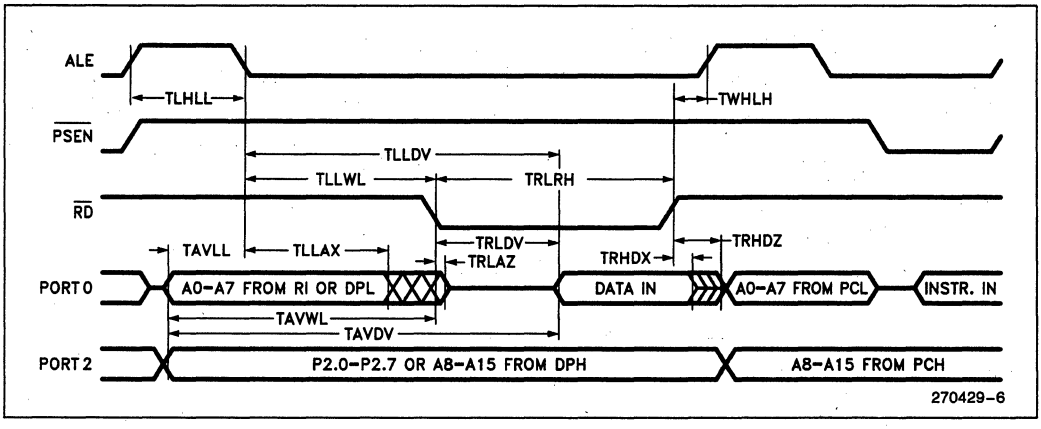
A.C. CHARACTERISTICS (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V); Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for All Other Outputs = 80 pF)

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

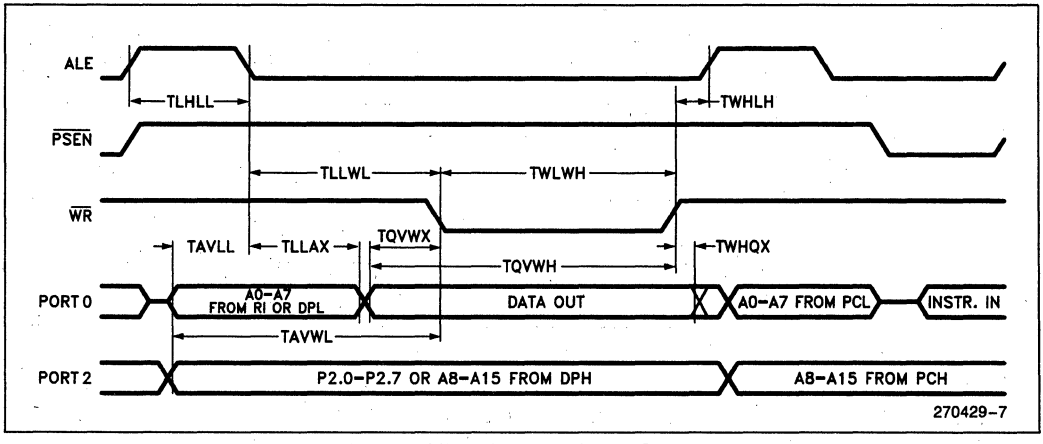
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency			3.5	12.0	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL−40		ns
TLLAX	Address Hold After ALE Low	48		TCLCL−35		ns
TLLIV	ALE Low to Valid Instruction In		233		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	58		TCLCL−25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	215		3TCLCL−35		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instruction In		125		3TCLCL−125	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		63		TCLCL−20	ns
TPXAV	$\overline{PSEN}$ to Address Valid	75		TCLCL−8		ns
TAVIV	Address to Valid Instruction In		302		5TCLCL−115	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		20		20	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL−70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		TCLCL−60		ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL−150		ns
TWHQX	Data Held After $\overline{WR}$	33		TCLCL−50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL−40	TCLCL+40	ns



External Program Memory Read Cycle



External Data Memory Read Cycle

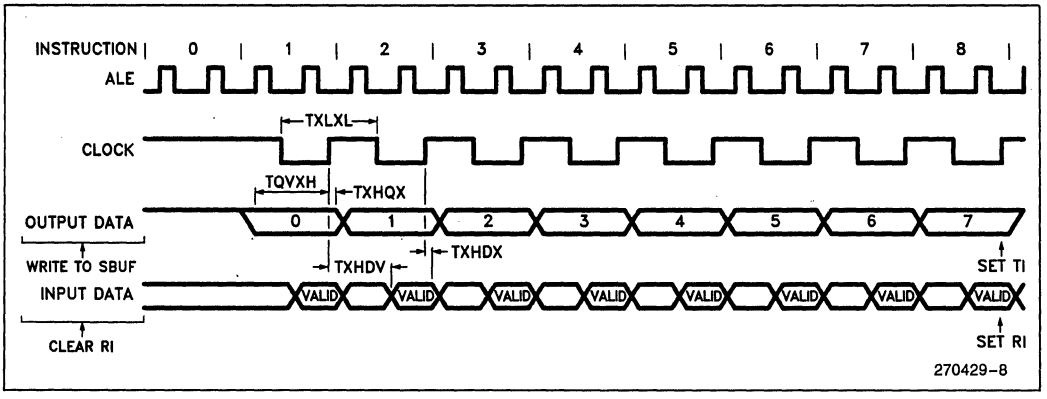


External Data Memory Write Cycle

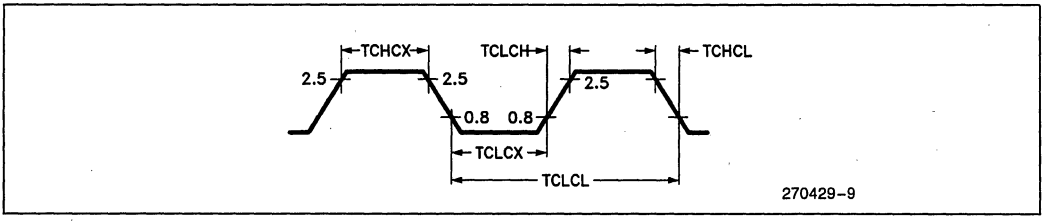
SERIAL PORT TIMING—SHIFT REGISTER MODE

TEST CONDITIONS T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns



Shift Register Mode Timing Waveforms

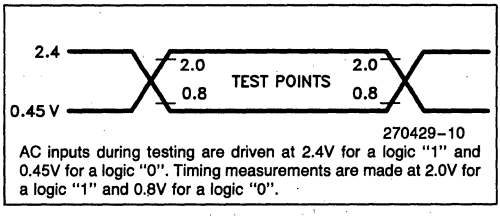


External Clock Drive Waveforms

EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	3.5	12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

A.C. TESTING INPUT/OUTPUT WAVEFORMS



AC inputs during testing are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

EPROM CHARACTERISTICS

Table 1 shows the logic levels for programming the Program Memory, the Encryption Table, and the Lock Bits and for reading the signature bytes.

Programming the EPROM

To be programmed, the 8752BH must be running with a 4 to 6 MHz oscillator. (The reason the oscillator needs to be running is that the internal bus is being used to transfer address and program data to appropriate internal registers.) The address of an EPROM location to be programmed is applied to Port 1 and pins P2.0 - P2.4 of Port 2, while the code byte to be programmed into that location is applied to Port 0. The other Port 2 and 3 pins, and RST, $\overline{PSEN}$, and $\overline{EA}/V_{PP}$ should be held at the "Program" levels indicated in Table 1. ALE/$\overline{PROG}$ is pulsed low to program the code byte into the addressed EPROM location. The setup is shown in Figure 5.

Normally $\overline{EA}/V_{PP}$ is held at a logic high until just before ALE/$\overline{PROG}$ is to be pulsed. Then $\overline{EA}/V_{PP}$ is raised to V_{PP}, ALE/$\overline{PROG}$ is pulsed low, and then $\overline{EA}/V_{PP}$ is returned to a valid high voltage. The voltage on the $\overline{EA}/V_{PP}$ pin must be at the valid $\overline{EA}/V_{PP}$ high level before a verify is attempted. Waveforms and detailed timing specifications are shown in later sections of this data sheet.

Note that the $\overline{EA}/V_{PP}$ pin must not be allowed to go above the maximum specified V_{PP} level for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The V_{PP} source should be well regulated and free of glitches.

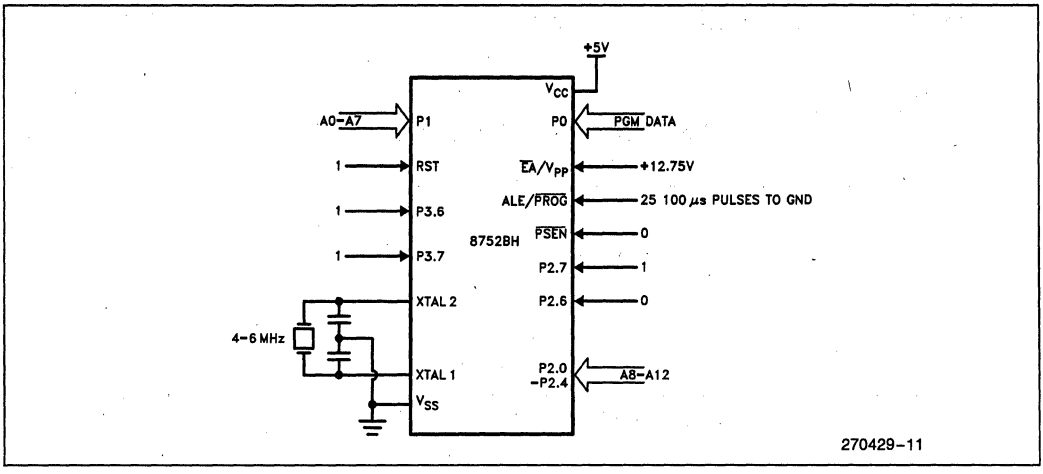


Figure 5. Programming the EPROM

Table 1. EPROM Programming Modes

MODE	RST	PSEN	ALE/PROG	EA/VPP	P2.7	P2.6	P3.6	P3.7
Program Code Data	1	0	0*	VPP	1	0	1	1
Verify Code Data	1	0	1	1	0	0	1	1
Program Encryption Table Use Addresses 0-1FH	1	0	0*	VPP	1	0	0	1
Program Lock x=1	1	0	0*	VPP	1	1	1	1
Bits (LBx) x=2	1	0	0*	VPP	1	1	0	0
Read Signature	1	0	1	1	0	0	0	0

NOTES:
"1" = Valid high for that pin
"0" = Valid low for that pin
"VPP" = +12.75V ±0.25V
*ALE/$\overline{PROG}$ is pulsed low for 100 uS for programming. (Quick-Pulse Programming™)

QUICK-PULSE PROGRAMMING™ ALGORITHM

The 8752BH can be programmed using the Quick-Pulse Programming™ Algorithm for microcontrollers. The features of the new programming method are a lower VPP (12.75 volts as compared to 21 volts) and a shorter programming pulse. It is possible to program the entire 8K Bytes of EPROM memory in less than 25 seconds with this algorithm!

To program the part using the new algorithm, VPP must be 12.75 ±0.25 Volts. ALE/$\overline{PROG}$ is pulsed low for 100 μseconds, 25 times as shown in Figure 6. Then, the byte just programmed may be verified. After programming, the entire array should be verified. The Program Lock features are programmed using the same method, but with the setup as shown in Table 1. The only difference in programming Lock features is that the Lock features cannot be directly verified. Instead, verification of programming is by observing that their features are enabled.

PROGRAM VERIFICATION

If the Lock Bits have not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired, either during or after the programming operation. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0 - P2.4. The other pins should be held at the "Verify" levels indicated in Table 1. The contents of the addressed location will come out on Port 0. External pullups are required on Port 0 for this operation. (If the Encryption Array in the EPROM has been programmed, the data present at Port 0 will be Code Data XNOR Encryption Data. The user must know the Encryption Array contents to manually "unencrypt" the data during verify.)

The setup, which is shown in Figure 7, is the same as for programming the EPROM except that pin P2.7 is held at a logic low, or may be used as an active low read strobe.

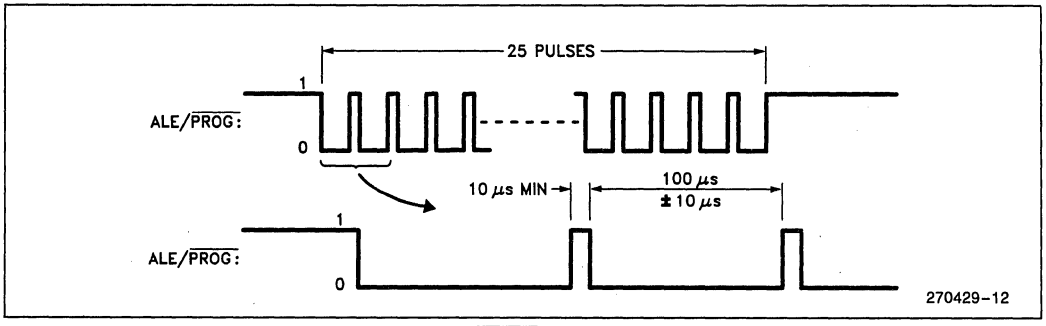


Figure 6. $\overline{PROG}$ Waveforms

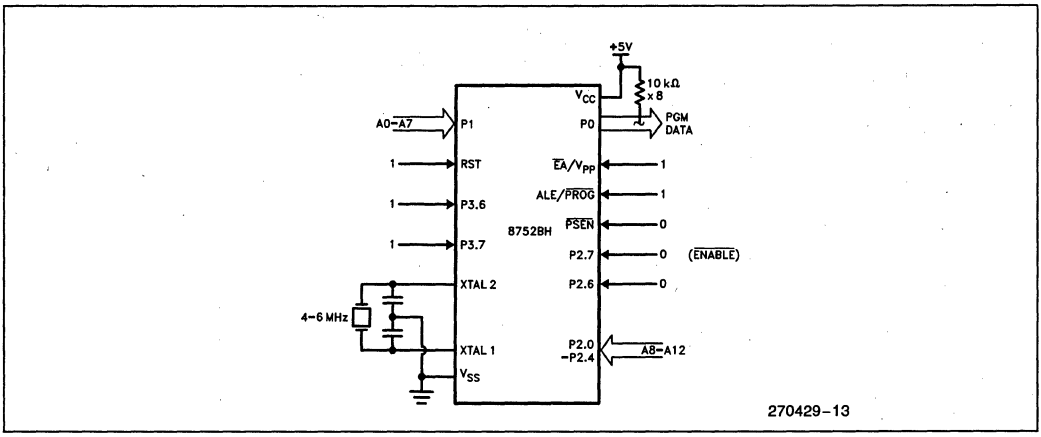


Figure 7. Verifying the EPROM

PROGRAM MEMORY LOCK

The two-level Program Lock system consists of 2 Lock bits and a 32-byte Encryption Array which are used to protect the program memory against software piracy.

ENCRYPTION ARRAY

Within the EPROM array are 32 bytes of Encryption Array that are initially unprogrammed (all 1s). Every time that a byte is addressed during a verify, 5 address lines are used to select a byte of the Encryption Array. This byte is then exclusive-NORed (XNOR) with the code byte, creating an Encrypted Verify byte. The algorithm, with the array in the unprogrammed state (all 1s), will return the code in its original, unmodified form.

It is recommended that whenever the Encryption Array is used, at least one of the Lock Bits be programmed as well.

LOCK BITS

Also included in the EPROM Program Lock scheme are two Lock Bits which function as shown in Table 2.

Erasing the EPROM also erases the Encryption Array and the Lock Bits, returning the part to full unlocked functionality.

To ensure proper functionality of the chip, the internally latched value of the $\overline{EA}$ pin must agree with its external state.

Table 2. Lock Bits and their Features

Lock Bits		Logic Enabled
LB1	LB2	
U	U	Minimum Program Lock features enabled. (Code Verify will still be encrypted by the Encryption Array)
P	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the EPROM is disabled
P	P	Same as above, but Verify is also disabled
U	P	Reserved for Future Definition

P = Programmed
U = Unprogrammed

READING THE SIGNATURE BYTES

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

(030H) = 89H indicates manufactured by Intel

(031H) = 52H indicates 8752BH

ERASURE CHARACTERISTICS

Erasure of the EPROM begins to occur when the 8752BH is exposed to light with wavelengths shorter than approximately 4,000 Angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room-level fluorescent lighting) could cause inadvertent erasure. If an application subjects the device to this type of exposure, it is suggested that an opaque label be placed over the window.

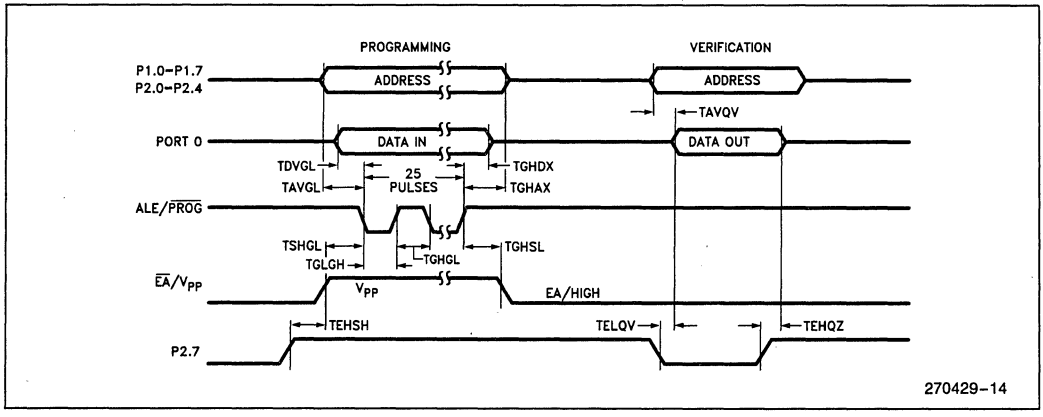
The recommended erasure procedure is exposure to ultraviolet light (at 2537 Angstroms) to an integrated dose of at lease 15 W-sec/cm. Exposing the EPROM to an ultraviolet lamp of 12,000 μW/cm rating for 30 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

(T_A = 21°C to 27°C, V_{CC} = 5.0V ±10%, V_{SS} = 0V)

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	12.5	13.0	V
I_{PP}	Programming Supply Current		50	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$ Low	48TCLCL		
TGHAX	Address Hold After $\overline{PROG}$	48TCLCL		
TDVGL	Data Setup to $\overline{PROG}$ Low	48TCLCL		
TGHDX	Data Hold After $\overline{PROG}$	48TCLCL		
TEHSH	P2.7 ($\overline{ENABLE}$) High to V_{PP}	48TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$ Low	10		μs
TGHSL	V_{PP} Hold After $\overline{PROG}$	10		μs
TGLGH	$\overline{PROG}$ Width	90	110	μs
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{ENABLE}$ Low to Data Valid		48TCLCL	
TEHQZ	Data Float After $\overline{ENABLE}$	0	48TCLCL	
TGHGL	$\overline{PROG}$ High to $\overline{PROG}$ Low	10		μs



EPROM Programming and Verification Waveforms

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -002 version of the 8752BH data sheet.

1. PLCC pin connection diagram was added.
2. Package table was added.
3. Timer/Counter 2 Design Consideration was added.
4. Design Consideration was added referring to previous designs using the 8751H.
5. Note 3 was added to DC Characteristics to explain the maximum current specification.
6. Signature Byte was corrected.
7. Data Sheet Revision Summary was added.

8752BH

EXPRESS

■ **Extended Temperature Range** ■ **Burn-In**

The Intel EXPRESS system offers enhancements to the operational specifications of the MCS®-51 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in, and an extended temperature range with or without burn-in.

With the commercial standard temperature range operational characteristics are guaranteed over the temperature range of 0°C to 70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

The optional burn-in is dynamic, for a minimum time of 160 hours at 125°C with V_{CC} = 5.5V ±0.25V, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets.

D.C. CHARACTERISTICS T_A = −40°C to +85°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IH}	Input High Voltage (Except XTAL2, RST)	2.1	V_{CC} + 0.5	V	

September 1988
Order Number: 270650-001

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range	Burn-In
P	plastic	commercial	no
D	cerdip	commercial	no
N	PLCC	commercial	no
R	LCC	commercial	no
TD	cerdip	extended	no
QP	plastic	commercial	yes
QD	cerdip	commercial	yes
LD	cerdip	extended	yes

Please note:

- Commercial temperature range is 0°C to 70°C. Extended temperature range is −40°C to +85°C.
- Burn-in is dynamic, for a minimum time of 160 hours at 125°C, V_{CC} = 5.5V ±0.25V, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).

Examples: N8752BH indicates 8752BH in a PLCC package and specified for commercial temperature range, without burn-in. LD8752BH indicates 8752BH in a cerdip package and specified for extended temperature range with burn-in.

80C51BH/80C51BH-1/80C51BH-2 CHMOS SINGLE-CHIP 8-BIT MICROCOMPUTER WITH FACTORY MASK-PROGRAMMABLE ROM

80C31BH/80C31BH-1/80C31BH-2 CHMOS SINGLE-CHIP 8-BIT CONTROL-ORIENTED CPU WITH RAM AND I/O

80C51BH/80C31BH—3.5 to 12 MHz, V_{CC} = 5V ±20%
80C51BH-1/80C31BH-1—3.5 to 16 MHz, V_{CC} = 5V ±20%
80C51BH-2/80C31BH-2—0.5 to 12 MHz, V_{CC} = 5V ±20%

- **Power Control Modes**
- **128 x 8-Bit RAM**
- **32 Programmable I/O Lines**
- **Two 16-Bit Timer/Counters**
- **64K Program Memory Space**
- **High Performance CHMOS Process**
- **Boolean Processor**
- **5 Interrupt Sources**
- **Programmable Serial Port**
- **64K Data Memory Space**

The MCS®-51 CHMOS products are fabricated on Intel's CHMOS III process and are functionally compatible with the standard MCS-51 HMOS and EPROM products. CHMOS III is a technology which combines the high speed and density characteristics of HMOS with the low power attributes of CHMOS. This combination expands the effectiveness of the powerful MCS-51 architecture and instruction set.

Like the MCS-51 HMOS versions, the MCS-51 CHMOS products have the following features: 4K byte of ROM (80C51BH/80C51BH-1/80C51BH-2 only); 128 bytes of RAM; 32 I/O lines; two 16-bit timer/counters; a five-source two-level interrupt structure; a full duplex serial port; and on-chip oscillator and clock circuitry. In addition, the MCS-51 CHMOS products have two software selectable modes of reduced activity for further power reduction—Idle and Power Down.

The Idle mode freezes the CPU while allowing the RAM, timer/counters serial port and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

October 1988
Order Number: 270064-006

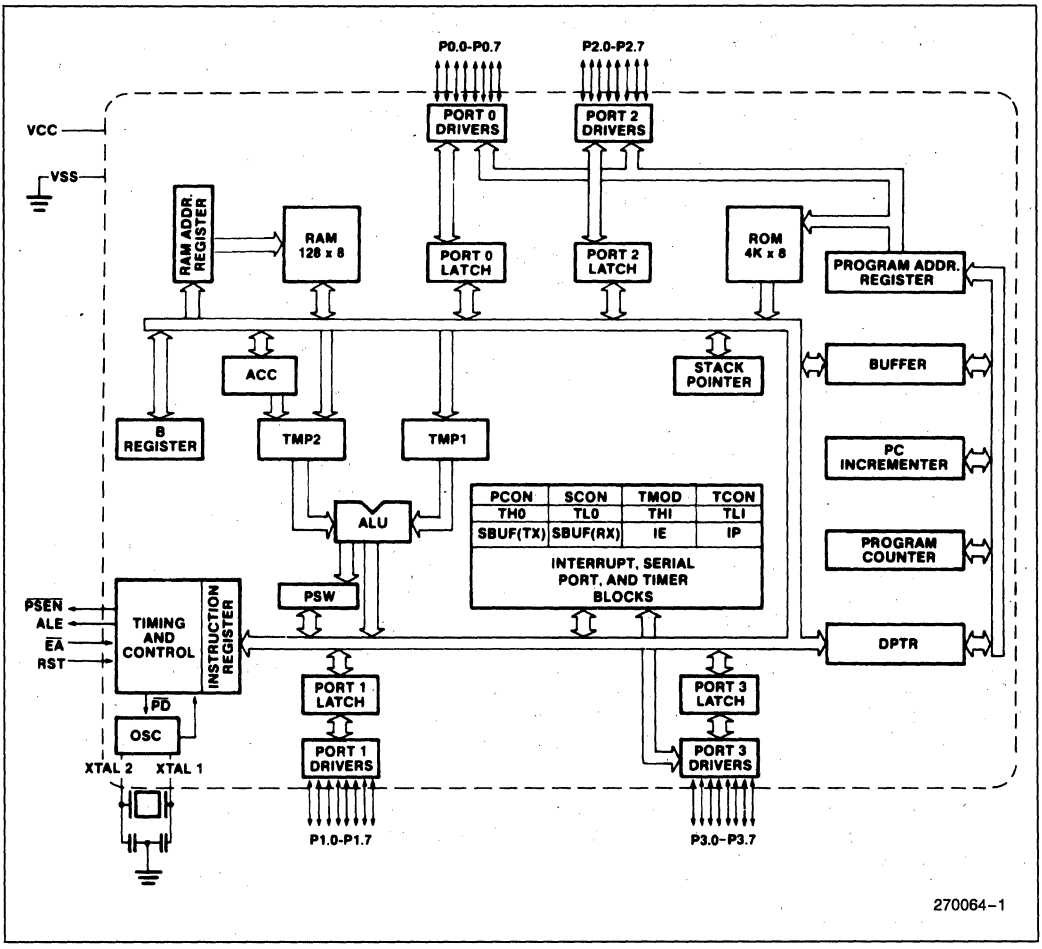


Figure 1. Block Diagram

IDLE MODE

In the Idle mode, the CPU puts itself to sleep while all the on chip peripherals stay active. The instruction that invokes the Idle mode is the last instruction executed in the normal operating mode before Idle mode is activated. The content of CPU, the on chip RAM, and all the Special Function Registers remain intact during this mode. The Idle mode can be terminated either by any enabled interrupt, at which time the process is picked up at the interrupt service routine and continued, or by a hardware reset which starts the processor the same as a power on reset.

POWER DOWN MODE

In the Power Down mode the oscillator is stopped, and the instruction that invokes Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

The only exit from Power Down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

The control bits for the reduced power modes are in the Special Function Register PCON.

NOTE:
For more detailed information on these reduced power modes refer to Application Note AP-252, "Designing with the 80C51BH".

Table 1. Status of the external pins during Idle and Power Down modes

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

PACKAGES

Part	Prefix	Package Type
80C51BH/ 80C31BH*	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-Pin PLCC

*The 80C51BH-1, 80C51BH-2, 80C31BH-1, and 80C31BH-2 have the same package types.

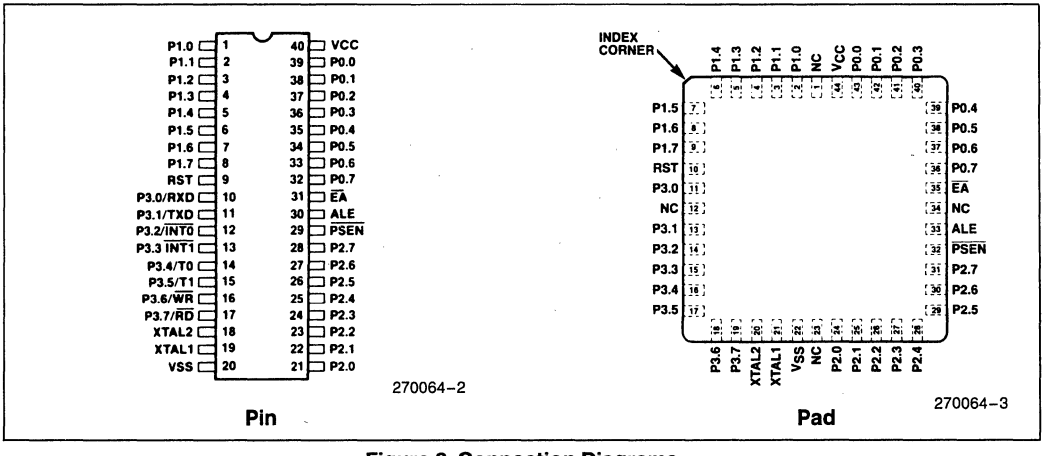


Figure 2. Connection Diagrams

PIN DESCRIPTIONS

V_{CC}

Supply voltage during normal, Idle, and Power Down operations.

V_{SS}

Circuit ground.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits Power-On reset using only an external capacitor to V_{CC}.

ALE

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$

Program Store Enable is the read strobe to external Program Memory.

When the 80C51BH is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory. $\overline{PSEN}$ is not activated during fetches from internal program memory.

$\overline{EA}$

External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the device to fetch code from external Program Memory locations starting at 0000H up to FFFFH. If $\overline{EA}$ is strapped to V_{CC} the device executes from internal Program Memory unless the program counter contains an address greater than 0FFFH.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock generator circuits.

XTAL2

Output from the inverting oscillator amplifier.

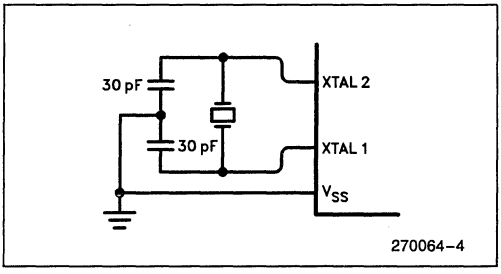


Figure 3. Crystal Oscillator

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillator for Microcontrollers".

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

Design Considerations

- At power on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up.
- Before entering the Power Down mode the contents of the Carry Bit and B.7 must be equal.
- When the Idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

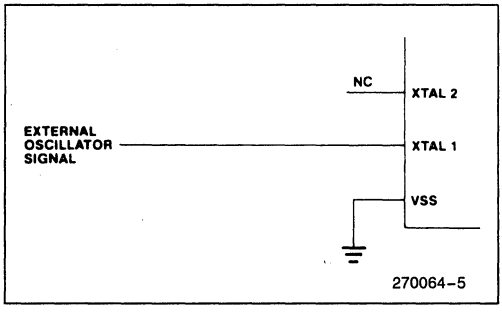


Figure 4. External Drive Configuration

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . 0°C to +70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on any Pin to V_{SS} . . . . . −0.5V to V_{CC} +0.5V
Voltage on V_{CC} to V_{SS} . . . . . −0.5V to 6.5V
Maximum I_{OL} per I/O pin . . . . . 15 mA
Power Dissipation 1.0W*

*This value is based on the maximum allowable die temperature and the thermal resistance of the package.

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ±20%; V_{SS} = 0V)

Symbol	Parameter	Min	Typ(3)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$)	−0.5		0.2 V_{CC} −0.1	V	
V_{IL1}	Input Low Voltage ($\overline{EA}$)	−0.5		0.2 V_{CC} − 0.3	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	0.2 V_{CC} + 0.9		V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage (6) (Ports 1, 2, 3)			0.45	V	I_{OL} = 1.6 mA (1)
V_{OL1}	Output Low Voltage (6) (Port 0, ALE, $\overline{PSEN}$)			0.45	V	I_{OL} = 3.2 mA (1)
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$)	2.4			V	I_{OH} = −60 μA V_{CC} = 5V ±10%
		0.75 V_{CC}			V	I_{OH} = −25 μA
		0.9 V_{CC}			V	I_{OH} = −10 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4			V	I_{OH} = −800 μA V_{CC} = 5V ±10%
		0.75 V_{CC}			V	I_{OH} = −300 μA
		0.9 V_{CC}			V	I_{OH} = −80 μA (2)
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)			−50	μA	V_{IN} = 0.45V
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)			−650	μA	V_{IN} = 2V
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)			±10	μA	0.45 < V_{IN} < V_{CC}
RRST	Reset Pulldown Resistor	50		150	KΩ	
CIO	Pin Capacitance			10	pF	Test Freq = 1 MHz, T_A = 25°C
I_{CC}	Power Supply Current:					
	Active Mode, 12 MHz (4)		11	20	mA	
	Idle Mode, 12 MHz (4)		1.7	5	mA	(5)
	Power Down Mode		5	50	μA	

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OLS} of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.
3. "Typicals" are based on a limited number of samples taken from early manufacturing lots and are not guaranteed. The values listed are at room temperature, 5V.
4. ICCMAX at other frequencies is given by
   Active Mode: ICCMAX = 1.47 × FREQ + 2.35
   Idle Mode: ICCMAX = 0.33 × FREQ + 1.05
where FREQ is the external oscillator frequency in MHz. ICCMAX is given in mA. See Figure 5.
5. See Figures 6 through 9 for I_{CC} test conditions. Minimum V_{CC} for Power Down is 2V.
6. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port-
   Port 0: 26 mA
   Ports1, 2, and 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

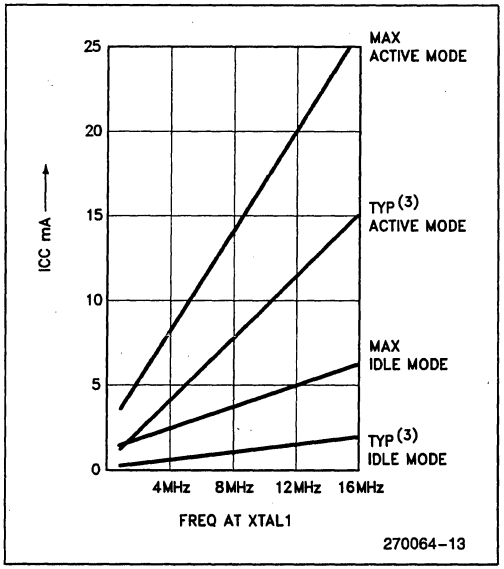


Figure 5. I_{CC} vs. Frequency. Valid only within frequency specifications of the device under test.

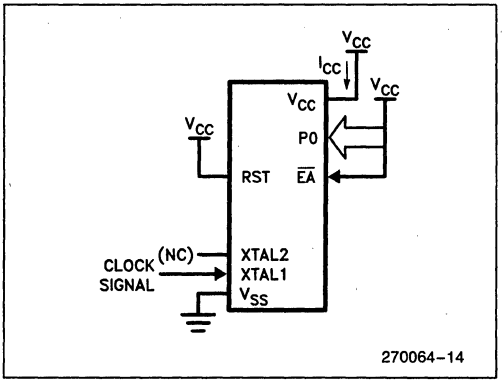


Figure 6. I_{CC} Test Condition, Active Mode. All other pins are disconnected.

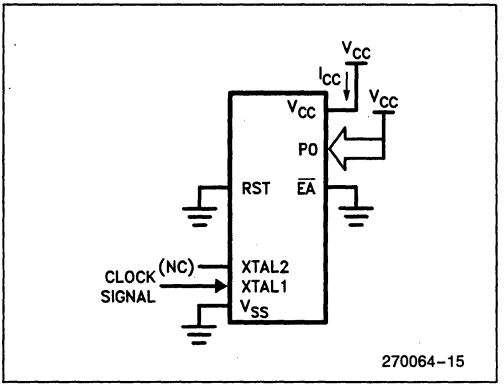


Figure 7. I_{CC} Test Condition, Idle Mode. All other pins are disconnected.

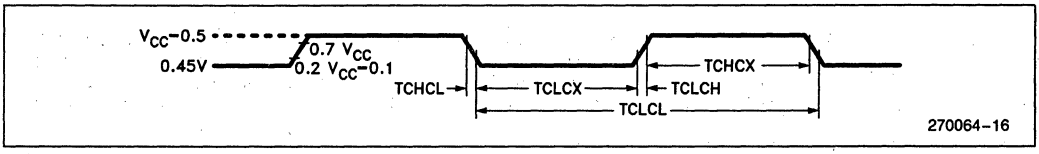


Figure 8. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

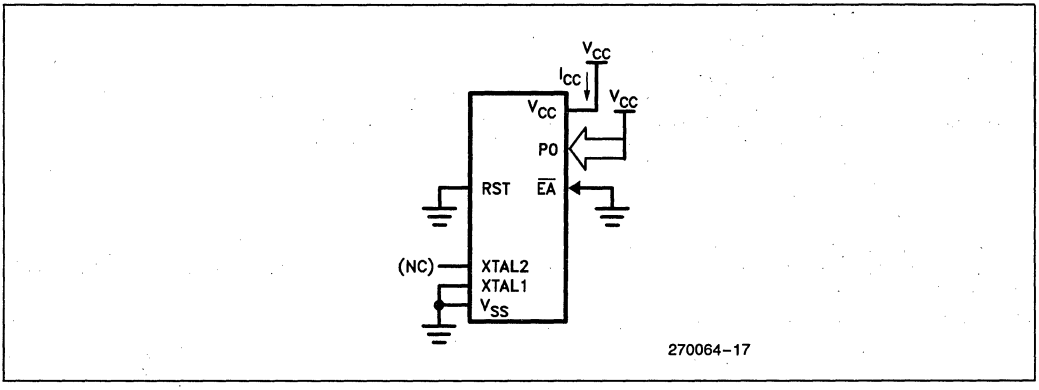


Figure 9. I_{CC} Test Condition, Power Down Mode. All other pins are disconnected. V_{CC} = 2V to 6V.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address.
C: Clock.
D: Input data.
H: Logic level HIGH.
I: Instruction (program memory contents).
L: Logic level LOW, or ALE.
P: $\overline{PSEN}$.
Q: Output data.
R: $\overline{RD}$ signal.
T: Time.
V: Valid.
W: $\overline{WR}$ signal.
X: No longer a valid logic level.
Z: Float.

EXAMPLE:

TAVLL = Time for Address Valid to ALE Low.
TLLPL = Time for ALE Low to $\overline{PSEN}$ Low.

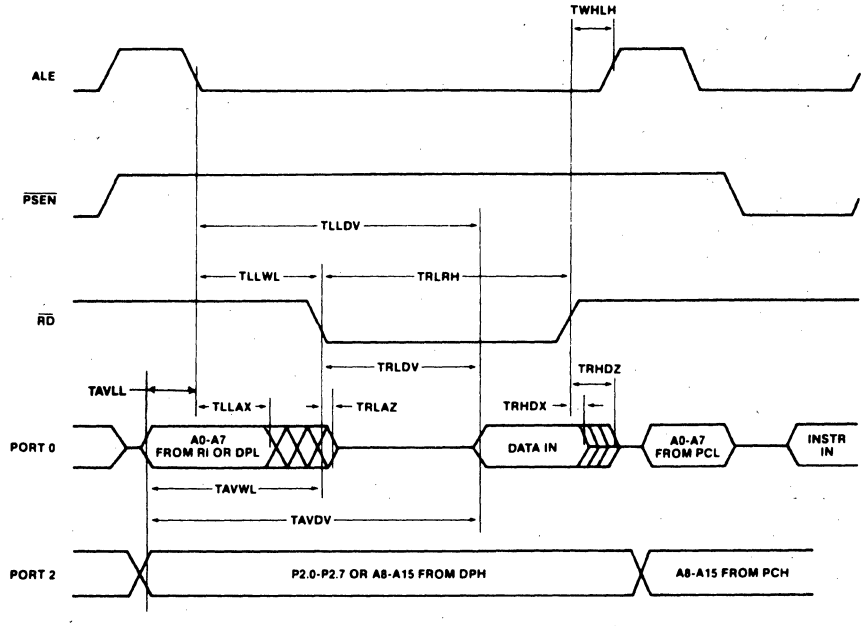
A.C. CHARACTERISTICS

(T_A = 0°C to 70°C, V_{CC} = 5V ±20%, V_{SS} = 0V, Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF, Load Capacitance for All Other Outputs = 80 pF)

EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

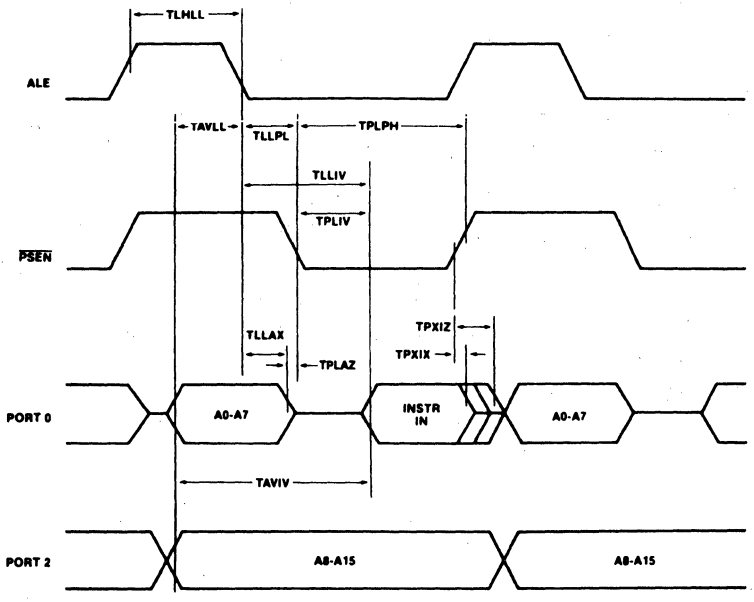
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 80C51BH/80C31BH 80C51BH-1/80C31BH-1 80C51BH-2/80C31BH-2			 3.5 3.5 0.5	 12 16 12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL − 40		ns
TAVLL	Address Valid to ALE Low	28		TCLCL − 55		ns
TLLAX	Address Hold After ALE Low	48		TCLCL − 35		ns
TLLIV	ALE Low to Valid Instr In		234		4TCLCL − 100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	43		TCLCL − 40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	205		3TCLCL − 45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr In		145		3TCLCL − 105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		59		TCLCL − 25	ns
TAVIV	Address to Valid Instr In		312		5TCLCL − 105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL − 100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL − 100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL − 165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL − 70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL − 150	ns
TAVDV	Address to Valid Data In		585		9TCLCL − 165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL − 50	3TCLCL + 50	ns
TAVWL	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL − 130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		TCLCL − 60		ns
TWHQX	Data Hold After $\overline{WR}$	33		TCLCL − 50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL − 40	TCLCL + 40	ns

EXTERNAL DATA MEMORY READ CYCLE



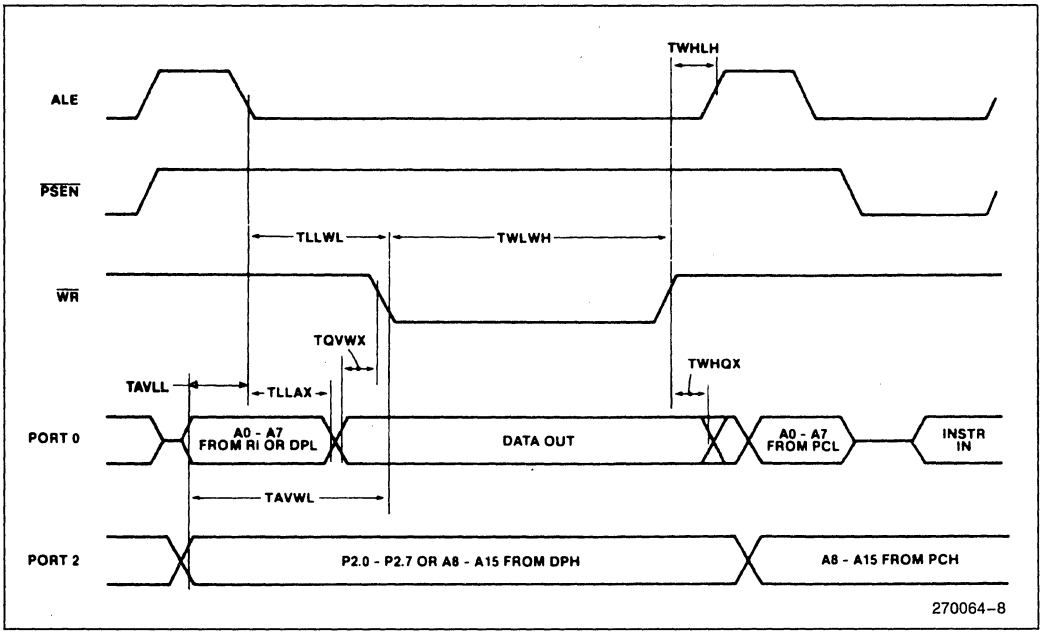
270064-6

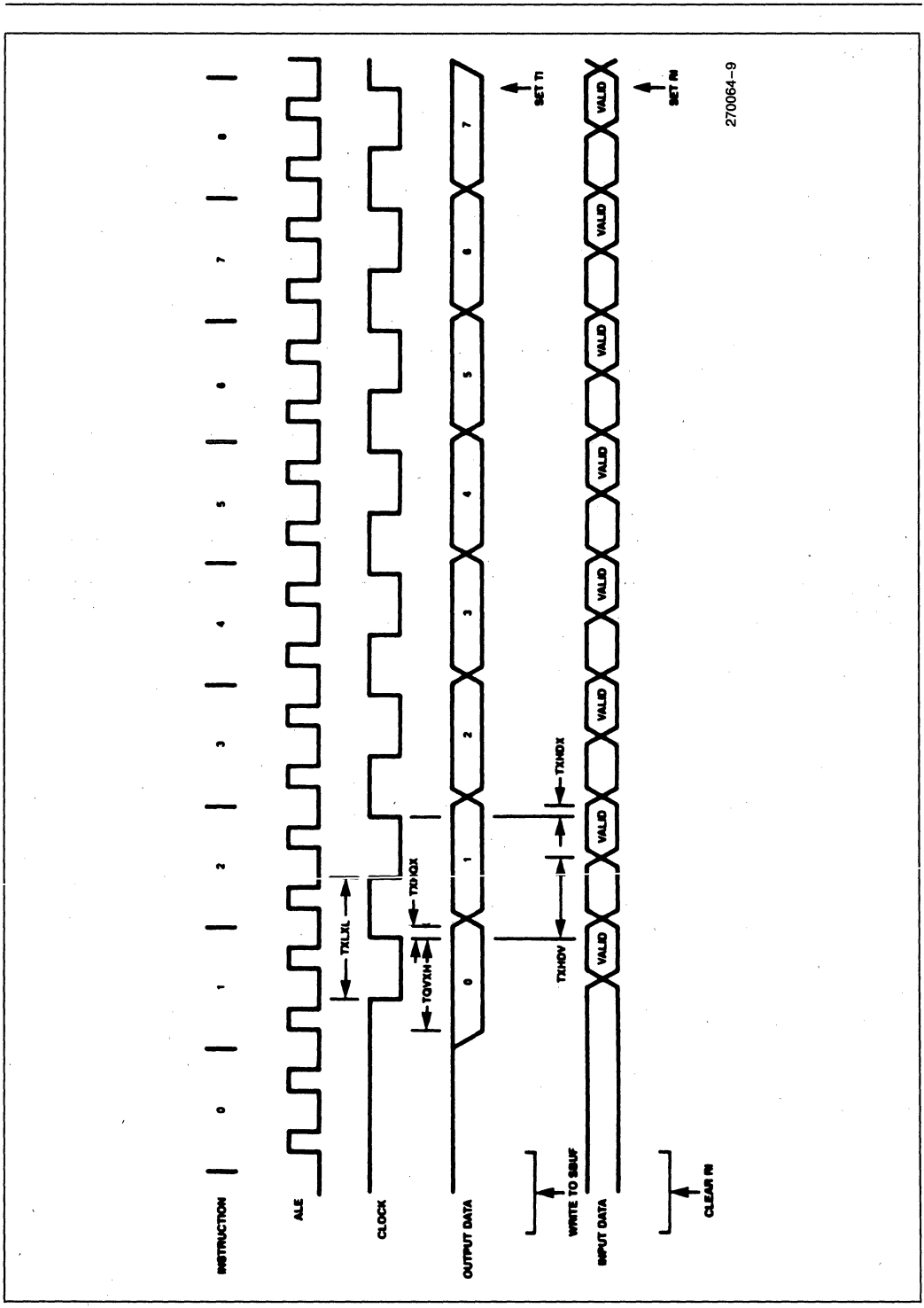
EXTERNAL PROGRAM MEMORY READ CYCLE



270064-7

EXTERNAL DATA MEMORY WRITE CYCLE





Shift Register Mode Timing Waveforms

EXTERNAL CLOCK DRIVE

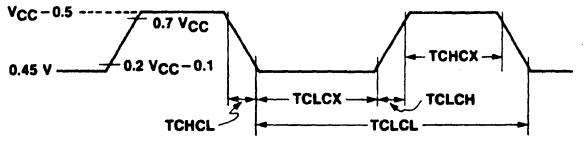
Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency 80C51BH/80C31BH 80C51BH-1/80C31BH-1 80C51BH-2/80C31BH-2	 3.5 3.5 0.5	 12 16 12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

SERIAL TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±20%; V_{SS} = 0V; Load Capacitance = 80 pF

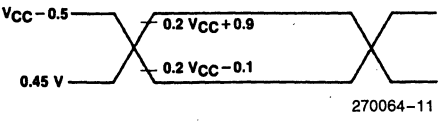
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL − 133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL − 117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL − 133	ns

EXTERNAL CLOCK DRIVE WAVEFORM



270064-10

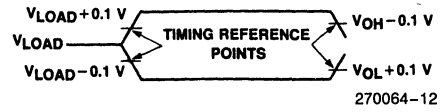
AC TESTING INPUT, OUTPUT WAVEFORMS



270064-11

AC Inputs during testing are driven at V_{CC} − 0.5 for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at V_{IH} min. for a logic "1" and V_{IL} max. for a logic "0".

FLOAT WAVEFORMS



270064-12

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -005 version of the 80C51BH data sheet:

1. Package table was added.
2. Note 6 on maximum current specifications added to DC Characteristics.
3. Data Sheet Revision Summary was added.

80C31BH/80C51BH

EXPRESS

- **Extended Temperature Range**
- **3.5 to 12 MHz V_{CC} = 5V ±20%**
- **Burn-In**

The Intel EXPRESS system offers enhancements to the operational specifications of the MCS®-51 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in and an extended temperature range with or without burn-in.

With the commercial standard temperature range, operational characteristics are guaranteed over the temperature range of 0°C to 70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

The optional burn-in is dynamic for a minimum time of 160 hours at 125°C with V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

September 1987
Order Number: 270218-002

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets.

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = 5V \pm 20\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage (Except EA)	-0.5	$0.2V_{CC} - 0.15$	V	
V_{IL1}	EA	$-0.5V$	$0.2V_{CC} - 0.35$	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	$0.2V_{CC} + 1$	$V_{CC} + 0.5$	V	
V_{IH1}	Input High Voltage to XTAL1, RST	$0.7V_{CC} + 0.1$	$V_{CC} + 0.5$	V	
I_{IL}	Logical 0 Input Current (Port 1, 2, 3)		-75	μA	$V_{in} = 0.45V$
I_{TL}	Logical 1 to 0 transition Current (Ports 1, 2, 3)		-750	μA	$V_{in} = 2.0V$

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range	Burn-In
P	Plastic	Commercial	No
D	Cerdip	Commercial	No
N	PLCC	Commercial	No
TP	Plastic	Extended	No
TD	Cerdip	Extended	No
TN	PLCC	Extended	No
QP	Plastic	Commercial	Yes
QD	Cerdip	Commercial	Yes
QN	PLCC	Commercial	Yes
LP	Plastic	Extended	Yes
LD	Cerdip	Extended	Yes
LN	PLCC	Extended	Yes

NOTE:
- Commercial temperature range is 0°C to 70°C. Extended temperature range is $-40°C$ to $+85°C$.
- Burn-in is dynamic for a minimum time of 160 hours at 125°C, $V_{CC} = 6.9V \pm 0.25V$, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).

Examples:

P80C31BH indicates 80C31BH in a plastic package and specified for commercial temperature range, without burn-in.
LD80C51BH indicates 80C51BH in a cerdip package and specified for extended temperature range with burn-in.

80C51 BHP
CHMOS SINGLE-CHIP 8-BIT MICROCOMPUTER WITH PROTECTED ROM

80C51BHP—3.5–12 MHz, V_{CC} = 5V ± 20%
80C51BHP-1—3.5–16 MHz, V_{CC} = 5V ± 20%
80C51BHP-2—0.5–12 MHz, V_{CC} = 5V ± 20%

- **Power Control Modes**
- **128 x 8-Bit RAM**
- **32 Programmable I/O Lines**
- **Two 16-Bit Timer/Counters**
- **4K Program Memory Space**
- **Protection Feature Protects ROM Parts Against Software Piracy**
- **High Performance CHMOS Process**
- **Boolean Processor**
- **5 Interrupt Sources**
- **Programmable Serial Port**
- **4K Data Memory Space (Expandable to 64K)**

The MCS®-51 family of CHMOS products is fabricated on Intel's CHMOS III process and is functionally compatible with the standard 8051 HMOS and EPROM products. CHMOS III is a technology which combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. This combination expands the effectiveness of the powerful 8051 architecture and instruction set.

Like the 8051 HMOS versions, the 80C51 BHP has the following features: 4K byte of ROM; 128 bytes of RAM; 32 I/O lines; two 16-bit timer/counters; a five-source two-level interrupt structure; a full duplex serial port; and on-chip oscillator and clock circuitry. In addition, the 80C51 BHP has two software selectable modes of reduced activity for further power reduction—Idle and Power Down.

The Idle mode freezes the CPU while allowing the RAM, timer/counters serial port and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

The 80C51BHP is identical to the 80C51BH with the exception of the Protection Feature. To incorporate this Protection Feature, program verification has been disabled and external memory accesses have been limited to 4K.

October 1988
Order Number: 270603-002

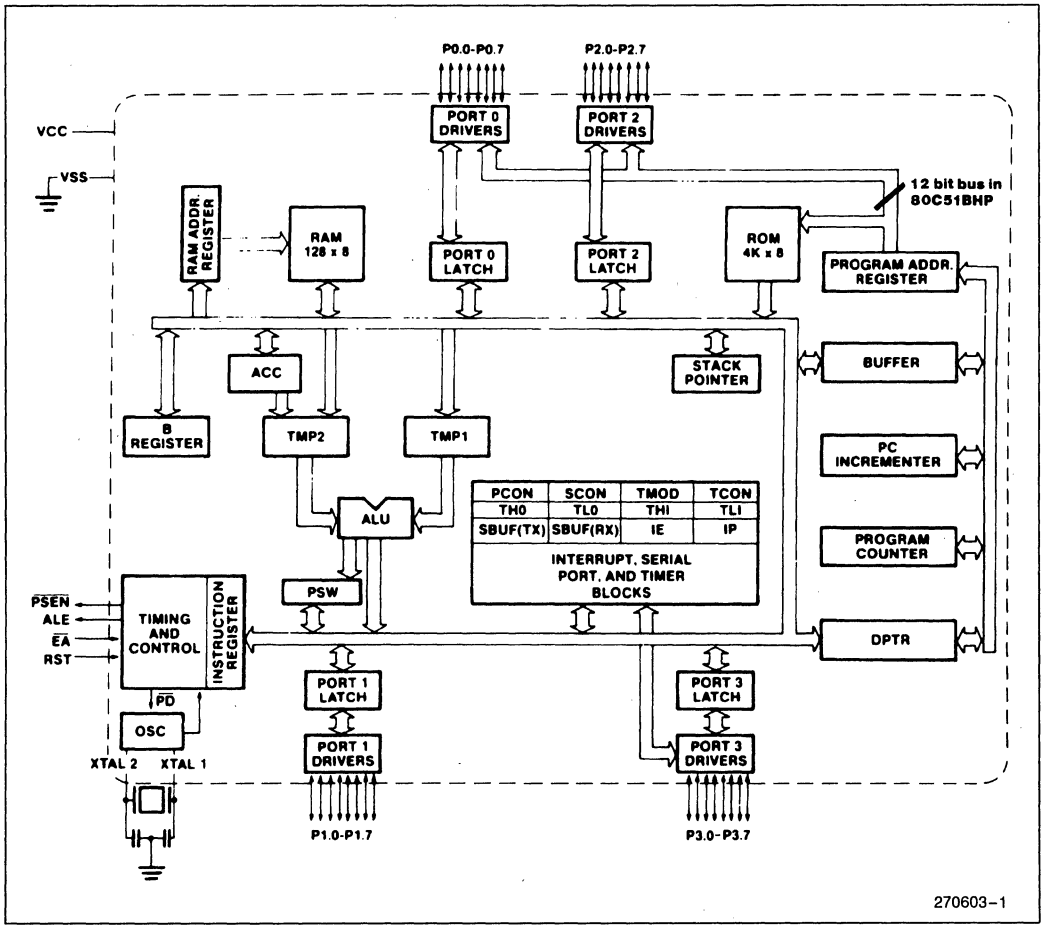


Figure 1. Block Diagram

IDLE MODE

In the Idle mode, the CPU puts itself to sleep while all the on chip peripherals stay active. The instruction that invokes the Idle mode is the last instruction executed in the normal operating mode before Idle mode is activated. The content of the on-chip RAM and all the Special Function Registers remain intact during this mode. The Idle mode can be terminated either by any enabled interrupt, at which time the process is picked up at the interrupt service routine and continued, or by a hardware reset which starts the processor the same as a power on reset.

POWER DOWN MODE

In the Power Down mode the oscillator is stopped, and the instruction that invokes Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

The only exit from Power Down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

The control bits for the reduced power modes are in the Special Function Register PCON.

NOTE:
For more detailed information on these reduced power modes refer to Application Note AP-252, "Designing with the 80C51BH".

PACKAGES

Part	Prefix	Package Type
80C51BHP	P	40-Pin Plastic DIP
	N	44-Pin PLCC

Table 1. Status of the external pins during Idle and Power Down modes

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

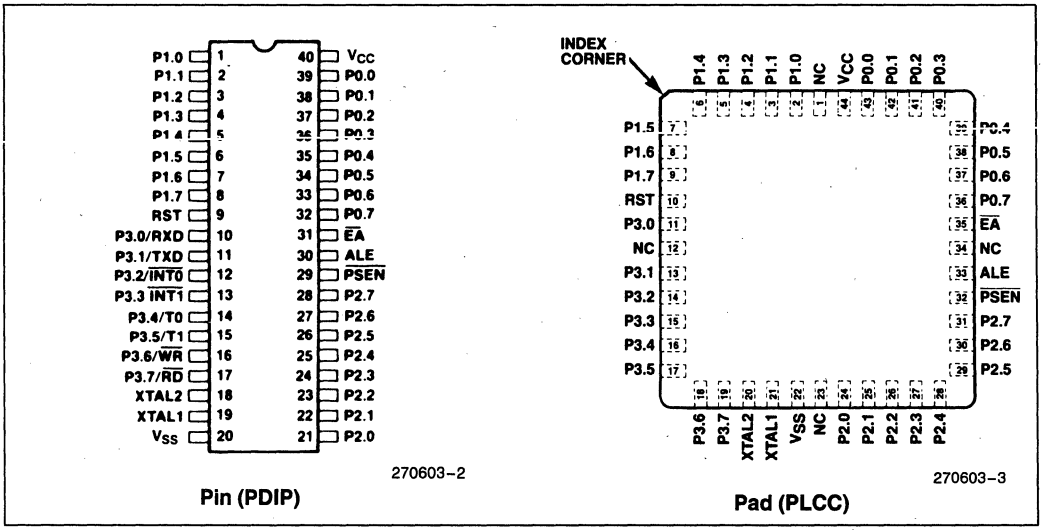


Figure 2. Connection Diagrams

PIN DESCRIPTIONS

V_{CC}

Supply voltage during normal, Idle, and Power Down operations.

V_{SS}

Circuit ground.

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. In the 80C51 BHP, Bits 2.4 through 2.7 are forced to 0, effectively limiting external data and code space to 4K each during external accesses (see Design Considerations). During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the 8051 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. An internal diffused resistor to V_{SS} permits Power-On reset using only an external capacitor to V_{CC}.

ALE

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$

Program Store Enable is the read strobe to external Program Memory.

When the device is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory. $\overline{PSEN}$ is not activated during fetches from internal program memory.

$\overline{EA}$

External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the device to fetch code from external Program Memory locations starting at 0000H up to FFFFH. If $\overline{EA}$ is strapped to V_{CC} the device executes from internal Program Memory unless the program counter contains an address greater than 0FFFH.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock generator circuits.

XTAL2

Output from the inverting oscillator amplifier.

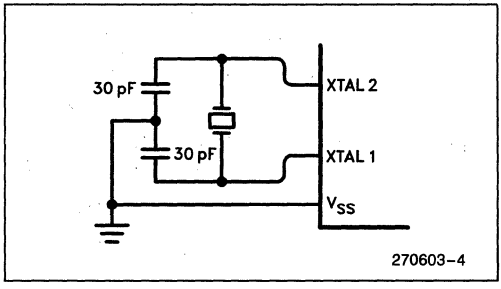


Figure 3. Crystal Oscillator

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillator for Microcontrollers".

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

Design Considerations

- The 80C51BHP cannot access external Program or Data memory above 4K. This means that the following instructions that use the Data Pointer only read/write data at address locations below 0FFFH:

  ```
  MOVX A, @DPTR
  MOVX @DPTR, A
  ```

  When the Data Pointer contains an address above the 4K limit, those locations will not be accessed. To access Data Memory above 4K, the MOVX @Ri, A or MOVX A, @Ri instructions must be used.
- Before entering the Power Down mode the contents of the Carry Bit and B.7 must be equal.
- When the Idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

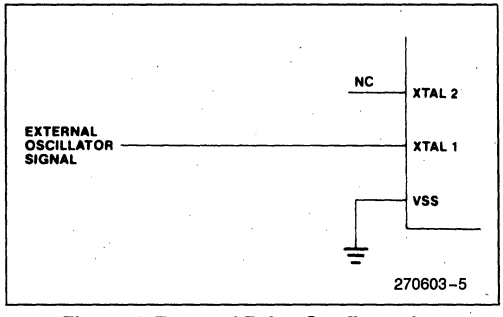


Figure 4. External Drive Configuration

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on any
Pin to V_{SS} .....−0.5V to V_{CC} +0.5V
Voltage on V_{CC} to V_{SS} .....−0.5V to 6.5V
Maximum I_{OL} per I/O Pin .....15 mA
Power Dissipation.....1.0W*

*This value is based on the maximum allowable die temperature and the thermal resistance of the package.

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS (T_A = 0°C to +70°C; V_{CC} = 5V ±20%; V_{SS} = 0V)

Symbol	Parameter	Min	Typ(3)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$)	−0.5		0.2 V_{CC} −0.1	V	
V_{IL1}	Input Low Voltage ($\overline{EA}$)	−0.5		0.2 V_{CC} − 0.3	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	0.2 V_{CC} + 0.9		V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage (6) (Ports 1, 2, 3)			0.45	V	I_{OL} = 1.6 mA (1)
V_{OL1}	Output Low Voltage (6) (Port 0, ALE, $\overline{PSEN}$)			0.45	V	I_{OL} = 3.2 mA (1)
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$)	2.4			V	I_{OH} = −60 μA V_{CC} = 5V ±10%
		0.75 V_{CC}			V	I_{OH} = −25 μA
		0.9 V_{CC}			V	I_{OH} = −10 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4			V	I_{OH} = −800 μA V_{CC} = 5V ±10%
		0.75 V_{CC}			V	I_{OH} = −300 μA
		0.9 V_{CC}			V	I_{OH} = −80 μA (2)
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)			−50	μA	V_{IN} = 0.45V
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)			−650	μA	V_{IN} = 2V
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)			±10	μA	0.45 < V_{IN} < V_{CC}
RRST	Reset Pulldown Resistor	50		150	KΩ	
CIO	Pin Capacitance			10	pF	Test Freq = 1 MHz, T_A = 25°C
I_{CC}	Power Supply Current:					
	Active Mode, 12 MHz (4)		11	20	mA	
	Idle Mode, 12 MHz (4)		1.7	5	mA	(5)
	Power Down Mode		5	50	μA	

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OLS} of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.
3. "Typicals" are based on a limited number of samples taken from early manufacturing lots and are not guaranteed. The values listed are at room temperature, 5V.
4. ICCMAX at other frequencies is given by
   Active Mode: ICCMAX = 1.47 × FREQ + 2.35
   Idle Mode: ICCMAX = 0.33 × FREQ + 1.05
where FREQ is the external oscillator frequency in MHz. ICCMAX is given in mA. See Figure 5.
5. See Figures 6 through 9 for I_{CC} test conditions. Minimum V_{CC} for Power Down is 2V.
6. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum I_{OL} per Port Pin:	10 mA
Maximum I_{OL} per 8-Bit Port —	
Port 0:	26 mA
Ports 1, 2 and 3:	15 mA
Maximum Total I_{OL} for all output pins:	71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

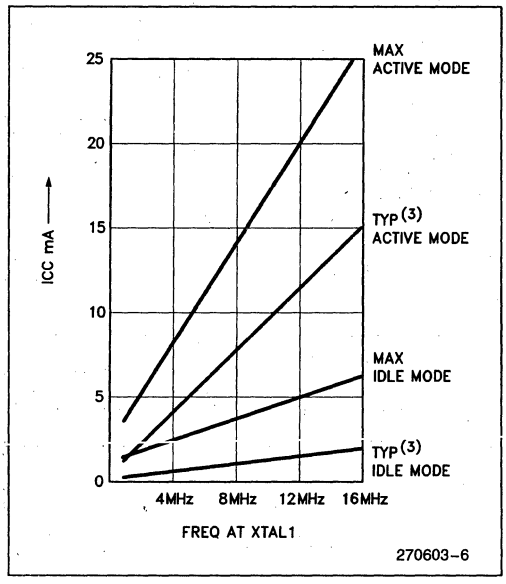


Figure 5. I_{CC} vs. Frequency. Valid only within frequency specifications of the device under test.

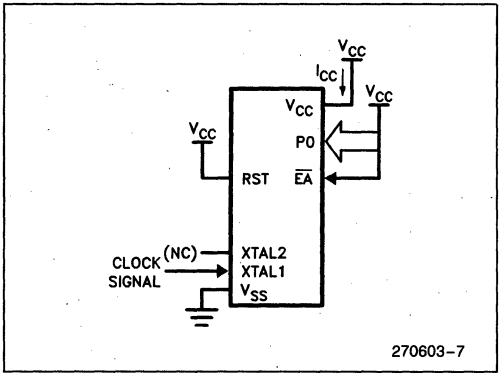


Figure 6. I_{CC} Test Condition, Active Mode. All other pins are disconnected.

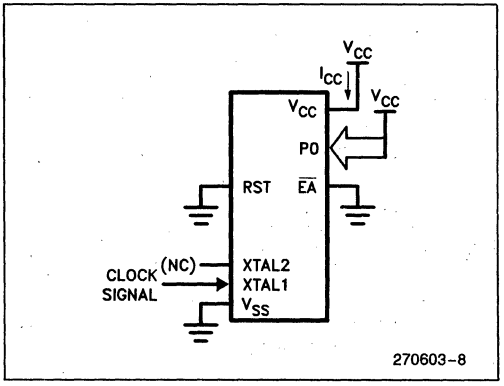


Figure 7. I_{CC} Test Condition, Idle Mode. All other pins are disconnected.

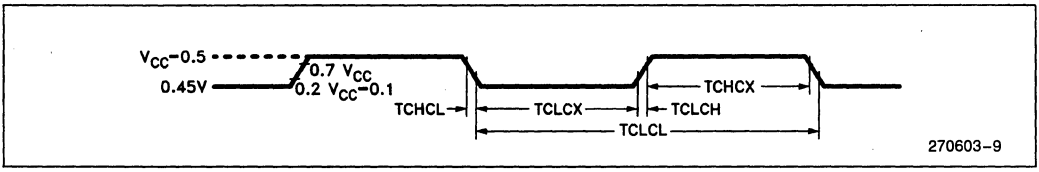


Figure 8. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

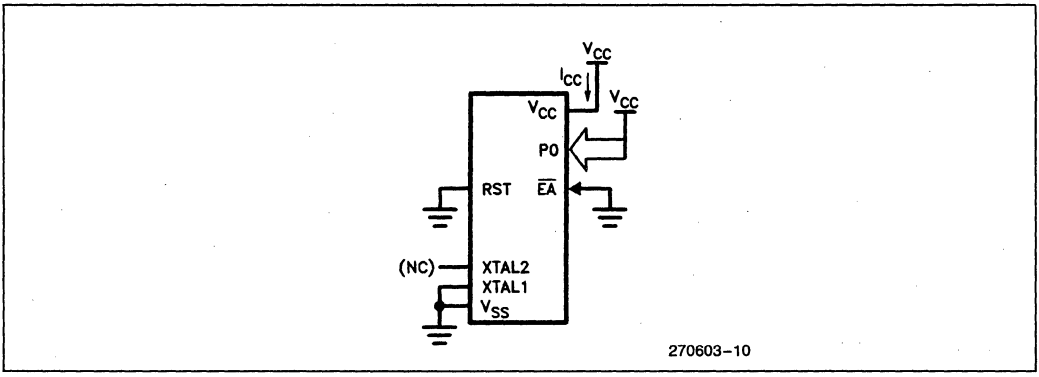


Figure 9. I_{CC} Test Condition, Power Down Mode. All other pins are disconnected. V_{CC} = 2V to 6V.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address.
C: Clock.
D: Input data.
H: Logic level HIGH.
I: Instruction (program memory contents).
L: Logic level LOW, or ALE.
P: $\overline{\text{PSEN}}$.
Q: Output data.
R: $\overline{\text{RD}}$ signal.
T: Time.
V: Valid.
W: $\overline{\text{WR}}$ signal.
X: No longer a valid logic level.
Z: Float.

EXAMPLE:

TAVLL = Time for Address Valid to ALE Low.
TLLPL = Time for ALE Low to $\overline{\text{PSEN}}$ Low.

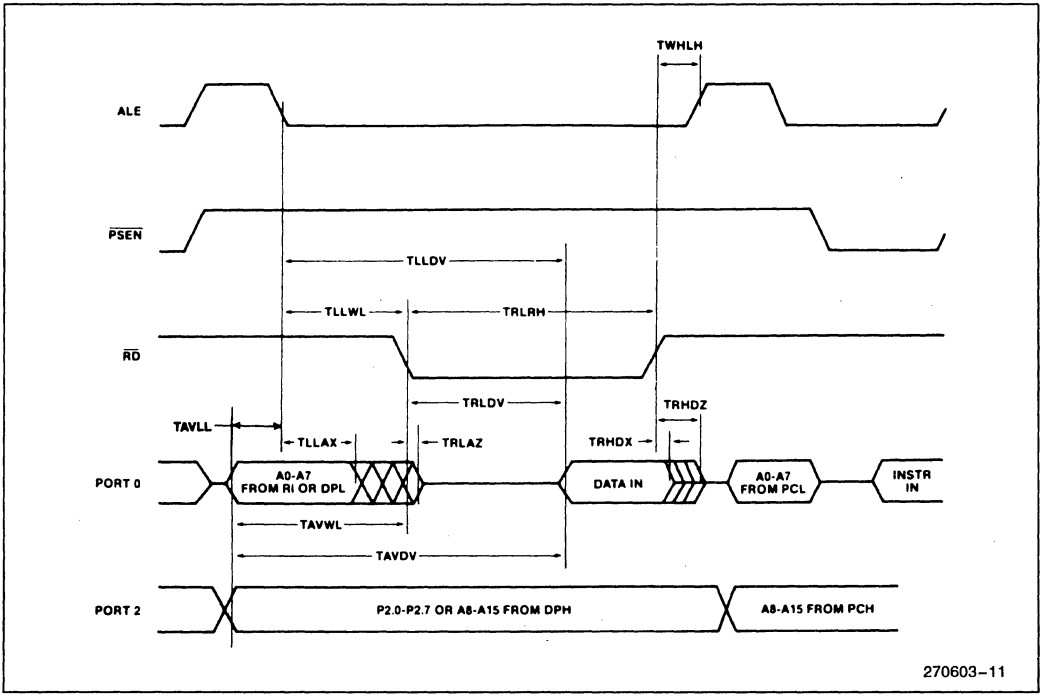
A.C. CHARACTERISTICS

(T_A = 0°C to 70°C, V_{CC} = 5V ±20%, V_{SS} = 0V, Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF, Load Capacitance for All Other Outputs = 80 pF)

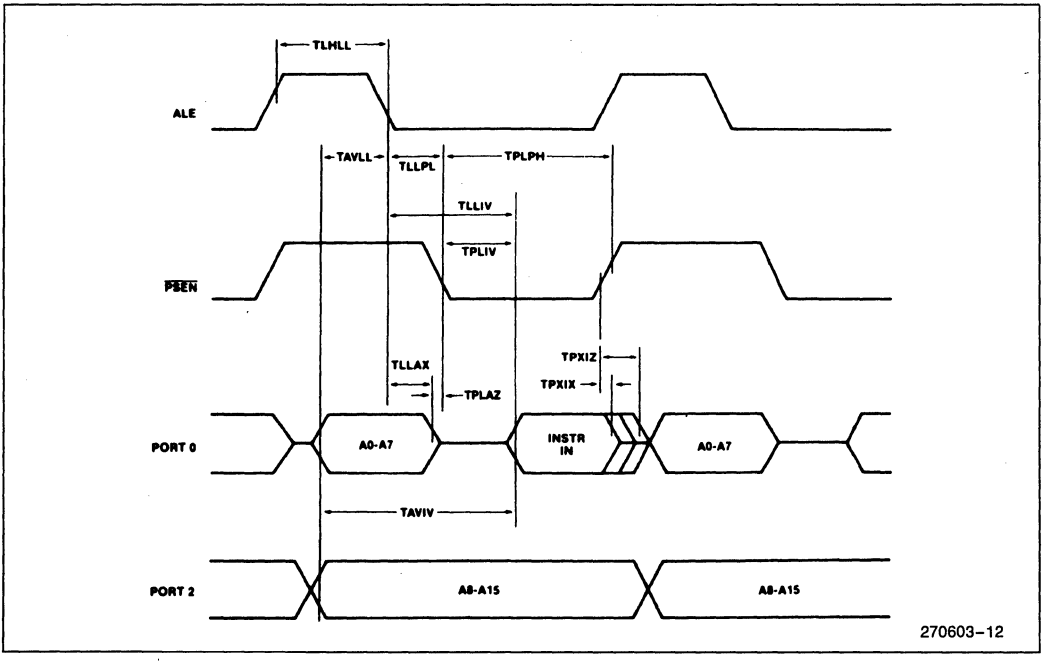
EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 80C51BH/80C31BH 80C51BH-1/80C31BH-1 80C51BH-2/80C31BH-2			 3.5 3.5 0.5	 12 16 12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL − 40		ns
TAVLL	Address Valid to ALE Low	28		TCLCL − 55		ns
TLLAX	Address Hold After ALE Low	48		TCLCL − 35		ns
TLLIV	ALE Low to Valid Instr In		234		4TCLCL − 100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	43		TCLCL − 40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	205		3TCLCL − 45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr In		145		3TCLCL − 105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		59		TCLCL − 25	ns
TAVIV	Address to Valid Instr In		312		5TCLCL − 105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL − 100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL − 100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL − 165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL − 70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL − 150	ns
TAVDV	Address to Valid Data In		585		9TCLCL − 165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL − 50	3TCLCL + 50	ns
TAVWL	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL − 130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		TCLCL − 60		ns
TWHQX	Data Hold After $\overline{WR}$	33		TCLCL − 50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL − 40	TCLCL + 40	ns

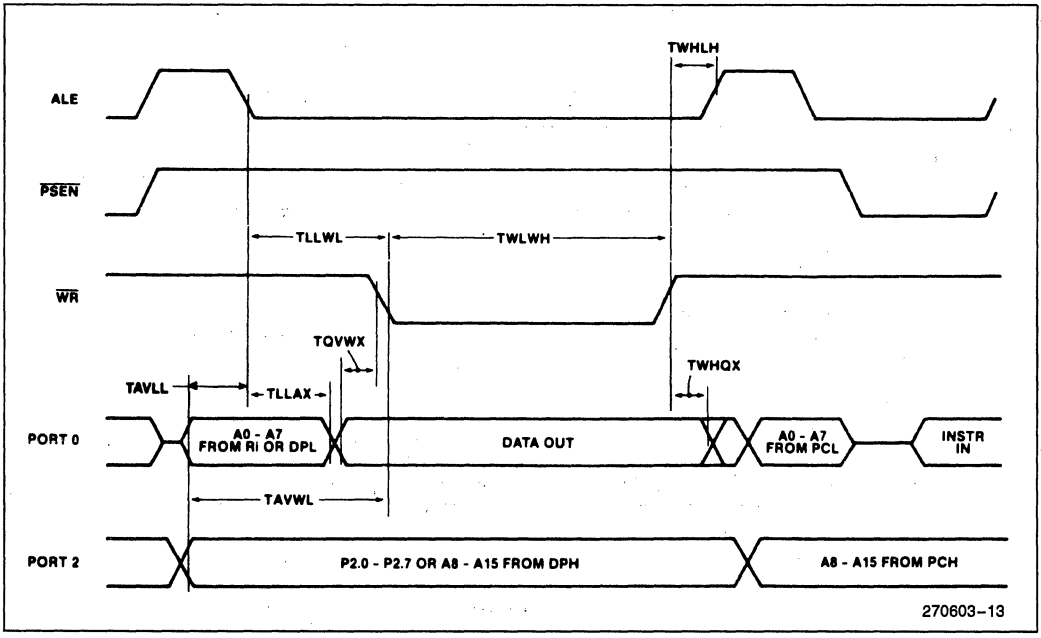
EXTERNAL DATA MEMORY READ CYCLE

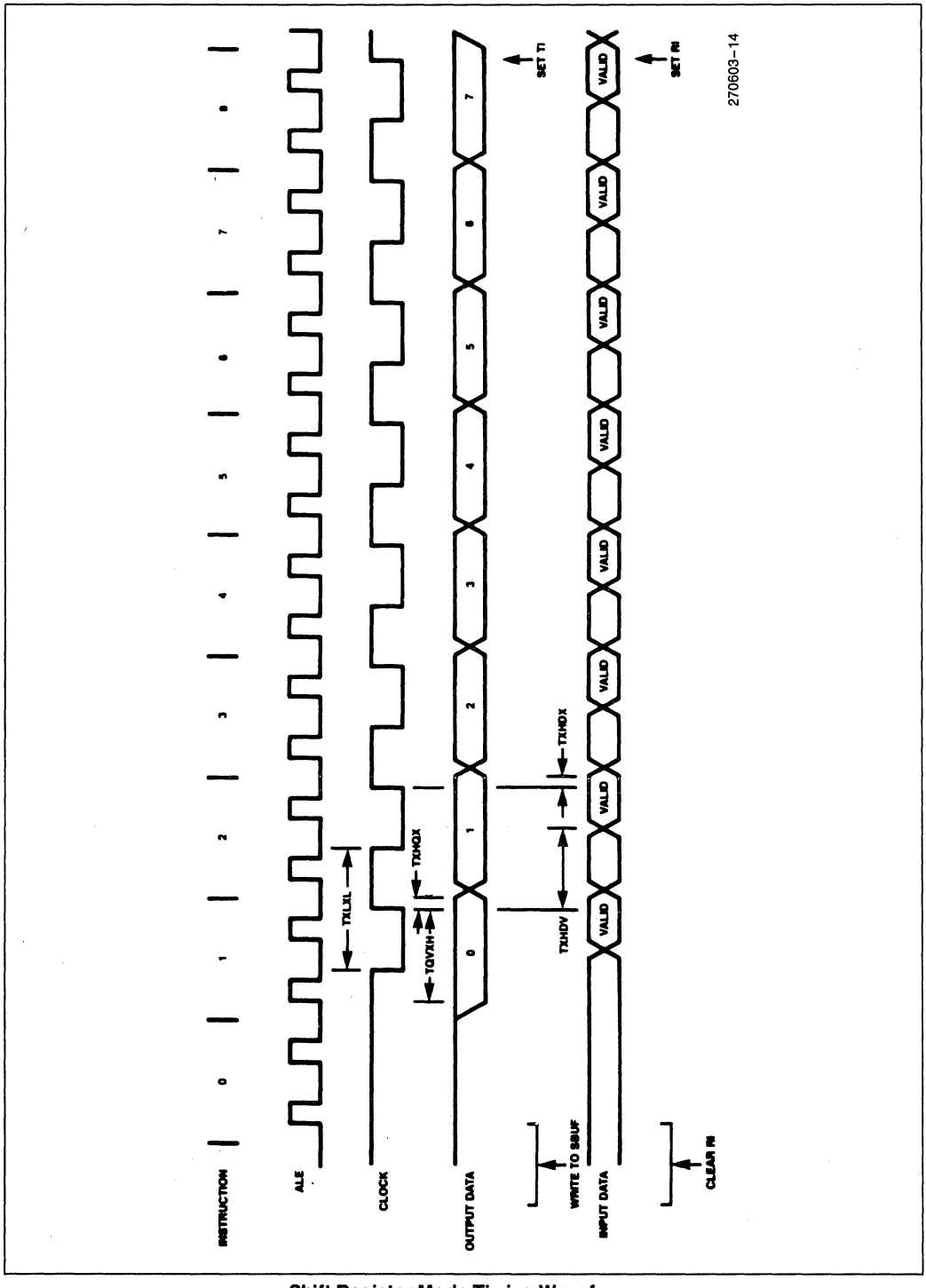


EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE





Shift Register Mode Timing Waveforms

EXTERNAL CLOCK DRIVE

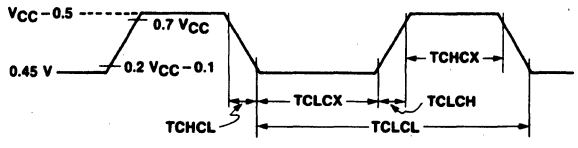
Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency 80C51BHP 80C51BHP-1 80C51BHP-2	 3.5 3.5 0.5	 12 16 12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

SERIAL TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±20%; V_{SS} = 0V; Load Capacitance = 80 pF

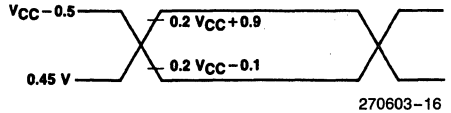
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL − 133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL − 117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL − 133	ns

EXTERNAL CLOCK DRIVE WAVEFORM



270603-15

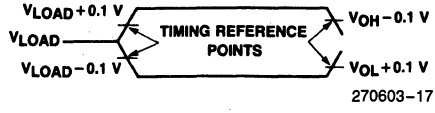
AC TESTING INPUT, OUTPUT WAVEFORMS



270603-16

AC Inputs during testing are driven at V_{CC} − 0.5 for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at V_{IH} min. for a logic "1" and V_{IL} max. for a logic "0".

FLOAT WAVEFORMS



270603-17

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -001 version of the 80C51BHP data sheet:

1. Package Table was added.
2. Note 6 on Maximum Current Specifications was added to D.C. Characteristics.
3. Data Sheet Revision Summary was added.

87C51/87C51-1/87C51-2
CHMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH 4K BYTES OF EPROM PROGRAM MEMORY

87C51—3.5 to 12 MHz, V_{CC} = 5V ± 10%
87C51-1—3.5 to 16 MHz, V_{CC} = 5V ± 10%
87C51-2—0.5 to 12 MHz, V_{CC} = 5V ± 10%

- **High Performance CHMOS EPROM**
- **Quick-Pulse Programming™ Algorithm**
- **2-Level Program Memory Lock**
- **Boolean Processor**
- **128-Byte Data RAM**
- **32 Programmable I/O Lines**
- **Two 16-Bit Timer/Counters**
- **5 Interrupt Sources**
- **Programmable Serial Channel**
- **TTL- and CMOS-Compatible Logic Levels**
- **64K External Program Memory Space**
- **64K External Data Memory Space**
- **IDLE and POWER DOWN Modes**
- **ONCE™ Mode Facilitates System Testing**
- **LCC, PLCC, and DIP Packaging Available**

The 87C51 is the EPROM version of the 80C51BH. It is fabricated on Intel's CHMOS II-E process. It contains 4K bytes of on-chip Program memory that can be electrically programmed, and can be erased by exposure to ultraviolet light.

The 87C51 EPROM array uses a modified Quick-Pulse programming algorithm, by which the entire 4K-byte array can be programmed in about 12 seconds.

The extremely low operating power, along with the two reduced power modes, Idle and Power Down, make this part very suitable for low power applications. The Idle mode freezes the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

October 1988
Order Number: 270147-005

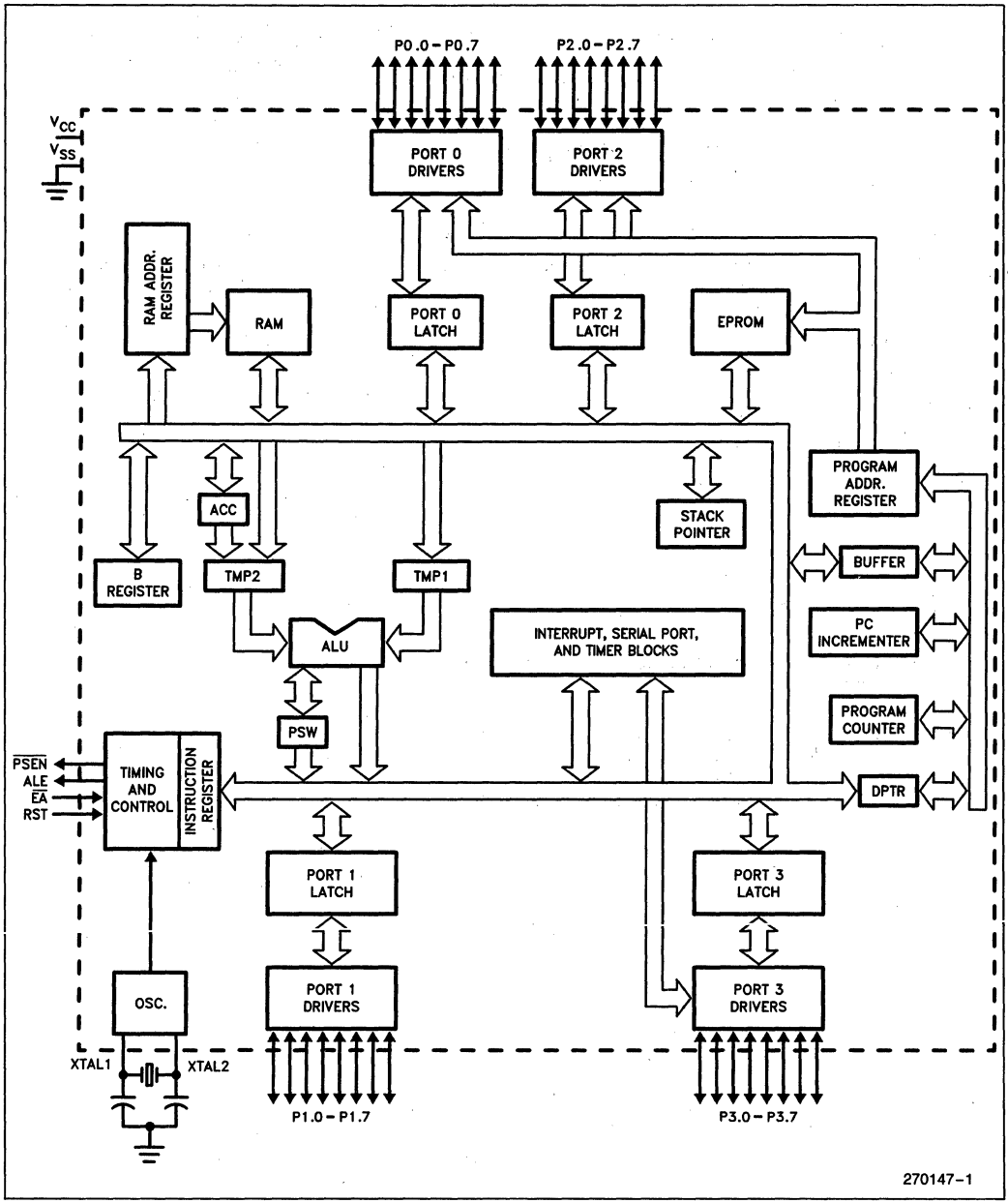


Figure 1. MCS®-51 Architectural Block Diagram

PACKAGES

Part	Prefix	Package Type
87C51/	P	40-Pin Plastic DIP
87C51-1/	D	40-Pin CERDIP
87C51-2	N	44-Pin PLCC

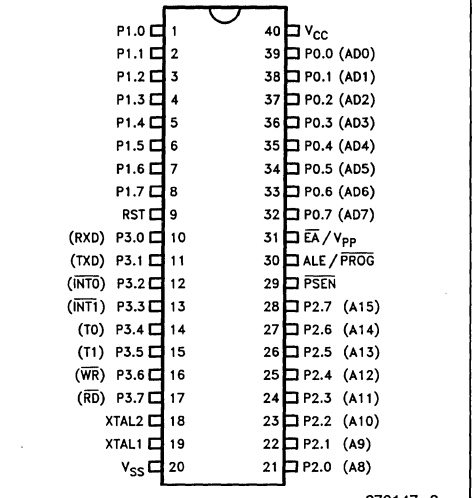


Figure 2. Pin Connections

PIN DESCRIPTION

V_{CC}: Supply voltage during normal, Idle, and Power Down operations.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external memory. In this application it uses strong internal pullups when emitting 1s.

Port 0 also receives the code bytes during EPROM programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during EPROM programming and program verification.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program memory and during accesses to external Data Memory that use 16-bit address (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s.

During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives some control signals and the high-order address bits during EPROM programming and program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Pin	Name	Alternate Function
P3.0	RXD	Serial input line
P3.1	TXD	Serial output line
P3.2	$\overline{INT0}$	External Interrupt 0
P3.3	$\overline{INT1}$	External Interrupt 1
P3.4	T0	Timer 0 external input
P3.5	T1	Timer 1 external input
P3.6	$\overline{WR}$	External Data Memory Write strobe
P3.7	$\overline{RD}$	External Data Memory Read strobe

Port 3 also receives some control signals for EPROM programming and program verification.

RST: Reset input. A logic high on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits a power-on reset to be generated using only an external capacitor to V_{CC}.

ALE/$\overline{PROG}$: Address Latch Enable output signal for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during EPROM programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$: Program Store Enable is the Read strobe to External Program Memory. When the 87C51 is executing from Internal Program Memory, $\overline{PSEN}$ is inactive (high). When the device is executing code from External Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to External Data Memory.

$\overline{EA}$/V_{PP}: External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the 87C51 to fetch code from External Program Memory locations starting at 0000H up to FFFFH. Note, however, that if either of the Lock Bits is programmed, the logic level at $\overline{EA}$ is internally latched during reset.

$\overline{EA}$ must be strapped to V_{CC} for internal program execution.

This pin also receives the 12.75V programming supply voltage (V_{PP}) during EPROM programming.

XTAL1: Input to the inverting oscillator amplifier and input to the internal clock generating circuits.

XTAL2: Output from the inverting oscillator amplifier.

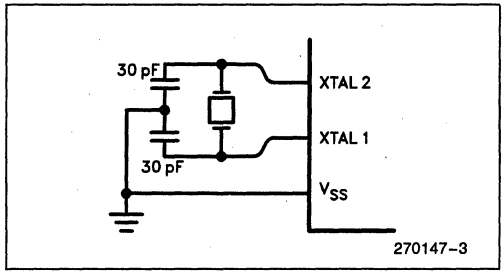


Figure 3. Using the On-Chip Oscillator

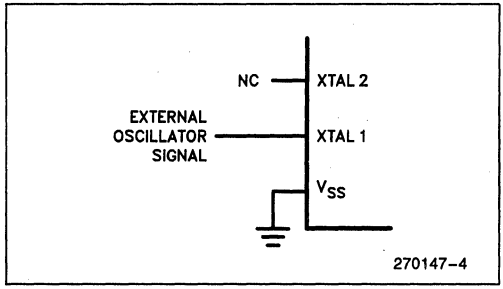


Figure 4. External Clock Drive

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3.

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

IDLE MODE

In Idle Mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the Special Functions Registers remain unchanged during this mode. The Idle Mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when Idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port

Table 1. Status of the external pins during Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

NOTE:
For more detailed information on the reduced power modes refer to current Embedded Controller Handbook, and Application Note AP-252, "Designing with the 80C51BH."

pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

POWER DOWN MODE

In the Power Down mode the oscillator is stopped, and the instruction that invokes Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

The only exit from Power Down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

DESIGN CONSIDERATIONS

Exposure to light when the device is in operation may cause logic errors. For this reason, it is suggested that an opaque label be placed over the window when the die is exposed to ambient light.

If using the 87C51 to prototype for the 80C51BH, consult the Design Considerations section of the 80C51BH data sheet.

PROGRAM MEMORY LOCK

The 87C51 contains two program memory lock schemes: Encrypted Verify and Lock Bits.

Encrypted Verify: The 87C51 implements a 32-byte EPROM array that can be programmed by the customer, and which can then be used to encrypt the program code bytes during EPROM verification. The EPROM verification procedure is performed as usual, except that each code byte comes out logically X-NORed with one of the 32 key bytes. The key bytes are gone through in sequence. Therefore, to read the ROM code, one has to know the 32 key bytes in their proper sequence.

Lock Bits: Also on the chip are two Lock Bits which can be left unprogrammed (U) or can be programmed (P) to obtain the following additional features:

Bit 1	Bit 2	Additional Features
U	U	none
P	U	• Externally fetched code can not access internal Program Memory. • Further programming disabled.
U	P	(Reserved for Future definition.)
P	P	• Externally fetched code can not access internal Program Memory. • Further programming disabled. • Program verification is disabled.

When Lock Bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

ONCE™ MODE

The ONCE ("on-circuit emulation") mode facilitates testing and debugging of systems using the 87C51 without the 87C51 having to be removed from the circuit. The ONCE mode is invoked by:

1. Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the 87C51 is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on $\overline{EA}$/V_{PP} Pin to V_{SS} .....0V to +13.0V
Voltage on Any Other Pin to V_{SS} ..−0.5V to +6.5V
Maximum I_{OL} per I/O Pin .....15 mA
Power Dissipation.....1.5W
(Based on package heat transfer limitations, not device power consumption).

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS: (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V)

Symbol	Parameter	Min	Typ(1)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$)	−0.5		.2V_{CC}−.1	V	
V_{IL1}	Input Low Voltage to $\overline{EA}$	0		.2V_{CC}−.3	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	.2V_{CC}+.9		V_{CC}+.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7V_{CC}		V_{CC}+.5	V	
V_{OL}	Output Low Voltage (Ports 1, 2, 3) (7)			0.45	V	I_{OL} = 1.6 mA (2)
V_{OL1}	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$) (7)			0.45	V	I_{OL} = 3.2 mA (2)
V_{OH}	Output High Voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$)	2.4			V	I_{OH} = −60 µA
		.75V_{CC}			V	I_{OH} = −25 µA
		.9V_{CC}			V	I_{OH} = −10 µA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4			V	I_{OH} = −800 µA
		.75 V_{CC}			V	I_{OH} = −300 µA
		.9V_{CC}			V	I_{OH} = −80 µA (3)
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)			−50	µA	V_{IN} = 0.45 V
I_{TL}	Logical 1-to-0 transition current (Ports 1, 2, 3)			−650	µA	(4)
I_{LI}	Input Leakage Current (Port 0)			±10	µA	V_{IN} = V_{IL} or V_{IH}
I_{CC}	Power Supply Current:					
	Active Mode @ 12 MHz (5)		11.5	25	mA	
	Idle Mode @ 12 MHz (5)		1.3	4	mA	(6)
	Power Down Mode		3	50	µA	
RRST	Internal Reset Pulldown Resistor	50		300	kΩ	
C_{IO}	Pin Capacitance			10	pF	

NOTES:
1. "Typicals" are based on a limited number of samples taken from early manufacturing lots and are not guaranteed. The values listed are at room temp, 5V.
2. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
3. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9V_{CC} specification when the address bits are stabilizing.
4. Pins of Ports 1, 2, and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V_{IN} is approximately 2V.
5. I_{CC}MAX at other frequencies is given by:

$$\text{Active Mode: } I_{CC}MAX = 0.94 \times FREQ + 13.71$$
$$\text{Idle Mode: } I_{CC}MAX = 0.14 \times FREQ + 2.31$$

where FREQ is the external oscillator frequency in MHz. I_{CC}MAX is given in mA. See Figure 5.

6. See Figures 6 through 9 for I_{CC} test conditions. Minimum V_{CC} for Power Down is 2V.
7. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port—
Port 0: 26 mA
Ports 1, 2, and 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink greater than the listed test conditions.

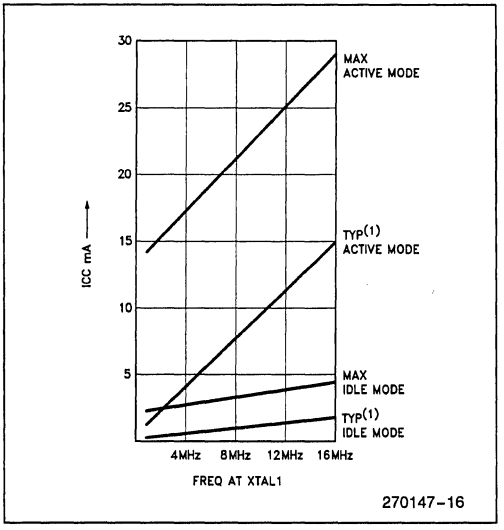


Figure 5. I_{CC} vs. FREQ. Valid only within frequency specifications of the device under test.

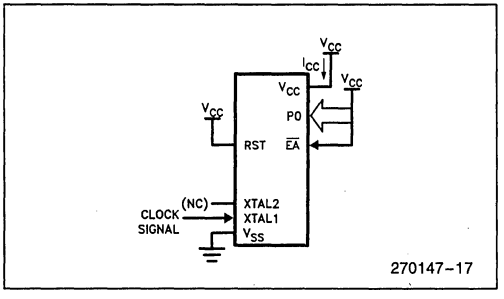


Figure 6. I_{CC} Test Condition, Active Mode. All other pins are disconnected.

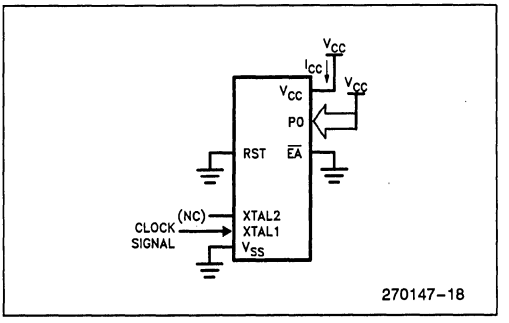


Figure 7. I_{CC} Test Condition, Idle Mode. All other pins are disconnected.

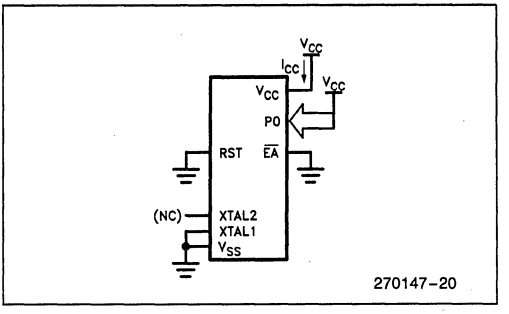


Figure 9. I_{CC} Test Condition, Power Down Mode. All other pins are disconnected. V_{CC} = 2V to 5.5V.

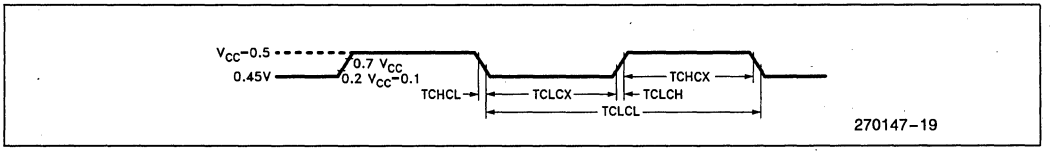


Figure 8. Clock Signal Waveform for I_{CC} tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A:Address.
C:Clock.
D:Input data.
H:Logic level HIGH.
I:Instruction (program memory contents).
L:Logic level LOW, or ALE.
P:$\overline{PSEN}$.
Q:Output data.
R:$\overline{RD}$ signal.
T:Time.
V:Valid.
W:$\overline{WR}$ signal.
X:No longer a valid logic level.
Z:Float.

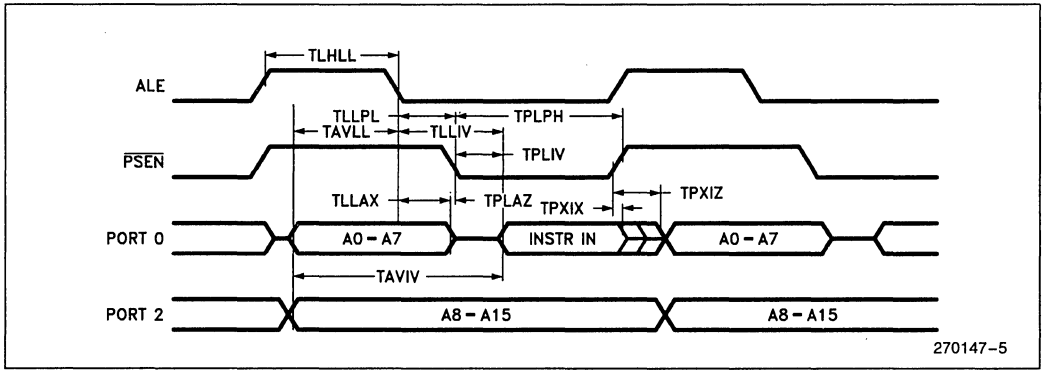
For example,

TAVLL = Time from Address Valid to ALE Low.
TLLPL = Time from ALE Low to PSEN Low.

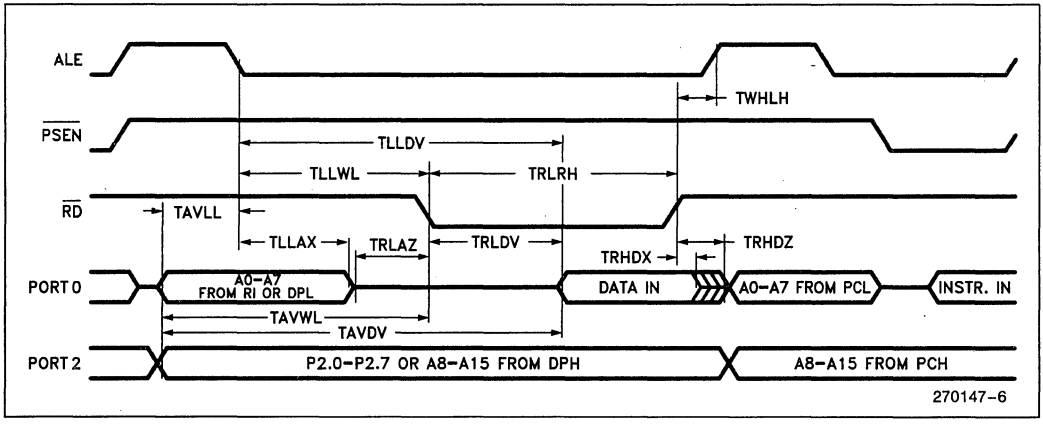
A.C. CHARACTERISTICS: (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF; Load Capacitance for All Other Outputs = 80 pF)

EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

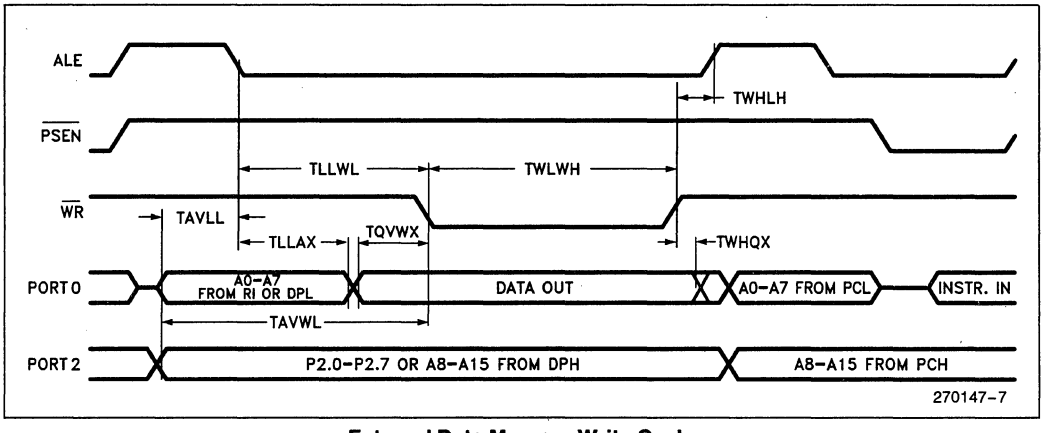
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 87C51 87C51-1 87C51-2			 3.5 3.5 0.5	 12 16 12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	28		TCLCL−55		ns
TLLAX	Address Hold After ALE Low	48		TCLCL−35		ns
TLLIV	ALE Low to Valid Instr In		234		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	43		TCLCL−40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	205		3TCLCL−45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr In		145		3TCLCL−105	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		59		TCLCL−25	ns
TAVIV	Address to Valid Instr In		312		5TCLCL−105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL−70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		TCLCL−60		ns
TWHQX	Data Hold After $\overline{WR}$	33		TCLCL−50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL−40	TCLCL+40	ns



External Program Memory Read Cycle



External Data Memory Read Cycle

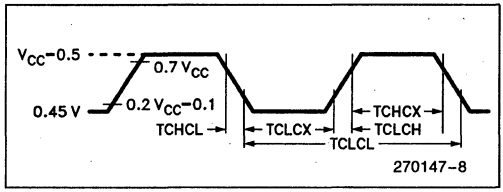


External Data Memory Write Cycle

EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency 87C51 87C51-1 87C51-2	 3.5 3.5 0.5	 12 16 12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

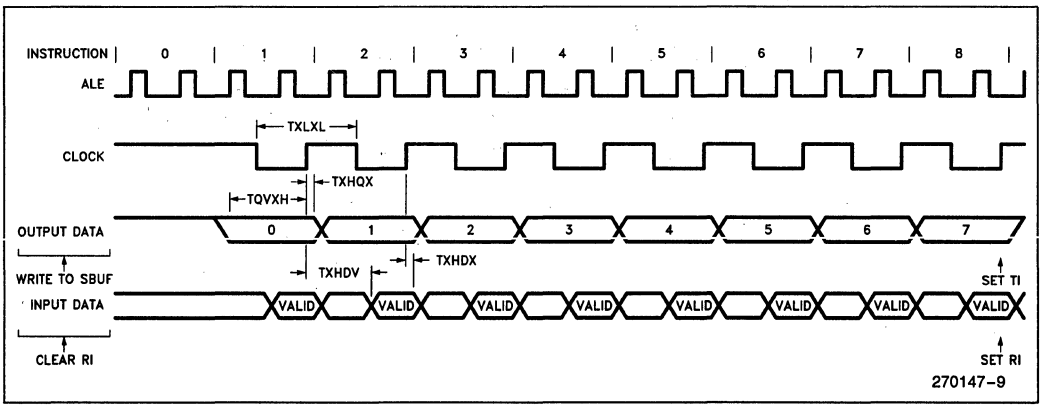
EXTERNAL CLOCK DRIVE WAVEFORM



SERIAL PORT TIMING—SHIFT REGISTER MODE

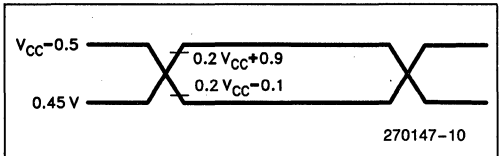
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold After Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns

SHIFT REGISTER MODE TIMING WAVEFORMS



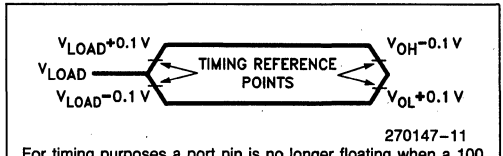
A.C. TESTING:

INPUT, OUTPUT WAVEFORMS



AC inputs during testing are driven at V_{CC} − 0.5 for a Logic "1" and 0.45V for a Logic "0." Timing measurements are made at V_{IH} min for a Logic "1" and V_{IL} max for a Logic "0".

FLOAT WAVEFORM



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

EPROM CHARACTERISTICS

The 87C51 is programmed by a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for V_{PP} (Programming Supply Voltage) and in the width and number of the ALE/$\overline{PROG}$ pulses.

The 87C51 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51 manufactured by Intel.

Table 2 shows the logic levels for reading the signature byte, and for programming the Program Memory, the Encryption Table, and the Lock Bits. The circuit configuration and waveforms for Quick-Pulse Programming™ are shown in Figures 10 and 11. Figure 12 shows the circuit configuration for normal Program Memory verification.

Table 2. EPROM Programming Modes

MODE	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.7	P2.6	P3.7	P3.6
Read Signature	1	0	1	1	0	0	0	0
Program Code Data	1	0	0*	V_{PP}	1	0	1	1
Verify Code Data	1	0	1	1	0	0	1	1
Pgm Encryption Table	1	0	0*	V_{PP}	1	0	1	0
Pgm Lock Bit 1	1	0	0*	V_{PP}	1	1	1	1
Pgm Lock Bit 2	1	0	0*	V_{PP}	1	1	0	0

NOTES:
"1" = Valid high for that pin
"0" = Valid low for that pin
V_{PP} = 12.75V ± 0.25V
V_{CC} = 5V ± 10% during programming and verification
*ALE/$\overline{PROG}$ receives 25 programming pulses while V_{PP} is held at 12.75V. Each programming pulse is low for 100 μS(±10 μS) and high for a minimum of 10 μS.

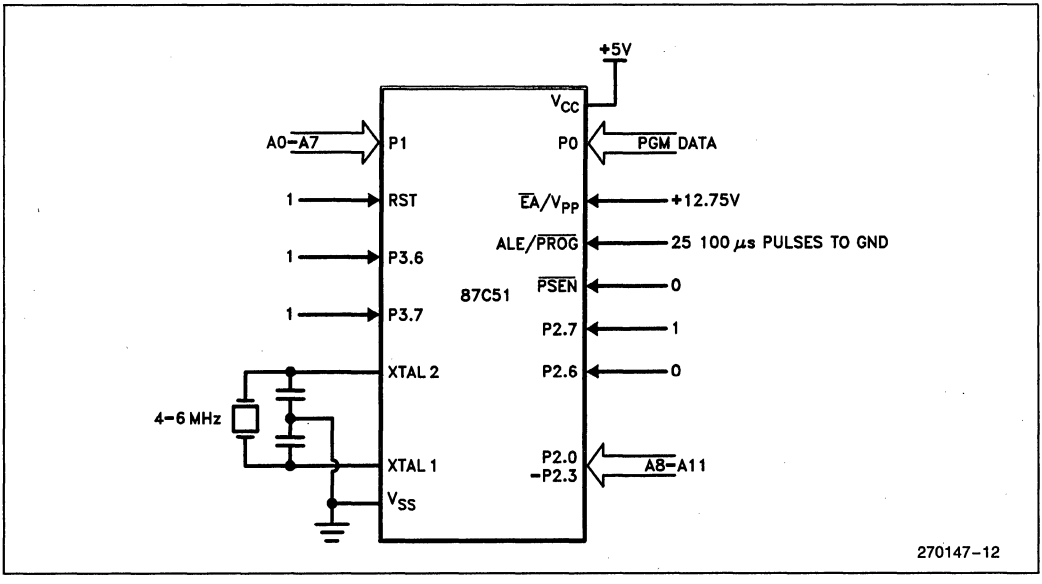


Figure 10. Programming Configuration

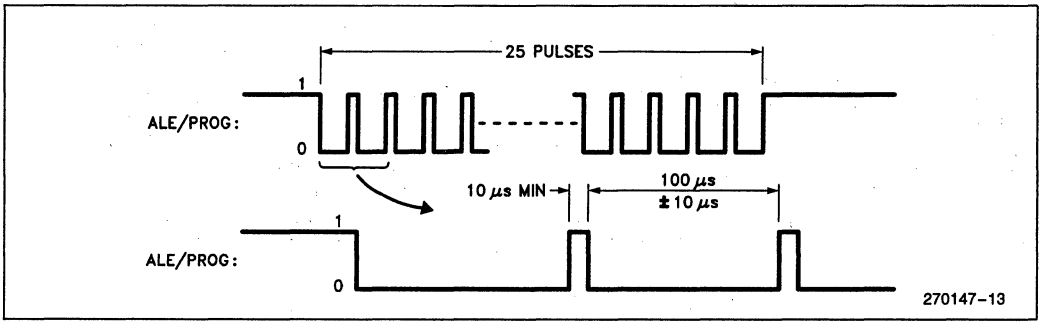


Figure 11. $\overline{\text{PROG}}$ Waveforms

Quick-Pulse Programming™

The setup for Microcontroller Quick-Pulse Programming™ is shown in Figure 10. Note that the 87C51 is running with a 4 to 6 MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to Ports 1 and 2, as shown in Figure 10. The code byte to be programmed into that location is applied to Port 0. RST, $\overline{\text{PSEN}}$, and pins of Ports 2 and 3 specified in Table 2 are held at the "Program Code Data" levels indicated in Table 2. Then ALE/$\overline{\text{PROG}}$ is pulsed low 25 times as shown in Figure 11.

To program the Encryption Table, repeat the 25-pulse programming sequence for addresses 0 through 1FH, using the "Pgm Encryption Table" levels. Don't forget that after the Encryption Table is programmed, verify cycles will produce only encrypted data.

To program the Lock Bits, repeat the 25-pulse programming sequence using the "Pgm Lock Bit" levels. After one Lock Bit is programmed, further programming of the Code Memory and Encryption Table is disabled. However, the other Lock Bit can still be programmed.

Note that the $\overline{\text{EA}}$/V_{PP} pin must not be allowed to go above the maximum specified V_{PP} level for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The V_{PP} source should be well regulated and free of glitches and overshoot.

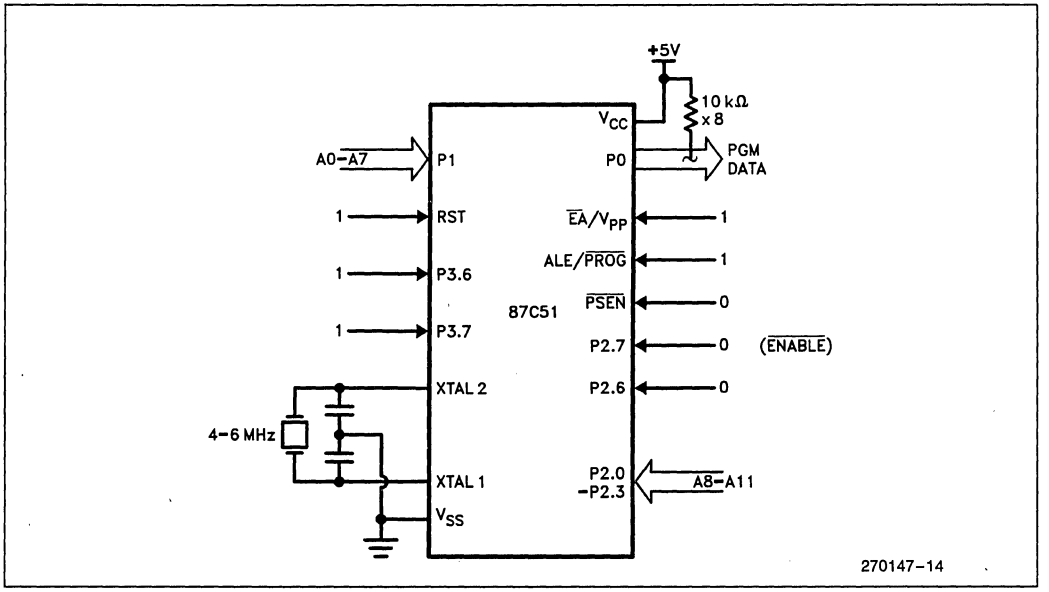


Figure 12. Program Verification

Program Verification

If Lock Bit 2 has not been programmed, the on-chip Program Memory can be read out for program verification. The address of the Program Memory location to be read is applied to Ports 1 and 2 as shown in Figure 12. The other pins are held at the "Verify Code Data" levels indicated in Table 2. The contents of the addressed location will be emitted on Port 0. External pullups are required on Port 0 for this operation. Detailed timing specifications are shown in later sections of this data sheet.

If the Encryption Table has been programmed, the data presented at Port 0 will be the Exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the Encryption Table contents in order to correctly decode the verification data. The Encryption Table itself can not be read out.

Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

(030H) = 89H indicates manufactured by Intel

(031H) = 57H indicates 87C51

Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 2, and which satisfies the timing specifications, is suitable.

Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 Angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. If an application subjects the device to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 Angstroms) to an integrated dose of at least 15 W-sec/cm^2. Exposing the EPROM to an ultraviolet lamp of 12,000 μW/cm^2 rating for 30 minutes, at a distance of about 1 inch, should be sufficient.

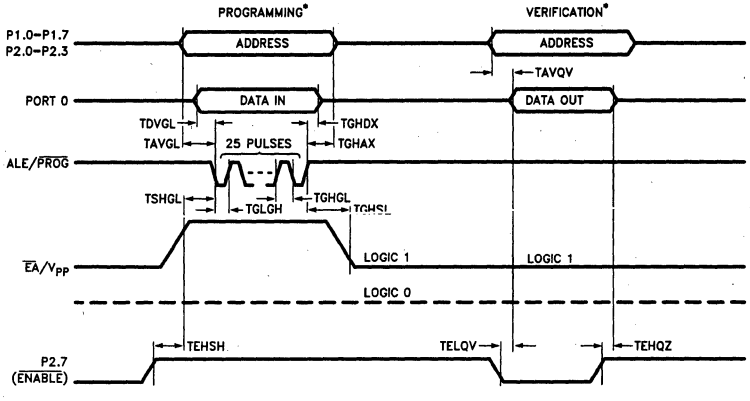
Erasure leaves the array in an all 1s state.

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS:

(T_A = 21°C to 27°C, V_{CC} = 5V ±10%, V_{SS} = 0V)

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	12.5	13.0	V
I_{PP}	Programming Supply Current		50	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$ Low	48TCLCL		
TGHAX	Address Hold After $\overline{PROG}$	48TCLCL		
TDVGL	Data Setup to $\overline{PROG}$ Low	48TCLCL		
TGHDX	Data Hold After $\overline{PROG}$	48TCLCL		
TEHSH	P2.7 ($\overline{ENABLE}$) High to V_{PP}	48TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
TGHSL	V_{PP} Hold After $\overline{PROG}$	10		µs
TGLGH	$\overline{PROG}$ Width	90	110	µs
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{ENABLE}$ Low to Data Valid		48TCLCL	
TEHQZ	Data Float After $\overline{ENABLE}$	0	48TCLCL	
TGHGL	$\overline{PROG}$ High to $\overline{PROG}$ Low	10		µs

EPROM Programming and Verification Waveforms



270147–15

*FOR PROGRAMMING CONDITIONS SEE FIGURE 10.
FOR VERIFICATION CONDITIONS SEE FIGURE 12.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -004 version of the 87C51BH data sheet:

1. Package table was added.
2. Note 7 on maximum current specifications added to DC Characteristics.
3. Data Sheet Revision Summary was added.

87C51
EXPRESS

- **Extended Temperature Range**
- **Burn-In**
- **3.5 MHz to 12 MHz V_{CC} = 5V ±10%**

The Intel EXPRESS system offers enhancements to the operational specifications of the MCS®-51 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in and an extended temperature range with or without burn-in.

With the commercial standard temperature range, operational characteristics are guaranteed over the temperature range of 0°C to +70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

The optional burn-in is dynamic for a minimum time of 160 hours at 125°C with V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

September 1988
Order Number: 270430-002

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets.

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage (Except EA)	-0.5	$0.2V_{CC} - 0.15$	V	
V_{IL1}	EA	0	$0.2V_{CC} - 0.35$	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	$0.2V_{CC} + 1$	$V_{CC} + 0.5$	V	
V_{IH1}	Input High Voltage to XTAL1, RST	$0.7V_{CC} + 0.1$	$V_{CC} + 0.5$	V	
I_{IL}	Logical 0 Input Current (Port 1, 2, 3)		-75	μA	$V_{IN} = 0.45V$
I_{TL}	Logical 1 to 0 transition Current (Ports 1, 2, 3)		-750	μA	$V_{IN} = 2.0V$
I_{CC}	Power Supply Current				(Note 1)
	Active Mode		35	mA	
	Idle Mode		6	mA	
	Power Down Mode		50	μA	

NOTE:
1. $V_{CC} = 4.5V-5.5V$, Frequency Range = 3.5 MHz–12 MHz.

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range(2)	Burn-In(3)
P	Plastic	Commercial	No
D	Cerdip	Commercial	No
N	PLCC	Commercial	No
TP	Plastic	Extended	No
TD	Cerdip	Extended	No
TN	PLCC	Extended	No
QP	Plastic	Commercial	Yes
QD	Cerdip	Commercial	Yes
QN	PLCC	Commercial	Yes
LP	Plastic	Extended	Yes
LD	Cerdip	Extended	Yes
LN	PLCC	Extended	Yes

NOTES:
2. Commercial temperature range is 0°C to +70°C. Extended temperature range is −40°C to +85°C.
3. Burn-in is dynamic for a minimum time of 160 hours at +125°C, V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).

Examples:

P87C51 indicates 87C51 in a plastic package and specified for commercial temperature range, without burn-in.
LD87C51 indicates 87C51 in a cerdip package and specified for extended temperature range with burn-in.

83C51FA/FB

9

intel

October 1988

Hardware Description of the 83C51FA and 83C51FB

Order Number: 270653-001

HARDWARE DESCRIPTION OF THE 83C51FA AND 83C51FB

CONTENTS

CONTENTS

CONTENTS

HARDWARE DESCRIPTION OF THE 83C51FA/FB

1.0 INTRODUCTION

The 80C51FA and 80C51FB are highly integrated 8-bit microcontrollers based on the MCS®-51 architecture. Their key feature is the programmable counter array (PCA) which is capable of measuring and generating pulse information on five I/O pins. Also included are an enhanced serial port for multi-processor communications, an up/down timer/counter, and a program lock scheme for the on-chip program memory. Since these products are CHMOS, they have two software selectable reduced power modes: Idle Mode and Power Down Mode. As a member of the MCS-51 family, the 80C51FA/FB are optimized for control applications.

This document presents a comprehensive description of the on-chip hardware features of the 8XC51FA/FB. It begins with a discussion of the on-chip memory and then discusses each of the peripherals as follows:

- 8K Bytes On-Chip EPROM/ROM (on the 87C51FA/83C51FA)
- 16K Bytes On-Chip EPROM/ROM (on the 87C51FB/83C51FB)
- 256 Bytes On-Chip Data RAM
- Special Function Registers (SFR)
- Four 8-bit bidirectional parallel ports
- Three 16-bit Timer/Counters with
  - — One Up/Down Timer/Counter
- Programmable Counter Array with
  - — Compare/Capture
  - — Software Timer
  - — High Speed Output
  - — Pulse Width Modulator
  - — Watchdog Timer
- Full-Duplex Programmable Serial Interface with
  - — Framing Error Detection
  - — Automatic Address Recognition
- Interrupt Structure with
  - — Seven interrupt sources
  - — Two priority levels
- Reduced Power Modes
  - — Idle Mode
  - — Power Down Mode

The 8XC51FA/FB uses the standard 8051 instruction set and is pin-for-pin compatible with the existing MCS-51 family of products. The 83C51FA/FB is the factory masked ROM device; the 80C51FA is the ROMless device; and the 87C51FA/FB is the EPROM device. The designation 8XC51FA/FB refers to any of these devices.

Figure 1 shows a functional block diagram of the 8XC51FA/FB.

2.0 MEMORY ORGANIZATION

All MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

2.1 Program Memory

The only difference between the 8XC51FA and 8XC51FB is the program memory size. The 8XC51FA has 8K bytes of internal ROM or EPROM, whereas the 8XC51FB has 16K bytes.

If the $\overline{EA}$ pin is connected to V_{SS}, all program fetches are directed to external memory. On the 83C51FA (or 87C51FA), if the $\overline{EA}$ pin is connected to V_{CC}, then program fetches to addresses 0000H through 1FFFH are directed to internal ROM and fetches to addresses 2000H through FFFFH are to external memory.

On the 83C51FB (or 87C51FB) if $\overline{EA}$ is connected to VCC, program fetches to addresses 0000H through 3FFFH are directed to internal ROM, and fetches to addresses 4000H through FFFFH are to external memory.

2.2 Data Memory

The 8XC51FA/FB implement 256 bytes of on-chip data RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means they have the same addresses, but they are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the CPU knows whether the access is to the upper 128 bytes of data RAM or to SFR space by the addressing mode used in the instruction. Instructions that use direct addressing access SFR space. For example,

```
MOV 0A0H, #data
```

accesses the SFR at location 0A0H (which is P2). Instructions that use indirect addressing access the upper 128 bytes of data RAM. For example,

```
MOV @R0, #data
```

where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H). Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

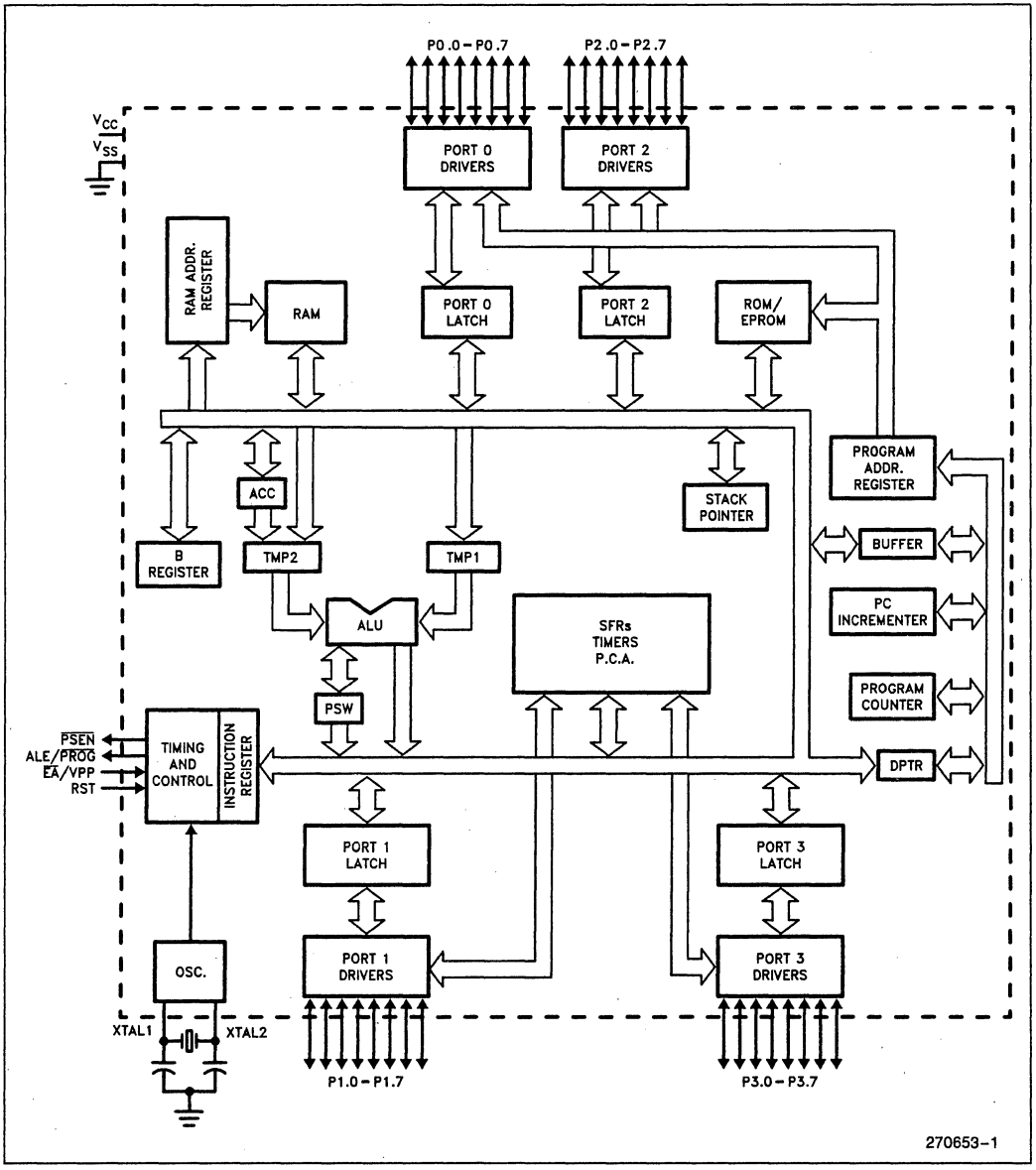


Figure 1. 8XC51FA/FB Functional Block Diagram

3.0 SPECIAL FUNCTION REGISTERS

A map of the on-chip memory area called the SFR (Special Function Register) space is shown in Table 1. Note that not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have no effect.

User software should not write 1s to these unimplemented locations, since they may be used in future MCS-51 products to invoke new features. In that case the reset or inactive values of the new bits will always be 0, and their active values will be 1.

The functions of the SFRs are outlined below. More information on the use of specific SFRs for each peripheral is included in the description of that peripheral.

Accumulator ACC is the Accumulator register. The mnemonics for Accumulator-Specific instructions, however, refer to the Accumulator simply as A.

Table 1. SFR Mapping and Reset Values

F8		CH 00000000	CCAP0H XXXXXXXX	CCAP1H XXXXXXXX	CCAP2H XXXXXXXX	CCAP3H XXXXXXXX	CCAP4H XXXXXXXX		FF
F0	* B 00000000								F7
E8		CL 00000000	CCAP0L XXXXXXXX	CCAP1L XXXXXXXX	CCAP2L XXXXXXXX	CCAP3L XXXXXXXX	CCAP4L XXXXXXXX		EF
E0	* ACC 00000000								E7
D8	CCON 00X00000	CMOD 00XXX000	CCAPM0 X0000000	CCAPM1 X0000000	CCAPM2 X0000000	CCAPM3 X0000000	CCAPM4 X0000000		DF
D0	* PSW 00000000								D7
C8	T2CON 00000000	T2MOD XXXXXXX0	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			CF
C0									C7
B8	* IP X0000000	SADEN 00000000							BF
B0	* P3 11111111								B7
A8	* IE 00000000	SADDR 00000000							AF
A0	* P2 11111111								A7
98	* SCON 00000000	* SBUF XXXXXXXX							9F
90	* P1 11111111								97
88	* TCON 00000000	* TMOD 00000000	* TL0 00000000	* TL1 00000000	* TH0 00000000	* TH1 00000000			8F
80	* P0 11111111	* SP 00000111	* DPL 00000000	* DPH 00000000				* PCON ** 00XX0000	87

* = Found in the 8051 core (See 8051 Hardware Description for explanations of these SFRs).
** = See description of PCON SFR. Bit PCON.4 is not affected by reset.
X = Undefined.

Table 2. PSW: Program Statue Word Register

PSW Address = 0D0H Reset Value = 0000 0000B

Bit Addressable

	CY	AC	F0	RS1	RS0	OV	—	P
Bit	7	6	5	4	3	2	1	0

Symbol	Function
CY	Carry flag.
AC	Auxiliary Carry flag. (For BCD Operations)
F0	Flag 0. (Available to the user for general purposes).
RS1	Register bank select bit 1.
RS0	Register bank select bit 0.

RS1	RS0	Working Register Bank and Address	
0	0	Bank 0	(00H–07H)
0	1	Bank 1	(08H–0FH)
1	0	Bank 2	(10H–17H)
1	1	Bank 3	(18H–1FH)

OV	Overflow flag.
—	User definable flag.
P	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the Accumulator, i.e., even parity.

B Register The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

Stack Pointer The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. The stack may reside anywhere in on-chip RAM. On reset, the Stack Pointer is initialized to 07H causing the stack to begin at location 08H.

Data Pointer The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address, but it may be manipulated as a 16-bit register or as two independent 8-bit registers.

Program Status Word The PSW register contains program status information as detailed in Table 2.

Ports 0 to 3 Registers P0, P1, P2, and P3 are the SFR latches of Port 0, Port 1, Port 2, and Port 3 respectively.

Timer Registers Register pairs (TH0, TL0), (TH1, TL1), and (TH2, TL2) are the 16-bit count registers for Timer/Counters 0, 1, and 2 respectively. Control and status bits are contained in registers TCON and TMOD for Timers 0 and 1 and in registers T2CON and T2MOD for Timer 2. The register pair (RCAP2H, RCAP2L) are the capture/reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Programmable Counter Array (PCA) Registers The 16-bit PCA timer/counter consists of registers CH and CL. Registers CCON and CMOD contain the control and status bits for the PCA. The CCAPMn (n = 0, 1, 2, 3, or 4) registers control the mode for each of the five PCA modules. The register pairs (CCAPnH, CCAPnL) are the 16-bit compare/capture registers for each PCA module.

Serial Port Registers The Serial Data Buffer, SBUF, is actually two separate registers: a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF initiates the transmission). When data is moved from SBUF, it comes from the receive buffer. Register SCON contains the control and status bits for the Serial Port. Registers SADDR and SADEN are used to define the Given and the Broadcast addresses for the Automatic Address Recognition feature.

Interrupt Registers The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the 7 interrupts in the IP register.

Power Control Register PCON controls the Power Reduction Modes. Idle and Power Down Modes.

4.0 PORT STRUCTURES AND OPERATION

All four ports in the 8XC51FA/FB are bidirectional. Each consists of a latch (Special Function Registers P0 through P3), an output driver, and an input buffer.

The output drivers of Ports 0 and 2, and the input buffers of Port 0, are used in accesses to external memory. In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content.

All the Port 1 and Port 3 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed in Table 3.

The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise the port pin is stuck at 0.

4.1 I/O Configurations

Figure 2 shows a functional diagram of a typical bit latch and I/O buffer in each of the four ports. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which clocks in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal. See the Read-Modify-Write Feature section.

As shown in Figure 2, the output drivers of Ports 0 and 2 are switchable to an internal ADDRESS and ADDRESS/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the P0 SFR gets 1s written to it.

Table 3. Alternate Port Functions

Port Pin	Alternate Function
P0.0/AD0–P0.7/AD7	Multiplexed Byte of Address/Data for External Memory
P1.0/T2	Timer 2 External Clock Input
P1.1/T2EX	Timer 2 Reload/Capture/Direction Control
P1.2/ECI	PCA External Clock Input
P1.3/CEX3	PCA Module 0 Capture Input, Compare/PWM Output
P1.4/CEX4	PCA Module 1 Capture Input, Compare/PWM Output
P1.5/CEX5	PCA Module 2 Capture Input, Compare/PWM Output
P1.6/CEX6	PCA Module 3 Capture Input, Compare/PWM Output
P1.7/CEX7	PCA Module 4 Capture Input, Compare/PWM Output
P2.0/A8–P2.7/A15	High Byte of Address for External Memory
P3.0/RXD	Serial Port Input
P3.1/TXD	Serial Port Output
P3.2/$\overline{\text{INT0}}$	External Interrupt 0
P3.3/$\overline{\text{INT}}$	External Interrupt 1
P3.4/T0	Timer 0 External Clock Input
P3.5/T1	Timer 1 External Clock Input
P3.6/$\overline{\text{WR}}$	Write Strobe for External Memory
P3.7/$\overline{\text{RD}}$	Read Strobe for External Memory

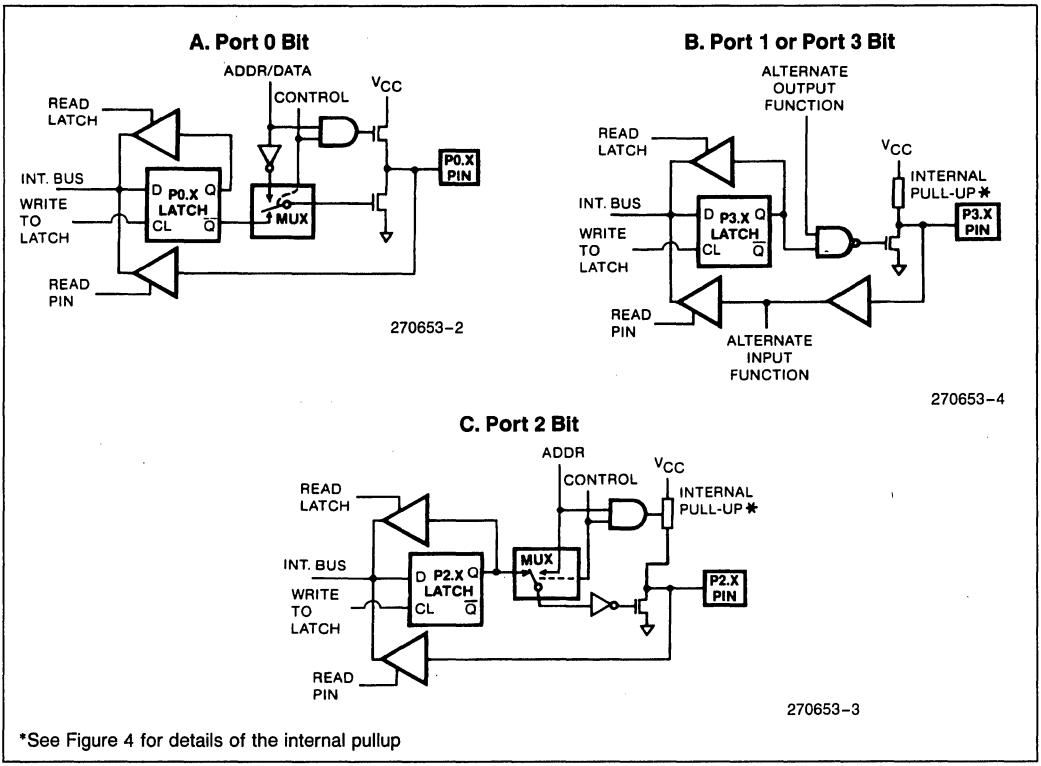


Figure 2. 8XC51FA/FB Port Bit Latches and I/O Buffers

Also shown in Figure 2 is that if a P1 or P3 latch contains a 1, then the output level is controlled by the signal labeled "alternate output function." The actual pin level is always available to the pin's alternate input function, if any.

Ports 1, 2, and 3 have internal pullups. Port 0 has open drain outputs. Each I/O line can be independently used as an input or an output (Ports 0 and 2 may not be used as general purpose I/O when being used as the ADDRESS/DATA BUS). To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. On Ports 1, 2, and 3, the pin is pulled high by the internal pullup, but can be pulled low by an external source.

Port 0 differs from the other ports in not having internal pullups. The pullup FET in the P0 output driver (see Figure 2) is used only when the Port is emitting 1s during external memory accesses. Otherwise the pullup FET is off. Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, which floats the pin and allows it to be used as a high-impedance input. Because Ports 1 through 3 have fixed internal pullups they are sometimes call "quasi-bidirectional" ports. When configured as inputs they pull high and will source current (IIL in the data sheets) when externally pulled low. Port 0, on the other hand, is considered "true" bidirectional, because it floats when configured as an input.

All the port latches have 1s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writing a 1 to it.

4.2 Writing to a Port

In the execution of an instruction that changes the value in a port latch, the new value arrives at the latch during State 6 Phase 2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during Phase 1 of any clock period. (During Phase 2 the output buffer holds the value it saw during the previous Phase 1). Consequently, the new value in the port latch won't actually appear at the output pin until the next Phase 1, which will be at S1P1 of the next machine cycle. Refer to Figure 3. For more information on internal timings refer to the CPU Timing section.

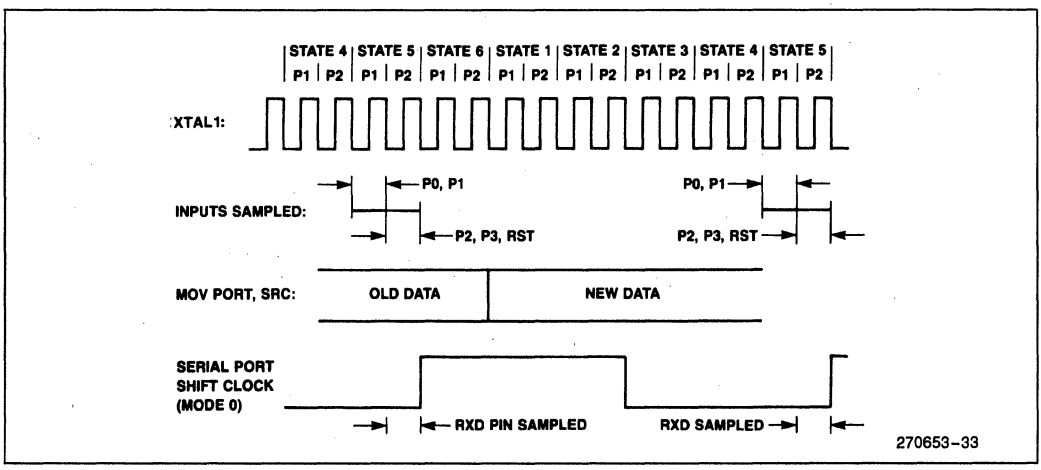


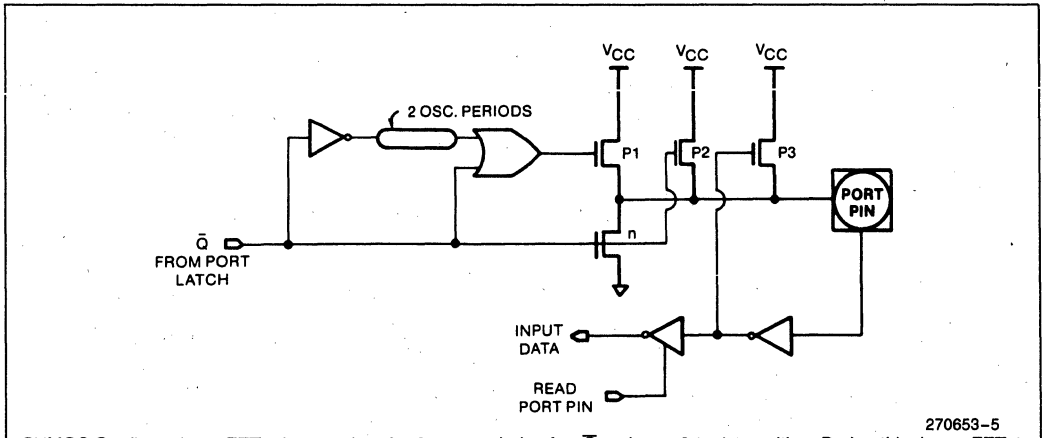
Figure 3. Port Operation

If the change requires a 0-to-1 transition in Ports 1, 2, and 3, an additional pullup is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pullup can source about 100 times the current that the normal pullup can. The internal pullups are field-effect transistors, not linear resistors. The pull-up arrangements are shown in Figure 4.

The pullup consists of three pFETs. Note that an n-channel FET (nFET) is turned on when a logical 1 is applied to its gate, and is turned off when a logical 0 is applied to its gate. A p-channel FET (pFET) is the opposite: it is on when its gate sees a 0, and off when its gate sees a 1.

pFET 1 in is the transistor that is turned on for 2 oscillator periods after a 0-to-1 transition in the port latch. A 1 at the port pin turns on pFET3 (a weak pull-up), through the invertor. This invertor and pFET form a latch which hold the 1.

If the pin is emitting a 1, a negative glitch on the pin from some external source can turn off pFET3, causing the pin to go into a float state. pFET2 is a very weak pullup which is on whenever the nFET is off, in traditional CMOS style. It's only about $1/10$ the strength of pFET3. Its function is to restore a 1 to the pin in the event the pin had a 1 and lost it to a glitch.



CHMOS Configuration. pFET 1 is turned on for 2 osc. periods after $\overline{Q}$ makes a 0-to-1 transition. During this time, pFET 1 also turns on pFET 3 through the inverter to form a latch which holds the 1. pFET 2 is also on. Port 2 is similar except that it holds the strong pullup on while emitting 1s that are address bits. (See text, "Accessing External Memory".)

Figure 4. Ports 1 and 3 Internal Pullup Configurations

4.3 Port Loading and Interfacing

The output buffers of Ports 1, 2, and 3 can each sink 1.6 mA at 0.45 V. These port pins can be driven by open-collector and open-drain outputs although 0-to-1 transitions will not be fast since there is little current pulling the pin up. An input 0 turns off pullup pFET3, leaving only the very weak pullup pFET2 to drive the transition.

In external bus mode, Port 0 output buffers can each sink 3.2 mA at 0.45 V. However, as port pins they require external pullups to be able to drive any inputs.

See the latest revision of the data sheet for design-in information.

4.4 Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Which ones do which? The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions. Listed below are the read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin:

ANL	(logical AND, e.g., ANL P1, A)
ORL	(logical OR, e.g., ORL P2, A)
XRL	(logical EX-OR, e.g., XRL P3, A)
JBC	(jump if bit = 1 and clear bit, e.g., JBC P1.1, LABEL)
CPL	(complement bit, e.g., CPL P3.0)
INC	(increment, e.g., INC P2)
DEC	(decrement, e.g., DEC P2)
DJNZ	(decrement and jump if not zero, e.g., DJNZ P3, LABEL)
MOV, PX.Y, C	(move carry bit to bit Y of Port X)
CLR PX.Y	(clear bit Y of Port X)
SETB PX.Y	(set bit Y of Port X)

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as a 0. Reading the latch rather than the pin will return the correct value of 1.

4.5 Accessing External Memory

Accesses to external memory are of two types: accesses to external Program Memory and accesses to external Data Memory. Accesses to external Program Memory use signal $\overline{PSEN}$ (program store enable) as the read strobe. Accesses to external Data Memory use $\overline{RD}$ or $\overline{WR}$ (alternate functions of P3.7 and P3.6) to strobe the memory. Refer to Figures 5 through 7.

Fetches from external Program Memory always use a 16-bit address. Accesses to external Data Memory can use either a 16-bit address (MOVX @ DPTR) or an 8-bit address (MOVX @ Ri).

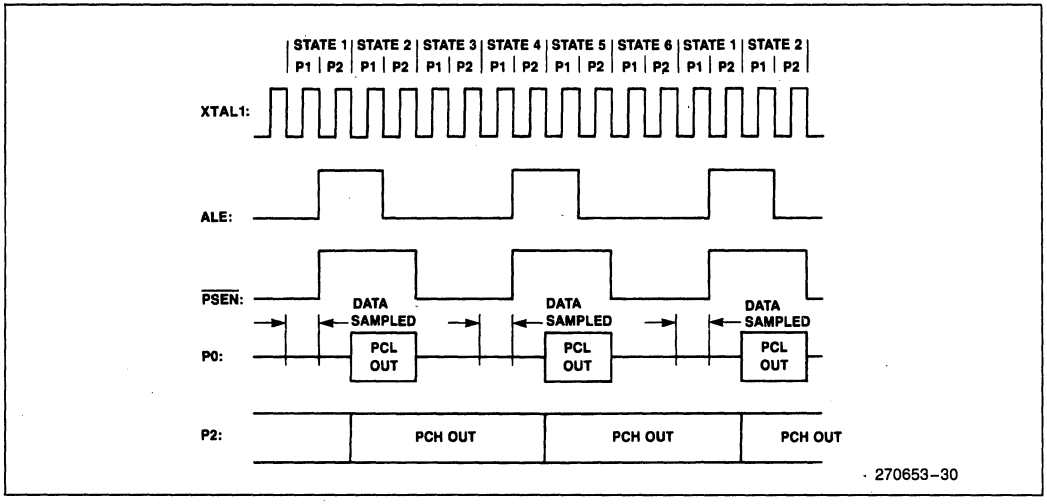


Figure 5. External Program Memory Fetches

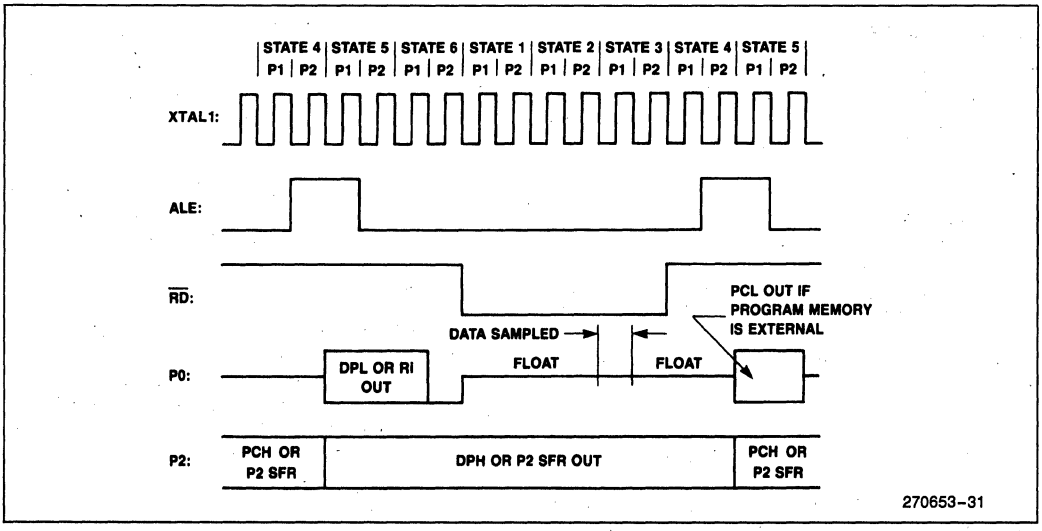


Figure 6. External Data Memory Read Cycle

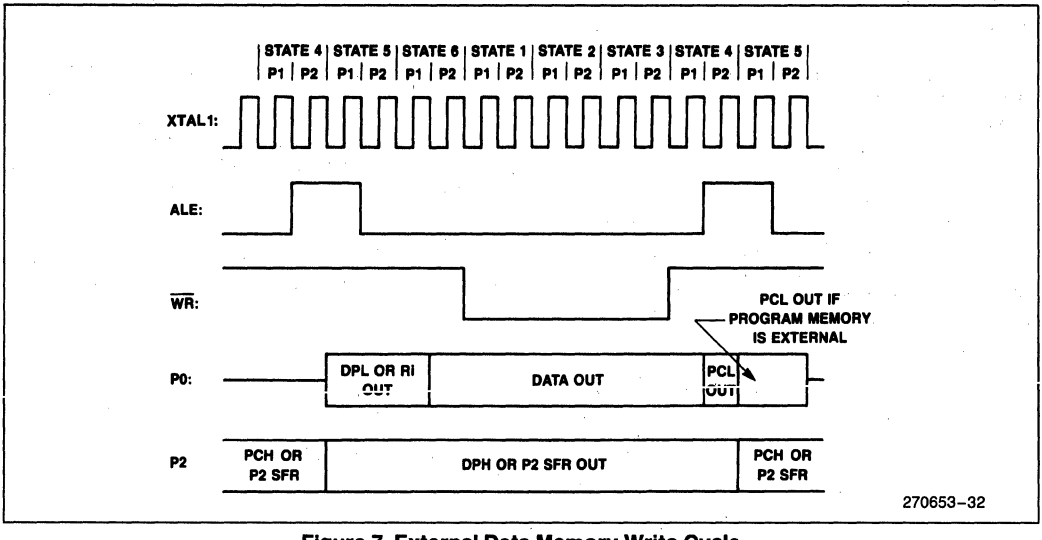


Figure 7. External Data Memory Write Cycle

Whenever a 16-bit address is used, the high byte of the address comes out on Port 2, where it is held for the duration of the read or write cycle. The Port 2 drivers use the strong pullups during the entire time that they are emitting address bits that are 1s. This occurs when the MOVX @ DPTR instruction is executed. During this time the Port 2 latch (the Special Function Register) does not have to contain 1s, and the contents of the Port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port 2 SFR will reappear in the next cycle.

If an 8-bit address is being used (MOVX @ Ri), the contents of the Port 2 SFR remain at the Port 2 pins throughout the external memory cycle. In this case, Port 2 pins can be used to page the external data memory.

In either case, the low byte of the address is time-multiplexed with the data byte on Port 0. The ADDRESS/DATA signal drives both FETs in the Port 0 output buffers. Thus, in external bus mode the Port 0 pins are not open-drain outputs and do not require external pullups. The ALE (Address Latch Enable) signal should be used to capture the address byte into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on Port 0 just before $\overline{WR}$ is activated, and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe ($\overline{RD}$) is deactivated.

During any access to external memory, the CPU writes 0FFH to the Port 0 latch (the Special Function Register), thus obliterating the information in the Port 0 SFR. Also, a MOV P0 instruction must not take place during external memory accesses. If the user writes to Port 0 during an external memory fetch, the incoming code byte is corrupted. Therefore, do not write to Port 0 if external program memory is used.

External Program Memory is accessed under two conditions:

1. Whenever signal $\overline{EA}$ is active, or
2. Whenever the program counter (PC) contains an address greater than 1FFFH (8K) for the 8XC51FA or 3FFFH (16K) for the 8XC51FB.

This requires that the ROMless versions have $\overline{EA}$ wired to V_{SS} enable the lower 8K or 16K program bytes to be fetched from external memory.

When the CPU is executing out of external Program Memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches they output the high byte of the PC with the Port 2 drivers using the strong pullups to emit bits that are 1s.

5.0 TIMERS/COUNTERS

The 8XC51FA/FB has three 16-bit Timer/Counters: Timer 0, Timer 1, and Timer 2. Each consists of two 8-bit registers, THx and TLx, (x = 0, 1, and 2). All three can be configured to operate either as timers or event counters.

In the Timer function, the TLx register is incremented every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin—T0, T1, or T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is $1/_{24}$ of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

In addition to the Timer or Counter selection, Timer 0 and Timer 1 have four operating modes from which to select: Modes 0 – 3. Timer 2 has three modes of operation: Capture, Auto-Reload, and Baud Rate Generator.

5.1 Timer 0 and Timer 1

The Timer or Counter function is selected by control bits C/$\overline{T}$ in the Special Function Register TMOD (Table 4). These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timer/Counters. Mode 3 operation is different for the two timers.

MODE 0

Either Timer 0 or Timer 1 in Mode 0 is an 8-bit Counter with a divide-by-32 prescaler. Figure 8 shows the Mode 0 operation for either timer.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TFx. The counted input is enabled to the Timer when TRx = 1 and either GATE = 0 or $\overline{INTx}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INTx}$, to facilitate pulse width measurements). TRx and TFx are

control bits in SFR TCON (Table 5). The GATE bit is in TMOD. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

The 13-bit register consists of all 8 bits of THx and the lower 5 bits of TLx. The upper 3 bits of TLx are indeterminate and should be ignored. Setting the run flag (TRx) does not clear these registers.

MODE 1

Mode 1 is the same as Mode 0, except that the Timer register uses all 16 bits. Refer to Figure 9. In this mode, THx and TLx are cascaded; there is no prescaler.

MODE 2

Mode 2 configures the Timer register as an 8-bit Counter (TLx) with automatic reload, as shown in Figure 10. Overflow from TLx not only sets TFx, but also reloads TLx with the contents of THx, which is preset by software. The reload leaves THx unchanged.

Table 4. TMOD: Timer/Counter Mode Control Register

TMOD Address = 89H Reset Value = 0000 0000B

Not Bit Addressable

	TIMER 1				TIMER 0			
	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Bit	7	6	5	4	3	2	1	0

Symbol	Function
GATE	Gating control when set. Timer/Counter 0 or 1 is enabled only while $\overline{INT0}$ or $\overline{INT1}$ pin is high and TR0 or TR1 control pin is set. When cleared, Timer 0 or 1 is enabled whenever TR0 or TR1 control bit is set.
C/$\overline{T}$	Timer or Counter Selector. Clear for Timer operation (input from internal system clock). Set for Counter operation (input from T0 or T1 input pin).

M1	M0	Operating Mode
0	0	8-bit Timer/Counter. THx with TLx as 5-bit prescaler.
0	1	16-bit Timer/Counter. THx and TLx are cascaded; there is no prescaler.
1	0	8-bit auto-reload Timer/Counter. THx holds a value which is to be reloaded into TLx each time it overflows.
1	1	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/Counter stopped.

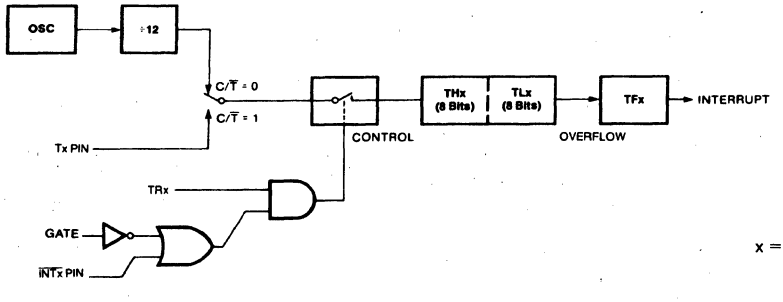


Figure 8. Timer/Counter 0 or 1 in Mode 0: 13-Bit Counter

Table 5. TCON: Timer/Counter Control Register

TCON Address = 88H Reset Value = 0000 0000B

Bit Addressable

	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF1	Timer 1 overflow Flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter 1 on/off.
TF0	Timer 0 overflow Flag. Set by hardware on Timer/Counter 0 overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter 0 on/off.
IE1	Interrupt 1 flag. Set by hardware when external interrupt 1 edge is detected (transmitted or level-activated). Cleared when interrupt processed only if transition-activated.
IT1	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupt 1.
IE0	Interrupt 0 flag. Set by hardware when external interrupt 0 edge is detected (transmitted or level-activated). Cleared when interrupt processed only if transition-activated.
IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupt 0.

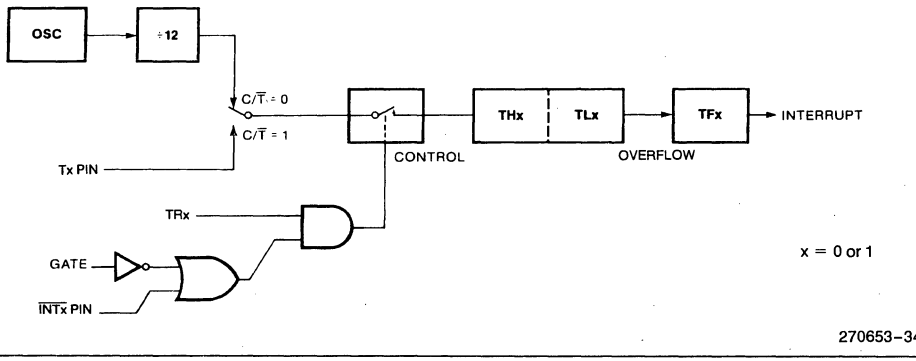


Figure 9. Timer/Counter 0 or 1 in Mode 1: 16-Bit Counter

MODE 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 11. TL0 uses the Timer 0 control bits: C/$\overline{T}$, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus TH0 now controls the Timer 1 interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in any application not requiring an interrupt.

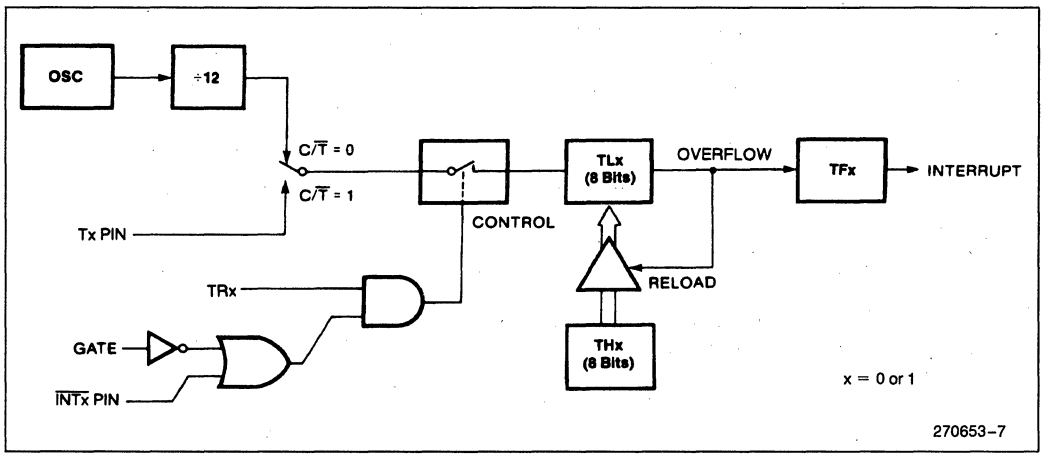


Figure 10. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

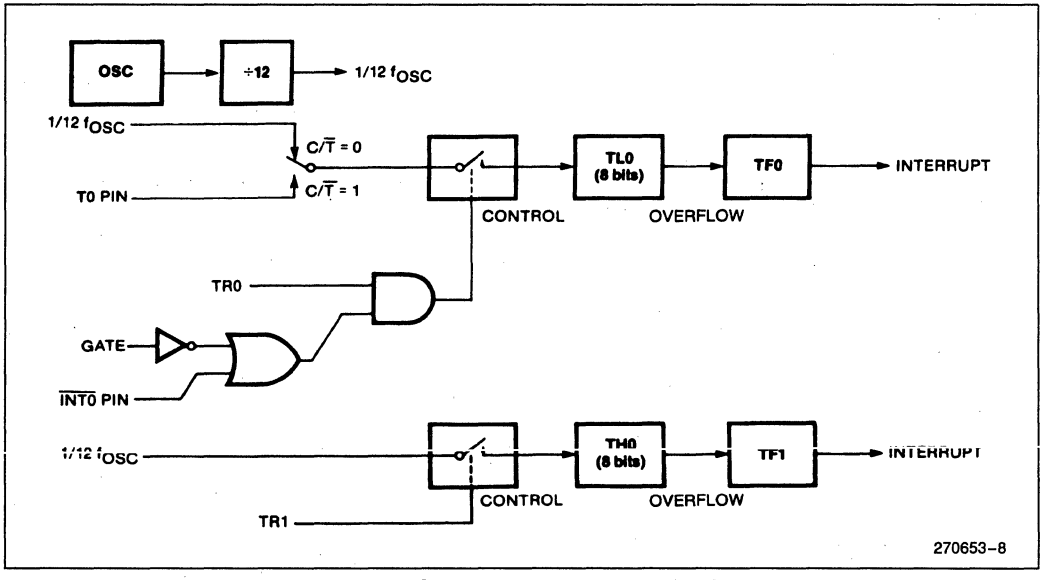


Figure 11. Timer/Counter 0 Mode 3: Two 8-Bit Counters

5.2 Timer 2

Timer 2 is a 16-bit Timer/Counter which can operate either as a timer or as an event counter. This is selected by bit C/$\overline{T2}$ in the Special Function Register T2CON (Table 7). It has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON as shown in Table 6.

Table 6. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	Mode
0	0	1	16-Bit Auto-Reload
0	1	1	16-Bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Table 7. T2CON: Timer/Counter 2 Control Register

T2CON Address = 0C8H Reset Value = 0000 0000B

Bit Addressable

	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/$\overline{T2}$	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12 or OSC/2 in baud rate generator mode). 1 = External event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

CAPTURE MODE

In the capture mode there are two options selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 still does the above, but with the added feature that a 1-to-0 tran-

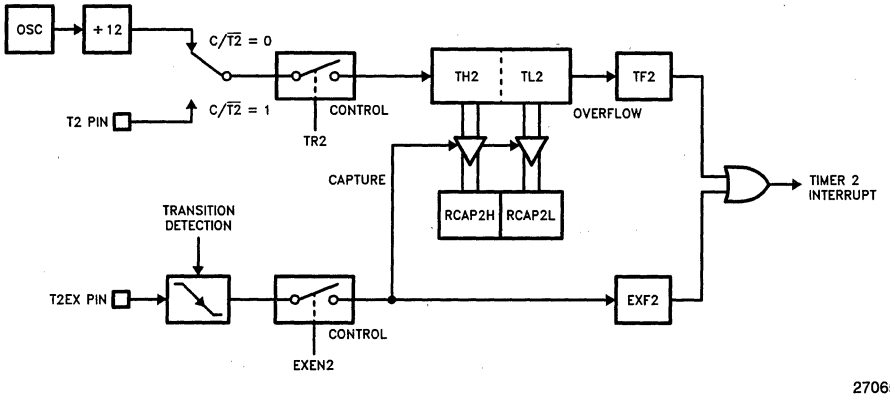


Figure 12. Timer 2 in Capture Mode

sition at external input T2EX causes the current value in the Timer 2 registers, TH2 and TL2, to be captured into registers RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 12.

AUTO-RELOAD MODE (UP OR DOWN COUNTER)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by a bit named DCEN (Down Counter Enable) located in the SFR T2MOD (see Table 8). Upon reset the DCEN bit is set to 0 so that Timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down depending on the value of the T2EX pin.

Figure 13 shows Timer 2 automatically counting up when DCEN = 0. In this mode there are two options selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Either the TF2 or EXF2 bit can generate the Timer 2 interrupt if it is enabled.

Table 8. T2MOD: Timer 2 Mode Control Register

T2MOD Address = 0C9H Reset Value = XXXX XXX0B

Not Bit Addressable

	—	—	—	—	—	—	—	DCEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
—	Not implemented, reserved for future use.*
DECN	When set, this bit allows Timer 2 to be configured as an up/down counter.

***NOTE:**
User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

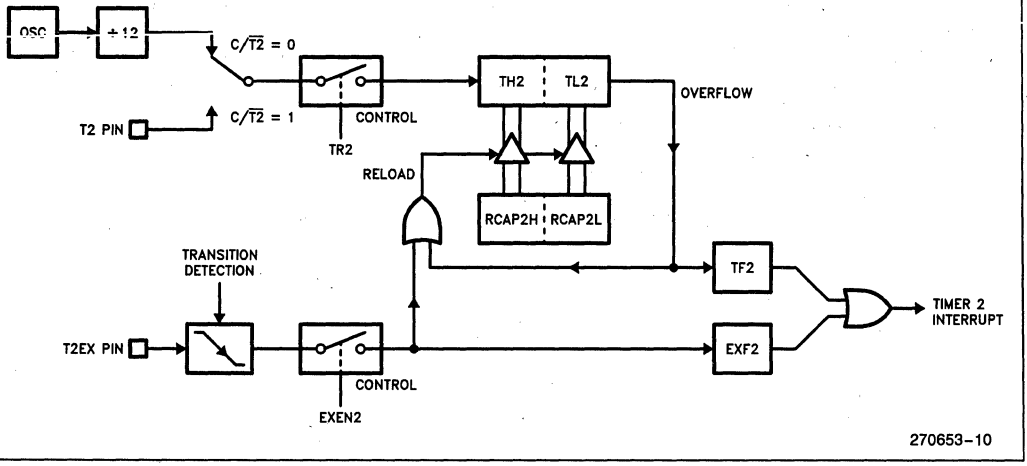


Figure 13. Timer 2 Auto Reload Mode (DCEN = 0)

Setting the DCEN bit enables Timer 2 to count up or down as shown in Figure 14. In this mode the T2EX pin controls the direction of count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit which can then generate an interrupt if it is enabled. This overflow also causes a the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. Now the timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows. This bit can be used as a 17th bit of resolution if desired. In this operating mode, EXF2 does not generate an interrupt.

BAUD RATE GENERATOR MODE

The baud rate generator mode is selected by setting the RCLK and/or TCLK bits in T2CON. Timer 2 in this mode will be described in conjunction with the serial port.

6.0 PROGRAMMABLE COUNTER ARRAY

The Programmable Counter Array (PCA) consists of a 16-bit timer/counter and five 16-bit compare/capture modules as shown in Figure 15. The PCA timer/counter serves as a common time base for the five modules and is the only timer which can service the PCA. Its clock input can be programmed to count any one of the following signals:

- oscillator frequency ÷ 12
- oscillator frequency ÷ 4
- Timer 0 overflow
- external input on ECI (P1.2).

Each compare/capture module can be programmed in any one of the following modes:

- rising and/or falling edge capture
- software timer
- high speed output
- pulse width modulator.

Module 4 can also be programmed as a watchdog timer.

When the compare/capture modules are programmed in the capture mode, software timer, or high speed output mode, an interrupt can be generated when the module executes its function. All five modules plus the PCA timer overflow share one interrupt vector (more about this in the PCA Interrupt section).

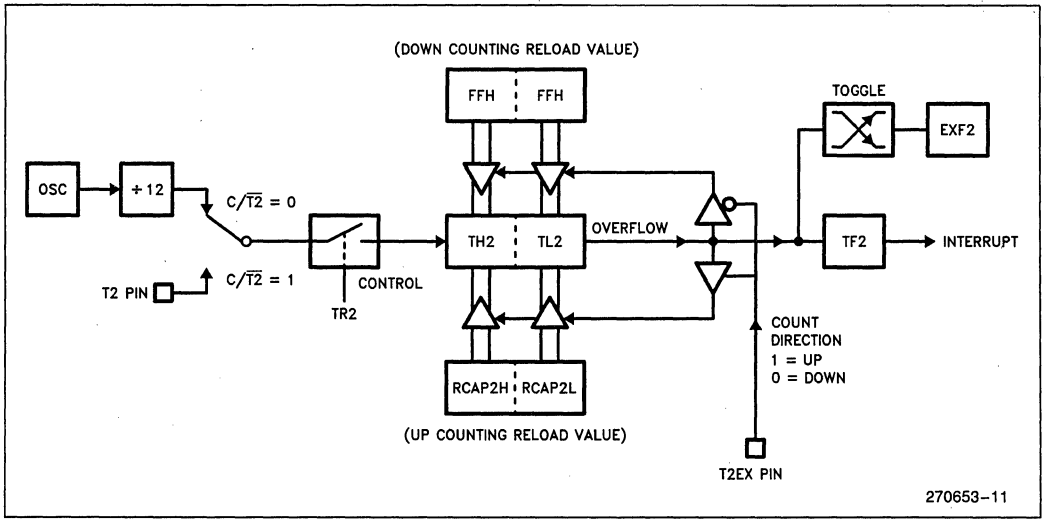


Figure 14. Timer 2 Auto Reload Mode (DCEN = 1)

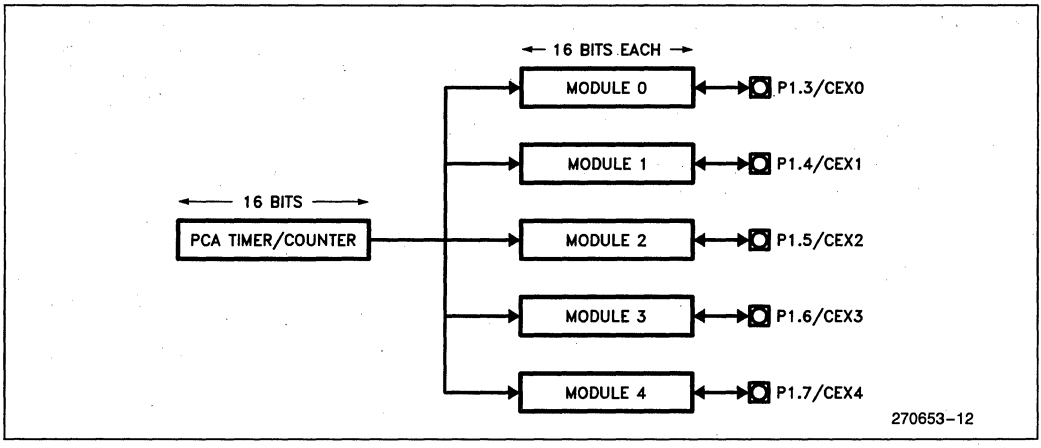


Figure 15. Programmable Counter Array

The PCA timer/counter and compare/capture modules share Port 1 pins for external I/O. These pins are listed below. If the port pin is not used for the PCA, it can still be used for standard I/O.

PCA Component	External I/O Pin
16-bit Counter	P1.2 / ECI
16-bit Module 0	P1.3 / CEX0
16-bit Module 1	P1.4 / CEX1
16-bit Module 2	P1.5 / CEX2
16-bit Module 3	P1.6 / CEX3
16-bit Module 4	P1.7 / CEX4

6.1 PCA 16-Bit Timer/Counter

The PCA has a free-running 16-bit timer/counter consisting of registers CH and CL (the high and low bytes of the count value). These two registers can be read or written to at any time. Figure 16 shows a block diagram of this timer. The clock input can be selected from the following four modes:

- Oscillator frequency ÷ 12
  The PCA timer increments once per machine cycle. With a 16 MHz crystal, the timer increments every 750 nanoseconds.
- Oscillator frequency ÷ 4
  The PCA timer increments three times per machine cycle. With a 16 MHz crystal, the timer increments every 250 nanoseconds.
- Timer 0 overflows
  The PCA timer increments whenever Timer 0 overflows. This mode allows a programmable input frequency to the PCA.
- External input
  The PCA timer increments when a 1-to-0 transition is detected on the ECI pin (P1.2). The maximum input frequency in this mode is oscillator frequency ÷ 8.

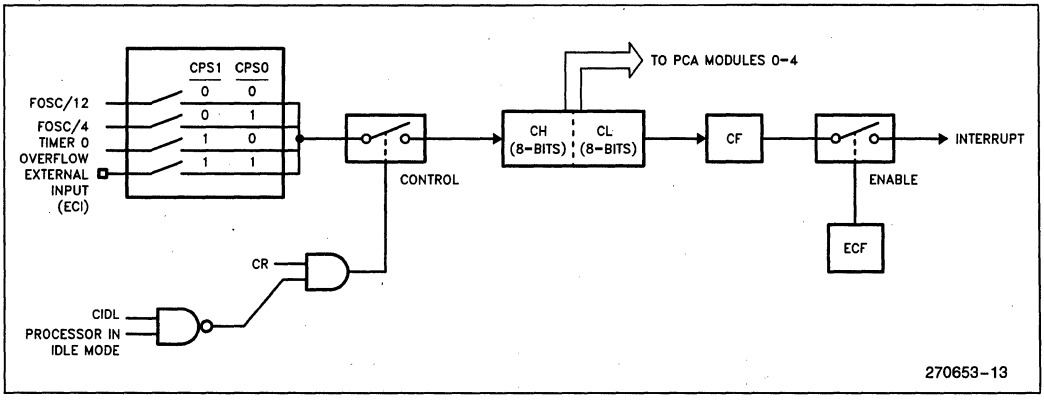


Figure 16. PCA Timer/Counter

The mode register CMOD contains the Count Pulse Select bits (CPS1 and CPS0) to specify the clock input. CMOD is shown in Table 9. This register also contains the ECF bit which enables the PCA counter overflow to generate the PCA interrupt. In addition, the user has the option of turning off the PCA timer during Idle Mode by setting the Counter Idle bit (CIDL). The Watchdog Timer Enable bit (WDTE) will be discussed in a later section.

The CCON register, shown in Table 10, contains two more bits which are associated with the PCA timer/counter. The CF bit gets set by hardware when the counter overflows, and the CR bit is set or cleared to turn the counter on or off. The other five bits in this register are the event flags for the compare/capture modules and will be discussed in the next section.

Table 9. CMOD: PCA Counter Mode Register

CMOD Address = 0D9H Reset Value = 00XX X000B

Not Bit Addressable

	CIDL	WDTE	—	—	—	CPS1	CPS0	ECF
Bit	7	6	5	4	3	2	1	0

Symbol	Function
CIDL	Counter Idle control: CIDL = 0 programs the PCA Counter to continue functioning during idle Mode. CIDL = 1 programs it to be gated off during idle.
WDTE	Watchdog Timer Enable: WDTE = 0 disables Watchdog Timer function on PCA Module 4. WDTE = 1 enables it.
—	Not implemented, reserved for future use.*
CPS1	PCA Count Pulse Select bit 1.
CPS0	PCA Count Pulse Select bit 0.

CPS1	CPS0	Selected PCA Input**
0	0	Internal clock, Fosc÷12
0	1	Internal clock, Fosc÷4
1	0	Timer 0 overflow
1	1	External clock at ECI/P1.2 pin (max. rate = Fosc÷8)

ECF	PCA Enable Counter Overflow interrupt: ECF = 1 enables CF bit in CCON to generate an interrupt. ECF = 0 disables that function of CF.

NOTE:
*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.
**Fosc = oscillator frequency

Table 10. CCON: PCA Counter Control Register

CCON Address = 0D8H Reset Value = 00X0 0000B

Bit Addressable

	CF	CR	—	CCF4	CCF3	CCF2	CCF1	CCF0
Bit	7	6	5	4	3	2	1	0

Symbol	Function
CF	PCA Counter Overflow flag. Set by hardware when the counter rolls over. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software but can only be cleared by software.
CR	PCA Counter Run control bit. Set by software to turn the PCA counter on. Must be cleared by software to turn the PCA counter off.
—	Not implemented, reserved for future use*.
CCF4	PCA Module 4 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.
CCF3	PCA Module 3 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.
CCF2	PCA Module 2 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.
CCF1	PCA Module 1 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.
CCF0	PCA Module 0 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.

***NOTE:**
User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

6.2 Capture/Compare Modules

Each of the five compare/capture modules has six possible functions it can perform:

— 16-bit Capture, positive-edge triggered
— 16-bit Capture, negative-edge triggered
— 16-bit Capture, both positive and negative-edge triggered
— 16-bit Software Timer
— 16-bit High Speed Output
— 8-bit Pulse Width Modulator.

In addition, module 4 can be used as a Watchdog Timer. The modules can be programmed in any combination of the different modes.

Each module has a mode register called CCAPMn (n = 0, 1, 2, 3, or 4) to select which function it will perform. The CCAPMn register is shown in Table 11. Note the ECCFn bit which enables the PCA interrupt when a module's event flag is set. The event flags (CCFn) are located in the CCON register and get set when a capture event, software timer, or high speed output event occurs for a given module.

Table 12 shows the combinations of bits in the CCAPMn register that are valid and have a defined function. Invalid combinations will produce undefined results.

Each module also has a pair of 8-bit compare/capture registers (CCAPnH and CCAPnL) associated with it. These registers store the time when a capture event occurred or when a compare event should occur. For the PWM mode, the high byte regiser CCAPnH controls the duty cycle of the waveform.

The next five sections describe each of the compare/capture modes in detail.

Table 11. CCAPMn: PCA Modules Compare/Capture Registers

CCAPMn (n = 0–4) Address: CCAPM0 0DAH, CCAPM1 0DBH, CCAPM2 0DCH, CCAPM3 0DDH, CCAPM4 0DEH — Reset Value = X000 0000B

Not Bit Addressable

	—	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
Bit	7	6	5	4	3	2	1	0

Symbol Function

—	Not implemented, reserved for future use*.
ECOMn	Enable Comparator. ECOMn = 1 enables the comparator function.
CAPPn	Capture Positive, CAPPn = 1 enables positive edge capture.
CAPNn	Capture Negative, CAPNn = 1 enables negative edge capture.
MATn	Match. When MATn = 1, a match of the PCA counter with this module's compare/capture register causes the CCFn bit in CCON to be set, flagging an interrupt.
TOGn	Toggle. When TOGn = 1, a match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle.
PWMn	Pulse Width Modulation Mode. PWMn = 1 enables the CEXn pin to be used as a pulse width modulated output.
ECCFn	Enable CCF interrupt. Enables compare/capture flag CCFn in the CCON register to generate an interrupt.

NOTE:
*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case,the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

Table 12. PCA Module Modes (CCAPMn Register)

—	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	Module Function
X	0	0	0	0	0	0	0	No operation
X	X	1	0	0	0	0	X	16-bit capture by a postive-edge trigger on CEXn
X	X	0	1	0	0	0	X	16-bit capture by a negative-edge trigger on CEXn
X	X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
X	1	0	0	1	0	0	X	16-bit Software Timer
X	1	0	0	1	1	0	X	16-bit High Speed Output
X	1	0	0	0	0	1	0	8-bit PWM
X	1	0	0	1	x	0	x	Watchdog Timer

X = Don't Care

6.3 16-Bit Capture Mode

Both positive and negative transitions can trigger a capture with the PCA. This gives the PCA the flexibility to measure periods, pulse widths, duty cycles, and phase differences on up to five separate inputs. Setting the CAPPn and/or CAPNn in the CCAPMn mode register select the input trigger—positive and/or negative transition—for module n. Refer to Figure 17.

The external input pins CEX0 through CEX4 are sampled for a transition. When a valid transition is detected (positive and/or negative edge), hardware loads the 16-bit value of the PCA timer (CH, CL) into the module's capture registers (CCAPnH, CCAPnL). The resulting value in the capture registers reflects the PCA timer value at the time a transition was detected on the CEXn pin.

Upon a capture, the module's event flag (CCFn) in CCON is set, and an interrupt is flagged if the ECCFn bit in the mode register CCAPMn is set. The PCA interrupt will then be generated if it is enabled. Since the hardware does not clear an event flag when the interrupt is vectored to, the flag must be cleared in software.

In the interrupt service routine, the 16-bit capture value must be saved in RAM before the next capture event occurs. A subsequent capture on the same CEXn pin will write over the first capture value in CCAPnH and CCAPnL.

6.4 16-Bit Software Timer Mode

In the compare mode, the 16-bit value of the PCA timer is compared with a 16-bit value pre-loaded in the module's compare registers (CCAPnH, CCAPnL). The comparison occurs three times per machine cycle in order to recognize the fastest possible clock input (i.e. ¼ x oscillator frequency). Setting the ECOMn bit in the mode register CCAPMn enables the comparator function as shown in Figure 18.

For the Software Timer mode, the MATn bit also needs to be set. When a match occurs between the PCA timer and the compare registers, a match signal is generated and the module's event flag (CCFn) is set. An interrupt is then flagged if the ECCFn bit is set. The PCA interrupt is generated only if it has been properly enabled. Software must clear the event flag before the next interrupt will be flagged.

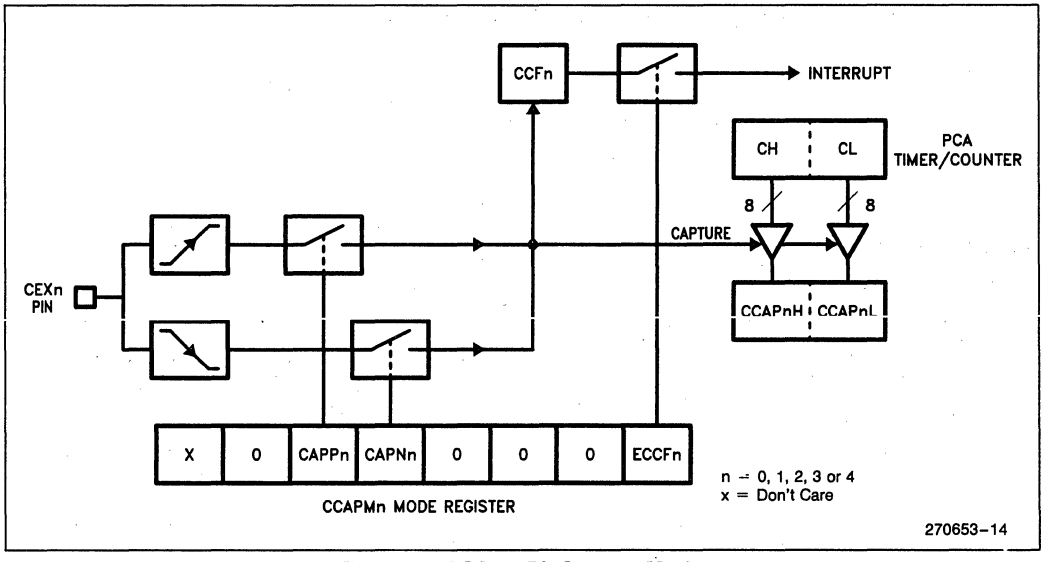


Figure 17. PCA 16-Bit Capture Mode

During the interrupt routine, a new 16-bit compare value can be written to the compare registers (CCAPnH and CCAPnL). *Notice, however, that a write to CCAPnL clears the ECOMn bit which temporarily disables the comparator function while these registers are being updated so an invalid match does not occur.* A write to CCAPnH sets the ECOMn bit and re-enables the comparator. For this reason, user software should write to CCAPnL first, then CCAPnH.

6.5 High Speed Output Mode

The High Speed Output (HSO) mode toggles a CEXn pin when a match occurs between the PCA timer and a pre-loaded value in a module's compare registers. For this mode, the TOGn bit needs to be set in addition to the ECOMn and MATn bits as seen in Figure 18. By setting or clearing the pin in software, the user can select whether the CEXn pin will change from a logical 0 to a logical 1 or vice versa. The user also has the option of flagging an interrupt when a match event occurs by setting the ECCFn bit.

The HSO mode is more accurate than toggling port pins in software because the toggle occurs *before* branching to an interrupt. That is, interrupt latency will not effect the accuracy of the output. If the user does not change the compare registers in an interrupt routine, the next toggle will occur when the PCA timer rolls over and matches the last compare value.

6.6 Watchdog Timer Mode

A Watchdog Timer is a circuit that automatically invokes a reset unless the system being watched sends regular hold-off signals to the Watchdog. These circuits are used in applications that are subject to electrical noise, power glitches, electrostatic discharges, etc., or where high reliability is required.

The Watchdog Timer function is only available on PCA module 4. In this mode, every time the count in the PCA timer matches the value stored in module 4's compare registers, an internal reset is generated. (See Figure 19.) The bit that selects this mode is WDTE in the CMOD register. Module 4 must be set up in either compare mode as a Software Timer or High Speed Output.

To hold off the reset, the user has three options:

(1) periodically change the compare value so it will never match the PCA timer,

(2) periodically change the PCA timer value so it will never match the compare value,

(3) disable the Watchdog by clearing the WDTE bit before a match occurs and then later re-enable it.

The first two options are more reliable because the Watchdog Timer is never disabled as in option #3. The second option is not recommended if other PCA modules are being used since this timer is the time base for all five modules. Thus, in most applications the first solution is the best option.

If a Watchdog Timer is not needed, module 4 can still be used in other modes.

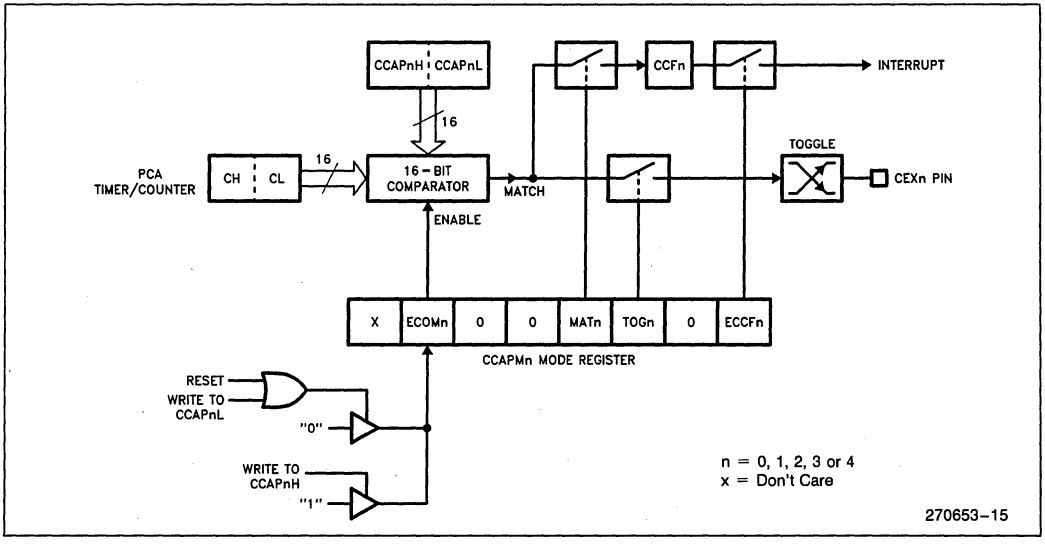


Figure 18. PCA 16-Bit Comparator Mode: Software Timer and High Speed Output

6.7 Pulse Width Modulator Mode

Any or all of the five PCA modules can be programmed to be a Pulse Width Modulator. The PWM output can be used to convert digital data to an analog signal by simple external circuitry. The frequency of the PWM depends on the clock sources for the PCA timer. With a 16 MHz crystal the maximum frequency of the PWM waveform is 15.6 KHz.

The PCA generates 8-bit PWMs by comparing the low byte of the PCA timer (CL) with the low byte of the module's compare registers (CCAPnL). Refer to Figure 20. When CL < CCAPnL the output is low. When CL ≥ CCAPnL the output is high. The value in CCAPnL controls the duty cycle of the waveform. To change the value in CCAPnL without output glitches, the user must write to the high byte register (CCAPnH). This value is then shifted by hardware into CCAPnL when CL rolls over from 0FFH to 00H which corresponds to the next period of the output.

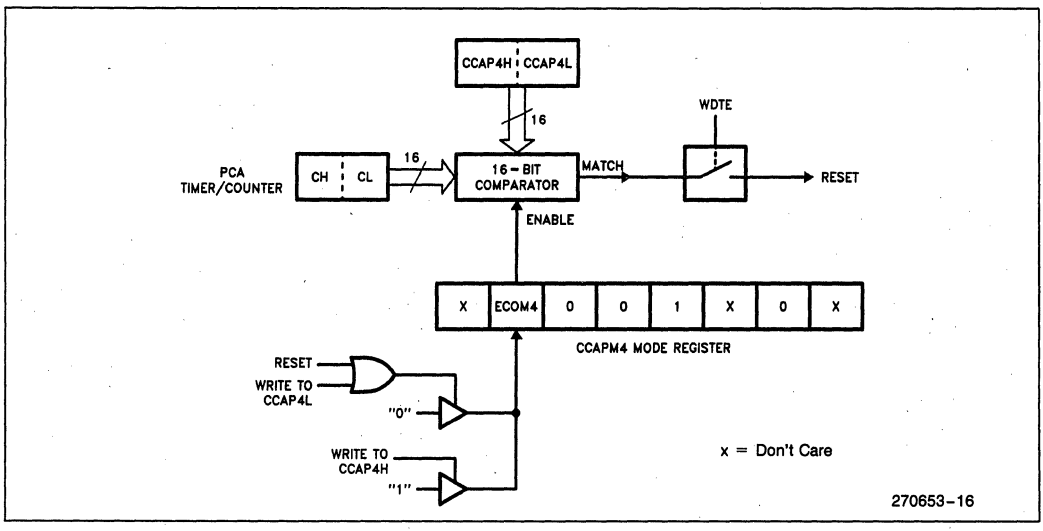


Figure 19. Watchdog Timer Mode

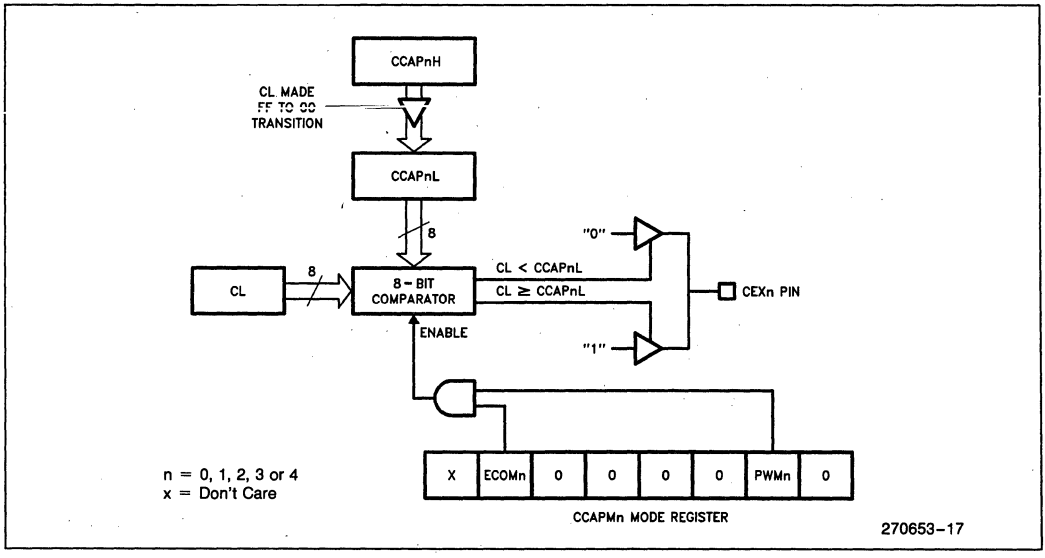


Figure 20. PCA 8-Bit PWM Mode

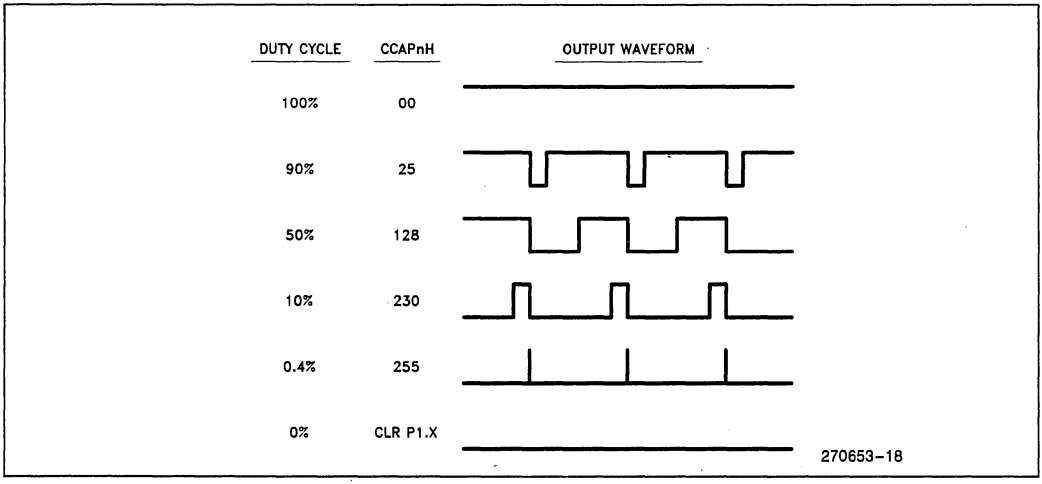


Figure 21. CCAPnH Varies Duty Cycle

CCAPnH can contain any integer from 0 to 255 to vary the duty cycle from a 100% to 0.4% (see Figure 21). A 0% duty cycle can be obtained by writing directly to the port pin with the CLR bit instruction.

7.0 SERIAL INTERFACE

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. (However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost). The serial port receive and transmit registers are both accessed through Special Function Register SBUF. Actually, SBUF is two separate registers, a transmit buffer and a receive buffer. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port control and status register is the Special Function Register SCON, shown in Table 13. This register contains the mode selection bits (SM0 and SM1); the SM2 bit for the multiprocessor modes (see Multiprocessor Communications section); the Receive Enable bit (REN); the 9th data bit for transmit and receive (TB8 and RB8); and the serial port interrupt bits (TI and RI).

The serial port can operate in 4 modes:

Mode 0: Serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Mode 1: 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). Refer to Fiugre 22. On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in SCON, while the stop bit is ignored. (The validity of the stop bit can be checked with Framing Error Detection.) The baud rate is programmable to either $1/32$ or $1/64$ the oscillator frequency.

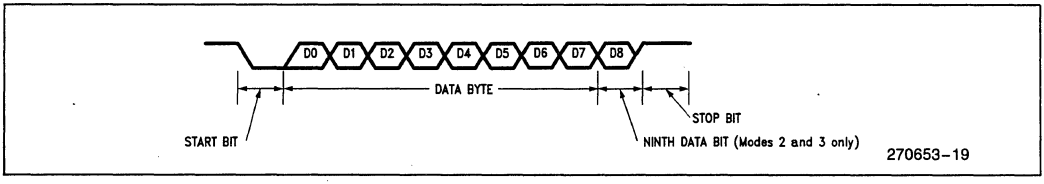


Figure 22. Data Frame: Modes 1, 2 and 3

Mode 3: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1. For more detailed information on each serial port mode, refer to the "Hardware Description of the 8051, 8052, and 80C51."

7.1 Framing Error Detection

Framing Error Detection allows the serial port to check for valid stop bits in modes 1, 2, or 3. A missing stop bit can be caused, for example, by noise on the serial lines, or transmission by two CPUs simultaneously.

If a stop bit is missing, a Framing Error bit FE is set. The FE bit can be checked in software after each reception to detect communication errors. Once set, the FE bit must be cleared in software. A valid stop bit will not clear FE.

The FE bit is located in SCON and shares the same bit address as SM0. Control bit SMOD0 in the PCON register (location PCON.6) determines whether the SM0 or FE bit is accessed. If SMOD0 = 0, then accesses to SCON.7 are to SM0. If SMOD0 = 1, then accesses to SCON.7 are to FE.

7.2 Multiprocessor Communications

Modes 2 and 3 provide a 9-bit mode to facilitate multiprocessor comunication. The 9th bit allows the controller to distinguish between address and data bytes. The 9th bit is set to 1 for address bytes and set to 0 for data bytes. When receiving, the 9th bit goes into RB8 in SCON. When transmitting, TB8 is set or cleared in software.

The serial port can be programmed such that when the stop bit is received the serial port interrupt will be activated only if the received byte is an address byte (RB8 = 1). This feature is enabled by setting the SM2 bit in SCON. A way to use this feature in multiprocessor systems is as follows.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. Remember, an address byte has its 9th bit set to 1, whereas a data byte has its 9th bit set to 0. All the slave processors should have their SM2 bits set to 1 so they will only be interrupted by an address byte. In fact, the 8XC51FA/FB has an Automatic Address Recognition feature which allows only the addressed slave to be interrupted. That is, the address comparison occurs in hardware, not software. (On the 8051 serial port, an address byte interrupts all slaves for an address comparison.)

The addressed slave then clears its SM2 bit and prepares to receive the data bytes that will be coming. The other slaves are unaffected by these data bytes. They are still waiting to be addressed since their SM2 bits are all set.

7.3 Automatic Address Recognition

Automatic Address Recognition reduces the CPU time required to service the serial port. Since the CPU is only interrupted when it receives its own address, the software overhead to compare addresses is eliminated. With this feature enabled in one of the 9-bit modes, the Receive Interrupt (RI) flag will only get set when the received byte corresponds to either a Given or Broadcast address.

The feature works the same way in the 8-bit mode (Mode 1) as in the 9-bit modes, except that the stop bit takes the place of the 9th data bit. If SM2 is set, the RI flag is set only if the received byte matches the Given or Broadcast Address and is terminated by a valid stop bit. Setting the SM2 bit has no effect in Mode 0.

The master can selectively communicate with groups of slaves by using the Given Address. Addressing all slaves at once is possible with the Broadcast Address. These addresses are defined for each slave by two Special Function Registers: SADDR and SADEN.

A slave's individual address is specified in SADDR. SADEN is a mask byte that defines don't-cares to form the Given Address. These don't-cares allow flexibility in the user-defined protocol to address one or more slaves at a time. The following is an example of how the user could define Given Addresses to selectively address different slaves.

Slave 1:

SADDR	=	1111	0001
SADEN	=	1111	1010
GIVEN	=	1111	0X0X

Slave 2:

SADDR	=	1111	0011
SADEN	=	1111	1001
GIVEN	=	1111	0XX1

Table 13. SCON: Serial Port Control Register

SCON Address = 98H Reset Value = 0000 0000B

Bit Addressable

SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
7	6	5	4	3	2	1	0

Bit: (SMOD0 = 0/1)*

Symbol	Function
FE	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0* bit must be set to enable access to the FE bit.
SM0	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)
SM1	Serial Port Mode Bit 1

SM0	SM1	Mode	Description	Baud Rate**
0	0	0	shift register	$F_{OSC}/12$
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	$F_{OSC}/64$ or $F_{OSC}/32$
1	1	3	9-bit UART	variable

SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.
REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
RB8	In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

NOTE:
*SMOD0 is located at PCON6.
**F_{OSC} = oscillator frequency

The SADEN byte are selected such that each slave can be addressed separately. Notice that bit 1 (LSB) is a don't-care for Slave 1's Given Address, but bit 1 = 1 for Slave 2. Thus, to selectively communicate with just Slave 1 the master must send an address with bit 1 = 0 (e.g. 1111 0000).

Similarly, bit 2 = 0 for Slave 1, but is a don't-care for Slave 2. Now to communicate with just Slave 2 an address with bit 2 = 1 must be used (e.g. 1111 0111).

Finally, for a master to communicate with both slaves at once the address must have bit 1 = 1 and bit 2 = 0. Notice, however, that bit 3 is a don't-care for both slaves. This allows two different addresses to select both slaves (1111 0001 or 1111 0101). If a third slave was added that required its bit 3 = 0, then the latter address could be used to communicate with Slave 1 and 2 but not Slave 3.

The master can also communicate with all slaves at once with the Broadcast Address. It is formed from the logical OR of the SADDR and SADEN registers with zeros defined as don't-cares. The don't-cares also allow

flexibility in defining the Broadcast Address, but in most applications a Broadcast Address will be 0FFH.

SADDR and SADEN are located at address A9H and B9H, respectively. On reset, the SADDR and SADEN registers are initialized to 00H which defines the Given and Broadcast Addresses as XXXX XXXX (all don't-cares). This assures the 8XC51FA/FB serial port to be backwards compatibility with other MCS®-51 products which do not implement Automatic Addressing.

7.4 Baud Rates

The baud rate in Mode 0 is fixed:

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{12}$$

The baud rate in Mode 2 depends on the value of bit SMOD1 in Special Function Register PCON. If SMOD1 = 0 (which is the value on reset), the baud rate is $1/64$ the oscillator frequency. If SMOD1 = 1, the baud rate is $1/32$ the oscillator frequency.

$$\text{Mode 2 Baud Rate} = 2^{\text{SMOD1}} \times \frac{\text{Oscillator Frequency}}{64}$$

The baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate, or by Timer 2 overflow rate, or by both (one for transmit and the other for receive).

7.5 Using Timer 1 to Generate Baud Rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD1 as follows:

$$\text{Modes 1 and 3 Baud Rate} = 2^{\text{SMOD1}} \times \frac{\text{Timer 1 Overflow Rate}}{32}$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In most applications, it is configured for "timer" operation in the auto-reload mode (high nibble of TMOD = 0010B). In this case, the baud rate is given by the formula:

$$\text{Modes 1 and 3 Baud Rate} = \frac{2^{\text{SMOD1}} \times \text{Oscillator Frequency}}{32 \times 12 \times [256 - (\text{TH1})]}$$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload.

Table 14 lists various commonly used baud rates and how they can be obtained from Timer 1.

7.6 Using Timer 2 to Generate Baud Rates

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 7). Note that the baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 23.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

Table 14. Timer 1 Generated Commonly Used Baud Rates

Baud Rate	f_{OSC}	SMOD	Timer 1		
			C/$\overline{T}$	Mode	Reload Value
Mode 0 Max: 1 MHz	12 MHz	X	X	X	X
Mode 2 Max: 375K	12 MHz	1	X	X	X
Modes 1, 3: 62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate as follows:

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either "timer" or "counter" operation. In most applications, it is configured for "timer" operation ($C/\overline{T2} = 0$). The "Timer" operation is different for Timer 2 when it's being used as a baud rate generator. Normally, as a timer, it increments every machine cycle (1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time ($\frac{1}{2}$ the oscillator frequency). The baud rate formula is given below:

$$\text{Modes 1 and 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 23. This figure is valid only if RCLK and/or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the Timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read, but shouldn't be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Table 15 lists commonly used baud rates and how they can be obtained from Timer 2.

Table 15. Timer 2 Generated Commonly Used Baud Rates

Baud Rate	Osc Freq	Timer 2	
		RCAP2H	RCAP2L
375K	12 MHz	FF	FF
9.6K	12 MHz	FF	D9
4.8K	12 MHz	FF	B2
2.4K	12 MHz	FF	64
1.2K	12 MHz	FE	C8
300	12 MHz	FB	1E
110	12 MHz	F2	AF
300	6 MHz	FD	8F
110	6 MHz	F9	57

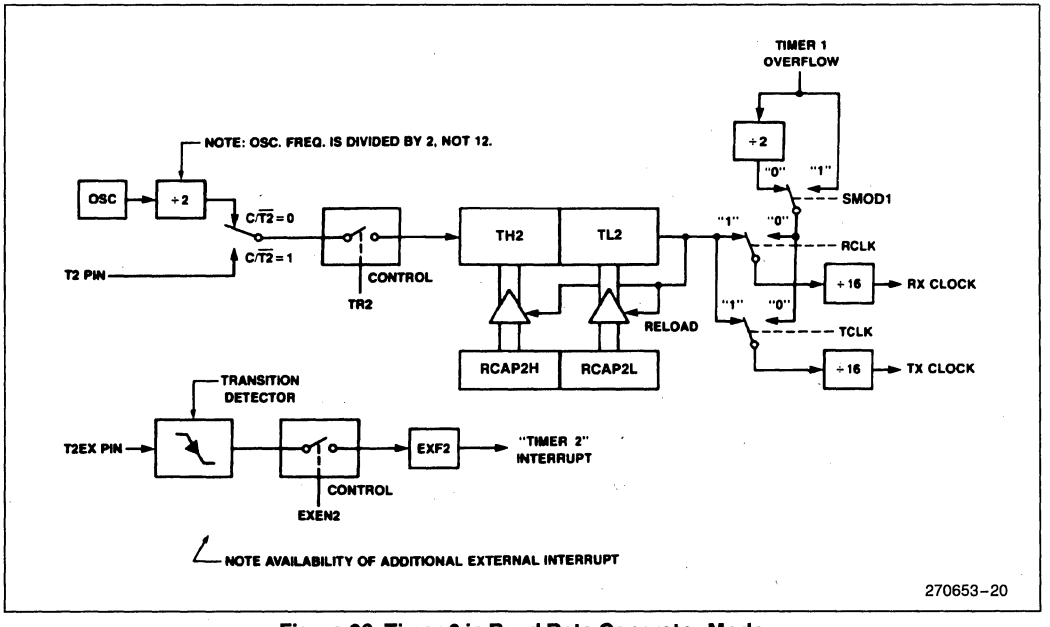


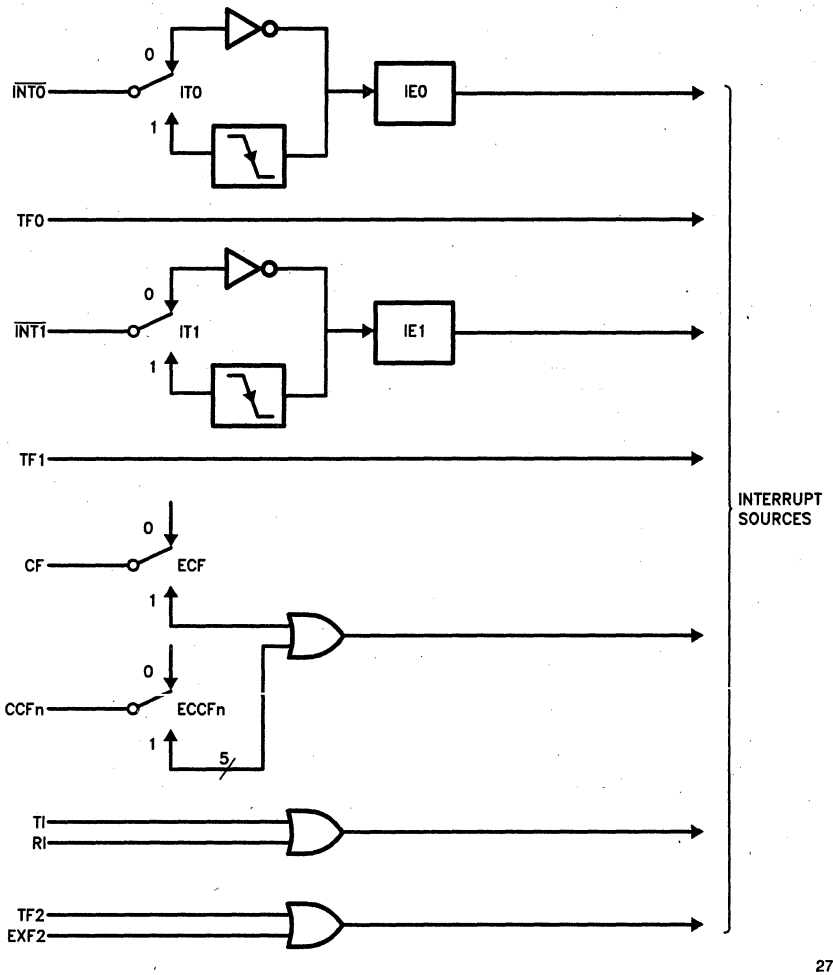
Figure 23. Timer 2 in Baud Rate Generator Mode

8.0 INTERRUPTS

The 8XC51FA/FB has a total of 7 interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), the PCA interrupt, and the serial port interrupt. These interrupts are all shown in Figure 24.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled in software.

Each of these interrupts will be briefly described followed by a discussion of the interrupt enable bits and the interrupt priority levels.



(See exceptions when Timer 2 is used as baud rate generator or an up/down counter.)

Figure 24. Interrupt Sources

8.1 External Interrupts

External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the $\overline{INTx}$ pin. If ITx = 1, external interrupt x is negative edge-triggered. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. These flags are cleared by hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called.

If external interrupt $\overline{INT0}$ or $\overline{INT1}$ is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

8.2 Timer Interrupts

Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1 in register TCON, which are set by a rollover in their respective Timer/Counter registers (except see Timer 0 in Mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

Timer 2 Interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared in software.

8.3 PCA Interrupt

The PCA interrupt is generated by the logical OR of CF, CCF0, CCF1, CCF2, CCF3, and CCF4 in register CCON. None of these flags is cleared by hardware when the service routine is vectored to. Normally the service routine will have to determine which bit flagged the interrupt and clear that bit in software. The PCA interrupt is enabled by bit EC in the Interrupt Enable register (see Table 16). In addition, the CF flag and each of the CCFn flags must also be enabled by bits ECF and ECCFn in registers CMOD and CCAPMn respectively, in order for that flag to be able to cause an interrupt.

8.4 Serial Port Interrupt

The serial port interrupt is generated by the logical OR of bits RI and TI in register SCON. Neither of these flags is cleared by hardware when the service routine is vectored to. The service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

8.5 Interrupt Enable

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in the Interrupt Enable (IE) register. (See Table 16.) Note that IE also contains a global disable bit, EA. If EA is set (1), the interrupts are individually enabled or disabled by their corresponding bits in IE. If EA is clear (0), all interrupts are disabled.

8.6 Priority Level Structure

Each interrupt source can also be individually programmed to one of two priority levels, by setting or clearing a bit in the Interrupt Priority (IP) register shown in Table 17. A low-priority interrupt can itself be interrupted by a higher priority interrupt, but not by another low-priority interrupt. A high priority interrupt cannot be interrupted by any other interrupt source.

Table 16. IE: Interrupt Enable Register

IE Address = 0A8H Reset Value = 0000 0000B

Bit Addressable

EA	EC	ET2	ES	ET1	EX1	ET0	EX0
7	6	5	4	3	2	1	0

Bit

Enable Bit = 1 enables the interrupt.
Enable Bit = 0 disables it.

Symbol	Function
EA	Global disable bit. If EA = 0, all Interrupts are disabled. If EA = 1, each Interrupt can be individually enabled or disabled by setting or clearing its enable bit.
EC	PCA Interrupt enable bit.
ET2	Timer 2 Interrupt enable bit.
ES	Serial Port Interrupt enable bit.
ET1	Timer 1 Interrupt enable bit.
EX1	External Interrupt 1 enable bit.
ET0	Timer 0 Interrupt enable bit.
EX0	External Interrupt 0 enable bit.

Table 17. IP: Interrupt Priority Registers

IP **Address = 0B8H** **Reset Value = X000 0000B**

Bit Addressable

—	PPC	PT2	PS	PT1	PX1	PT0	PX0
7	6	5	4	3	2	1	0

Bit

Priority Bit = 1 assigns high priority
Priority Bit = 0 assigns low priority

Symbol	Function
—	Not implemented, reserved for future use.*
PPC	PCA Interrupt priority bit.
PT2	Timer 2 Interrupt priority bit.
PS	Serial Port Interrupt priority bit.
PT1	Timer 1 Interrupt priority bit.
PX1	External Interrupt 1 priority bit.
PT0	Timer 0 Interrupt priority bit.
PX0	External Interrupt 0 priority bit.

NOTE:
*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the **same** priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence shown in Table 18.

Note that the "priority within level" structure is only used to resolve *simultaneous requests of the same priority level.*

Table 18. Interrupt Priority within Level Polling Sequence

1 (Highest)	$\overline{\text{INT0}}$
2	Timer 0
3	$\overline{\text{INT1}}$
4	Timer 1
5	PCA
6	Serial Port
7 (Lowest)	Timer 2

How Interrupts are Handled

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. The Timer 2 interrupt cycle is slightly different, as described in the Response Time section. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

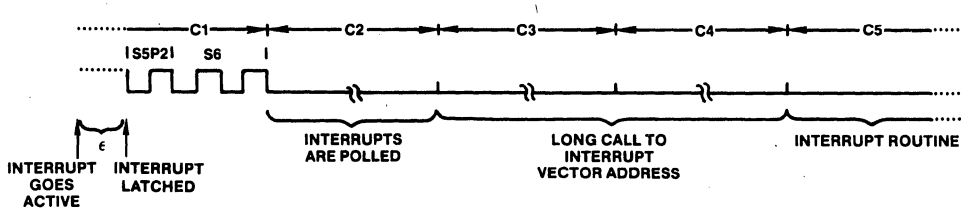
1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any write to IE or IP, then at least one more instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. If the interrupt flag for a *level-sensitive* external interrupt is active but not being responded to for one of the above conditions and is not *still* active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in Figure 25.

Note that if an interrupt of a higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 25, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine having been executed.



This is the fastest possible response when C2 is the final cycle of an instruction other than RETI or write IE or IP.

Figure 25. Interrupt Response Timing Diagram

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. The hardware-generated LCALL pushes the contents of the Program Counter onto the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to, as shown in Table 19.

Table 19. Interrupt Vector Address

Interrupt Source	Interrupt Request Bits	Cleared by Hardware	Vector Address
$\overline{INT0}$	IE0	No (level) Yes (trans.)	0003H
TIMER 0	TF0	Yes	000BH
$\overline{INT1}$	IE1	No (level) Yes (trans.)	0013H
TIMER 1	TF1	Yes	001BH
SERIAL PORT	RI, TI	No	0023H
TIMER 2	TF2, EXF2	No	002BH
PCA	CF, CCFn (n = 0-4)	No	0033H

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking interrupt was still in progress.

Note that the starting addresses of consecutive interrupt service routines are only 8 bytes apart. That means if consecutive interrupts are being used (IE0 and TF0, for example, or TF0 and IE1), and if the first interrupt routine is more than 7 bytes long, then that routine will have to execute a jump to some other memory location where the service routine can be completed without overlapping the starting address of the next interrupt routine.

8.7 Response Time

The $\overline{INT0}$ and $\overline{INT1}$ levels are inverted and latched into the Interrupt Flags IE0 and IE1 at S5P2 of every machine cycle. Similarly, the Timer 2 flag EXF2 and the Serial Port flags RI and TI are set at S5P2. The values are not actually polled by the circuitry until the next machine cycle.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag TF2 is set at S2P2 and is polled in the same cycle in which the timer overflows.

If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapses between activation of an external interrupt request and the beginning of execution of the service routine's first instruction. Figure 25 shows interrupt response timing.

A longer response time would result if the request is blocked by one of the 3 previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles, since the longest instructions (MUL and DIV) are only 4 cycles long, and if the instruction in progress is RETI or write to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one or more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single-interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

9.0 RESET

The reset input is the RST pin, which has a Schmitt Trigger input. A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods) *while the oscillator is running*. The CPU responds by generating an internal reset, with the timing shown in Figure 26.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins, ALE, and $\overline{PSEN}$ will maintain their current activities for the 19 oscillator periods after a logic 1 has been sampled at the RST pin; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

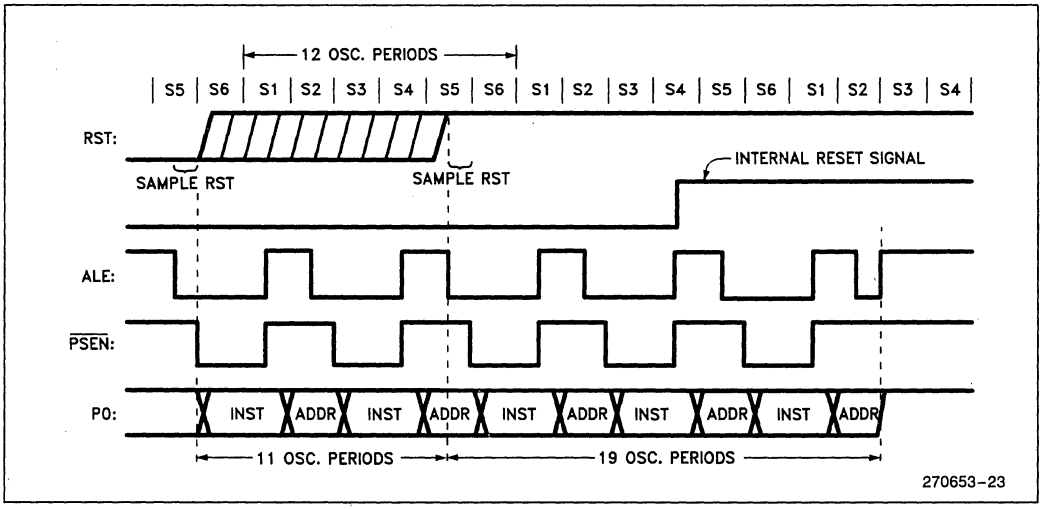


Figure 26. Reset Timing

While the RST pin is high, the port pins, ALE and $\overline{PSEN}$ are weakly pulled high. After RST is pulled low, it will take 1 to 2 machine cycles for ALE and $\overline{PSEN}$ to start clocking. For this reason, other devices can not be synchronized to the internal timings of the 8XC51FA/FB.

Driving the ALE and $\overline{PSEN}$ pins to 0 while reset is active could cause the device to go into an indeterminate state.

The internal reset algorithm redefines all the SFRs. Table 1 lists the SFRs and their reset values. The internal RAM is not affected by reset. On power up the RAM content is indeterminate.

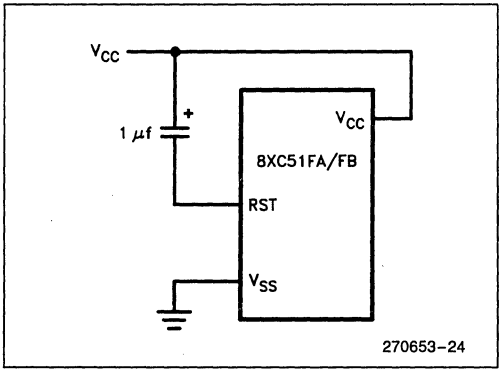


Figure 27. Power on Reset Circuitry

9.1 Power-On Reset

For CHMOS devices, when VCC is turned on, an automatic reset can be obtained by connecting the RST pin to VCC through a 1 μF capacitor (Figure 27). The CHMOS devices do not require an external resistor like the HMOS devices because they have an internal pull-down on the RST pin.

When power is turned on, the circuit holds the RST pin high for an amount of time that depends on the capacitor value and the rate at which it charges. To ensure a valid reset the RST pin must be held high long enough to allow the oscillator to start up plus two machine cycles.

On power up, V_{CC} should rise within approximately ten milliseconds. The oscillator start-up time will depend on the oscillator frequency. For a 10 MHz crystal, the start-up time is typically 1 msec. For a 1 MHz crystal, the start-up time is typically 10 msec.

With the given circuit, reducing V_{CC} quickly to 0 causes the RST pin voltage to momentarily fall below 0V. However, this voltage is internally limited and will not harm the device.

Note that the port pins will be in a random state until the oscillator has started and the internal reset algorithm has written Is to them.

Powering up the device without a valid reset could cause the CPU to start executing instructions from an indeterminate location. This is because the SFRs, specifically the Program Counter, may not get properly initialized.

10.0 POWER-SAVING MODES OF OPERATION

For applications where power consumption is critical, the 8XC51FA/FB provides two power reducing modes of operation: Idle and Power Down. The input through which backup power is supplied during these operations is V_{CC}. Figure 28 shows the internal circuitry which implements these features. In the Idle mode (IDL = 1), the oscillator continues to run and the Interrupt, Serial Port, PCA, and Timer blocks continue to be clocked, but the clock signal is gated off to the CPU. In Power Down (PD = 1), the oscillator is frozen. The Idle and Power Down modes are activated by setting bits in Special Function Register PCON (Table 20).

10.1 Idle Mode

An instruction that sets PCON.0 causes that to be the last instruction executed before going into the Idle mode. In the Idle mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer, and Serial Port functions. The PCA can be programmed either to pause or continue operating during Idle (refer to the PCA section for more details). The CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical states they had at the time Idle was activated. ALE and $\overline{PSEN}$ hold at logic high levels.

There are two ways to terminate the Idle Mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into Idle.

The flag bits (GF0 and GF1) can be used to give an indication if an interrupt occurred during normal operation or during Idle. For example, an instruction that activates Idle can also set one or both flag bits. When Idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset needs to be held active for only two machine cycles (24 oscillator periods) to complete the reset.

The signal at the RST pin clears the IDL bit directly and asynchronously. At this time the CPU resumes program execution from where it left off; that is, at the instruction following the one that invoked the Idle Mode. As shown in Figure 26, two or three machine cycles of program execution may take place before the

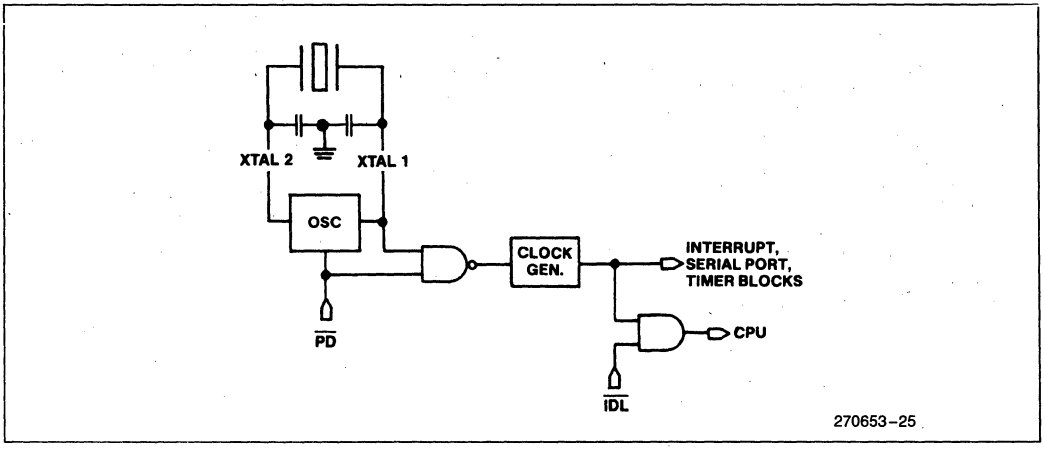


Figure 28. Idle and Power Down Hardware

Table 20. PCON: Power Control Register

PCON Address = 87H Reset Value = 00XX 0000B

Not Bit Addressable

	SMOD1	SMOD0	—	POF	GF1	GF0	PD	IDL
Bit	7	6	5	4	3	2	1	0

Symbol	Function
SMOD1	Double Baud rate bit. When set to a 1 and Timer 1 is used to generate baud rates, and the Serial Port is used in modes 1, 2, or 3.
SMOD0	When set, Read/Write accesses to SCON.7 are to the FE bit. When clear, Read/Write accesses to SCON.7 are to the SM0 bit.
—	Not implemented, reserved for future use.*
POF	Power Off Flag. Set by hardware on the rising edge of V_{CC}. Set or cleared by software. This flag allows detection of a power failure caused reset. V_{CC} must remain above 3V to retain this bit.
GF1	General-purpose flag bit.
GF0	General-purpose flag bit.
PD	Power Down bit. Setting this bit activates Power Down operation.
IDL	Idle mode bit. Setting this bit activates idle modes operation. If 1s are written to PD and IDL at the same time, PD takes precedence.

NOTE:
*User software should not write 1s to unimplemented bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate

internal reset algorithm takes control. On-chip hardware inhibits access to the internal RAM during this time, but access to the port pins is not inhibited. To eliminate the possibility of unexpected outputs at the port pins, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external Data RAM.

10.2 Power Down Mode

An instruction that sets PCON.1 causes that to be the last instruction executed before going into the Power Down mode. In this mode the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Special Function Registers are held. The port pins output the values held by their respective SRFs, and ALE and $\overline{PSEN}$ output lows. In Power Down V_{CC} can be reduced to as low as 2V. Care must be taken, however, to ensure that V_{CC} is not reduced before Power Down is invoked.

The 8XC51FA/FB can exit Power Down with either a hardware reset or external interrupt. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 msec).

With an external interrupt, $\overline{INT0}$ or $\overline{INT1}$ must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator and bringing the pin back high completes the exit. After the RETI instruction is executed in the interrupt service routine, the next instruction will be the one following the instruction that put the device in Power Down.

10.3 Power Off Flag

The Power Off Flag (POF) is set by hardware when V_{CC} rises from 0 to 5 Volts. POF can also be set or cleared by software. This allows the user to distinguish between a "cold start" reset and a "warm start" reset.

A cold start reset is one that is coincident with V_{CC} being turned on to the device after it was turned off. A warm start reset occurs while V_{CC} is still applied to the device and could be generated, for example, by a Watchdog Timer or an exit from Power Down.

Immediately after reset, the user's software can check the status of the POF bit. POF = 1 would indicate a cold start. The software then clears POF and commences its tasks. POF = 0 immediately after reset would indicate a warm start.

V_{CC} must remain above 3 volts for POF to retain a 0.

11.0 EPROM VERSIONS

The 87C51FA/FB uses the fast "Quick-Pulse" programming™ algorithm. The devices program at V_{PP} = 12.75V (and V_{CC} = 5.0V) using a series of twenty-five 100 μs PROG pulses per byte programmed. This results in a total programming time of approximately 26 seconds for the 87C51FA's 8K bytes and approximately 50 seconds for the 87C51FB's 16K bytes.

11.1 Two-Level Program Memory Lock

In some microcontroller applications it is desirable that the Program Memory be secure from software piracy. The 8XC51FA/FB has a two-level program lock feature which protect the code of the on-chip EPROM or ROM. The two-level scheme consists of a 32-byte encryption array and two lock bits.

Encryption Array: Within the EPROM/ROM are 32 bytes of Encryption Array that are initially unprogrammed (all 1's). The user can program the Encryption Array to encrypt the program code bytes during EPROM/ROM verification. The verification procedure is performed as usual except that each code byte comes out exclusive-NOR'ed (XNOR) with one of the key bytes. Therefore, to read the ROM code the user has to know the 32 key bytes in their proper sequence.

Unprogrammed bytes have the value 0FFH. So if the Encryption Array is left unprogrammed, all the key bytes have the value 0FFH. Since any code byte XNORed with 0FFH leaves the byte unchanged, leaving the Encryption Array unprogrammed in effect bypasses the encryption feature.

Program Lock Bits: Also included in the Program Lock scheme are two Lock Bits which can be programmed as shown in Table 21.

Erasing the EPROM also erases the Encryption Array and the Lock Bits, returning the part to full functionality.

Exposure to Light: The EPROM window must be covered with an opaque label when the device is in operation. This is not so much to protect the EPROM array from inadvertent erasure, but to protect the RAM and other on-chip logic. Allowing light to impinge on the silicon die while the device is operating can cause logical malfunction.

Table 21. EPROM Lock Bits

Program Lock Bits		Logic Enabled
LB1	LB2	
U	U	No Program Lock features enabled. (Code Verify will still be encrypted by the Encryption Array.)
P	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and futher programming of EPROM is disabled.
P	P	Same as above, but Verify is also disabled (option available on EPROM only)
U	P	Reserved for Future Definition

12.0 ONCE MODE

The ONCE (ON-Circuit Emulation) mode facilitates testing and debugging of systems using the 8XC51FA/FB without having to remove the device from the circuit. The ONCE mode is invoked by:

1. Pulling ALE low while the device is in reset and $\overline{PSEN}$ is high;
2. Holding ALE low as RST is deactivated.

While the device is in ONCE mode, the Port 0 pins go into a float state, and the other port pins, ALE, and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit.

Normal operation is restored after a valid reset is applied.

13.0 ON-CHIP OSCILLATOR

The on-chip oscillator for the CHMOS devices, shown in Figure 29, consists of a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator. In this application the crystal is operating in its fundamental response mode as an inductive reactance in parallel resonance with capacitance external to the crystal (Figure 30).

The oscillator on the CHMOS devices can be turned off under software control by setting the PD bit in the PCON register. The feedback resistor R_f in Figure 29 consists of paralleled n- and p-channel FETs controlled by the PD bit, such that R_f is opened when

PD = 1. The diodes D1 and D2, which act as clamps to V_{CC} and V_{SS}, are parasitic to the R_f FETs.

The crystal specifications and capacitance values (C1 and C2 in Figure 30) are not critical. 30 pF can be used in these positions at any frequency with good quality crystals. In general, crystals used with these devices typically have the following specifications:

ESR (Equivalent Series Resistance)	see Figure 32
C_O (shunt capacitance)	7.0 pF maximum
C_L (load capacitance)	30 pF ±3 pF
Drive Level	1 MW

Frequency, tolerance, and temperature range are determined by the system requirements.

A ceramic resonator can be used in place of the crystal in cost-sensitive applications. When a ceramic resonator is used, C1 and C2 are normally selected as higher values, typically 47 pF. The manufacturer of the ceramic resonator should be consulted for recommendations on the values of these capacitors.

A more in-depth discussion of crystal specifications, ceramic resonators, and the selection of values for C1 and C2 can be found in Application Note AP-155, "Oscillators for Microcontrollers" in the Embedded Control Applications handbook.

To drive the CHMOS parts with an external clock source, apply the external clock signal to XTAL1 and leave XTAL2 floating as shown in Figure 31. This is an important difference from the HMOS parts. With HMOS, the external clock source is applied to XTAL2, and XTAL1 is grounded.

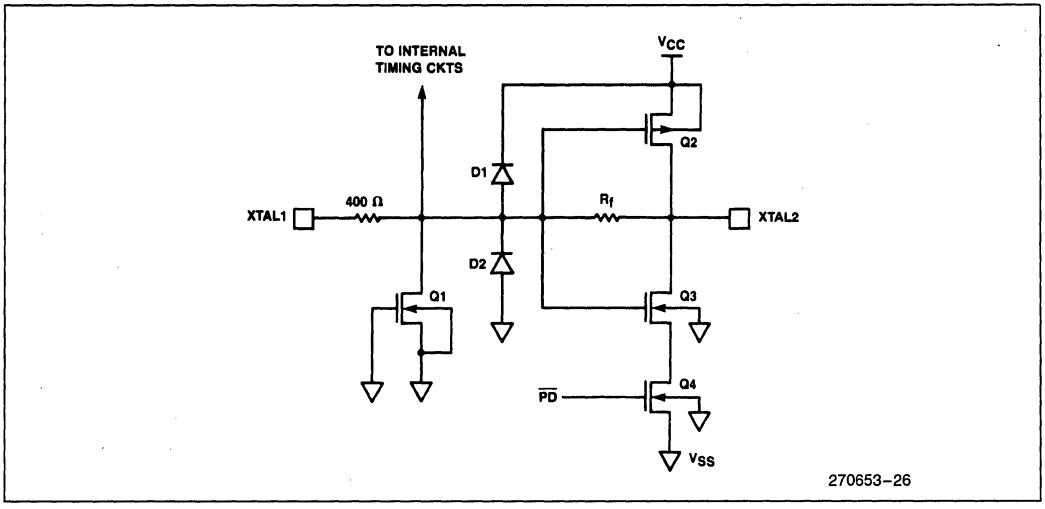


Figure 29. On-Chip Oscillator Circuitry

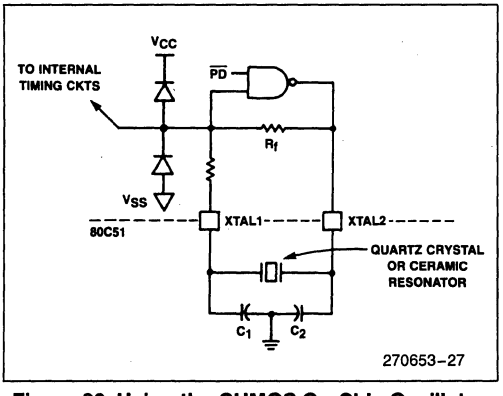


Figure 30. Using the CHMOS On-Chip Oscillator

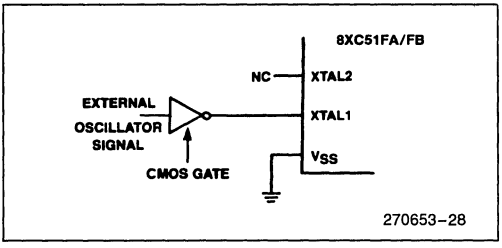


Figure 31. Driving the CHMOS Parts with an External Clock Source

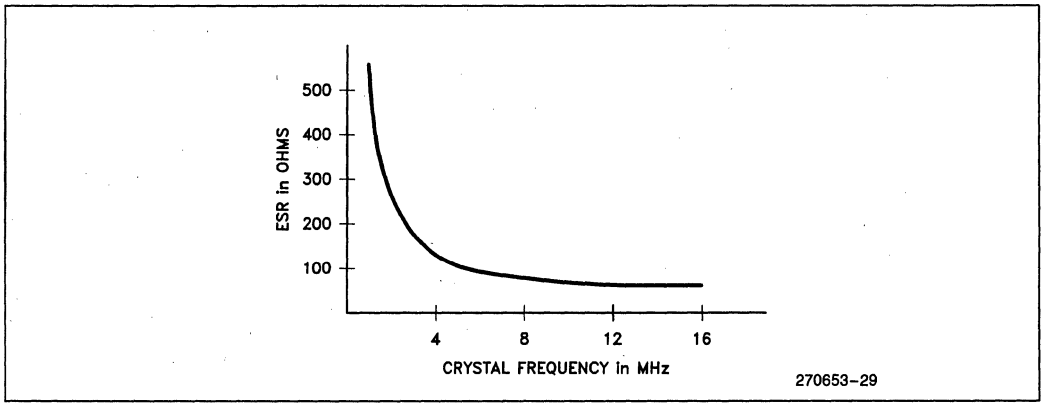


Figure 32. ESR vs Frequency

14.0 CPU TIMING

The internal clock generator defines the sequence of states that make up a machine cycle. A machine cycle consists of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1 microsecond if the oscillator frequency is 12 MHz. Each state is then divided into a Phase 1 and Phase 2 half.

Figure 3 and Figures 5 through 7 show when the various strobe and port signals are clocked internally. The figures do not show rise and fall times of the signals, nor do they show propagation delays between the XTAL1 signal and events at other pins.

Rise and fall times are dependent on the external loading that each pin must drive. They are approximately 10 nsec, measured between 0.8V and 2.0V.

Propagation delays are different for different pins. For a given pin they vary with pin loading, temperature, V_{CC}, and manufacturing lot. If the XTAL1 waveform is taken as the timing reference, propagation delays may vary from 25 to 125 nsec.

The AC Timings section of the data sheets do not reference any timing to the XTAL1 waveform. Rather, they relate the critical edges of control and input signals to each other. The timings published in the data sheets include the effects of propagation delays under the specified test condition.

ADDITIONAL REFERENCES

The following application notes provide supplemental information to this document and can be found in the *Embedded Control Applications* handbook.

1. AP-125 "Designing Microcontroller Systems for Electrically Noisy Environments"
2. AP-155 "Oscillators for Microcontrollers"
3. AP-252 "Designing with the 80C51BH"
4. AP-410 "Enhanced Serial Port on the 83C51FA"
5. AP-415 "83C51FA/FB PCA Cookbook"
6. AB-41 "Software Serial Port Implemented with the PCA"
7. AP-425 "Small DC Motor Control"

83C51FA/80C51FA
CHMOS SINGLE-CHIP 8-BIT MICROCOMPUTER

83C51FA—8K Bytes of Factory Mask Programmable ROM
80C51FA—CPU with RAM and I/O
83C51FA/80C51FA—3.5 MHz to 12 MHz, V_{CC} = 5V ± 10%
83C51FA-1/80C51FA-1—3.5 MHz to 16 MHz, V_{CC} = 5V ± 10%
83C51FA-2/80C51FA-2—0.5 MHz to 12 MHz, V_{CC} = 5V ± 10%

- High Performance CHMOS EPROM
- Three 16-Bit Timer/Counters
  - Timer 2 is an Up/Down Timer/Counter
- Programmable Counter Array with:
  - High Speed Output,
  - Compare/Capture,
  - Pulse Width Modulator,
  - Watchdog Timer capabilities
- Program Lock System
- 256 Bytes of On-Chip Data RAM
- Boolean Processor
- 32 Programmable I/O Lines
- 7 Interrupt Sources
- Programmable Serial Channel with:
  - Framing Error Detection
  - Automatic Address Recognition
- TTL and CMOS Compatible Logic Levels
- 64K External Program Memory Space
- 64K External Data Memory Space
- MCS®-51 Fully Compatible Instruction Set
- Power Saving Idle and Power Down Modes
- ONCE™ (On-Circuit Emulation) Mode

MEMORY ORGANIZATION

PROGRAM MEMORY: Up to 8K bytes of the program memory can reside in the on-chip ROM (83C51FA only). In addition the device can address up to 64K of program memory external to the chip.

DATA MEMORY: This microcontroller has a 256 x 8 on-chip RAM. In addition it can address up to 64K bytes of external data memory.

The Intel 83C51FA is a single-chip control oriented microcontroller which is fabricated on Intel's reliable CHMOS III technology. Being a member of the 8051 family, the 83C51FA uses the same powerful instruction set, has the same architecture, and is pin for pin compatible with the existing MCS-51 products. The 83C51FA is an enhanced version of the 80C51BH. It's added features make it an even more powerful microcontroller for applications that require Pulse Width Modulation, High Speed I/O, and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multi-processor communications.

For the remainder of this document, the 83C51FA and 80C51FA will be referred to as the 83C51FA.

October 1988
Order Number: 270538-002

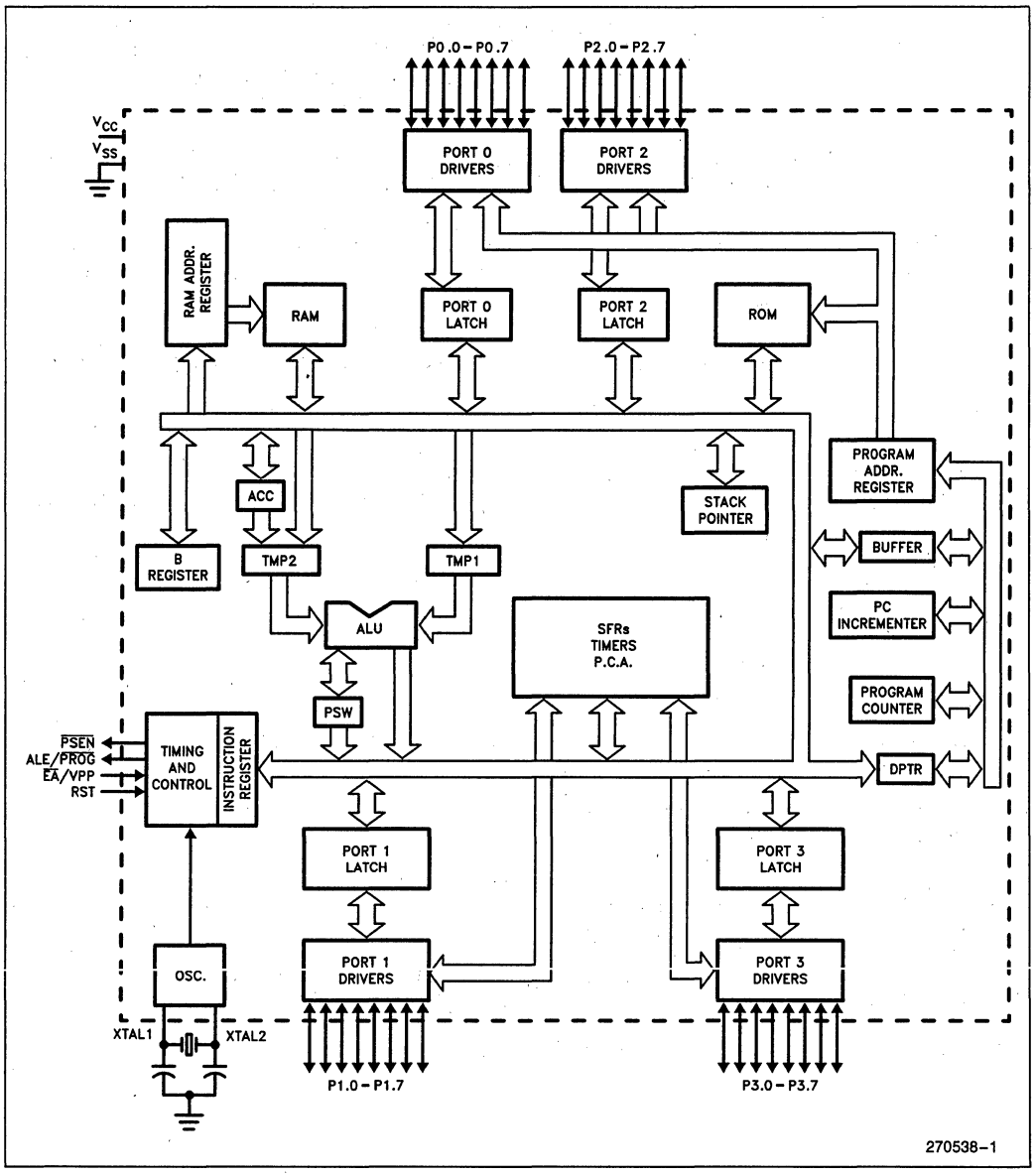
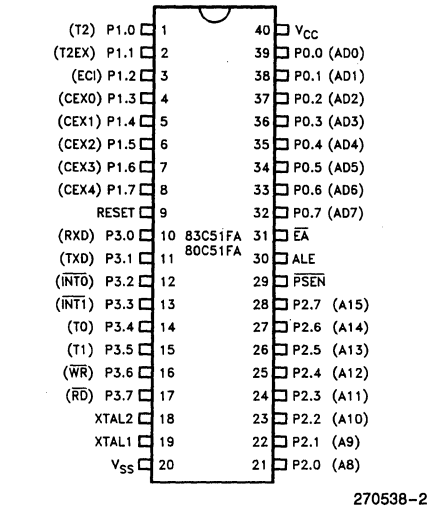


Figure 1. 83C51FA Block Diagram

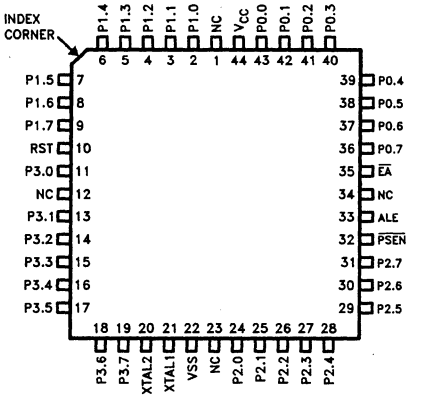
PACKAGES

Part	Prefix	Package Type
83C51FA	P	40-Pin Plastic DIP
80C51FA	D	40-Pin CERDIP
	N	44-Pin PLCC



270538-2

DIP



270538-19

PLCC

Figure 2. Pin Connections

PIN DESCRIPTIONS

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit, open drain, bidirectional I/O port. As an output port each pin can sink several LS TTL inputs. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting1's, and can source and sink several LS TTL inputs.

Port 0 outputs the code bytes during program verification on the 83C51FA. External pullup resistors are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can drive LS TTL inputs. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

In addition, Port 1 serves the functions of the following special features of the 83C51FA:

Port Pin	Alternate Function
P1.0	T2 (External Count Input to Timer/ Counter 2)
P1.1	T2EX (Timer/Counter 2 Capture/ Reload Trigger and Direction Control)
P1.2	ECI (External Count Input to the PCA)
P1.3	CEX0 (External I/O for Compare/ Capture Module 0)
P1.4	CEX1 (External I/O for Compare/ Capture Module 1)
P1.5	CEX2 (External I/O for Compare/ Capture Module 2)
P1.6	CEX3 (External I/O for Compare/ Capture Module 3)
P1.7	CEX4 (External I/O for Compare/ Capture Module 4)

Port 1 receives the low-order address bytes during ROM verification.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can drive LS TTL inputs. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Some Port 2 pins receive the high-order address bits during program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can drive LS TTL inputs. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits a power-on reset with only a capacitor connected to V_{CC}.

ALE: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$: Program Store Enable is the read strobe to external Program Memory.

When the 80C51FA is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory.

$\overline{EA}/V_{PP}$: External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the device to fetch code from external Program Memory locations 0000H to 0FFFFH. Note, however, that if either of the Program Lock bits are programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of a inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 floats, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the data sheet must be observed.

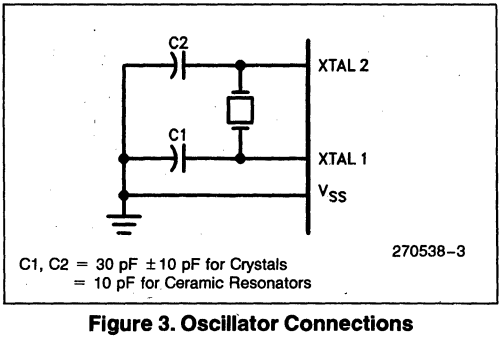


Figure 3. Oscillator Connections

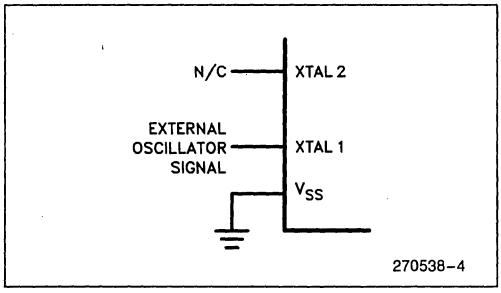


Figure 4. External Clock Drive Configuration

IDLE MODE

The user's software can invoke the Idle Mode. When the microcontroller is in this mode, power consumption is reduced. The Special Function Registers and the onboard RAM retain their values during Idle, but the processor stops executing instructions. Idle Mode will be exited if the chip is reset or if an enabled interrupt occurs. The PCA timer/counter can optionally be left running or paused during Idle Mode.

POWER DOWN MODE

To save even more power, a Power Down mode can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 83C51FA either a hardware reset or an external interrupt can cause an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

DESIGN CONSIDERATION

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

ONCE™ MODE

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 83C51FA without the 83C51FA having to be removed from the circuit. The ONCE Mode is invoked by:

1) Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2) Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the 83C51FA is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

Table 1. Status of the External Pins during Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

NOTE:
For more detailed information on the reduced power modes refer to current Embedded Controller Handbook, and Application Note AP-252, "Designing with the 80C51BH."

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on EA/V_{PP} Pin to V_{SS} .....0V to +6.5V
Voltage on Any Other Pin to V_{SS} ..−0.5V to +6.5V
Maximum I_{OL} per I/O Pin .....15 mA
Power Dissipation.....1.5W
(based on PACKAGE heat transfer limitations, not device power consumption)

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

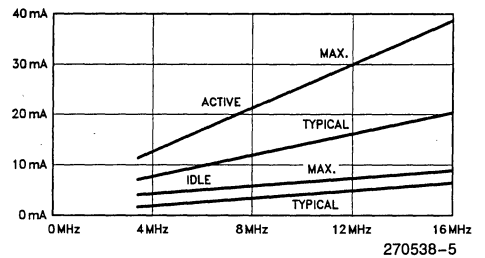
NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS: T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter	Min	Typical (4)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except $\overline{EA}$)	−0.5		0.2 V_{CC}−0.1	V	
V_{IL1}	Input Low Voltage $\overline{EA}$	0		0.2 V_{CC}−0.3	V	
V_{IH}	Input High Voltage (Except XTAL1, RST)	0.2 V_{CC}+0.9		V_{CC}+0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}		V_{CC}+0.5	V	
V_{OL}	Output Low Voltage (5) (Ports 1, 2 and 3)			0.3 0.45 1.0	V V V	I_{OL} = 100 μA I_{OL} = 1.6 mA (1) I_{OL} = 3.5 mA
V_{OL1}	Output Low Voltage (5) (Port 0, ALE/$\overline{PROG}$, $\overline{PSEN}$)			0.3 0.45 1.0	V V V	I_{OL} = 200 μA I_{OL} = 3.2 mA (1) I_{OL} = 7.0 mA
V_{OH}	Output High Voltage (Ports 1, 2 and 3 ALE/$\overline{PROG}$ and $\overline{PSEN}$)	V_{CC}−0.3 V_{CC}−0.7 V_{CC}−1.5			V V V	I_{OH} = −10 μA I_{OH} = −30 μA (2) I_{OH} = −60 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	V_{CC}−0.3 V_{CC}−0.7 V_{CC}−1.5			V V V	I_{OH} = −200 μA I_{OH} = −3.2 mA (2) I_{OH} = −7.0 mA
I_{IL}	Logical 0 Input Current (Ports 1, 2, and 3)		−10	−50	μA	V_{IN} = 0.45V
I_{LI}	Input leakage Current (Port 0 and $\overline{EA}$)		0.02	±10	μA	V_{IN} = V_{IL} or V_{IH}
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, and 3)		−265	−650	μA	V_{IN} = 2V
RRST	RST Pulldown Resistor	40	100	225	KΩ	
CIO	Pin Capacitance			10	pF	@1MHz, 25°C
I_{CC}	Power Supply Current: Running at 12 MHz (Figure 5) Idle Mode at 12 MHz (Figure 5) Power Down Mode		 15 5 5	 30 7.5 75	 mA mA μA	(Note 3)

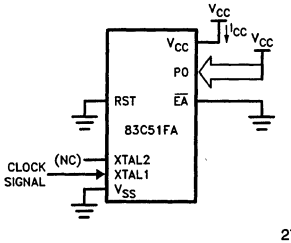
NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In applications where capacitance loading exceeds 100 pFs, the noise pulse on the ALE signal may exceed 0.8V. In these cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an Address Latch with a Schmitt Trigger Strobe input.
2. Capacitive loading on Ports 0 and 2 cause the V_{OH} on ALE and $\overline{PSEN}$ to drop below the 0.9 V_{CC} specification when the address lines are stabilizing.
3. See Figures 6–9 for test conditions. Minimum V_{CC} for power down is 2V.
4. Typicals are based on limited number of samples, and are not guaranteed. The values listed are at room temperature and 5V.
5. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port -
   Port 0: 26 mA
   Ports 1, 2, and 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.



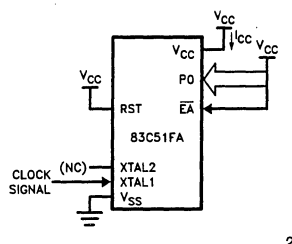
I_{CC} Max at other frequencies is given by:
Active Mode
I_{CC} MAX = 2.2 × FREQ + 3.1
Idle Mode
I_{CC} MAX = 0.49 × FREQ + 1.6
Where FREQ is in MHz, I_{CC}MAX is given in mA.

Figure 5. I_{CC} vs Frequency



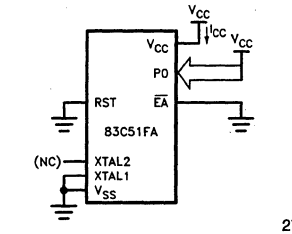
All other pins disconnected
TCLCH = TCHCL = 5 ns

Figure 7. I_{CC} Test Condition Idle Mode



All other pins disconnected
TCLCH = TCHCL = 5 ns

Figure 6. I_{CC} Test Condition, Active Mode



All other pins disconnected

Figure 8. I_{CC} Test Condition, Power Down Mode. V_{CC} = 2.0V to 6.0V.

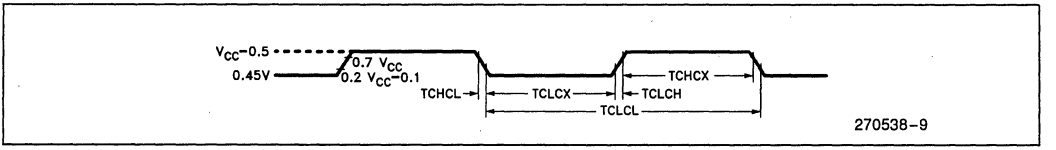


Figure 9. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address
C: Clock
D: Input Data
H: Logic level HIGH
I: Instruction (program memory contents)
L: Logic level LOW, or ALE
P: $\overline{PSEN}$
Q: Output Data
R: $\overline{RD}$ signal
T: Time
V: Valid
W: $\overline{WR}$ signal
X: No longer a valid logic level
Z: Float

For example,

TAVLL = Time from Address Valid to ALE Low
TLLPL = Time from ALE Low to $\overline{PSEN}$ Low

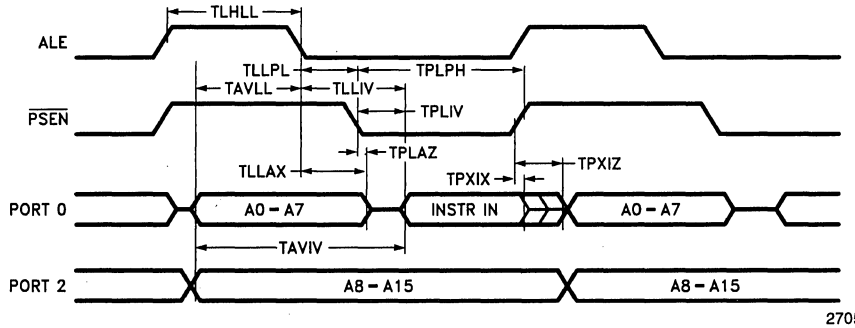
A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±20%, V_{SS} = 0V, Load Capacitance for Port 0, ALE and $\overline{PSEN}$ = 100 pF, Load Capacitance for All Other Outputs = 80 pF

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

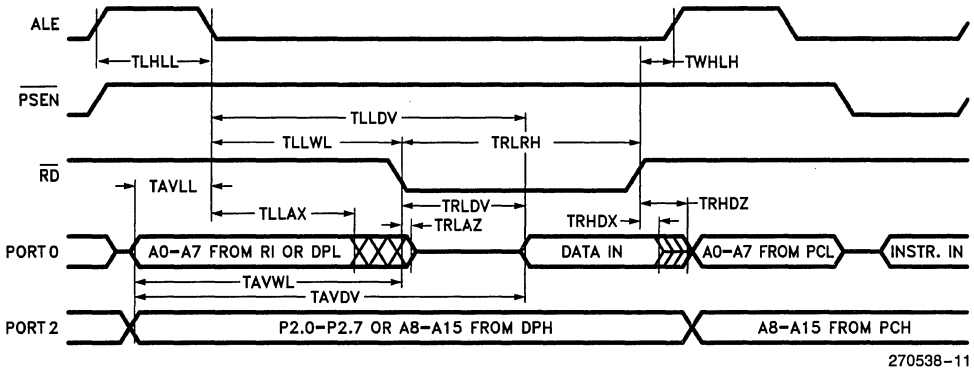
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 83C51FA 83C51FA-1 83C51FA-2			 3.5 3.5 0.5	 12 16 12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL−40		ns
TLLAX	Address Hold After ALE Low	53		TCLCL−30		ns
TLLIV	ALE Low to Valid Instruction In		234		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	53		TCLCL−30		ns
TPLPH	$\overline{PSEN}$ Pulse Width	205		3TCLCL−45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instruction In		145		3TCLCL−105	ns
TPXIX	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instruction Float After $\overline{PSEN}$		59		TCLCL−25	ns
TAVIV	Address to Valid Instruction In		312		5TCLCL−105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		107		2TCLCL−60	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	33		TCLCL−50		ns
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL−50		ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL−70		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL−40	TCLCL+40	ns

EXTERNAL PROGRAM MEMORY READ CYCLE



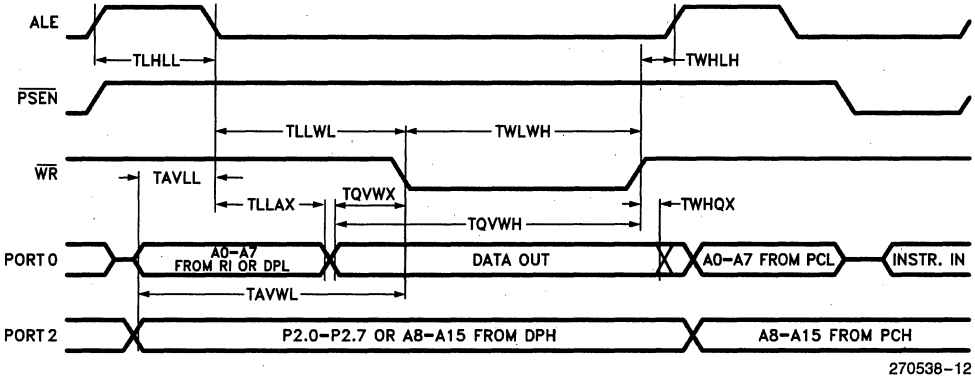
270538-10

EXTERNAL DATA MEMORY READ CYCLE



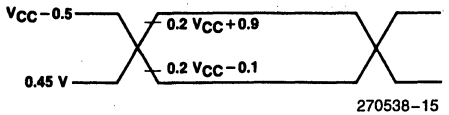
270538-11

EXTERNAL DATA MEMORY WRITE CYCLE



A.C. TESTING INPUT

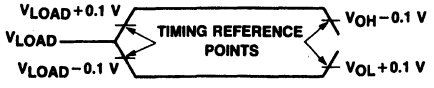
Input, Output Waveforms



270538-15

AC Inputs during testing are driven at $V_{CC}-0.5V$ for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at V_{IH} min for a Logic "1" and V_{OL} max for a Logic "0".

Float Waveforms



270538-16

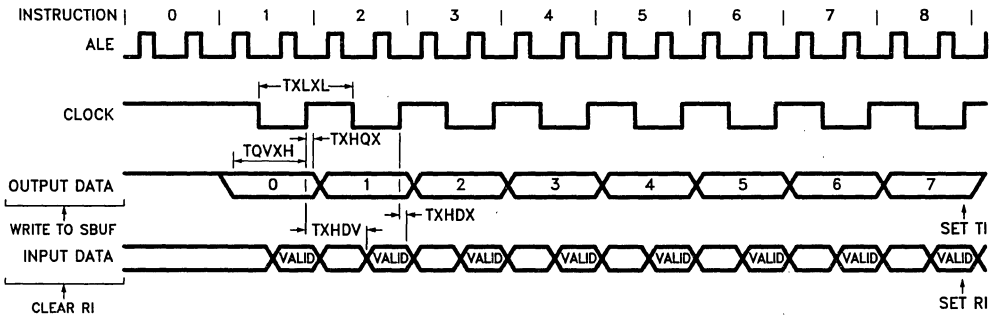
For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

SERIAL PORT TIMING—SHIFT REGISTER MODE

Test Conditions: $T_A = 0°C$ to $+70°C$; $V_{CC} = 5V \pm 20\%$; $V_{SS} = 0V$; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold after Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns

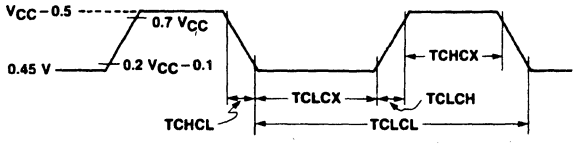
SHIFT REGISTER MODE TIMING WAVEFORMS



EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency			MHz
	83C51FA/80C51FA	3.5	12	
	83C51FA-1/80C51FA-1	3.5	16	
	83C51FA-2/80C51FA-2	0.5	12	
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

EXTERNAL CLOCK DRIVE WAVEFORM



ROM CHARACTERISTICS

Table 2 shows the logic levels for verifying the code data and reading the signature bytes on the 83C51FA.

Table 2. ROM Modes

Mode	RST	$\overline{PSEN}$	ALE	$\overline{EA}$	P2.7	P2.6	P3.6	P3.7
Verify Code Data	1	0	1	1	0	0	1	1
Read Signature	1	0	1	1	0	0	0	0

NOTES:
"1" = Valid high for that pin
"0" = Valid low for that pin

Program Verification

If the Program Lock Bit has not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0–P2.4. The other pins should be held at the "Verify" levels indicated in Table 2. The contents of the addressed locations will come out on Port 0. External pullups are required on Port 0 for this operation.

If the Encryption Array in the ROM has been programmed, the data present at Port 0 will be Code Data XNOR Encryption Data. The user must know the Encryption Array contents to manually "unencrypt" the data during verify.

Figure 10 shows the setup for verifying the program memory.

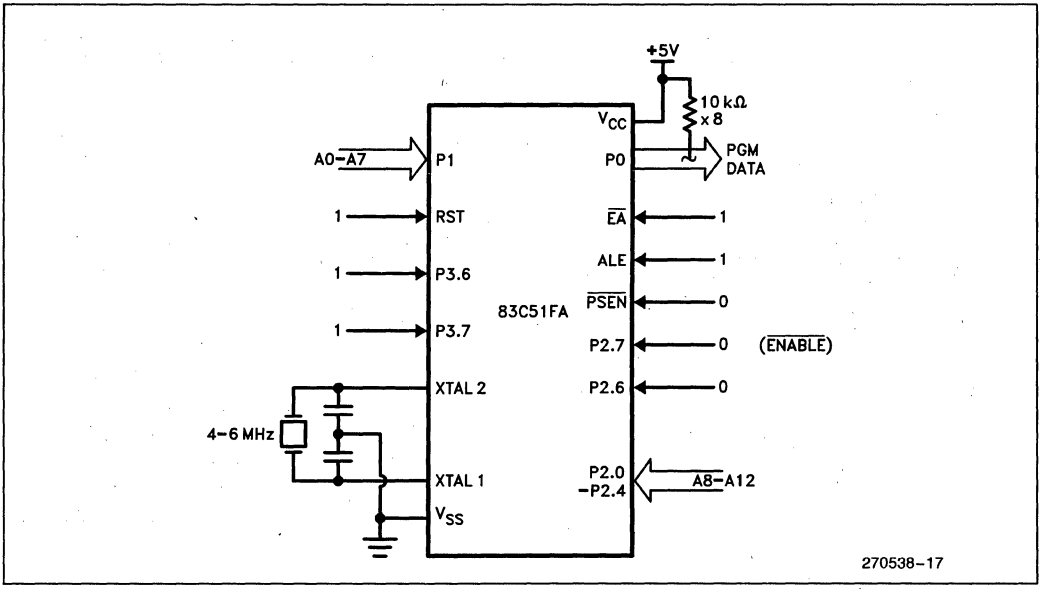


Figure 10. Verifying the ROM

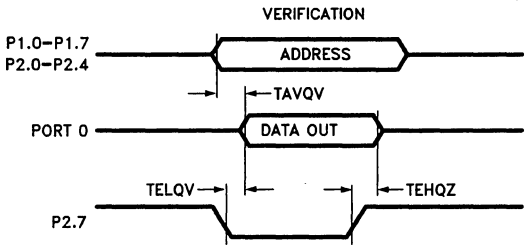
ROM VERIFICATION CHARACTERISTICS

T_A = 21°C to 27°C; V_{CC} = 5V ± 0.25V; V_{SS} = 0V

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVQV	Address to Data Valid		48TCLCL	
TELQV	ENABLE Low to Data Valid		48TCLCL	
TEHQZ	Data Float after ENABLE	0	48TCLCL	

ROM VERIFICATION WAVEFORMS



ROM Program Lock

The Program Lock system consists of one Program Lock bit and a 32 byte Encryption Array which are used to protect the program memory against software piracy.

Table 3 outlines the features of programming the Lock Bit.

Encryption Array

Within the ROM array are 32 bytes of Encryption Array that are initially unprogrammed (all 1's). Every time that a byte is addressed during a verify, 5 address lines are used to select a byte of the Encryption Array. This byte is then exclusive-NOR'ed (XNOR) with the code byte, creating an Encrypted Verify byte. The algorithm, with the array in the unprogrammed state (all 1's), will return the code in it's original, unmodified form.

Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

(030H) = 89H indicates manufacture by Intel
(031H) = 53H indicates 83C51FA

Table 3. Program Lock Bit and its Features

Program Lock Bit LB1	Logic Enabled
U	No Program Lock features enabled. (Code Verify will still be encrypted by the Encryption Array.)
P	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -001 version of the 83C51FA/80C51FA data sheet:

1. Data sheet was upgraded from ADVANCE INFORMATION to PRELIMINARY.
2. The old device name (83C252/80C252) was removed from the title.
3. PLCC pin connection diagram was added.
4. Package table was added.
5. Exit from Power Down Mode was clarified.
6. Maximum I_{OL} per I/O pin was added to the ABSOLUTE MAXIMUM RATINGS.
7. Note 4 was added to explain the maximum safe current spec.
8. I_{pd} was improved from 100 μA to 75 μA.
9. Typical DC characteristics were added for: I_{IL}, I_{LI}, I_{TL}, RRST, I_{CC}.
10. Note 5 was added to explain the test conditions for typical values.
11. Maximum clock frequency was added to the AC table.
12. Timing spec's improved for:

    TAVLL changed from TCLCL-55 to TCLCL-40
    TLLAX changed from TCLCL-35 to TCLCL-30
    TLLPL changed from TCLCL-40 to TCLCL-30
    TRHDZ changed from TCLCL-70 to TCLCL-60
    TQVWX changed from "Address Valid Before WR" to "Data Valid to WR Transition", and changed from TCLCL-60 to TCLCL-50
    TQVWH was added.
13. Data sheet revision summary was added.

83C51FA/80C51FA

EXPRESS

83C51FA/80C51FA—3.5 MHz to 12 MHz, V_{CC} = 5V ±10%
83C51FA-1/80C51FA-1—3.5 MHz to 16 MHz, V_{CC} = 5V ±10%
83C51FA-2/80C51FA-2—0.5 MHz to 12 MHz, V_{CC} = 5V ±10%

- **Extended Temperature Range**
- **Burn-In**

The Intel EXPRESS system offers enhancements to the operational specifications of the MCS®-51 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in and an extended temperature range with or without burn-in.

With the commercial standard temperature range, operational characteristics are guaranteed over the temperature range of 0°C to 70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

The optional burn-in is dynamic for a minimum time of 168 hours at 125°C with V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

August 1988
Order Number: 270620-001

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets.

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
I_{IL}	Logical 0 Input Current (Port 1, 2, 3)		−75	μA	$V_{in} = 0.45V$
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	$V_{CC} - 1.5$		V	$I_{OH} = -6.0$ mA

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range	Burn-In
P	Plastic	Commercial	No
D	Cerdip	Commercial	No
N	PLCC	Commercial	No
TP	Plastic	Extended	No
TD	Cerdip	Extended	No
TN	PLCC	Extended	No
LP	Plastic	Extended	Yes
LD	Cerdip	Extended	Yes
LN	PLCC	Extended	Yes

NOTE:
- Commercial temperature range is 0°C to 70°C. Extended temperature range is −40°C to +85°C.
- Burn-in is dynamic for a minimum time of 168 hours at 125°C, $V_{CC} = 6.9V \pm 0.25V$, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).

Examples:

P83C51FA indicates 83C51FA in a plastic package and specified for commercial temperature range, without burn-in.
LD80C51FA indicates 80C51FA in a cerdip package and specified for extended temperature range with burn-in.

87C51FA
CHMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH PROGRAMMABLE COUNTER ARRAY, UP/DOWN COUNTER, 8K BYTES USER PROGRAMMABLE EPROM

- High Performance CHMOS EPROM
- Power Control Modes
- Three 16-Bit Timer/Counters
- Programmable Counter Array with:
  - High Speed Output,
  - Compare/Capture,
  - Pulse Width Modulator,
  - Watchdog Timer capabilities
- Up/Down Timer/Counter
- Two Level Program Lock System
- 8K On-Chip EPROM
- 256 Bytes of On-Chip Data RAM
- Quick Pulse Programming™ Algorithm
- Boolean Processor
- 32 Programmable I/O Lines
- 7 Interrupt Sources
- Programmable Serial Channel with:
  - Framing Error Detection
  - Automatic Address Recognition
- TTL Compatible Logic Levels
- 64K External Program Memory Space
- 64K External Data Memory Space
- MCS®-51 Fully Compatible Instruction Set
- Power Saving Idle and Power Down Modes
- ONCE™ (On-Circuit Emulation) Mode

MEMORY ORGANIZATION

PROGRAM MEMORY: Up to 8K bytes of the program memory can reside in the on-chip EPROM. In addition the device can address up to 64K of program memory external to the chip.

DATA MEMORY: This microcontroller has a 256 x 8 on-chip RAM. In addition it can address up to 64K bytes of external data memory.

The Intel 87C51FA is a single-chip control oriented microcontroller which is fabricated on Intel's reliable CHMOS II-E technology. Being a member of the MCS®-51 family, the 87C51FA uses the same powerful instruction set, has the same architecture, and is pin for pin compatible with the existing MCS-51 products. The 87C51FA is an enhanced version of the 87C51. It's added features make it an even more powerful microcontroller for applications that require Pulse Width Modulation, High Speed I/O, and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multi-processor communications.

October 1988
Order Number: 270258-003

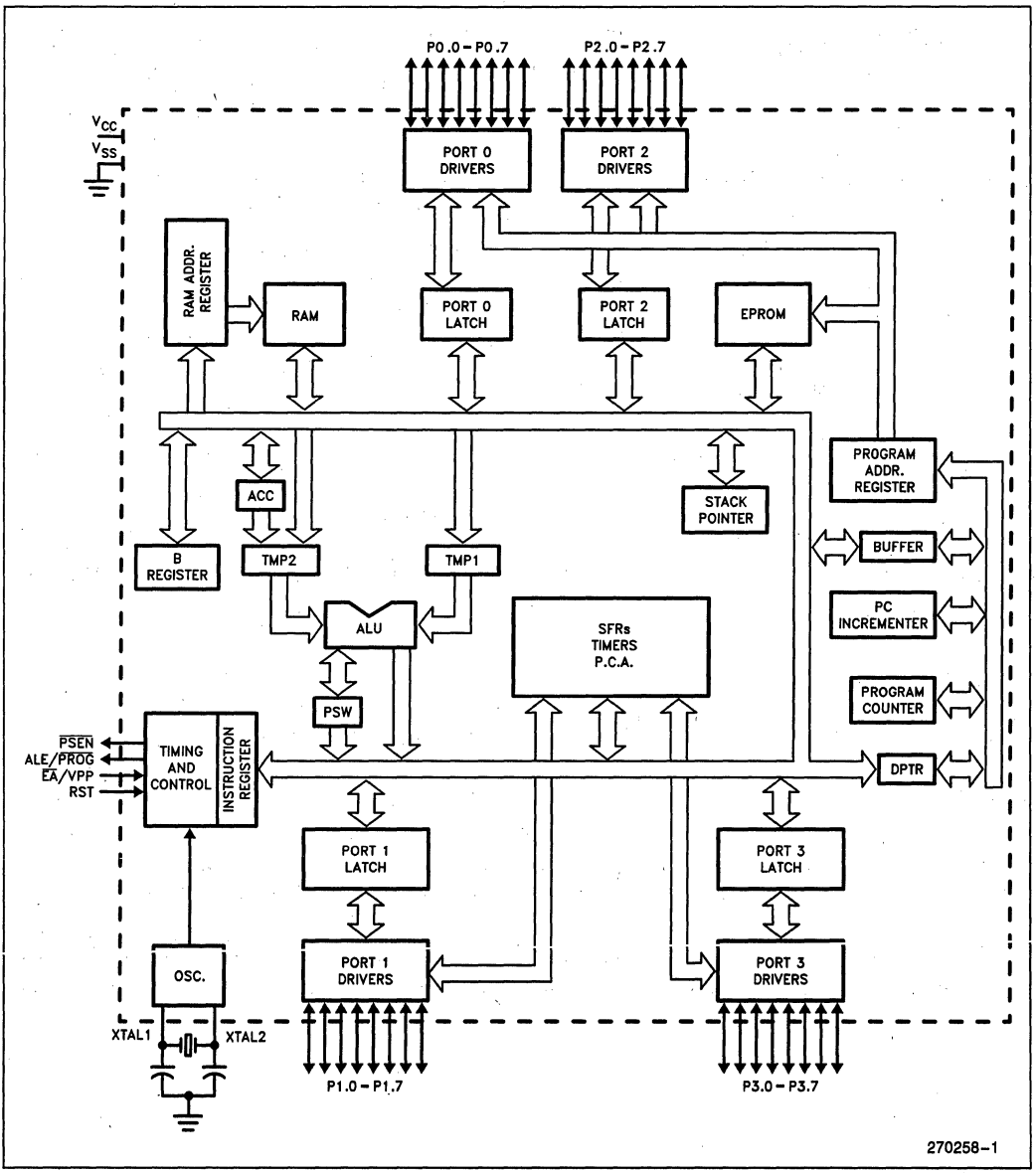


Figure 1. 87C51FA Block Diagram

PACKAGES

Part	Prefix	Package Type
87C51FA	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-PIN PLCC

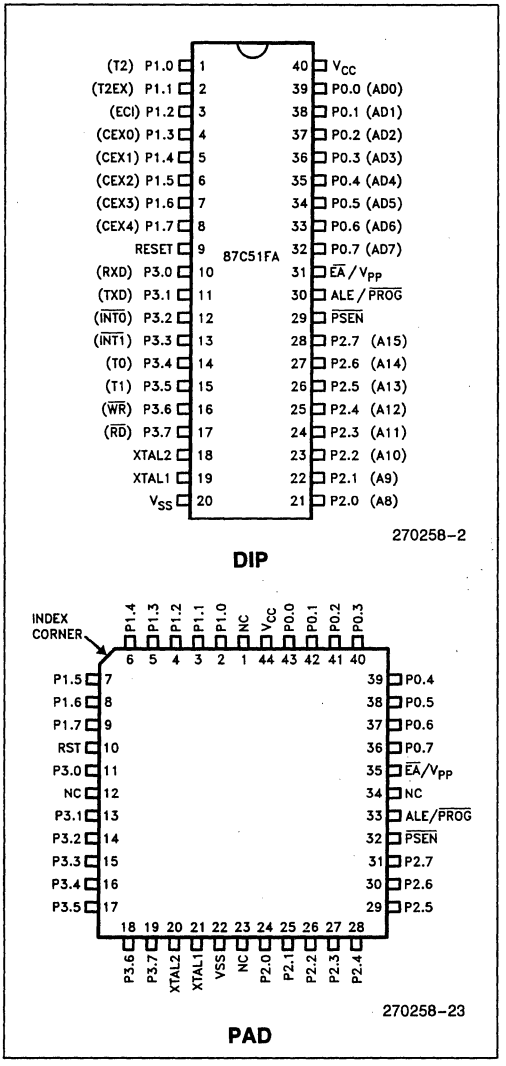


Figure 2. Pin Connections

PIN DESCRIPTIONS

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit, open drain, bidirectional I/O port. As an output port each pin can sink several LS TTL inputs. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting1's, and can source and sink several LS TTL inputs.

Port 0 also receives the code bytes during EPROM programming, and outputs the code bytes during program verification. External pullup resistors are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can drive LS TTL inputs. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

In addition, Port 1 serves the functions of the following special features of the 87C51FA:

Port Pin	Alternate Function
P1.0	T2 (External Count Input to Timer/ Counter 2)
P1.1	T2EX (Timer/Counter 2 Capture/ Reload Trigger and Direction Control)
P1.2	ECI (External Count Input to the PCA)
P1.3	CEX0 (External I/O for Compare/ Capture Module 0)
P1.4	CEX1 (External I/O for Compare/ Capture Module 1)
P1.5	CEX2 (External I/O for Compare/ Capture Module 2)
P1.6	CEX3 (External I/O for Compare/ Capture Module 3)
P1.7	CEX4 (External I/O for Compare/ Capture Module 4)

Port 1 receives the low-order address bytes during EPROM programming and verifying.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can drive LS TTL inputs. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Some Port 2 pins receive the high-order address bits during EPROM programming and program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can drive LS TTL inputs. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits a power-on reset with only a capacitor connected to V_{CC}.

ALE: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin (ALE/$\overline{PROG}$) is also the program pulse input during EPROM programming for the 87C51FA.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

Throughout the remainder of this data sheet, ALE will refer to the signal coming out of the ALE/$\overline{PROG}$ pin, and the pin will be referred to as the ALE/$\overline{PROG}$ pin.

$\overline{PSEN}$: Program Store Enable is the read strobe to external Program Memory.

When the 87C51FA is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory.

$\overline{EA}$/V_{PP}: External Access enable. $\overline{EA}$ must be strapped to VSS in order to enable the device to fetch code from external Program Memory locations 0000H to 0FFFH. Note, however, that if either of the Program Lock bits are programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the programming supply voltage (V_{PP}) during EPROM programming.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of a inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 floats, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the data sheet must be observed.

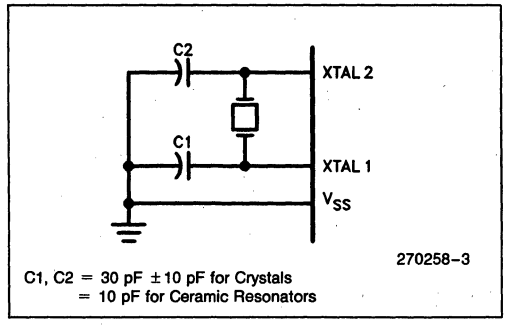


Figure 3. Oscillator Connections

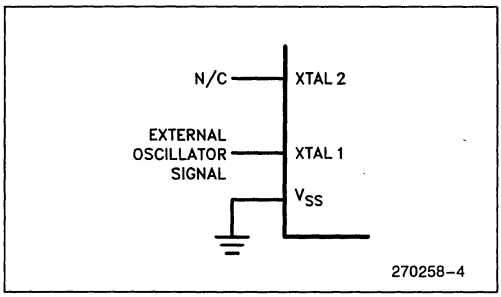


Figure 4. External Clock Drive Configuration

IDLE MODE

The user's software can invoke the Idle Mode. When the microcontroller is in this mode, power consumption is reduced. The Special Function Registers and the onboard RAM retain their values during Idle, but the processor stops executing instructions. Idle Mode will be exited if the chip is reset or if an enabled interrupt occurs. The PCA timer/counter can optionally be left running or paused during Idle Mode.

POWER DOWN MODE

To save even more power, a Power Down mode can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 87C51FA either hardware reset or external interrupt can cause an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

DESIGN CONSIDERATION

- Ambient light is known to affect the internal RAM contents during operation. If the 87C51FA application requires the part to be run under ambient lighting, an opaque label should be placed over the window to exclude light.
- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

ONCE™ MODE

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 87C51FA without the 87C51FA having to be removed from the circuit. The ONCE Mode is invoked by:

1) Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2) Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the 87C51FA is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

Table 1. Status of the External Pins during Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

NOTE:
For more detailed information on the reduced power modes refer to current Embedded Controller Handbook, and Application Note AP-252, "Designing with the 80C51BH."

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on EA/V_{PP} Pin to V_{SS} .....0V to +13.0V
Voltage on Any Other Pin to V_{SS} ..−0.5V to +6.5V
Maximum I_{OL} per I/O Pin .....15 mA
Power Dissipation.....1.5W
(based on PACKAGE heat transfer limitations, not device power consumption)

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

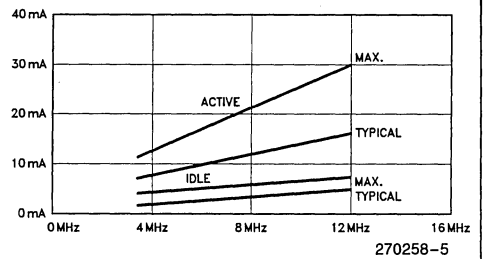
NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS: (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V)

Symbol	Parameter	Min	Typical (Note 4)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5		0.2 V_{CC}−0.1	V	
V_{IL1}	Input Low Voltage $\overline{EA}$	0		0.2 V_{CC}−0.3	V	
V_{IH}	Input High Voltage (Except XTAL1, RST, $\overline{EA}$)	0.2 V_{CC}+0.9		V_{CC}+0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}		V_{CC}+0.5	V	
V_{OL}	Output Low Voltage(5) (Ports 1, 2 and 3)			0.3 0.45 1.0	V	I_{OL} = 100μA I_{OL} = 1.6 mA(1) I_{OL} = 3.5 mA
V_{OL1}	Output Low Voltage(5) (Port 0, ALE/$\overline{PROG}$, $\overline{PSEN}$)			0.3 0.45 1.0	V	I_{OL} = 200 μA I_{OL} = 3.2 mA(1) I_{OL} = 7.0 mA
V_{OH}	Output High Voltage (Ports 1, 2 and 3 ALE/$\overline{PROG}$ and $\overline{PSEN}$)	V_{CC} − 0.3 V_{CC} −0.7 V_{CC} −1.5			V V V	I_{OH} = −10 μA I_{OH} = −30 μA(2) I_{OH} = −60 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	V_{CC} −0.3 V_{CC} −0.7 V_{CC} −1.5			V V V	I_{OH} = −200 μA I_{OH} = −3.2 mA(2) I_{OH} = −7.0 mA
I_{IL}	Logical 0 Input Current (Ports 1, 2, and 3)		−10	−50	μA	V_{IN} = 0.45V
I_{LI}	Input leakage Current (Port 0)		0.02	±10	μA	V_{IN} = V_{IL} or V_{IH}
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, and 3)		−265	−650	μA	V_{IN} = 2V
RRST	RST Pulldown Resistor	40	100	225	KΩ	
CIO	Pin Capacitance			10	pF	@1MHz, 25°C
I_{CC}	Power Supply Current: Running at 12 MHz (Figure 5) Idle Mode at 12 MHz (Figure 5) Power Down Mode		 15 5 5	 30 7.5 75	 mA mA μA	(Note 3)

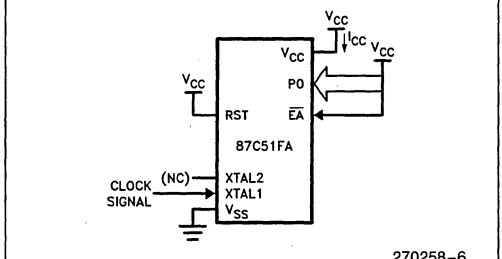
NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In applications where capacitance loading exceeds 100 pFs, the noise pulse on the ALE signal may exceed 0.8V. In these cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an Address Latch with a Schmitt Trigger Strobe input.
2. Capacitive loading on Ports 0 and 2 cause the V_{OH} on ALE and $\overline{PSEN}$ to drop below the 0.9 V_{CC} specification when the address lines are stabilizing.
3. See Figures 6–9 for test conditions. Minimum V_{CC} for power down is 2V.
4. Typicals are based on limited number of samples, and are not guaranteed. The values listed are at room temperature and 5V.
5. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port -
   Port 0: 26 mA
   Ports 1, 2, and 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.



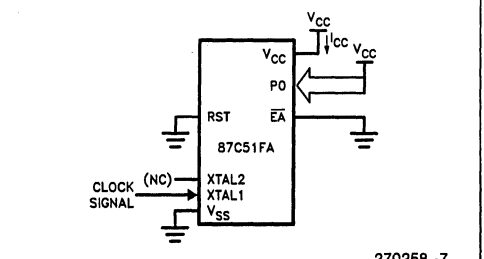
I_{CC} Max at other frequencies is given by:
Active Mode
I_{CC} MAX = 2.2 × FREQ + 3.1
Idle Mode
I_{CC} MAX = 0.49 × FREQ + 1.6
Where FREQ is in MHz, I_{CC}MAX is given in mA.

Figure 5. I_{CC} vs Frequency



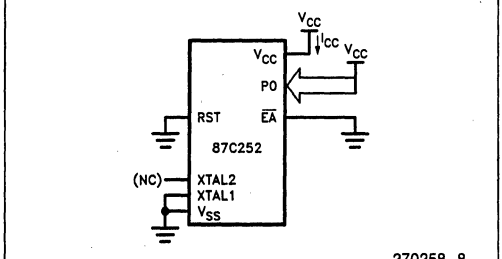
All other pins disconnected
TCLCH = TCHCL = 5 ns

Figure 6. I_{CC} Test Condition, Active Mode



All other pins disconnected
TCLCH = TCHCL = 5 ns

Figure 7. I_{CC} Test Condition Idle Mode



All other pins disconnected

Figure 8. I_{CC} Test Condition, Power Down Mode. V_{CC} = 2.0V to 5.5V.

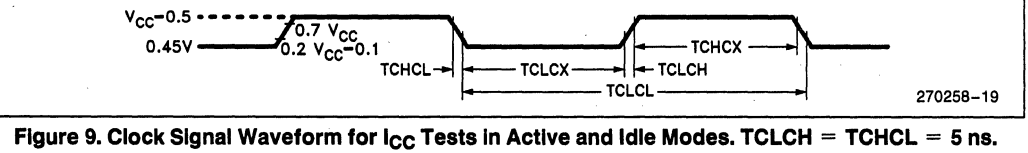


Figure 9. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address
C: Clock
D: Input Data
H: Logic level HIGH
I: Instruction (program memory contents)
L: Logic level LOW, or ALE
P: $\overline{PSEN}$
Q: Output Data
R: $\overline{RD}$ signal
T: Time
V: Valid
W: $\overline{WR}$ signal
X: No longer a valid logic level
Z: Float

For example,

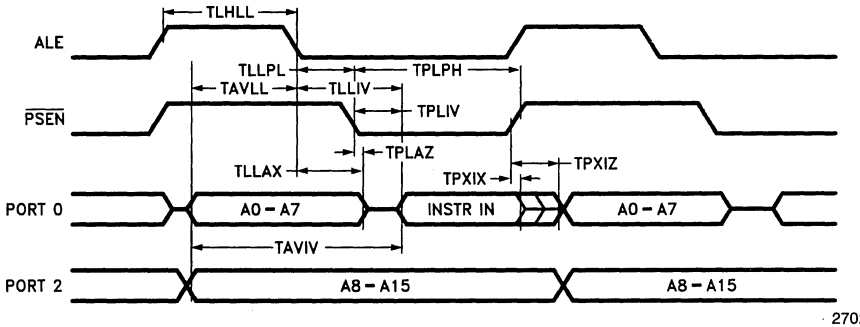
TAVLL = Time from Address Valid to ALE Low
TLLPL = Time from ALE Low to PSEN Low

A.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = 5V ±10%, V_{SS} = 0V, Load Capacitance for Port 0, ALE/$\overline{PROG}$ and PSEN = 100 pF, Load Capacitance for All Other Outputs = 80 pF)

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

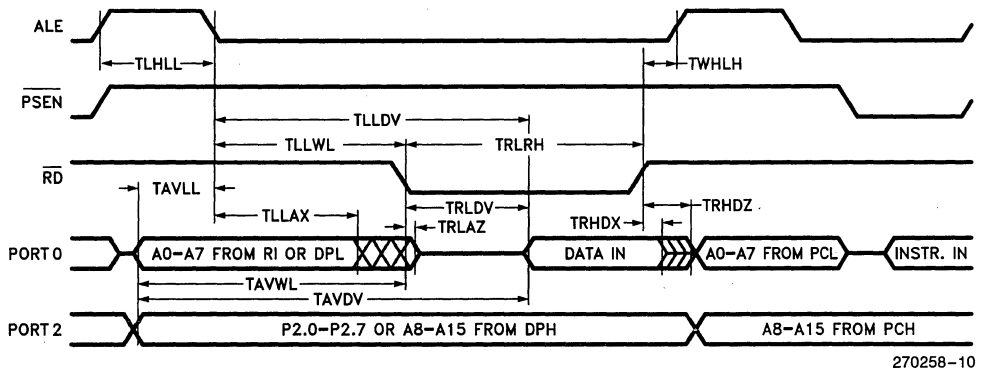
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency			3.5	12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL−40		ns
TLLAX	Address Hold After ALE Low	53		TCLCL−30		ns
TLLIV	ALE Low to Valid Instruction In		234		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	53		TCLCL−30		ns
TPLPH	$\overline{PSEN}$ Pulse Width	205		3TCLCL−45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instruction In		145		3TCLCL−105	ns
TPXIX	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instruction Float After $\overline{PSEN}$		59		TCLCL−25	ns
TAVIV	Address to Valid Instruction In		312		5TCLCL−105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		107		2TCLCL−60	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address Valid to $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	33		TCLCL−50		ns
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL−50		ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL−150		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL−40	TCLCL+40	ns

EXTERNAL PROGRAM MEMORY READ CYCLE



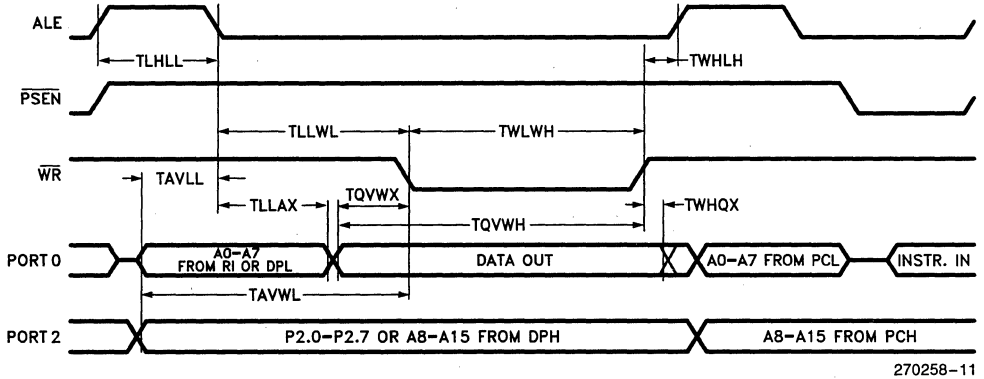
270258-9

EXTERNAL DATA MEMORY READ CYCLE



270258-10

EXTERNAL DATA MEMORY WRITE CYCLE



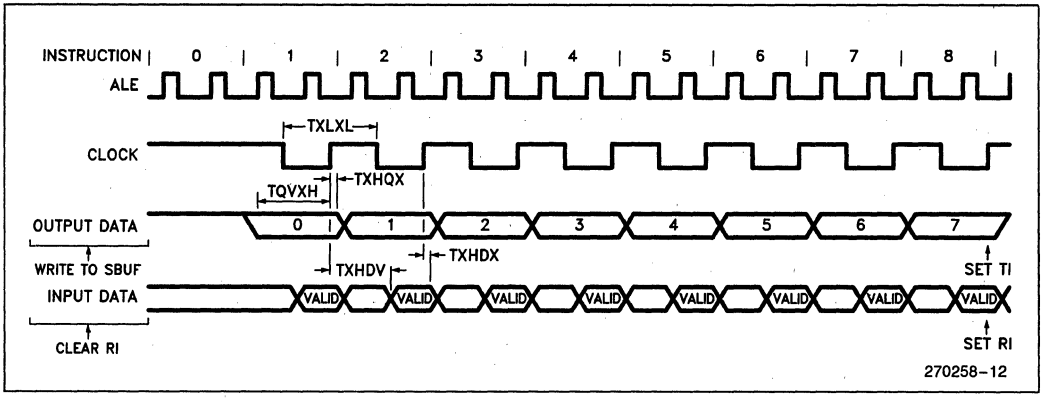
270258-11

SERIAL PORT TIMING - SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold after Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns

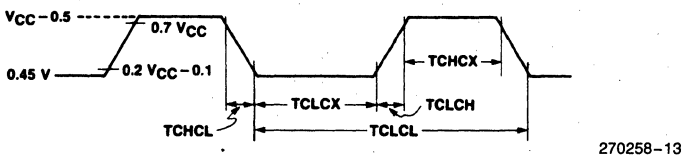
SHIFT REGISTER MODE TIMING WAVEFORMS



EXTERNAL CLOCK DRIVE

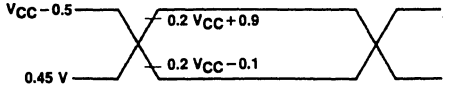
Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	3.5	12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

EXTERNAL CLOCK DRIVE WAVEFORM



A.C. TESTING INPUT

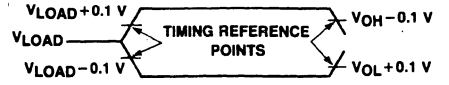
Input, Output Waveforms



270258-14

AC Inputs during testing are driven at $V_{CC}-0.5V$ for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at V_{IH} min for a Logic "1" and V_{OL} max for a Logic "0".

Float Waveforms



270258-15

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

EPROM CHARACTERISTICS

Table 2 shows the logic levels for programming the Program Memory, the Encryption Table, and the Lock Bits and for reading the signature bytes.

Table 2. EPROM Programming Modes

Mode	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.7	P2.6	P3.6	P3.7
Program Code Data	1	0	0*	V_{PP}	1	0	1	1
Verify Code Data	1	0	1	1	0	0	1	1
Program Encryption Table Use Addresses 0-1FH	1	0	0*	V_{PP}	1	0	0	1
Program Lock x=1	1	0	0*	V_{PP}	1	1	1	1
Bits (LBx) x=2	1	0	0*	V_{PP}	1	1	0	0
Read Signature	1	0	1	1	0	0	0	0

NOTES:
"1" = Valid high for that pin
'0" = Valid low for that pin
"VPP" = +12.75V ±0.25V
* ALE/$\overline{PROG}$ is pulsed low for 100 μs for programming. (Quick-Pulse Programming™)

PROGRAMMING THE EPROM

To be programmed, the part must be running with a 4 to 6 MHz oscillator. (The reason the oscillator needs to be running is that the internal bus is being used to transfer address and program data to appropriate internal EPROM locations.) The address of an EPROM location to be programmed is applied to Port 1 and pins P2.0 - P2.4 of Port 2, while the code byte to be programmed into that location is applied to Port 0. The other Port 2 and 3 pins, RST $\overline{PSEN}$, and $\overline{EA}/V_{PP}$ should be held at the "Program" levels indicated in Table 2. ALE/$\overline{PROG}$ is pulsed low to program the code byte into the addressed EPROM location. The setup is shown in Figure 10.

Normally $\overline{EA}/V_{PP}$ is held at logic high until just before ALE/$\overline{PROG}$ is to be pulsed. Then $\overline{EA}/V_{PP}$ is raised to V_{PP}, ALE/$\overline{PROG}$ is pulsed low, and then $\overline{EA}/V_{PP}$ is returned to a valid high voltage. The voltage on the $\overline{EA}/V_{PP}$ pin must be at the valid $\overline{EA}/V_{PP}$ high level before a verify is attempted. Waveforms and detailed timing specifications are shown in later sections of this data sheet.

Note that the $\overline{EA}/V_{PP}$ pin must not be allowed to go above the maximum specified V_{PP} level for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The V_{PP} source should be well regulated and free of glitches.

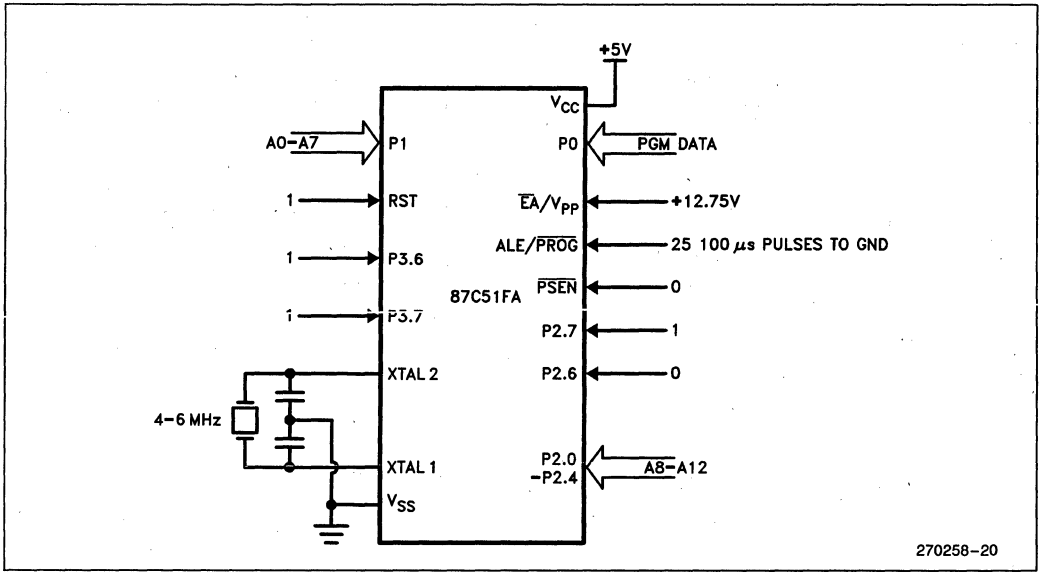


Figure 10. Programming the EPROM

Quick-Pulse Programming™ Algorithm

The 87C51FA can be programmed using the Quick-Pulse Programming™ Algorithm for microcontrollers. The features of the new programming method are a lower V_{PP} (12.75V as compared to 21V) and a shorter programming pulse. It is possible to program the entire 8K Bytes of EPROM memory in less than 25 seconds with this algorithm!

To program the part using the new algorithm, V_{PP} must be 12.75V ±0.25V. ALE/$\overline{PROG}$ is pulsed low for 100 µs, 25 times as shown in Figure 11. Then, the byte just programmed may be verified. After programming, the entire array should be verified. The Program Lock features are programmed using the same method, but with the setup as shown in Table 2. The only difference in programming Program Lock features is that the Program Lock features cannot be directly verified. Instead, verification of programming is by observing that their features are enabled.

Program Verification

If the Program Lock Bits have not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired, either during or after the programming operation. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0 - P2.4. The other pins should be held at the "Verify" levels indicated in Table 3. The contents of the addressed locations will come out on Port 0. External pullups are required on Port 0 for this operation.

If the Encryption Array in the EPROM has been programmed, the data present at Port 0 will be Code Data XNOR Encryption Data. The user must know the Encryption Array contents to manually "unencrypt" the data during verify.

The setup, which is shown in Figure 12, is the same as for programming the EPROM except that pin P2.7 is held at a logic low, or may be used as an active-low read strobe.

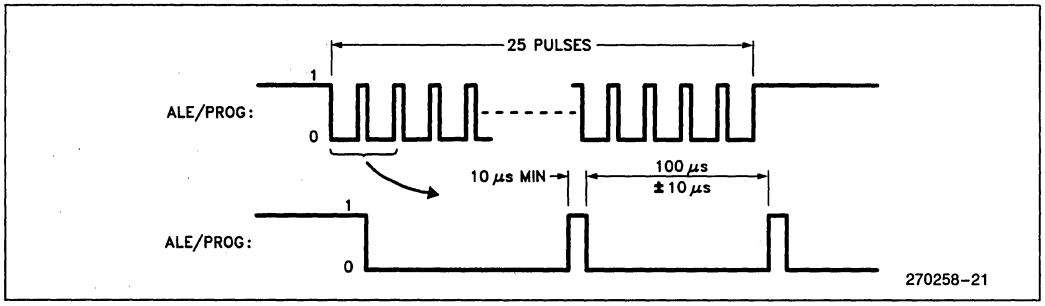


Figure 11. $\overline{\text{PROG}}$ Waveforms

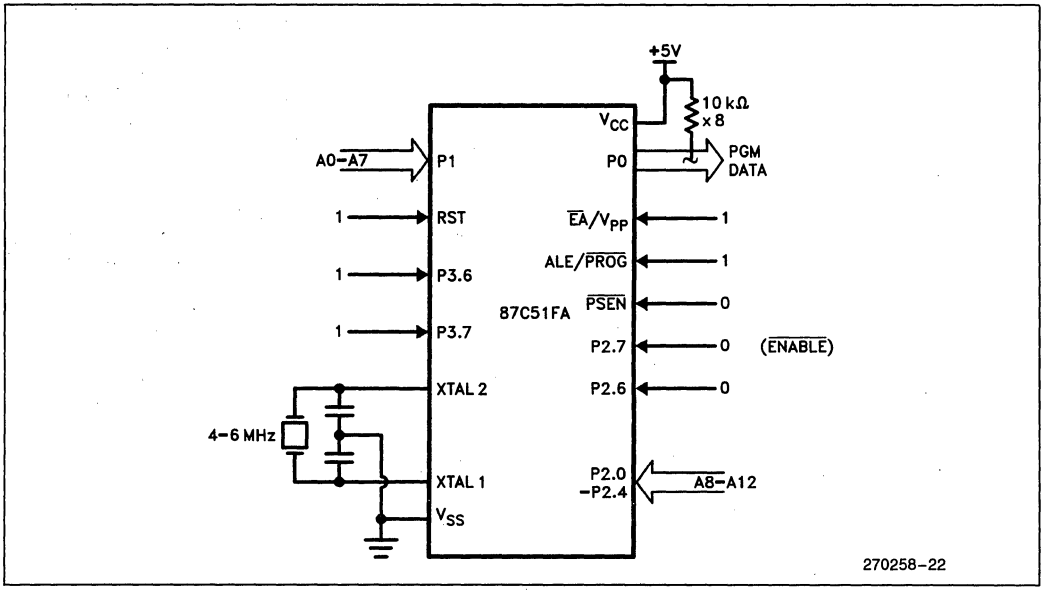


Figure 12. Verifying the EPROM

EPROM Program Lock

The two-level Program Lock system consists of two Program Lock bits and a 32 byte Encryption Array which are used to protect the program memory against software piracy.

Encryption Array

Within the EPROM array are 32 bytes of Encryption Array that are initially unprogrammed (all 1's). Every time that a byte is addressed during a verify, 5 address lines are used to select a byte of the Encryption Array. This byte is then exclusive-NOR'ed (XNOR) with the code byte, creating an Encrypted Verify byte. The algorithm, with the array in the unprogrammed state (all 1's), will return the code in it's original, unmodified form.

Program Lock Bits

Also included in the EPROM Program Lock scheme are two Program Lock Bits which are programmed as shown in Table 2.

Table 3 outlines the features of programming the Lock Bits.

Erasing the EPROM also erases the Encryption Array and the Program Lock Bits, returning the part to full functionality.

Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

(030H) = 89H indicates manufacture by Intel
(031H) = 50H indicates 87C51FA

Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelength shorter than approximately 4,000 Angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room-level fluorescent lighting) could cause inadvertent erasure. If an application subjects the device to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 Angstroms) to an integrated dose of at least 15 W-sec/cm. Exposing the EPROM to an ultraviolet lamp of 12,000 μW/cm rating for 30 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the all EPROM Cells in a 1's state.

Table 3. Program Lock Bits and their Features

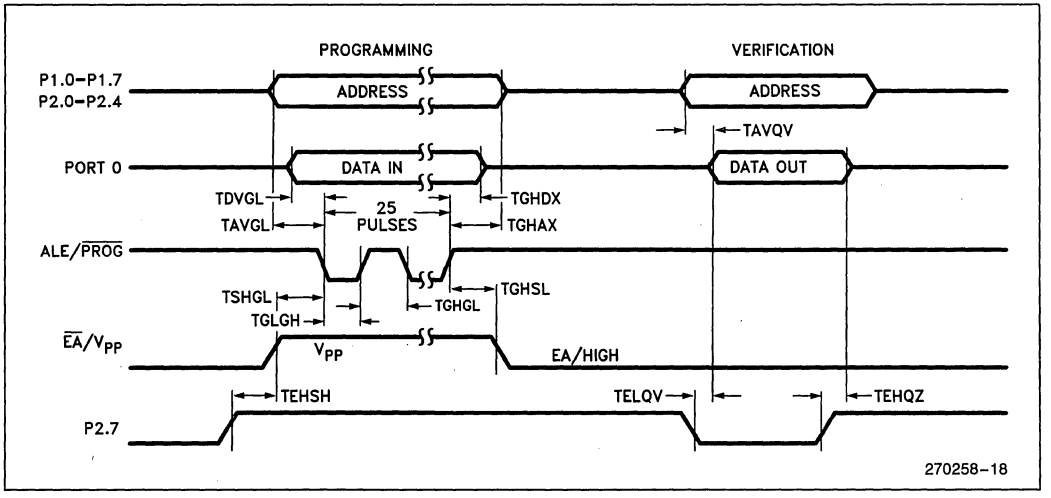
Program Lock Bits LB1	LB2	Logic Enabled
U	U	No Program Lock features enabled. (Code Verify will still be encrypted by the Encryption Array.)
P	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the EPROM is disabled.
P	P	Same as above, but Verify is also disabled
U	P	Reserved for Future Definition

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

(T_A = 21°C to 27°C; V_{CC} = 5V ± 0.25V; V_{SS} = 0V)

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	12.5	13.0	V
I_{PP}	Programming Supply Current		50	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$ Low	48TCLCL		
TGHAX	Address Hold after $\overline{PROG}$	48TCLCL		
TDVGL	Data Setup to $\overline{PROG}$ Low	48TCLCL		
TGHDX	Data Hold after $\overline{PROG}$	48TCLCL		
TEHSH	P2.7 (ENABLE) High to V_{PP}	48TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
TGHSL	V_{PP} Hold after $\overline{PROG}$	10		µs
TGLGH	$\overline{PROG}$ Width	90	110	µs
TAVQV	Address to Data Valid		48TCLCL	
TELQV	ENABLE Low to Data Valid		48TCLCL	
TEHQZ	Data Float after ENABLE	0	48TCLCL	
TGHGL	$\overline{PROG}$ High to $\overline{PROG}$ Low	10		µs

EPROM PROGRAMMING AND VERIFICATION WAVEFORMS



DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -002 version of the 87C51FA data sheet:

1. Data sheet was upgraded from ADVANCE INFORMATION to PRELIMINARY.
2. The old device name (87C252) was removed from the title.
3. PLCC pin connection diagram was added.
4. Package table was added.
5. Exit from Power Down Mode was clarified.
6. Maximum I_{OL} per I/O pin was added to ABSOLUTE MAXIMUM RATINGS.
7. Note 4 was added to explain the maximum safe current spec.
8. I_{PD} was improved from 100 μA to 75 μA.
9. Typical DC characteristics were added for: I_{IL}, I_{LI}, I_{TL}, RRST and I_{CC}.
10. Note 5 was added to explain the test conditions for typical values.
11. Timing spec's improved for:
    - TAVLL changed from TCLCL−55 to TCLCL−40
    - TLLAX changed from TCLCL−35 to TCLCL−30
    - TLLPL changed from TCLCL−40 to TCLCL−30
    - TRHDZ changed from TCLCL−70 to TCLCL−60
    - TQVWX changed from "Address Valid Before WR" to "Data Valid to WR Transition" and changed from TCLCL−60 to TCLCL−50
    - TQVWH was added.
12. Data sheet revision summary was added.
13. EA Leakage current not spec'ed.

87C51FA

EXPRESS

- **Extended Temperature Range**
- **3.5 MHz to 12 MHz V_{CC} = 5V ±10%**
- **Burn-In**

The Intel EXPRESS system offers enhancements to the operational specifications of the 8051 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in and an extended temperature range with or without burn-in.

With the commercial standard temperature range, operational characteristics are guaranteed over the temperature range of 0°C to +70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

The optional burn-in is dynamic for a minimum time of 168 hours at 125°C with V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

June 1988
Order Number: 270619-001

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets.

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
I_{IL}	Logical 0 Input Current (Port 1, 2, 3)		−75	μA	$V_{IN} = 0.45V$
I_{LI}	Input Leakage Current (Port 0 and $\overline{EA}$)		±15	μA	$V_{IN} = V_{IL}$ or V_{IH}
I_{TL}	Logical 1 to 0 transition Current (Ports 1, 2, 3)		−750	μA	$V_{IN} = 2.0V$
I_{CC}	Power Supply Current				(Note 1)
	Active Mode		35	mA	
	Idle Mode		7.5	mA	
	Power Down Mode		150	μA	

NOTE:
1. $V_{CC} = 4.5V–5.5V$, Frequency Range = 3.5 MHz–12 MHz.

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range(2)	Burn-In(3)
P	Plastic	Commercial	No
D	Cerdip	Commercial	No
N	PLCC	Commercial	No
TP	Plastic	Extended	No
TD	Cerdip	Extended	No
TN	PLCC	Extended	No
LP	Plastic	Extended	Yes
LD	Cerdip	Extended	Yes
LN	PLCC	Extended	Yes

NOTES:
2. Commercial temperature range is 0°C to +70°C. Extended temperature range is −40°C to +85°C.
3. Burn-in is dynamic for a minimum time of 168 hours at +125°C, V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).

Examples:

P87C51FA indicates 87C51FA in a plastic package and specified for commercial temperature range, without burn-in.
LD87C51FA indicates 87C51FA in a cerdip package and specified for extended temperature range with burn-in.

83C51FB
CHMOS SINGLE-CHIP 8-BIT MICROCOMPUTER

83C51FB—16K bytes of Factory Mask Programmable ROM
83C51FB—3.5 MHz to 12 MHz, V_{CC} = 5V ± 10%
83C51FB-1—3.5 MHz to 16 MHz, V_{CC} = 5V ± 10%
83C51FB-2—0.5 MHz to 12 MHz, V_{CC} = 5V ± 10%

- High Performance CHMOS EPROM
- Three 16-Bit Timer/Counters
- Programmable Counter Array with:
  - — High Speed Output,
  - — Compare/Capture,
  - — Pulse Width Modulator,
  - — Watchdog Timer Capabilities
- Up/Down Timer/Counter
- Program Lock System
- 16K bytes of On-Chip Program ROM
- 256 Bytes of On-Chip Data RAM
- Boolean Processor
- 32 Programmable I/O Lines
- 7 Interrupt Sources
- Programmable Serial Channel with:
  - — Framing Error Detection
  - — Automatic Address Recognition
- TTL and CMOS Compatible Logic Levels
- 64K bytes External Program Memory Space
- 64K bytes External Data Memory Space
- MCS®-51 Fully Compatible Instruction Set
- Power Saving Idle and Power Down Modes
- ONCE™ (On-Circuit Emulation) Mode

MEMORY ORGANIZATION

PROGRAM MEMORY: Up to 16K bytes of the program memory can reside in the on-chip ROM. In addition the device can address up to 64K bytes of program memory external to the chip.

DATA MEMORY: This microcontroller has a 256 x 8 on-chip RAM. In addition it can address up to 64K bytes of external data memory.

The Intel 83C51FB is a single-chip control-oriented microcontroller which is fabricated on Intel's reliable CHMOS IV technology. Being a member of the MCS-51 family, the 83C51FB uses the same powerful instruction set, has the same architecture, and is pin-for-pin upward compatible with the existing MCS-51 products. The 83C51FB is an enhanced version of the 80C51BH. Its added features make it an even more powerful microcontroller for applications that require Pulse Width Modulation, High Speed I/O, and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multi-processor communications.

September 1988
Order Number: 270598-002

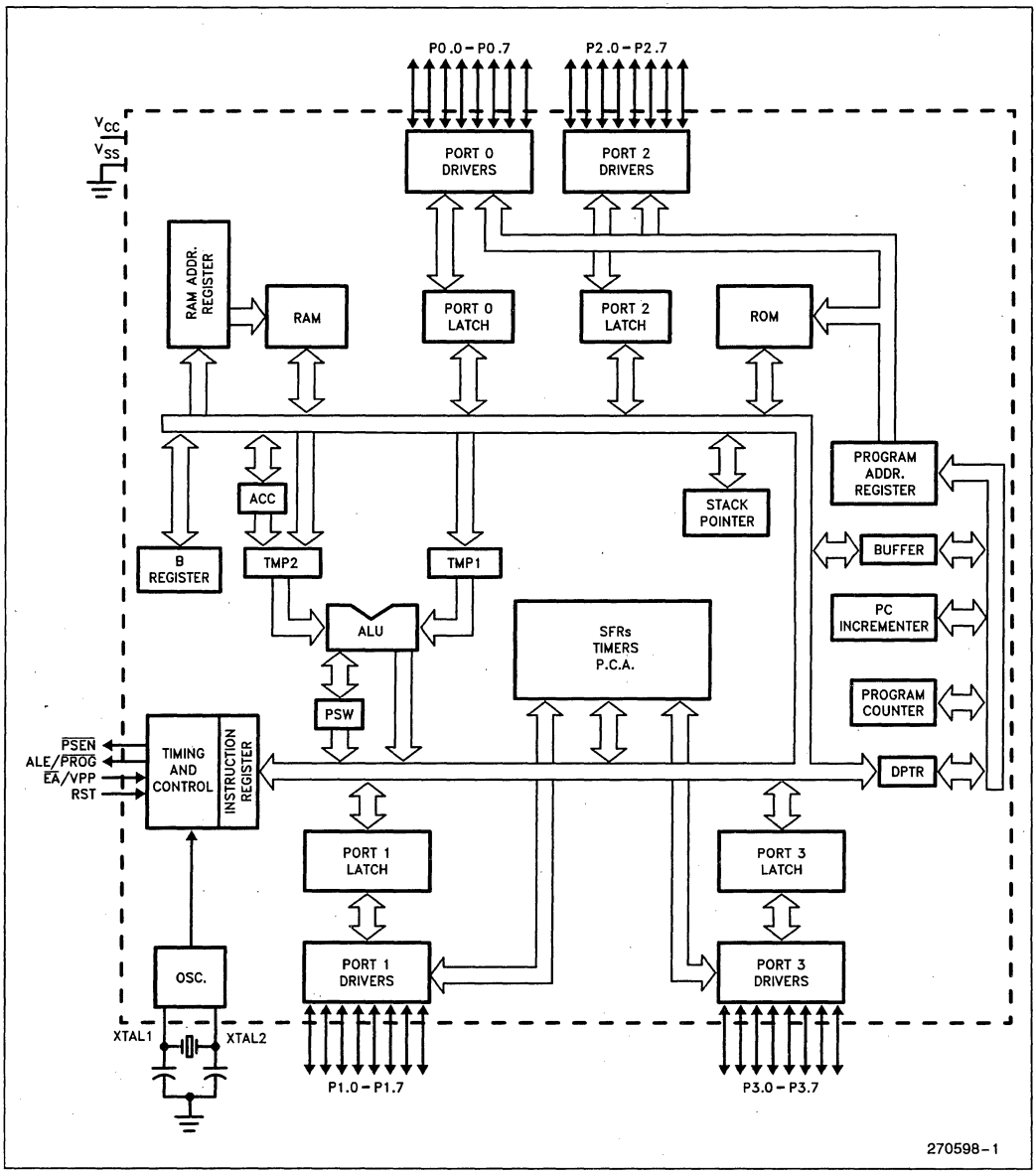
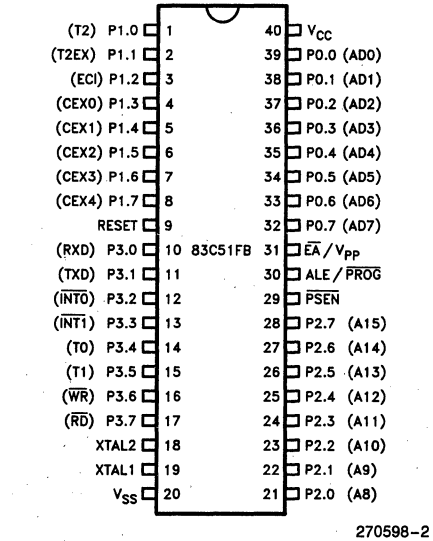


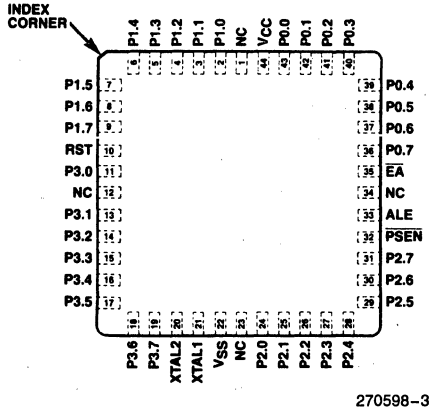
Figure 1. 83C51FB Block Diagram

PACKAGES

Part	Prefix	Package Type
83C51FB	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-Pin PLCC



DIP



PLCC

Figure 2. Connection Diagrams

PIN DESCRIPTIONS

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit, open drain, bidirectional I/O port. As an output port each pin can sink several LS TTL inputs. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's, and can source and sink several LS TTL inputs.

Port 0 outputs the code bytes during program verification on the 83C51FB. External pullup resistors are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can drive LS TTL inputs. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

In addition, Port 1 serves the functions of the following special features of the 83C51FB:

Port Pin	Alternate Function
P1.0	T2 (External Count Input to Timer/ Counter 2)
P1.1	T2EX (Timer/Counter 2 Capture/ Reload Trigger and Direction Control)
P1.2	ECI (External Count Input to the PCA)
P1.3	CEX0 (External I/O for Compare/ Capture Module 0)
P1.4	CEX1 (External I/O for Compare/ Capture Module 1)
P1.5	CEX2 (External I/O for Compare/ Capture Module 2)
P1.6	CEX3 (External I/O for Compare/ Capture Module 3)
P1.7	CEX4 (External I/O for Compare/ Capture Module 4)

Port 1 receives the low-order address bytes during ROM verification.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can drive LS TTL inputs. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Some Port 2 pins receive the high-order address bits during program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can drive LS TTL inputs. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the 8051 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits a power-on reset with only a capacitor connected to V_{CC}.

ALE: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$: Program Store Enable is the read strobe to external Program Memory.

When the 83C51FB is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory.

$\overline{EA}$: External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the device to fetch code from external Program Memory locations 0000H to 0FFFFH. Note, however, that if the Program Lock bit is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of a inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 floats, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the data sheet must be observed.

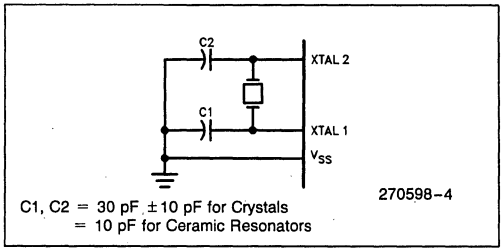


Figure 3. Oscillator Connections

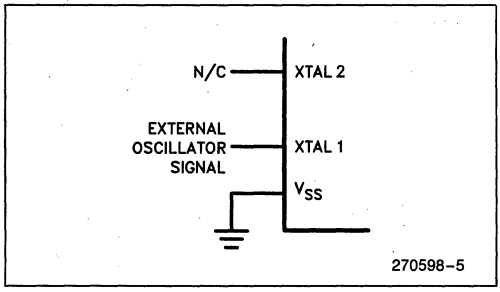


Figure 4. External Clock Drive Configuration

IDLE MODE

The user's software can invoke the Idle Mode. When the microcontroller is in this mode, power consumption is reduced. The Special Function Registers and the onboard RAM retain their values during Idle, but the processor stops executing instructions. Idle Mode will be exited if the chip is reset or if an enabled interrupt occurs. The PCA timer/counter can optionally be left running or paused during Idle Mode.

POWER DOWN MODE

To save even more power, a Power Down mode can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 83C51FB either a hardware reset or external interrupt can cause an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

DESIGN CONSIDERATION

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

ONCE™ MODE

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 83C51FB without the 83C51FB having to be removed from the circuit. The ONCE Mode is invoked by:

1) Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2) Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the 83C51FB is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

Table 1. Status of the External Pins during Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

NOTE:
For more detailed information on the reduced power modes refer to current Embedded Controller Handbook, and Application Note AP-252, "Designing with the 80C51BH."

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on $\overline{EA}$ Pin to V_{SS} .....0V to +13.0V
Voltage on Any Other Pin to V_{SS} ..−0.5V to +6.5V
Maximum I_{OL} Per I/O Pin .....15 mA
Power Dissipation.....1.5W
(based on PACKAGE heat transfer limitations, not device power consumption)

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

D.C. CHARACTERISTICS: (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V)

Symbol	Parameter	Min	Typical (Note 5)	Max (Note 5)	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5		0.2 V_{CC}−0.1	V	
V_{IL1}	Input Low Voltage $\overline{EA}$	0		0.2 V_{CC}−0.3	V	
V_{IH}	Input High Voltage (Except XTAL1, RST, $\overline{EA}$)	0.2 V_{CC}+0.9		V_{CC}+0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}		V_{CC}+0.5	V	
V_{OL}	Output Low Voltage(Note 6) (Ports 1, 2, and 3)			0.3	V	I_{OL} = 100 μA (Note 1)
				0.45	V	I_{OL} = 1.6 mA (Note 1)
				1.0	V	I_{OL} = 3.5 mA (Notes 1, 4)
V_{OL1}	Output Low Voltage (Note 6) (Port 0, ALE, $\overline{PSEN}$)			0.3	V	I_{OL} = 200 μA (Note 1)
				0.45	V	I_{OL} = 3.2 mA (Note 1)
				1.0	V	I_{OL} = 7.0 mA (Notes 1, 4)
V_{OH}	Output High Voltage (Ports 1, 2, and 3)	V_{CC}−0.3			V	I_{OH} = −10 μA
		V_{CC}−0.7			V	I_{OH} = −30 μA
		V_{CC}−1.5			V	I_{OH} = −60 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)	V_{CC}−0.3			V	I_{OH} = −200 μA
		V_{CC}−0.7			V	I_{OH} = −3.2 mA
		V_{CC}−1.5			V	I_{OH} = −7.0 mA (Note 4)
I_{IL}	Logical 0 Input Current (Ports 1, 2, and 3)			−50	μA	V_{IN} = 0.45V
I_{LI}	Input Leakage Current (Port 0)			±10	μA	0 < V_{IN} < V_{CC} − 0.3V
I_{LI1}	Input Leakage Current ($\overline{EA}$)			TBD	μA	0 < V_{IN} < V_{CC} − 0.3V
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, and 3)			−650	μA	V_{IN} = 2V
RRST	RST Pulldown Resistor	40		225	KΩ	
CIO	Pin Capacitance			10	pF	@1 MHz, 25°C
I_{CC}	Power Supply Current:					(Note 3)
	Running at 12 MHz (Figure 5)		20	40	mA	
	Idle Mode at 12 MHz (Figure 5)		5	10	mA	
	Power Down Mode		15	100	μA	

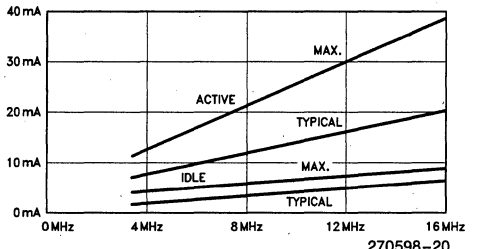
D.C. CHARACTERISTICS: (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V) (Continued)

NOTES:

1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In applications where capacitance loading exceeds 100 pFs, the noise pulse on the ALE signal may exceed 0.8V. In these cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an Address Latch with a Schmitt Trigger Strobe input.
2. Capacitive loading on Ports 0 and 2 cause the V_{OH} on ALE and $\overline{PSEN}$ to drop below the 0.9 V_{CC} specification when the address lines are stabilizing.
3. See Figures 6–9 for test conditions. Minimum V_{CC} for power down is 2V.
4. Care must be taken not to exceed the maximum allowable power dissipation.
5. Typicals are based on limited number of samples and are not guaranteed. The values listed are at room temperature and 5V.
6. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

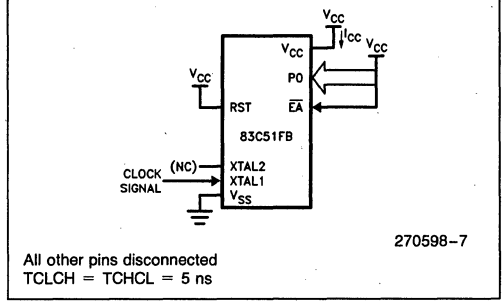
Maximum I_{OL} per port pin:	10 mA
Maximum I_{OL} per 8-bit port-	
Port 0:	26 mA
Ports 1, 2, and 3:	15 mA
Maximum total I_{OL} for all output pins:	71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.



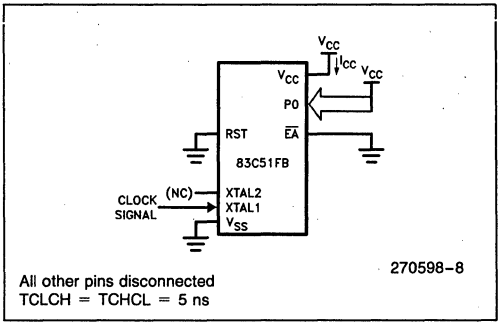
I_{CC} Max at other frequencies is given by:
Active Mode
I_{CC} MAX = 2.2 × FREQ + 3.1
Idle Mode
I_{CC} MAX = 0.49 × FREQ + 1.6
Where FREQ is in MHz, I_{CC}MAX is given in mA.

Figure 5. I_{CC} vs Frequency



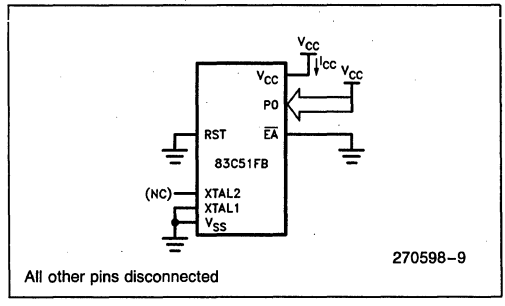
All other pins disconnected
TCLCH = TCHCL = 5 ns

Figure 6. I_{CC} Test Condition, Active Mode



All other pins disconnected
TCLCH = TCHCL = 5 ns

Figure 7. I_{CC} Test Condition Idle Mode



All other pins disconnected

Figure 8. I_{CC} Test Condition, Power Down Mode. V_{CC} = 2.0V to 5.5V.

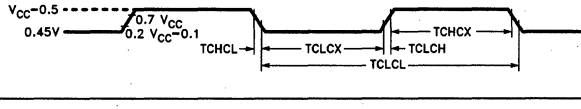


Figure 9. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address
C: Clock
D: Input Data
H: Logic level HIGH
I: Instruction (program memory contents)
L: Logic level LOW, or ALE
P: $\overline{PSEN}$
Q: Output Data
R: $\overline{RD}$ signal
T: Time
V: Valid
W: $\overline{WR}$ signal
X: No longer a valid logic level
Z: Float

For example,

TAVLL = Time from Address Valid to ALE Low
TLLPL = Time from ALE Low to $\overline{PSEN}$ Low

A.C. CHARACTERISTICS

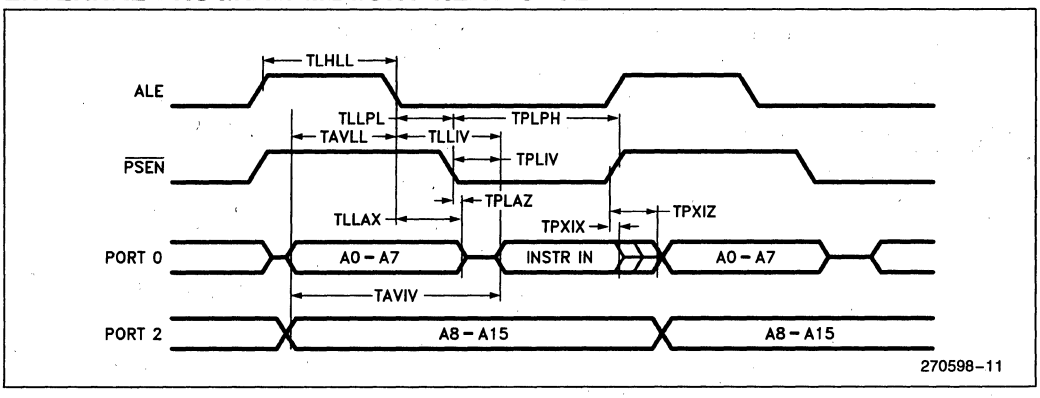
T_A = 0°C to +70°C, V_{CC} = 5V ±10%, V_{SS} = 0V, Load Capacitance for Port 0, ALE and $\overline{PSEN}$ = 100 pF, Load Capacitance for All Other Outputs = 80 pF

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

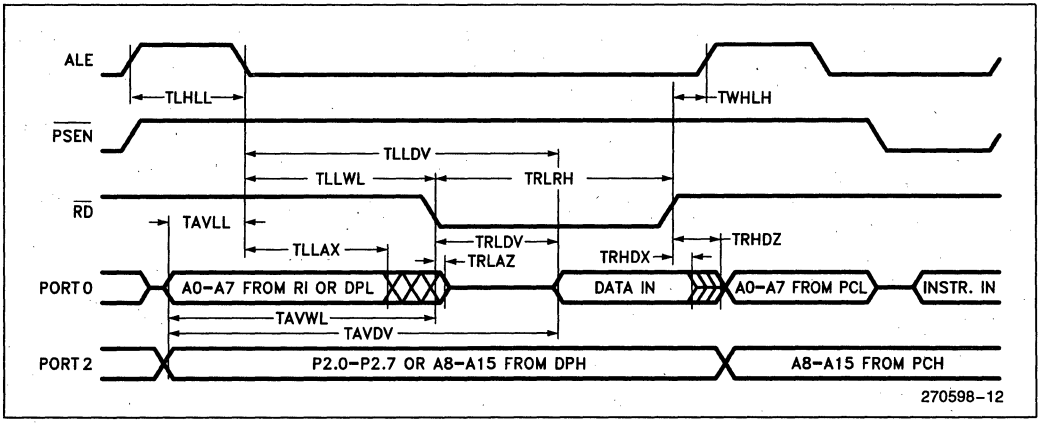
EXTERNAL PROGRAM MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 83C51FB 83C51FB-1 83C51FB-2			 3.5 3.5 0.5	 12 16 12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL−40		ns
TLLAX	Address Hold After ALE Low	53		TCLCL−30		ns
TLLIV	ALE Low to Valid Instruction In		234		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	53		TCLCL−30		ns
TPLPH	$\overline{PSEN}$ Pulse Width	205		3TCLCL−45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instruction In		145		3TCLCL−105	ns
TPXIX	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instruction Float After $\overline{PSEN}$		59		TCLCL−25	ns
TAVIV	Address to Valid Instruction In		312		5TCLCL−105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		107		2TCLCL−60	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid before $\overline{WR}$	33		TCLCL−50		ns
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL−50		ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL−150		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL−40	TCLCL+40	ns

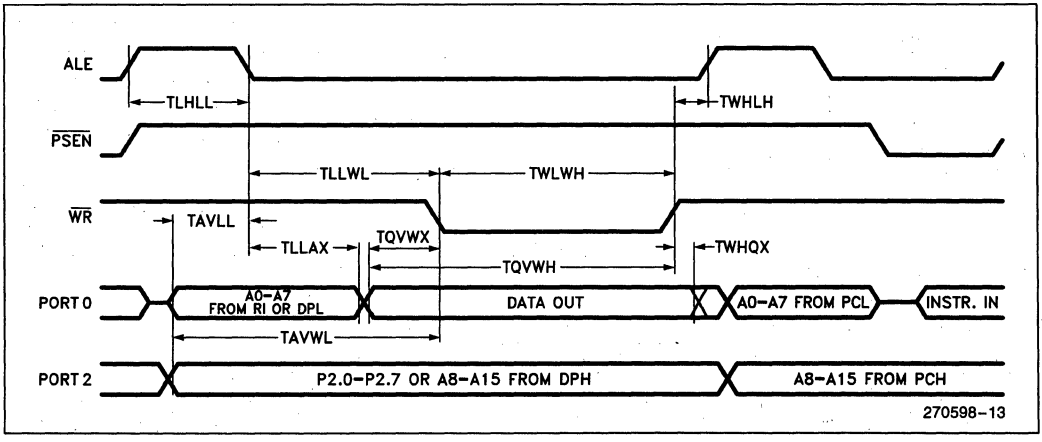
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY READ CYCLE

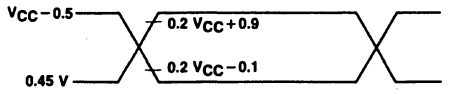


EXTERNAL DATA MEMORY WRITE CYCLE



A.C. TESTING INPUT

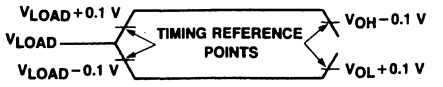
Input, Output Waveforms



270598-21

AC inputs during testing are driven at $V_{CC}-0.5V$ for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at V_{IH} min for a Logic "1" and V_{IL} max for a Logic "0".

Float Waveforms



270598-15

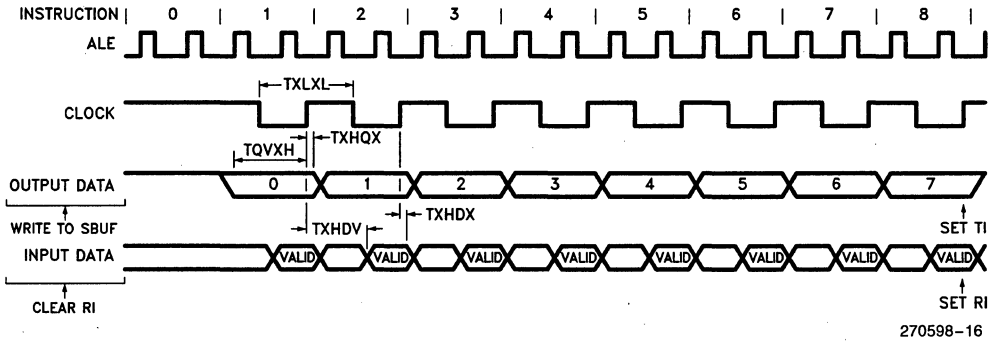
For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

SERIAL PORT TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold after Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns

SHIFT REGISTER MODE TIMING WAVEFORMS

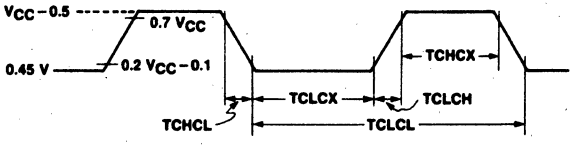


270598-16

EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency 83C51FB 83C51FB-1 83C51FB-2	 3.5 3.5 0.5	 12 16 12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

EXTERNAL CLOCK DRIVE WAVEFORM



270598-17

ROM CHARACTERISTICS

Table 2 shows the logic levels for verifying the code data and reading the signature bytes on the 83C51FB.

Table 2. ROM Modes

Mode	RST	$\overline{PSEN}$	ALE	$\overline{EA}$	P2.7	P2.6	P3.6	P3.7
Verify Code Data	1	0	1	1	0	0	1	1
Read Signature	1	0	1	1	0	0	0	0

NOTES:
"1" = Valid high for that pin
"0" = Valid low for that pin

Program Verification

If the Program Lock Bits have not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0–P2.5. The other pins should be held at the "Verify" levels indicated in Table 2. The contents of the addressed locations will come out on Port 0. External pullups are required on Port 0 for this operation.

If the Encryption Array in the ROM has been programmed, the data present at Port 0 will be Code Data XNOR Encryption Data. The user must know the Encryption Array contents to manually "unencrypt" the data during verify.

Figure 10 shows the setup for verifying the program memory.

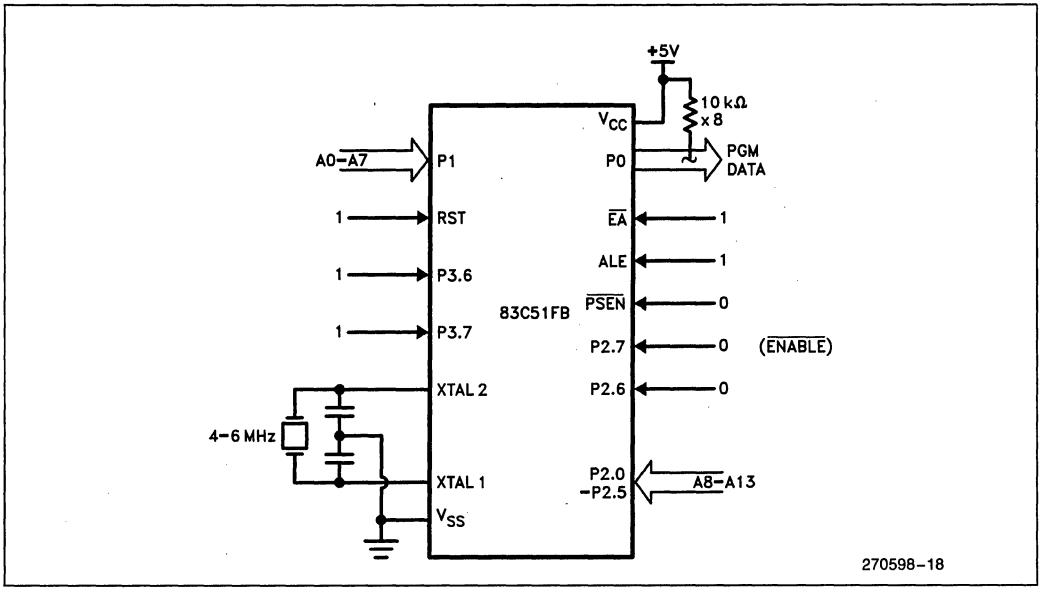


Figure 10. Verifying the ROM

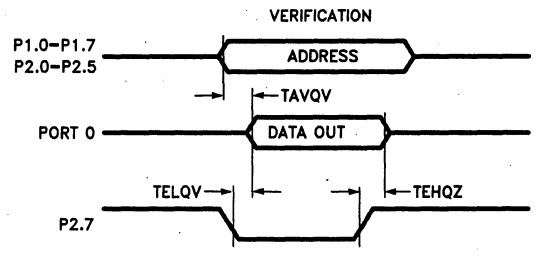
ROM VERIFICATION CHARACTERISTICS

T_A = 21°C to 27°C; V_{CC} = 5V ± 0.25V; V_{SS} = 0V

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVQV	Address to Data Valid		48TCLCL	
TELQV	ENABLE Low to Data Valid		48TCLCL	
TEHQZ	Data Float after ENABLE	0	48TCLCL	

ROM VERIFICATION WAVEFORMS



ROM Program Lock

The Program Lock system consists of one Program Lock bit and a 32 byte Encryption Array which are used to protect the program memory against software piracy.

Table 3 outlines the features of programming the Lock Bit.

Table 3. Program Lock Bit and their Features

Program Lock Bit LB1	Logic Enabled
U	No Program Lock features enabled. (Code Verify will still be encrypted by the Encryption Array.)
P	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset.

Encryption Array

Within the ROM array are 32 bytes of Encryption Array that are initially unprogrammed (all 1's). Every time that a byte is addressed during a verify, 5 address lines are used to select a byte of the Encryption Array. This byte is then exclusive-NOR'ed (XNOR) with the code byte, creating an Encrypted Verify byte. The algorithm, with the array in the unprogrammed state (all 1's), will return the code in it's original, unmodified form.

Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

(030H) = 89H indicates manufacture by Intel
(031H) = 5EH indicates 83C51FB

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -001 version of the 83C51FB data sheet:

1. Package table was added.
2. Note 4 was added to explain the maximum safe current spec.
3. Maximum I_{OL} per I/O pin was added to the ABSOLUTE MAXIMUM RATING.
4. Typical values for I_{CC} table were added.
5. Note 5 was added to explain the test conditions for typical values.
6. I_{CC} vs Frequency (Figure 5) was changed to resemble the 87C51FB data sheet.
7. Timing specs improved for:
   TLLAX changed from TCLCL-35 to TCLCL-30
   TLLPL changed from TCLCL-40 to TCLCL-30
   TRHDZ changed from TCLCL-70 to TCLCL-60
   TQVWH was added.
   TQVWX changed from TCLCL-60 to TCLCL-50
8. A.C. TESTING INPUT figure and specs were changed to match the 87C51FB.
9. Data sheet revision summary was added.

87C51FB
CHMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH 16K BYTES USER PROGRAMMABLE EPROM

87C51FB—3.5 MHz to 12 MHz, V_{CC} = 5V ± 10%
87C51FB-1—3.5 MHz to 16 MHz, V_{CC} = 5V ± 10%
87C51FB-2—0.5 MHz to 12 MHz, V_{CC} = 5V ± 10%

- High Performance CHMOS EPROM
- Three 16-Bit Timer/Counters
- Programmable Counter Array with:
  - — High Speed Output,
  - — Compare/Capture,
  - — Pulse Width Modulator,
  - — Watchdog Timer capabilities
- Up/Down Timer/Counter
- Two Level Program Lock System
- 16K On-Chip EPROM
- 256 Bytes of On-Chip Data RAM
- Quick Pulse Programming™ Algorithm
- Boolean Processor
- 32 Programmable I/O Lines
- 7 Interrupt Sources
- Programmable Serial Channel with:
  - — Framing Error Detection
  - — Automatic Address Recognition
- TTL and CMOS Compatible Logic Levels
- 64K External Program Memory Space
- 64K External Data Memory Space
- MCS®-51 Fully Compatible Instruction Set
- Power Saving Idle and Power Down Modes
- ONCE™ (On-Circuit Emulation) Mode

MEMORY ORGANIZATION

PROGRAM MEMORY: Up to 16K bytes of the program memory can reside in the on-chip EPROM. In addition the device can address up to 64K of program memory external to the chip.

DATA MEMORY: This microcontroller has a 256 x 8 on-chip RAM. In addition it can address up to 64K bytes of external data memory.

The Intel 87C51FB is a single-chip control-oriented microcontroller which is fabricated on Intel's reliable CHMOS III-E technology. Being a member of the MCS-51 family, the 87C51FB uses the same powerful instruction set, has the same architecture, and is pin for pin compatible with the existing MCS-51 family of products. The 87C51FB is an enhanced version of the 87C51. It's added features make it an even more powerful microcontroller for applications that require Pulse Width Modulation, High Speed I/O, and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multi-processor communications.

October 1988
Order Number: 270563-002

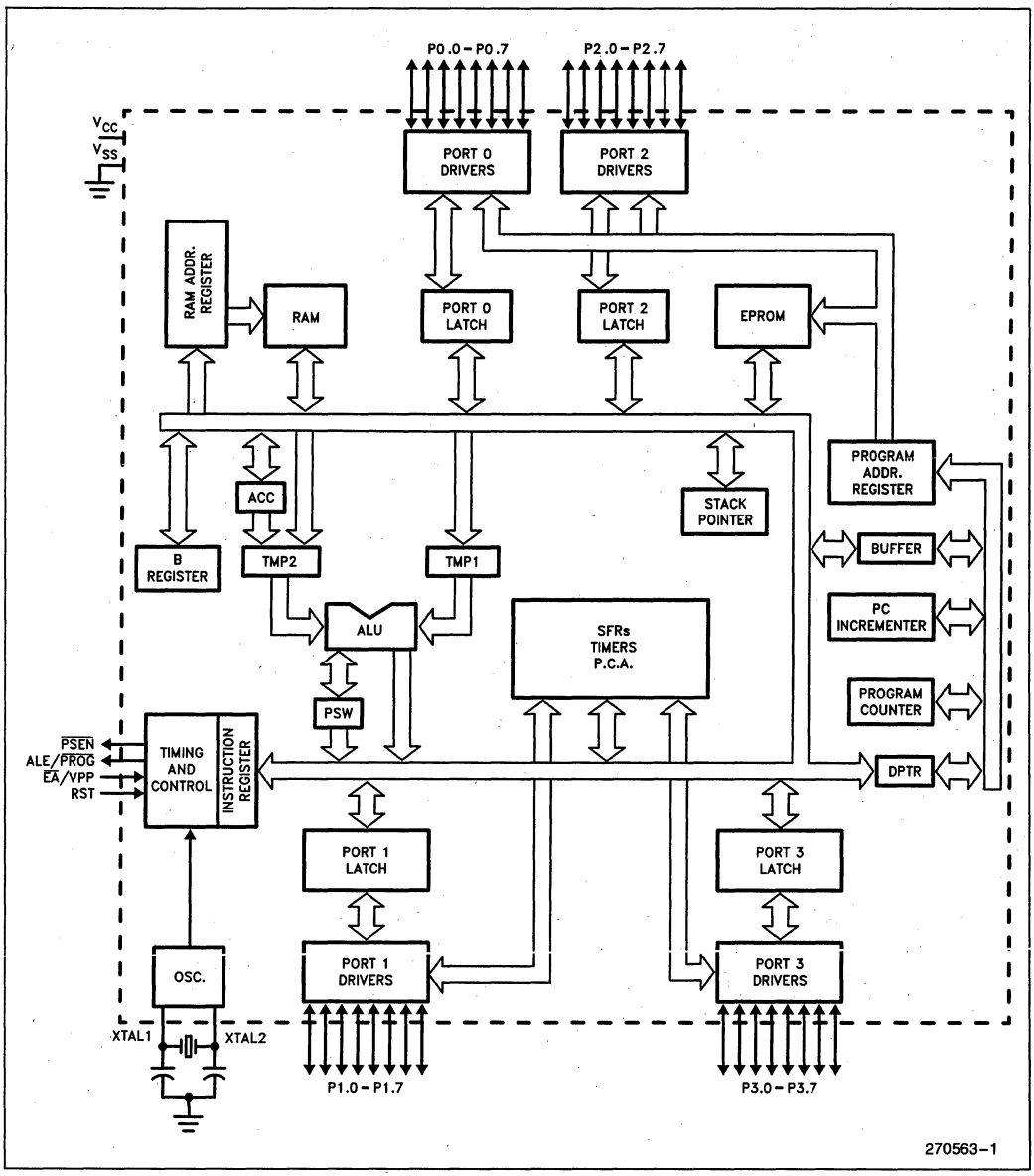


Figure 1. 87C51FB Block Diagram

PACKAGES

Part	Prefix	Package Type
87C51FB	P	40-Pin Plastic DIP
	D	40-Pin CERDIP
	N	44-Pin PLCC

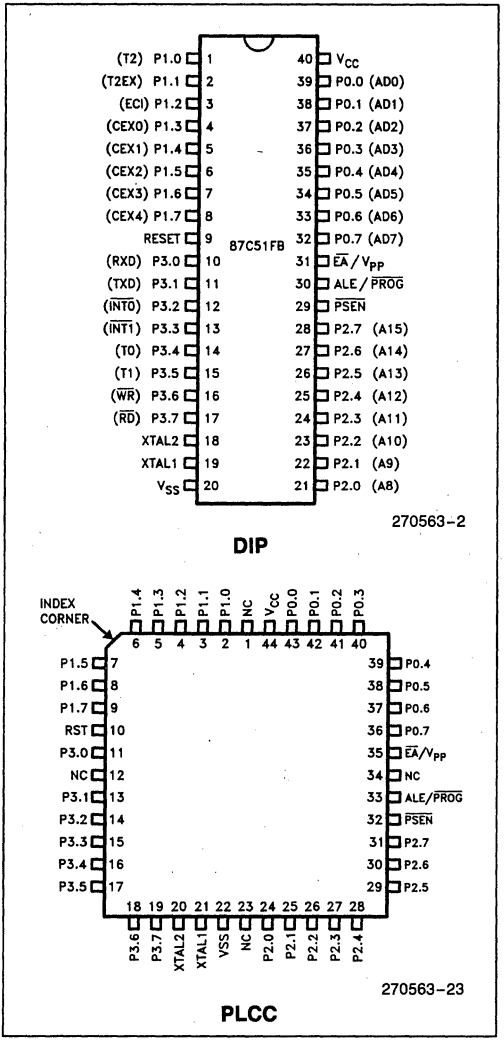


Figure 2. Pin Connections

PIN DESCRIPTIONS

V_{CC}: Supply voltage.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit, open drain, bidirectional I/O port. As an output port each pin can sink several LS TTL inputs. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting1's, and can source and sink several LS TTL inputs.

Port 0 also receives the code bytes during EPROM programming, and outputs the code bytes during program verification. External pullup resistors are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can drive LS TTL inputs. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

In addition, Port 1 serves the functions of the following special features of the 87C51FB:

Port Pin	Alternate Function
P1.0	T2 (External Count Input to Timer/ Counter 2)
P1.1	T2EX (Timer/Counter 2 Capture/ Reload Trigger and Direction Control)
P1.2	ECI (External Count Input to the PCA)
P1.3	CEX0 (External I/O for Compare/ Capture Module 0)
P1.4	CEX1 (External I/O for Compare/ Capture Module 1)
P1.5	CEX2 (External I/O for Compare/ Capture Module 2)
P1.6	CEX3 (External I/O for Compare/ Capture Module 3)
P1.7	CEX4 (External I/O for Compare/ Capture Module 4)

Port 1 receives the low-order address bytes during EPROM programming and verifying.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can drive LS TTL inputs. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Some Port 2 pins receive the high-order address bits during EPROM programming and program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can drive LS TTL inputs. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the 8051 Family, as listed below:

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST: Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits a power-on reset with only a capacitor connected to V_{CC}.

ALE: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin (ALE/$\overline{PROG}$) is also the program pulse input during EPROM programming for the 87C51FB.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

Throughout the remainder of this data sheet, ALE will refer to the signal coming out of the ALE/$\overline{PROG}$ pin, and the pin will be referred to as the ALE/$\overline{PROG}$ pin.

$\overline{PSEN}$: Program Store Enable is the read strobe to external Program Memory.

When the 87C51FB is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory.

$\overline{EA}$/V_{PP}: External Access enable. $\overline{EA}$ must be strapped to VSS in order to enable the device to fetch code from external Program Memory locations 0000H to 0FFFFH. Note, however, that if either of the Program Lock bits are programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the programming supply voltage (V_{PP}) during EPROM programming.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of a inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 floats, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the data sheet must be observed.

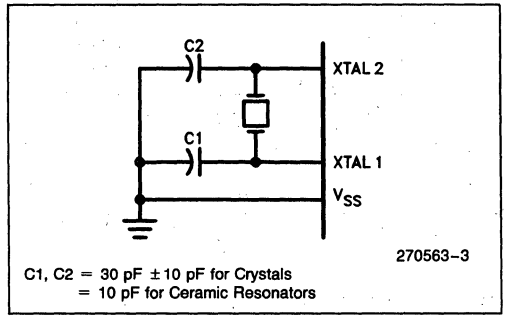


Figure 3. Oscillator Connections

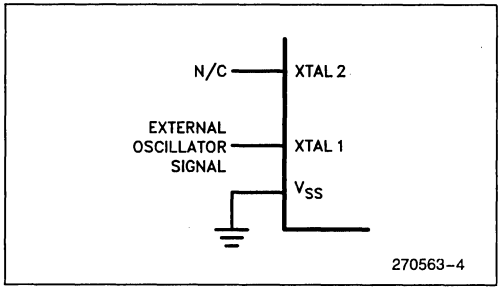


Figure 4. External Clock Drive Configuration

IDLE MODE

The user's software can invoke the Idle Mode. When the microcontroller is in this mode, power consumption is reduced. The Special Function Registers and the onboard RAM retain their values during Idle, but the processor stops executing instructions. Idle Mode will be exited if the chip is reset or if an enabled interrupt occurs. The PCA timer/counter can optionally be left running or paused during Idle Mode.

POWER DOWN MODE

To save even more power, a Power Down mode can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is terminated.

On the 87C51FB either a hardware reset or an external interrupt can cause an exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and on-chip RAM to retain their values.

To properly terminate Power down the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

DESIGN CONSIDERATION

- Ambient light is known to affect the internal RAM contents during operation. If the 87C51FB application requires the part to be run under ambient lighting, an opaque label should be placed over the window to exclude light.
- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

ONCE™ MODE

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 87C51FB without the 87C51FB having to be removed from the circuit. The ONCE Mode is invoked by:

1) Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2) Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the 87C51FB is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

Table 1. Status of the External Pins during Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

NOTE:
For more detailed information on the reduced power modes refer to current Embedded Controller Handbook, and Application Note AP-252, "Designing with the 80C51BH."

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on EA/V_{PP} Pin to V_{SS} .....0V to +13.0V
Voltage on Any Other Pin to V_{SS} ..−0.5V to +6.5V
Maximum I_{OL} Per I/O Pin .....15 mA
Power Dissipation.....1.5W
(based on PACKAGE heat transfer limitations, not device power consumption)

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

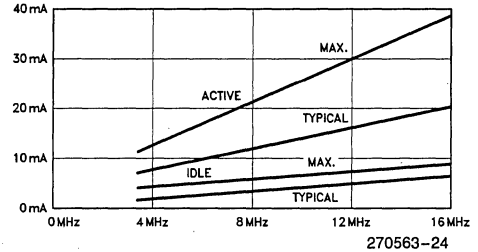
ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

D.C. CHARACTERISTICS: (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V)

Symbol	Parameter	Min	Typ (Note 5)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5		0.2 V_{CC}−0.1	V	
V_{IL1}	Input Low Voltage $\overline{EA}$	0		0.2 V_{CC}−0.3	V	
V_{IH}	Input High Voltage (Except XTAL1, RST, $\overline{EA}$)	0.2 V_{CC}+0.9		V_{CC}+0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}		V_{CC}+0.5	V	
V_{OL}	Output Low Voltage **(Note 6)** (Ports 1, 2, and 3)			0.3	V	I_{OL} = 100 μA (Note 1)
				0.45	V	I_{OL} = 1.6 mA (Note 1)
				1.0	V	I_{OL} = 3.5 mA (Notes 1, 4)
V_{OL1}	Output Low Voltage **(Note 6)** (Port 0, ALE, $\overline{PSEN}$)			0.3	V	I_{OL} = 200 μA (Note 1)
				0.45	V	I_{OL} = 3.2 mA (Note 1)
				1.0	V	I_{OL} = 7.0 mA (Note 1, 4)
V_{OH}	Output High Voltage (Ports 1, 2, and 3)	V_{CC}−0.3			V	I_{OH} = −10 μA
		V_{CC}−0.7			V	I_{OH} = −30 μA
		V_{CC}−1.5			V	I_{OH} = −60 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)	V_{CC}−0.3			V	I_{OH} = −200 μA
		V_{CC}−0.7			V	I_{OH} = −3.2 mA
		V_{CC}−1.5			V	I_{OH} = −7.0 mA (Note 4)
I_{IL}	Logical 0 Input Current (Ports 1, 2, and 3)			−50	μA	V_{IN} = 0.45V
I_{LI}	Input leakage Current (Port 0)			±10	μA	0 < V_{IN} < V_{CC}−0.3V
I_{LI1}	Input Leakage Current ($\overline{EA}$)			TBD	μA	0 < V_{IN} < V_{CC}−0.3V
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, and 3)			−650	μA	V_{IN} = 2V
RRST	RST Pulldown Resistor	40		225	KΩ	
CIO	Pin Capacitance			10	pF	@1 MHz, 25°C
I_{CC}	Power Supply Current:					(Note 3)
	Running at 12 MHz (Figure 5)		20	40	mA	
	Idle Mode at 12 MHz (Figure 5)		5	10	mA	
	Power Down Mode		15	100	μA	

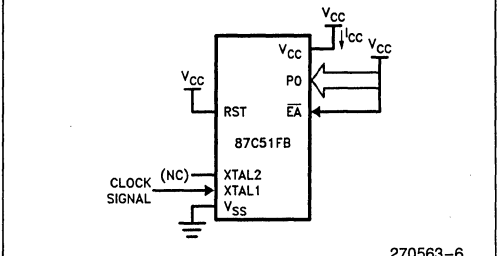
NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operations. In applications where capacitance loading exceeds 100 pFs, the noise pulse on the ALE signal may exceed 0.8V. In these cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an Address Latch with a Schmitt Trigger Strobe input.
2. Capacitive loading on Ports 0 and 2 cause the V_{OH} on ALE and $\overline{PSEN}$ to drop below the 0.9 V_{CC} specification when the address lines are stabilizing.
3. See Figures 6–9 for test conditions. Minimum V_{CC} for Power Down is 2V.
4. Care must be taken not to exceed the maximum allowable power dissipation.
5. Typicals are based on limited number of samples and are not guaranteed. The values listed are at room temperature and 5V.
6. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10mA
   Maximum I_{OL} per 8-bit port—
   Port 0: 26 mA
   Ports 1, 2 and 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.



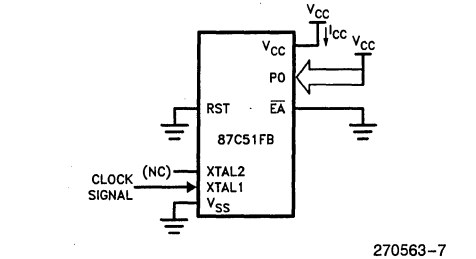
I_{CC} Max at other frequencies is given by:
Active Mode
I_{CC} MAX = 2.2 × FREQ + 3.1
Idle Mode
I_{CC} MAX = 0.49 × FREQ + 1.6
Where FREQ is in MHz, I_{CC} MAX is given in mA.

Figure 5. I_{CC} vs Frequency



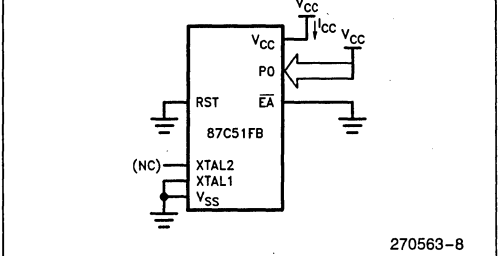
All other pins disconnected
TCLCH = TCHCL = 5 ns

Figure 6. I_{CC} Test Condition, Active Mode



All other pins disconnected
TCLCH = TCHCL = 5 ns

Figure 7. I_{CC} Test Condition Idle Mode



All other pins disconnected

Figure 8. I_{CC} Test Condition, Power Down Mode. V_{CC} = 2.0V to 5.5V.

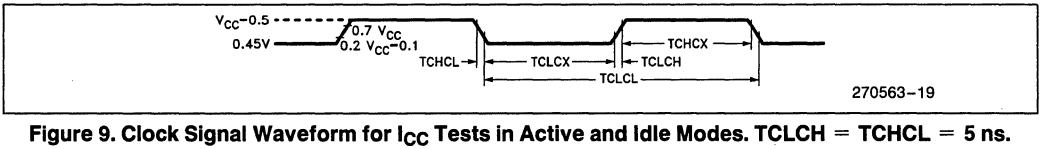


Figure 9. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address
C: Clock
D: Input Data
H: Logic level HIGH
I: Instruction (program memory contents)
L: Logic level LOW, or ALE
P: $\overline{PSEN}$
Q: Output Data
R: $\overline{RD}$ signal
T: Time
V: Valid
W: $\overline{WR}$ signal
X: No longer a valid logic level
Z: Float

For example,

TAVLL = Time from Address Valid to ALE Low
TLLPL = Time from ALE Low to $\overline{PSEN}$ Low

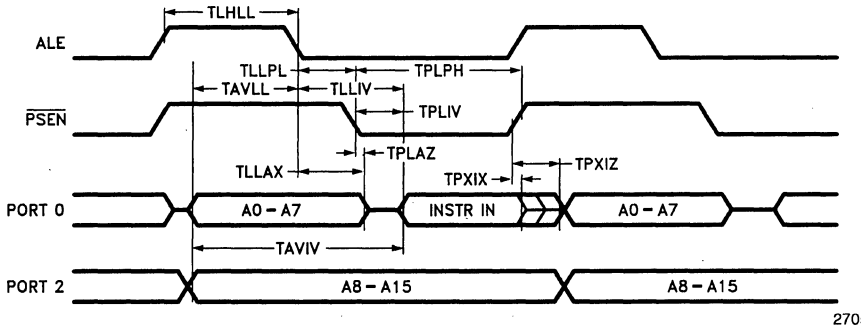
A.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = 5V ±10%, V_{SS} = 0V, Load Capacitance for Port 0, ALE/$\overline{PROG}$ and $\overline{PSEN}$ = 100 pF, Load Capacitance for All Other Outputs = 80 pF)

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

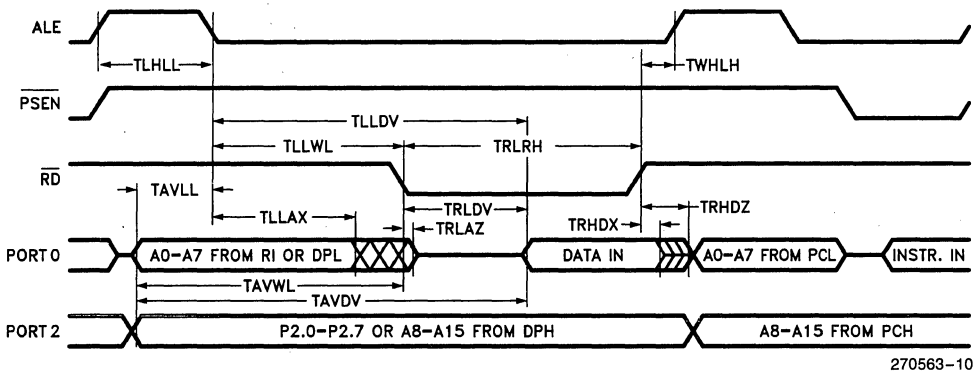
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency			0.5	16	MHz
TLHLL	ALE Pulse Width	127		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL−40		ns
TLLAX	Address Hold After ALE Low	53		TCLCL−30		ns
TLLIV	ALE Low to Valid Instruction In		234		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	53		TCLCL−30		ns
TPLPH	$\overline{PSEN}$ Pulse Width	205		3TCLCL−45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instruction In		145		3TCLCL−105	ns
TPXIX	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instruction Float After $\overline{PSEN}$		59		TCLCL−25	ns
TAVIV	Address to Valid Instruction In		312		5TCLCL−105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		252		5TCLCL−165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		107		2TCLCL−60	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL−150	ns
TAVDV	Address to Valid Data In		585		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address Valid to $\overline{WR}$ Low	203		4TCLCL−130		ns
TQVWX	Data Valid before $\overline{WR}$	33		TCLCL−50		ns
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL−50		ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL−150		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL−40	TCLCL+40	ns

EXTERNAL PROGRAM MEMORY READ CYCLE



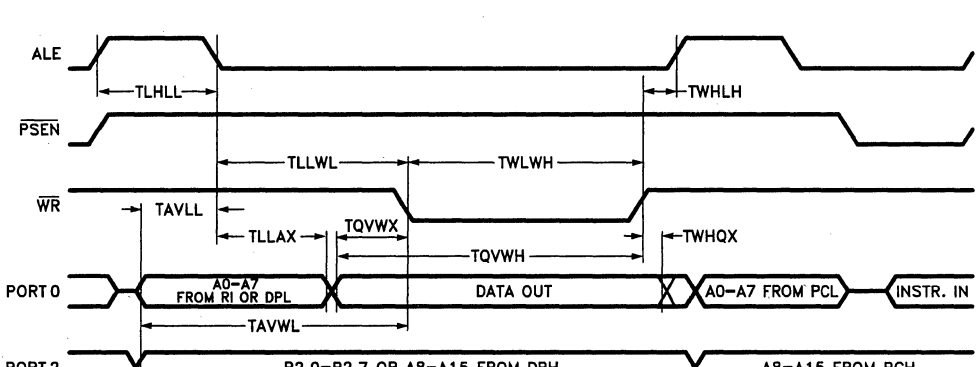
270563-9

EXTERNAL DATA MEMORY READ CYCLE



270563-10

EXTERNAL DATA MEMORY WRITE CYCLE



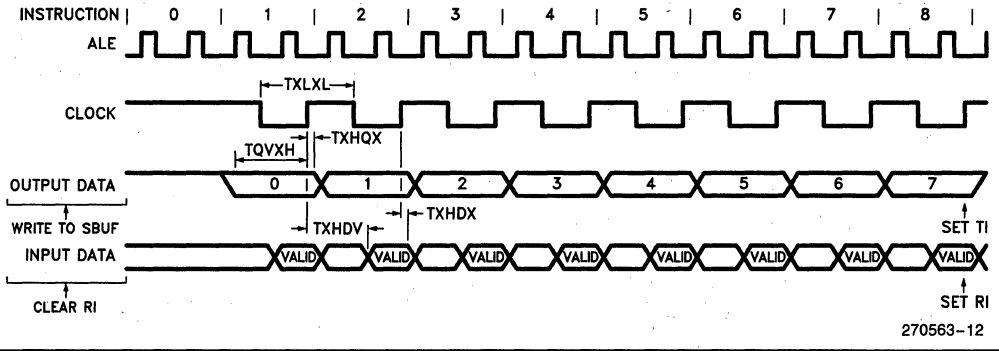
270563-11

SERIAL PORT TIMING - SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL−133		ns
TXHQX	Output Data Hold after Clock Rising Edge	50		2TCLCL−117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL−133	ns

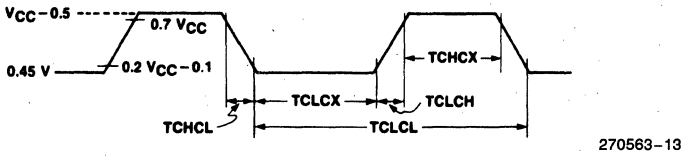
SHIFT REGISTER MODE TIMING WAVEFORMS



EXTERNAL CLOCK DRIVE

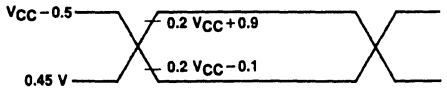
Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency 87C51FB 87C51FB-1 87C51FB-2	 3.5 3.5 0.5	 12 16 12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

EXTERNAL CLOCK DRIVE WAVEFORM



A.C. TESTING INPUT

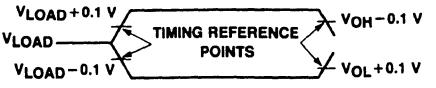
Input, Output Waveforms



270563-14

AC Inputs during testing are driven at $V_{CC}-0.5V$ for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at V_{IH} min for a Logic "1" and V_{IL} max for a Logic "0".

Float Waveforms



270563-15

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

EPROM CHARACTERISTICS

Table 2 shows the logic levels for programming the Program Memory, the Encryption Table, and the Lock Bits and for reading the signature bytes.

Table 2. EPROM Programming Modes

Mode	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.7	P2.6	P3.6	P3.7
Program Code Data	1	0	0*	V_{PP}	1	0	1	1
Verify Code Data	1	0	1	1	0	0	1	1
Program Encryption Table Use Addresses 0-1FH	1	0	0*	V_{PP}	1	0	0	1
Program Lock x=1	1	0	0*	V_{PP}	1	1	1	1
Bits (LBx) x=2	1	0	0*	V_{PP}	1	1	0	0
Read Signature	1	0	1	1	0	0	0	0

NOTES:
"1" = Valid high for that pin
"0" = Valid low for that pin
"VPP" = +12.75V ±0.25V
* ALE/$\overline{PROG}$ is pulsed low for 100 μs for programming. (Quick-Pulse Programming™)

PROGRAMMING THE EPROM

To be programmed, the part must be running with a 4 to 6 MHz oscillator. (The reason the oscillator needs to be running is that the internal bus is being used to transfer address and program data to appropriate internal EPROM locations.) The address of an EPROM location to be programmed is applied to Port 1 and pins P2.0 - P2.5 of Port 2, while the code byte to be programmed into that location is applied to Port 0. The other Port 2 and 3 pins, RST, $\overline{PSEN}$, and $\overline{EA}/V_{PP}$ should be held at the "Program" levels indicated in Table 2. ALE/$\overline{PROG}$ is pulsed low to program the code byte into the addressed EPROM location. The setup is shown in Figure 10.

Normally $\overline{EA}/V_{PP}$ is held at logic high until just before ALE/$\overline{PROG}$ is to be pulsed. Then $\overline{EA}/V_{PP}$ is raised to V_{PP}, ALE/$\overline{PROG}$ is pulsed low, and then $\overline{EA}/V_{PP}$ is returned to a valid high voltage. The voltage on the $\overline{EA}/V_{PP}$ pin must be at the valid $\overline{EA}/V_{PP}$ high level before a verify is attempted. Waveforms and detailed timing specifications are shown in later sections of this data sheet.

Note that the $\overline{EA}/V_{PP}$ pin must not be allowed to go above the maximum specified V_{PP} level for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The V_{PP} source should be well regulated and free of glitches.

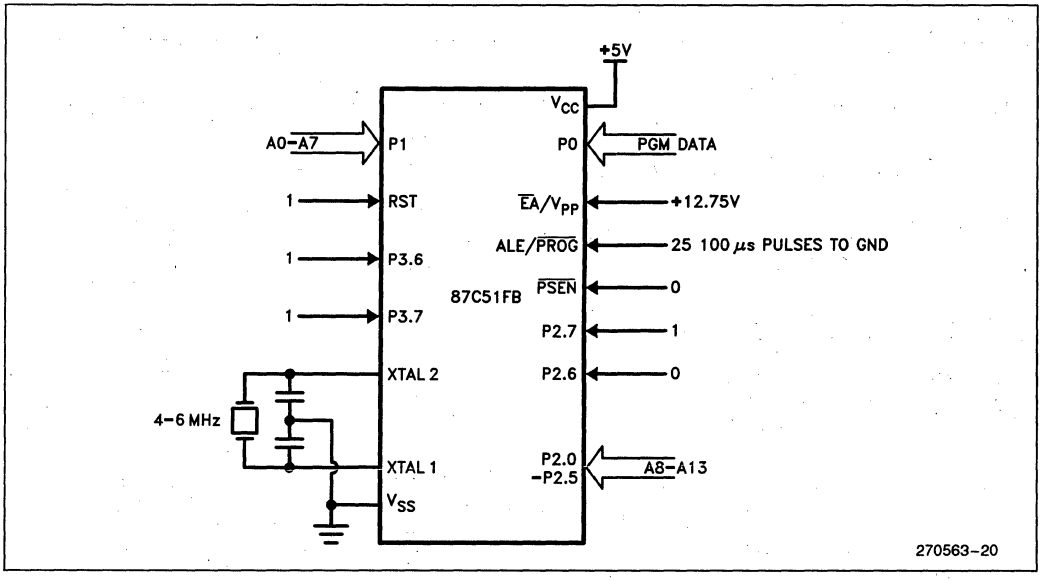


Figure 10. Programming the EPROM

Quick-Pulse Programming™ Algorithm

The 87C51FB can be programmed using the Quick-Pulse Programming™ Algorithm for microcontrollers. The features of the new programming method are a lower V_{PP} (12.75V as compared to 21V) and a shorter programming pulse. It is possible to program the entire 16K Bytes of EPROM memory in less than 50 seconds with this algorithm!

To program the part using the new algorithm, V_{PP} must be 12.75V ±0.25V. ALE/$\overline{PROG}$ is pulsed low for 100 μs, 25 times as shown in Figure 11. Then, the byte just programmed may be verified. After programming, the entire array should be verified. The Program Lock features are programmed using the same method, but with the setup as shown in Table 2. The only difference in programming Program Lock features is that the Program Lock features cannot be directly verified. Instead, verification of programming is by observing that their features are enabled.

Program Verification

If the Program Lock Bits have not been programmed, the on-chip Program Memory can be read out for verification purposes, if desired, either during or after the programming operation. The address of the Program Memory location to be read is applied to Port 1 and pins P2.0 - P2.5. The other pins should be held at the "Verify" levels indicated in Table 2. The contents of the addressed locations will come out on Port 0. External pullups are required on Port 0 for this operation.

If the Encryption Array in the EPROM has been programmed, the data present at Port 0 will be Code Data XNOR Encryption Data. The user must know the Encryption Array contents to manually "unencrypt" the data during verify.

The setup, which is shown in Figure 12, is the same as for programming the EPROM except that pin P2.7 is held at a logic low, or may be used as an active-low read strobe.

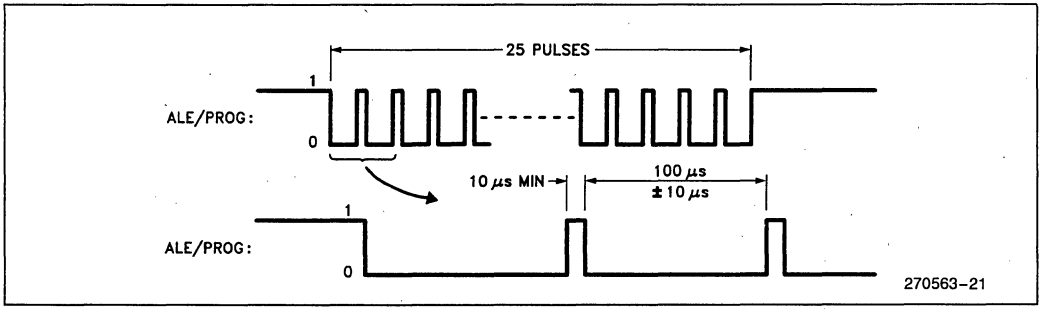


Figure 11. $\overline{\text{PROG}}$ Waveforms

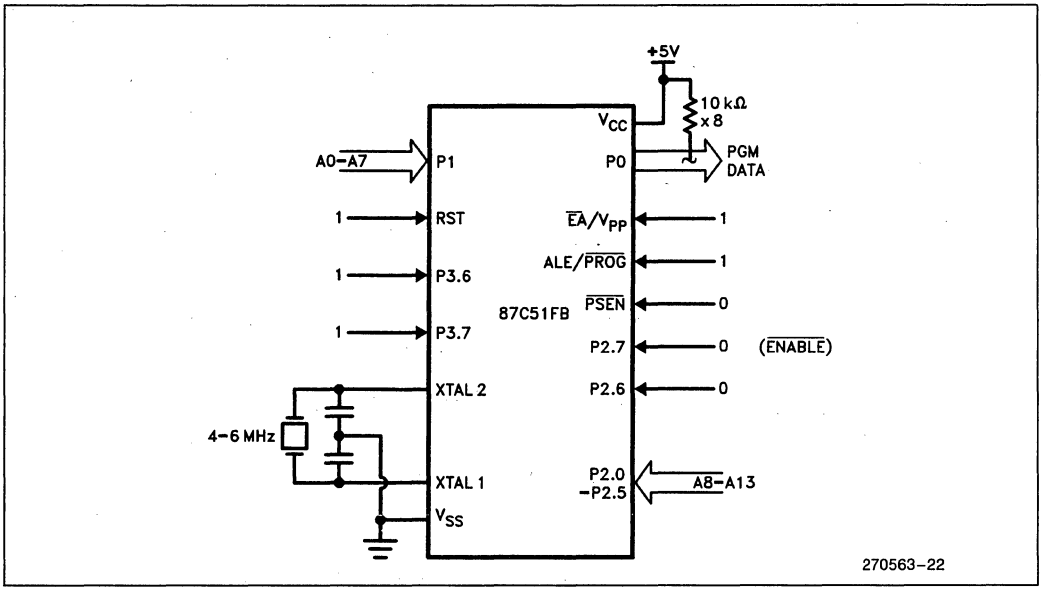


Figure 12. Verifying the EPROM

EPROM Program Lock

The two-level Program Lock system consists of two Program Lock bits and a 32 byte Encryption Array which are used to protect the program memory against software piracy.

Encryption Array

Within the EPROM array are 32 bytes of Encryption Array that are initially unprogrammed (all 1's). Every time that a byte is addressed during a verify, 5 address lines are used to select a byte of the Encryption Array. This byte is then exclusive-NOR'ed (XNOR) with the code byte, creating an Encrypted Verify byte. The algorithm, with the array in the unprogrammed state (all 1's), will return the code in it's original, unmodified form.

Program Lock Bits

Also included in the EPROM Program Lock scheme are two Program Lock Bits which are programmed as shown in Table 2.

Table 3 outlines the features of programming the Lock Bits.

Erasing the EPROM also erases the Encryption Array and the Program Lock Bits, returning the part to full functionality.

Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

(030H) = 89H indicates manufacture by Intel
(031H) = 5FH indicates 87C51FB

Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelength shorter than approximately 4,000 Angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room-level fluorescent lighting) could cause inadvertent erasure. If an application subjects the device to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 Angstroms) to an integrated dose of at least 15 W-sec/cm^2. Exposing the EPROM to an ultraviolet lamp of 12,000 μW/cm^2 rating for 30 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves all the EPROM Cells in a 1's state.

Table 3. Program Lock Bits and their Features

Program Lock Bits LB1	LB2	Logic Enabled
U	U	No Program Lock features enabled. (Code Verify will still be encrypted by the Encryption Array.)
P	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the EPROM is disabled.
P	P	Same as above, but Verify is also disabled
U	P	Reserved for Future Definition

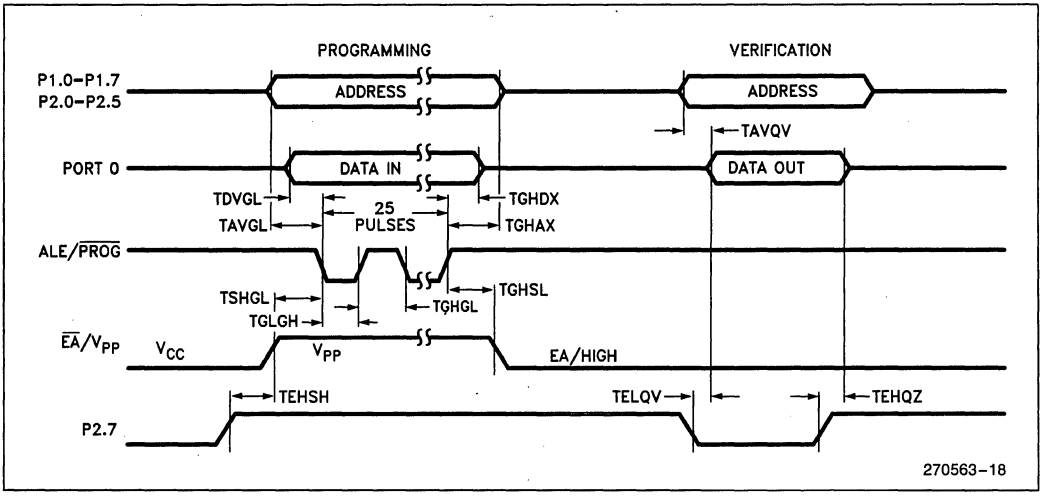
EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

(T_A = 21°C to 27°C; V_{CC} = 5V ± 0.25V; V_{SS} = 0V)

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	12.5	13.0	V
I_{PP}	Programming Supply Current		50	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$ Low	48TCLCL		
TGHAX	Address Hold after $\overline{PROG}$	48TCLCL		
TDVGL	Data Setup to $\overline{PROG}$ Low	48TCLCL		
TGHDX	Data Hold after $\overline{PROG}$	48TCLCL		
TEHSH	P2.7 (ENABLE) High to V_{PP}	48TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
TGHSL	V_{PP} Hold after $\overline{PROG}$	10		µs
TGLGH	$\overline{PROG}$ Width	90	110	µs
TAVQV	Address to Data Valid		48TCLCL	
TELQV	ENABLE Low to Data Valid		48TCLCL	
TEHQZ	Data Float after ENABLE	0	48TCLCL	
TGHGL	$\overline{PROG}$ High to $\overline{PROG}$ Low	10		µs

EPROM PROGRAMMING AND VERIFICATION WAVEFORMS

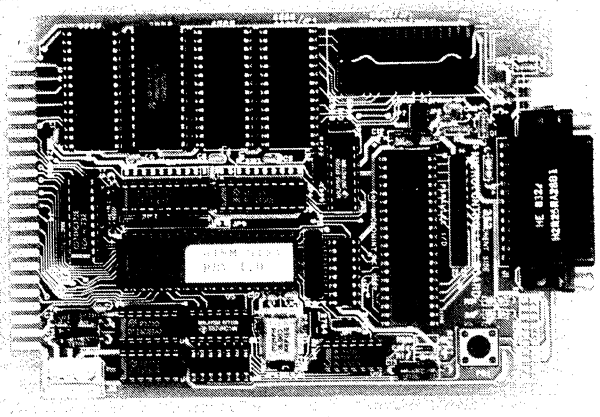


DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -001 version of the 87C51FB data sheet:

1. Title changed to include -1 and -2 version of the device.
2. PLCC pin connection diagram was added.
3. Package table was added.
4. Exit from power down mode was clarified.
5. Maximum I_{OL} per I/O pin was added to the ABSOLUTE MAXIMUM RATING.
6. Note 6 was added to explain the maximum safe current spec.
7. Typical values for ICC table were added.
8. Note 5 was added to explain the test conditions for typical values.
9. Timing specs improved for:
   - TLLAX changed from TCLCL-35 to TCLCL-30
   - TLLPL changed from TCLCL-40 to TCLCL-30
   - TRHDZ changed from TCLCL-70 to TCLCL-60
   - TQVWX changed from TCLCL-60 to TCLCL-50
   - TQVWH was added.
10. Data sheet revision summary was added.

EV80C51FA EVALUATION BOARD



LOW COST CODE EVALUATION TOOL

Intel's EV80C51FA evaluation board provides a hardware environment for code execution and software debugging at a relatively low cost. The board features the 80C51FA single chip, CHMOS*, 8-bit microcontroller, the newest member of the industry standard 8051 family. The board allows the user to take full advantage of the power of the 8051. The EV80C51FA provides up to 16 MHz execution of a user's code. Plus, its memory (ROMsim) can be reconfigured to match the user's planned memory system, allowing for exact analysis of code execution speeds in a particular application.

Popular features such as single-step program execution and sixteen software breakpoints are standard on the EV80C51FA. Intel provides a complete code development environment using assembly language (ASM-51) as well as Intel's high-level language PL/M-51 to accelerate development schedules.

The evaluation board is hosted on an IBM PC** or BIOS-compatible clone, already a standard development solution in most of today's engineering environments. The source code for the on-board monitor (written in ASM-51) is public domain. The program is about 2K bytes and can be easily modified to be included in the user's target hardware. In this way, the provided PC host software can be used throughout the development phase.

EV80C51FA FEATURES

- Up to 16 MHz Execution Speed
- 8K Bytes of ROMsim
- Totally CMOS, low power board
- Concurrent Interrogation of Memory and Registers
- Sixteen Software Breakpoints
- Program Step Mode
- High-Level Language Support
- RS-232-C Communication Link

FULL SPEED EXECUTION

The EV80C51FA executes the user's code from on-board ROMsim at up to 16 MHz. By changing crystals on the 80C51FA any slower execution speed can be evaluated. However, the board's host interface timing is affected by this crystal change, and therefore the monitor code requires minor modifications.

8K BYTES OF ROMSIM

The board comes with 8K bytes of SRAM to be used as ROMsim for the user's code and as data memory if needed.

intel

*CHMOS is a patented Intel process.
**IBM PC, XT, AT and DOS are registered trademarks of International Business Machines Corporation.

MARCH 1988
Order Number: 270592-001

TOTALLY CMOS BOARD

The EV80C51FA board is built totally with CMOS components. Its power consumption is therefore very low, requiring 5 volts at only 120 mA. The board also requires +/- 12 volts at 0.1 mA.

CONCURRENT INTEROGATION OF MEMORY AND REGISTERS

The monitor for the EV80C51FA allows the user to read and modify internal registers and external memory while the user's code is running in the board.

SIXTEEN SOFTWARE BREAKPOINTS

There are sixteen breakpoints available which automatically substitute an LCALL instruction for a user's instruction at the breakpoint location. The substitution occurs when execution is started. If the code is halted or a breakpoint is reached, the user's code is restored into the ROMsim.

PROGRAM STEP MODE

The stepping mode redirects the external interrupt 0 vector for use by the monitor. All other interrupts are available to the user, and will function as normal. External interrupt 0 is returned to the user after stepping.

HIGH LEVEL LANGUAGE SUPPORT

The host software for the EV80C51FA board is able to load absolute object code generated by ASM-51, PL/M-51 or RL-51, which are available from Intel.

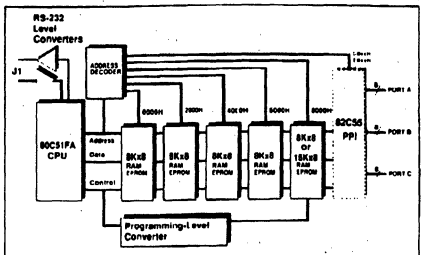
RS-232-C COMMUNICATION LINK

The EV80C51FA communicates with the host using the on chip UART of the 80C51FA. Timer 2 is used as the baud-rate generator to reach 9600 baud with a 16 MHz crystal. Timer 1 could be used as the baud-rate generator if a different crystal frequency is selected. This is outlined in the supplied User's Manual.

PERSONAL COMPUTER REQUIREMENTS

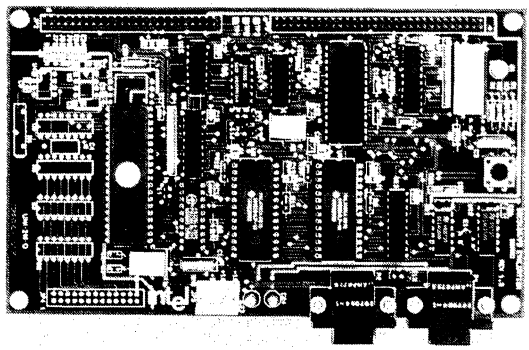
The EV80C51FA Evaluation Board is hosted on an IBM PC**, XT**, AT** or BIOS-compatible clone. The PC must meet the following minimum requirements:

- 512K Bytes of Memory
- One 360K Byte floppy Disk Drive
- PC DOS** 3.1 or Later
- A Serial Port (COM1 or COM2) at 9600 Baud
- ASM-51 or PL/M-51
- A text editor such as AEDIT



Block Diagram of the EV80C51FA Board

EV80C51FB EVALUATION BOARD



LOW COST CODE EVALUATION TOOL

Intel's EV80C51FB evaluation board provides a hardware environment for code execution and software debugging at a relatively low cost. The board features the 80C51FB or 80C51FA, single chip, CHMOS*, 8-bit microcontrollers, the newest members of the industry standard 8051 family. The board allows the user to take full advantage of the power of the 8051. The EV80C51FB provides up to 16 MHz execution of a user's code. Plus, its memory (ROMsim) can be reconfigured to match the user's planned memory system, allowing for exact analysis of code execution speeds in a particular application.

Popular features such as a single line assembler/disassembler, single-step program execution and sixteen software breakpoints are standard on the EV80C51FB. Intel provides a complete code development environment using assembly language (ASM-51) as well as Intel's high-level language PL/M-51 to accelerate development schedules.

The evaluation board is hosted on an IBM PC** or BIOS-compatible clone, already a standard development solution in most of today's engineering environments. The source code for the on-board monitor (written in ASM-51) is public domain. The program is about 3K bytes and can be easily modified to be included in the user's target hardware. In this way, the provided PC host software can be used throughout the development phase.

EV80C51FB FEATURES

- **Up to 16 MHz Execution Speed**
- **16K Bytes of ROMsim**
- **Flexible Chip-Select Controller**
- **Totally CMOS, low power board**
- **Concurrent Interrogation of Memory and Registers**
- **Sixteen Software Breakpoints**
- **Program Step Mode**
- **High-Level Language Support**
- **Single Line Assembler/Disassembler**
- **RS-232-C Communication Link**

FULL SPEED EXECUTION

The EV80C51FB executes the user's code from on-board ROMsim at up to 16 MHz. By changing crystals on the 80C51FB any slower execution speed can be evaluated. The board's host interface timing is not affected by this crystal change.

16K BYTES OF ROMSIM

The board comes with 16K bytes of SRAM to be used as ROMsim for the user's code and as data memory if needed.

intel

*CHMOS is a patented Intel process.
**IBM PC, XT, AT and DOS are registered trademarks of International Business Machines Corporation.

AUGUST 1988
ORDER NUMBER: 270618-001

FLEXIBLE MEMORY DECODING

By changing the Programable Logic Device (PLD) on the board, the memory on the board can be made to look like the memory system planned for the user's hardware application. The PLD controls the chip-select inputs on the board with 64 byte boundaries of resolution.

TOTALLY CMOS BOARD

The EV80C51FB board is built totally with CMOS components. Its power consumption is therefore very low, requiring 5 volts at only 225 mA. If the on board LED's are disabled, the current drops to only 80 mA. The board also requires +/- 12 volts at 10 mA.

CONCURRENT INTEROGATION OF MEMORY AND REGISTERS

The monitor for the EV80C51FB allows the user to read and modify internal registers and external memory while the user's code is running in the board.

SIXTEEN SOFTWARE BREAKPOINTS

There are sixteen breakpoints available which automatically substitute an LCALL instruction for a user's instruction at the breakpoint location. The substitution occurs when execution is started. If the code is halted or a breakpoint is reached, the user's code is restored into the ROMsim.

PROGRAM STEP MODE

The stepping mode redirects the external interrupt 0 vector for use by the monitor. All other interrupts are available to the user, and will function as normal. External interrupt 0 is returned to the user after stepping.

HIGH LEVEL LANGUAGE SUPPORT

The host software for the EV80C51FB board is able to load absolute object code generated by ASM-51, PL/M-51 or RL-51, which are available from Intel.

SINGLE LINE ASSEMBLER/DISASSEMBLER

The host has a Single Line Assembler, and a Disassembler, to simplify modification and examination of code loaded on the board.

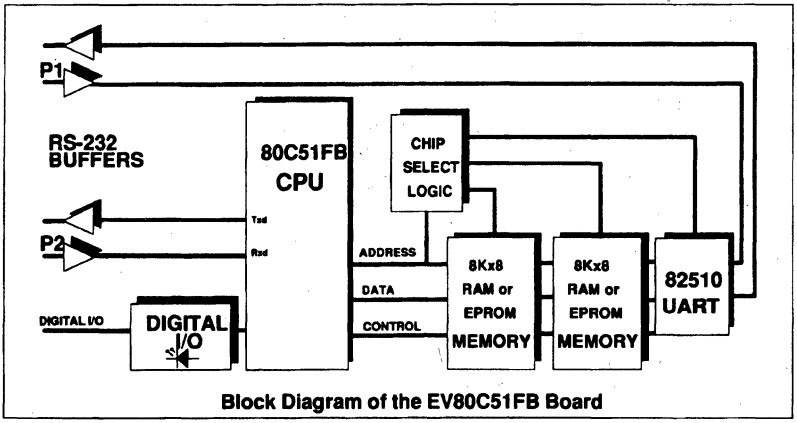
RS-232-C COMMUNICATION LINK

The EV80C51FB communicates with the host using an Intel 82510 UART provided on board. This frees the on-chip UART of the 80C51FB or 80C51FA for the user's application.

PERSONAL COMPUTER REQUIREMENTS

The EV80C51FB Evaluation Board is hosted on an IBM PC**, XT**, AT** or BIOS compatible clone. The PC must meet the following minimum requirements:

- 512K Bytes of Memory
- One 360K Byte floppy Disk Drive
- PC DOS** 3.1 or Later
- A Serial Port (COM1 or COM2) at 9600 Baud
- ASM-51 or PL/M-51
- A text editor such as AEDIT



Block Diagram of the EV80C51FB Board

Printed in USA/TI293/888/10K/IL SM
Embedded Controllers

8XC51GA

10

HARDWARE DESCRIPTION OF THE 8XC51GA

INTRODUCTION

The 8XC51GA is an 8-bit control-oriented microcontroller based on the 8051 architecture. The 8XC51GA is an enhanced version of the 8XC51BH and incorporates many new features.

- 8-Channel 8-Bit A/D Converter
- 16-Bit Watchdog Timer
- Oscillator Fail Detect Logic
- Half-Duplex, Synchronous Serial Interface
- 7 Interrupt Sources
- 400 mV Hysteresis on Ports 1 and 3 Inputs

Other features available on the 8XC51GA which are also on the 8XC51BH include:

- 4 Kbytes of EPROM (87C51GA) or ROM (83C51GA) Program Memory
- 128 Bytes of Data RAM
- Idle and Power Down Modes
- Full-Duplex, Asynchronous Serial Interface
- Two 16-Bit Programmable Timer/Counters
- 32 Programmable I/O Lines

The 8XC51GA uses the standard 8051 instruction set and is compatible with existing 80C51 family of products. The 83C51GA is the factory masked ROM device; the 80C51GA is the ROMless device; and the 87C51GA is the EPROM device. The designation, 8XC51GA, refers to any of the three GA devices.

It is assumed that the reader is familiar with the 8051 architecture. The following sections discuss only the differences between the 8XC51GA and the standard 80C51. For more detailed information on the 8051, consult the "Hardware Description of the 8051, 8052 and 80C51" chapter in the Intel *Embedded Controller Handbook*.

OVERVIEW OF THE A/D CONVERTER

The 8XC51GA A/D Converter is an 8-bit device with 8 inputs in the 48- and 52-pin packages. It features user selectable internal Sample and Hold and conversion speed control circuitry. The AD Converter operates in both normal and idle modes with a nominal conversion speed of 22 μs (130 states) at 12 MHz and an accuracy of ± 1 LSB. Separate voltage reference (V_{REF}) and analog ground (AGND) signals are bonded out to external pins. The 4 low-order analog inputs are multiplexed with the 4 low-order Port 1 inputs.

A/D Special Function Registers

There are two Special Function Registers associated with the A/D operation. Writing to the A/D control register ADCON controls the start of the A/D Conversion, the enabling or disabling of the internal sample and hold, the number of states taken for conversion, and the analog input channel selection. Reading the ADCON register yields the current status of the A/D converter operation.

The A/D conversion result is stored in the result register, ADRES, a read-only register which is cleared before the start of any conversion.

The A/D Control Register, ADCON, contains the bits necessary for controlling the A/D conversion process.

Bit	7	6	5	4	3	2	1	0
ADCON (Write)	X	X	SHD	RCS	GO	CH2	CH1	CH0

Address = 97H
Not Bit Addressable

Reset Value = XX000000H

Symbol	Position	Function
—	ADCON.7	Not Used
—	ADCON.6	Not Used
SHD*	ADCON.5	Sample-and-Hold Bit. Writing a 0 enables S/H. Writing a 1 disables S/H.
RCS*	ADCON.4	Reduce Conversion States bit. Writing a 1 reduces the number of states for conversion. Writing a 0 causes a normal number of conversion states.
GO	ADCON.3	Start A/D Conversion Bit. Writing a 1 starts the A/D conversion process. Writing a 0 means conversion process not started (or cancelled).
CH2	ADCON.2	High-order bit for analog input channel selection. Value = 4
CH1	ADCON.1	Mid-order bit for analog input channel selection. Value = 2
CH0	ADCON.0	Low-order bit for analog input channel selection. Value = 1

*See SHD/RLS/States Chart

Figure 1. ADCON: A/D Control Register

Reading the ADCON register yields the complete status of the A/D conversion process as follows:

Bit	7	6	5	4	3	2	1	0
ADCON (Read)	X	AIF	SHD	RCS	STA	CH2	CH1	CH0

Address = 97H
Not Bit Addressable

Reset Value = XX000000H

Symbol	Position	Function
	ADCON.7	Not Used
AIF	ADCON.6	A/D Interrupt Flag: Set to a 1 at the end of a conversion. A 0 indicates idle or conversion in progress.
SHD	ADCON.5	Sample/Hold Disabled status bit. A 1 means disabled. A 0 means enabled.
RCS	ADCON.4	Reduced Conversion States status bit. A 1 means reduced number of states. A 0 indicates normal number of states.
STA	ADCON.3	Conversion Status bit. A 1 indicates A/D conversion in progress. A 0 means A/D Idle.
CH2	ADCON.2	High-order input channel address bit.
CH1	ADCON.1	Mid-order input channel address bit.
CH0	ADCON.0	Low-order input channel address bit.

Figure 2. ADCON: A/D Status Register

The A/D Result register ADRES is a binary-coded result of the last A/D conversion and ranges from 00 to FF Hex.

Bit	7	6	5	4	3	2	1	0
ADRES (Read Only)	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
	MSB							LSB

Address = 84H
Not Bit Addressable

Reset Value = 01111111B

Writing to the read-only ADRES register should be avoided.

A/D Conversion Speed Control

The purpose of a sample and hold capacitor is to minimize the effect of external noise by isolating the signal input from the signal source during an A/D conversion. Unfortunately, capacitor leakage can lead to inaccuracies in the A/D conversion at very low clock speeds. For this reason, the sample and hold capacitor can be taken out of the circuit under program control and the number of cycles needed for the conversion can be reduced under program control. Thus, the conversion time for designs using slow clocks is reduced. Note that when running at full speed, both the sample and hold capacitors and the normal number of states should be enabled for the conversion. The following table indicates the number of states needed for a conversion with all possible combinations of the SHD (Sample-and-Hold Disabled) and RCS (Reduced Clock States) bits in the ADCON register.

SHD	RCS	Number of States
0	0	130 (22 μs @ 12 MHz)
0	1	75
1	0	235
1	1	131

A/D Interrupt

The A/D interrupt flag, AIF, is set following each A/D conversion cycle. Bit IE.6 of the 8XC51GA Interrupt Enable Register and Bit IP.6 of the Interrupt Priority Register are assigned to the A/D interrupt. If the A/D interrupt is enabled by IE.6, then IP.6 is examined before service to determine the priority level. AIF is cleared when the system vectors to the interrupt service routine address 0033H. For further information on the A/D interrupt, refer to the "Interrupts" section.

Analog Input Channels

The input pins for the four low-order analog inputs, ACH0–3, are shared with the four low-order digital inputs, P1.0–3. This restricts P1.0–3 as inputs only; they cannot be used as outputs. The four high-order analog inputs, ACH4–7 are not shared and exist as discrete inputs in the 48-pin and 52-pin packages. Note that the high-order four bits of Port 1, P1.4–7, can be used as normal outputs.

PORTS

Except for the shared input portion of Port 1 all the other port functions are essentially the same as in the 80C51 with the following enhancements:

P1.4–P1.7, Port 2, and Port 3 in the 8XC51GA reset to output high asynchronously in order to guarantee known states even in the absence of an active internal clock. (The 80C51 ports reset synchronously which requires that the clock be running). In the 8XC51GA, Port 0 is floated asynchronously at reset. If output highs are required, then external pull-up resistors should be installed on Port 0. P1.0–P1.3 are high-impedance (floating), input-only type inputs.

For improved noise margin, Ports 1 and 3 have Schmitt trigger inputs with minimum hysteresis of 400 mV. TTL compatibility is maintained.

WATCHDOG TIMER (WDT)

The Watchdog Timer (WDT) provides the ability to recover from hardware or software malfunctions by forcing the part into reset. The WDT is a 16-bit counter which must be cleared by software before the counter reaches the maximum value of FFFFH. Otherwise, the WDT generates an internal reset signal. The WDT reset signal is logically ORed with the Oscillator Fail Detect reset signal to generate an asynchronous reset which has a 4 machine cycle duration. The WDT operates in both normal and idle modes. The counter is cleared and initiated by reset or a software clear. A software clear consists of writing the sequence 1EH and E1H to the Watchdog Timer Control Register, WDTCON.

Three Special Function Registers are allocated for the WDT. The software WDT clear sequence of 1EH and E1H is written to WDTCON, a write-only register. The other two SFRs are allocated to the read-only timer registers, WDTLB (Watchdog Timer Lower Byte) and WDTUB (Watchdog Timer Upper Byte). The following chart indicates the SFR addresses of these three registers:

The Watchdog Timer is automatically disabled during Power Down Mode. It cannot be disabled during Normal and Idle Modes and is active anytime the oscillator is running. The external RESET pin is not driven upon a WDT generated reset.

SFR Name	Address	Function
WDTCON	A6H	Writing sequence 1EH and E1H clears the watchdog timer registers to 0s.
WDTUB	96H	Reading this address yields the contents of the upper byte of the WDT.
WDTLB	86H	Reading this address yields the contents of the lower byte of the WDT.

OSCILLATOR FAIL DETECT (OFD)

The Oscillator Fail Detect Circuit triggers a reset (for 4 machine cycles) if the oscillator frequency is below the trigger frequency (range of 20 KHz to 400 KHz). The reset is removed when the oscillator frequency is higher than the trigger frequency. The OFD can be disabled by software by writing the sequence E1H and 1EH to the OFDCON register. Writing anything to OFDCON except the disable sequence, E1H and 1EH, will have no effect.

Before going into the Power Down Mode, the OFD must be disabled or the OFD will force the 8XC51GA out of Power Down. Once the OFD has been disabled, it can only be enabled again by a RESET, which is a necessary step to come out of the Power Down Mode. The OFD cannot be enabled under software control.

Bit	7	6	5	4	3	2	1	0
OFDCON (Read/Write)	—	—	—	—	—	—	—	OFDS

Address = A5H
Not Bit Addressable
OFDS = 0: OFD Active
OFDS = 1: OFD Disabled
Reset Value = XXXXXXX0H

SERIAL EXPANSION PORT (SEP)

In addition to the existing serial port of the 8XC51, a half-duplex synchronous serial interface is provided in the 8XC51GA. Two pins, SEPIO and SEPCLK, are dedicated for the interface. The SEPIO pin is used for transmission or reception of 8-bit packets of serial data, and the SEPCLK outputs the synchronizing clock signal. Four clock frequencies and four serial interface timing modes are provided through the SEP Control Register.

Three SFRs are used for the Serial Expansion Port. The SEPCON register controls the operation of the SEP while the SEPSTA register returns the status of the SEP operation. The data is exchanged through the SEPDAT register.

Bit	7	6	5	4	3	2	1	0
SEPCON (Read/Write)	X	X	SEPE	SEPREN	CLKP	CLKPH	SEPS1	SEPS0

Address = D7H
Not Bit Addressable
Reset Value = XX000000B

Symbol	Position	Function
—	SEPCON.7	Not Used
—	SEPCON.6	Not Used
SEPE	SEPCON.5	SEP Enable: 1 = Enable, 0 = Disable with SEPIO and SEPCLK tri-stated
SEPREN	SEPCON.4	SEP Receive Enable: 1 = Enable 0 = Disable
CLKP	SEPCON.3	Clock Polarity: 0 = Idle Polarity is Low 1 = Idle Polarity is High
CLKPH	SEPCON.2	Clock Phase: 0 = Start Data Sample on First SEPCLK Edge 1 = Start Data Sample on SEPCLK Edge Half Phase Later
SEPS1	SEPCON.1	See SEPS1/SEPS0 Chart, Figure 4
SEPS0	SEPCON.0	See SEPS1/SEPS0 Chart, Figure 4

Figure 3. SEPCON: Serial Expansion Port Control Register

SEPS1	SEPS0	XTAL Divided by	Freq. (@12 MHz)
0	0	12	1.000 MHz
0	1	24	500 KHz
1	0	48	250 KHz
1	1	96	125 KHz

Figure 4. User Selectable Clock Frequencies

Reset disables the SEP by resetting the enable bit, SEPE. When SEPE is set by the software, the SEPCLK will assume the idle state controlled by the CLKP bit. If CLKP = 0 the idle state of the SEPCLK clock output is low, and if CLKP = 1 the idle state of the SEPCLK output is high.

The CLKPH bit controls the point in time at which the input data is sampled. If CLKPH = 0 the data is sampled for 8 cycles starting from the first SEPCLK transition edge. If CLKPH = 1 the data is sampled for 8 cycles starting from the transition edge one-half phase later from the first SEPCLK transition edge. The four combinations of the CLKP and CLKPH bits allow four different serial interface timings as shown in the following diagram. No matter which timing is chosen, the data will always be transmitted a half cycle ahead of the sampling edge. (See Figure 5.)

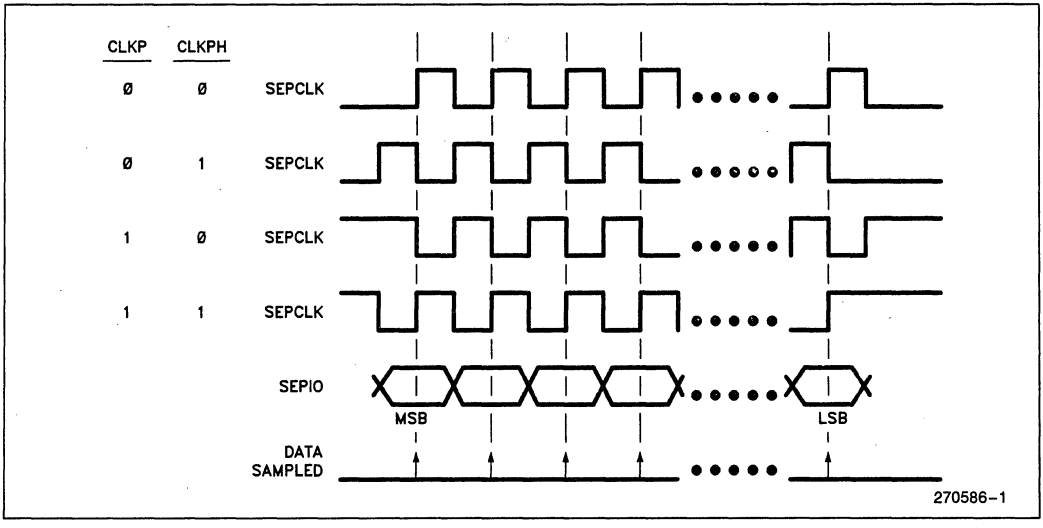


Figure 5. SEP Clock Waveforms

Bit	7	6	5	4	3	2	1	0
SEPSTA (Read/Write)	X	X	X	SEPFWR	SEPFRD	SEPIF	SEPIP	SEPIE

Address = F7H
Not Bit Addressable

Reset Value = XXX00000B

Symbol	Position	Function
—	SEPSTA.7	Reserved
—	SEPSTA.6	Reserved
—	SEPSTA.5	Reserved
SEPFWR	SEPSTA.4	SEPFWR = 1: SEPDAT Read/Write Attempted During Data Transmission
SEPFRD	SEPSTA.3	SEPFRD = 1: SEPDAT Read/Write Attempted During Data Reception
SEPIF	SEPSTA.2	SEPIF = 1: Interrupt Flag Set upon Completion of Data Transmission or Reception SEPIF = 0: Interrupt Flag Cleared
SEPIP	SEPSTA.1	SEPIP = 1: SEP Interrupt Priority is High SEPIP = 0: SEP Interrupt Priority is Low
SEPIE	SEPSTA.0	SEPIE = 1: SEP Interrupt Enabled SEPIE = 0: SEP Interrupt Disabled

Figure 6. SEPSTA: Serial Expansion Port Status Register

Bit	7	6	5	4	3	2	1	0
SEPDAT (Read/Write)	D7	D6	D5	D4	D3	D2	D1	D0

MSB LSB
Address E7H Reset Value = XXXXXXXXB
Not Bit Addressable

Transmitting

The SEPIO pin will float until transmit is initiated by writing to the SEPDAT register. Note that the Receive Enable Bit SEPREN must be cleared before transmitting. The data byte that is written to SEPDAT will be shifted out through the SEPIO pin, MSB first. At the same time, the synchronous clock SEPCLK will be output (8 cycles). If an attempt to read or write is made to the SEPDAT register during a transmit operation the Fault Write Bit SEPFWR will be set. The transmit operation will still be completed, and the SEPIF bit will be set. SEPFWR can be cleared by software or by a reset.

Receiving

Data reception is initiated by setting the SEPREN bit in the SEPCON SFR. The SEPCLK outputs the synchronizing clock, and the data received on the SEPIO pin is shifted into SEPDAT. The SEPREN bit is automatically cleared after 8 bits have been received. The Read Fault bit SEPFRD is set when a read or write to the SEPDAT register is attempted during a receive operation. The data reception will be completed and the false operation will be ignored. The SFRFRD bit can be cleared by software or a reset. Note that the input data must be stable during the SEPCLK pulse train. The source of the transmitted data during a receive operation has no control over the clock.

SEP Interrupt

At the end of either a transmission or reception, the SEP interrupt flag SEPIF is set, and if the SEPIE bit equals a 1 then an interrupt is generated. The SEPIF bit can be set regardless of the state of the SEPIE bit, but SEPIF must be cleared by software.

INTERRUPTS

The A/D interrupt and the Serial Expansion Port interrupt have been added to the five standard 8XC51 interrupts for a total of seven. When an A/D conversion is completed, the A/D interrupt flag AIF is set. (AIF is located in the ADCON register). When the A/D interrupt vectors to address 0033H, this flag is cleared by hardware. The A/D interrupt can be enabled or disabled by the control bit EAI in the IE register (Figure 7). The priority level can be set by the control bit PAI in the IP register (Figure 8).

At the end of data transmission or reception in the Serial Expansion Port, the SEP interrupt flag SEPIF is set. The SEP interrupt is enabled or disabled by the SEPIE bit in the SEPSTA register. The priority level can be set by the SEPIP bit in the SEPSTA register.

Interrupt Enable Register (IE)

Bit	7	6	5	4	3	2	1	0
IE (Read/Write)	EA	EAI	X	ES	ET1	EX1	ET0	EX0

Address = A8H
Bit Addressable

Reset Value = 00X00000B

Symbol	Position	Function
EA	IE.7	EA = 0: Interrupts Globally Disabled EA = 1: Interrupts Enabled by the Individual Interrupt Control Bits
EAI	IE.6	EAI = 0: A/D Interrupt Disabled EAI = 1: A/D Interrupt Enabled if EA = 1
X	IE.5	Reserved for 8052 Timer 2 Interrupt
ES	IE.4	ES = 0: Serial Port Interrupt Disabled ES = 1: Serial Port Interrupt Enabled if EA = 1
ET1	IE.3	ET1 = 0: Timer 1 Interrupt Disabled ET0 = 1: Timer 1 Interrupt Enabled if EA = 1
EX1	IE.2	EX1 = 0: External Interrupt 1 Disabled EX1 = 1: External Interrupt 1 Enabled if EA = 1
ET0	IE.1	ET0 = 0: Timer 0 Interrupt Disabled ET0 = 1: Timer 0 Interrupt Enabled if EA = 1
EX0	IE.0	EX0 = 0: External Interrupt 0 Disabled EX0 = 1: External Interrupt 0 Enabled if EA = 1

Figure 7. Interrupt Enable Register

Interrupt Priority Register (IP)

Bit	7	6	5	4	3	2	1	0
IP (Read/Write)	X	PAI	X	PS	PT1	PX1	PT0	PX0

Address = B8H
Bit Addressable

Reset Value = X0X00000B

Symbol	Position	Function
X	IP.7	Reserved
PAI	IP.6	A/D Interrupt Priority Bit PAI = 1: High Priority PAI = 0: Low Priority
X	IP.5	Reserved for 8052 Timer 2 Interrupt
PS	IP.4	Serial Port Interrupt Priority Bit PS = 1: High Priority PS = 0: Low Priority
PT1	IP.3	Timer 1 Interrupt Priority Bit PT1 = 1: High Priority PT1 = 0: Low Priority
PX1	IP.2	External Interrupt 1 Priority Bit PX1 = 1: High Priority PX1 = 0: Low Priority
PT0	IP.1	Timer 0 Interrupt Priority Bit PT0 = 1: High Priority PT0 = 0: Low Priority
PX0	IP.0	External Interrupt 0 Priority Bit PX0 = 1: High Priority PX0 = 0: Low Priority

Figure 8. Interrupt Priority Register

Interrupt Priority

The interrupts are divided into two hardware priority levels depending on the state of the interrupt priority bit in the IP register. This divides the interrupts into two groups, high and low priority. Also within each priority level there is a second priority structure determined by the internal polling sequence. These priorities are given below.

Interrupt Source	Vector Address	Priority Within Level
INT0	0003H	0 (Highest)
TIMER 0	000BH	1
A/D	0033H	2
INT1	0013H	3
Serial Expansion Port	003BH	4
TIMER 1	001BH	5
Serial Port	0023H	6 (Lowest)

Note that the vector addresses for the A/D and SEP interrupts are backwards compatible with the 8XC51 interrupts.

POWER REDUCTION MODES

The two power reduction modes, Idle and Power Down, are similar to the 8XC51 with the following additions:

In addition to the CPU being disabled and the External Interrupts, Serial Port, and Timers being enabled during Idle Mode, the A/D Converter, Watchdog Timer, and the Oscillator Fail Detect circuitry are also enabled.

During Power Down Mode, all functions are suspended while maintaining the status of the CPU, memory, and I/O. The only exit from Power Down on the 8XC51GA is a hardware reset. Note that the Oscillator Fail Detect must be disabled before going into Power Down Mode. Otherwise, the OFD logic will cause a reset when the oscillator is stopped. This would immediately bring the part out of Power Down Mode.

SPECIAL FUNCTION REGISTERS

The following table indicates the layout of the Special Function Registers including the addresses and initial values immediately following reset.

Note that not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses will, in general, return random data, and write accesses will have no effect.

User software should not write 1s to these unimplemented locations, since they may be used in future 8051 products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0, and their active values will be 1.

Table 1. Special Function Register Memory Map and Values after Reset

FBH									FFH
F0H	*B 00000000							SEPSTA XXX00000	F7H
E8H									EFH
E0H	*ACC 00000000							SEPDAT XXXXXXXX	E7H
D8H									DFH
D0H	*PSW 00000000							SEPCON XX000000	D7H
C8H									CFH
C0H									C7H
B8H	*IP X0X00000								BFH
B0H	*P3 11111111								B7H
A8H	*IE 00X00000								AFH
A0H	*P2 11111111					OFDCON XXXXXX0	WDTCON XXXXXXXX		A7H
98H	* SCON 00000000	* SBUF XXXXXXXX							9FH
90H	* P1 11111111					WDTDIS XXXXXXX0	WDTUB 00000000	ADCON XX000000	97H
88H	* TCON 00000000	* TMOD 00000000	* TL0 00000000	*TL1 00000000	* TH0 00000000	* TH1 00000000			8FH
80H	* P0 11111111	* SP 00000111	* DPL 00000000	* DPH 00000000	ADRES 01111111		WDTLB 00000000	* PCON ** 0XXX0000	87H

NOTES:
* = Found in the 8051 core *(See 8051 Hardware Description for explanations of these SFRs)
** = See description of PCON SFR. Bit PCON.4 is not affected by reset.
X = Undefined

87C51GA/87C51GA-1/87C51GA-2 CHMOS 8-BIT MICROCONTROLLER WITH A/D CONVERTER AND 4 KBYTES OF EPROM

87C51GA—3.5 MHz to 12 MHz, V_{CC} = 5V ±10%
87C51GA-1—3.5 MHz to 16 MHz, V_{CC} = 5V ±10%
87C51GA-2—0.5 MHz to 12 MHz, V_{CC} = 5V ±10%

- 8-Channel 8-Bit A/D Converter
- Oscillator Fail Detect
- 16-Bit Watchdog Timer
- Synchronous Serial Channel
- 2-Level Program Memory Lock
- Boolean Processor
- 128-Byte Data RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- 7 Interrupt Sources
- High Performance CHMOS EPROM
- TTL- and CMOS-Compatible Logic
- Quick-Pulse Programming™
- Full-Duplex Serial Channel
- 64K External Program Memory Space
- 64K External Data Memory Space
- Idle and Power Down Modes
- ONCE™ Mode Facilitates System Testing
- DIP, CERQUAD and PLCC Packaging Available
- Hysteresis on Ports 1 and 3
- Program Memory Lock

The 87C51GA combines all the features of the 87C51 with these additional enhancements: an 8-channel, 8-bit A/D converter; a 16-bit watchdog timer; oscillator fail detect circuitry; and a half-duplex synchronous serial port. The 87C51 family features include: 4 Kbytes of EPROM; 128 bytes of RAM; 32 I/O lines; two 16-bit programmable timer/counters; a seven source two-level interrupt structure; a full-duplex serial port; on-chip oscillator and clock circuitry; and two power reduction modes. The 80C51GA is the ROMless part and the 83C51GA is the masked ROM part.

The 87C51GA is fabricated on Intel's CHMOS II-E process and is functionally compatible with the standard 8051 Family of HMOS and EPROM products. CHMOS II-E is a technology which combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. This combination expands the effectiveness of the powerful 8051 architecture and instruction set.

The 87C51GA EPROM array uses a modified Quick-Pulse Programming Algorithm, by which the entire 4-Kbyte array can be programmed in about 12 seconds. The on-chip Program Memory is electrically programmed and can be erased by exposure to ultra-violet light.

The extremely low operating power, along with the two software selectable reduced power modes, Idle and Power Down, make this part very suitable for low power applications. The Idle mode freezes the CPU while allowing the RAM, timer/counters, serial port, A/D converter, watchdog timer, oscillator fail detect and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

October 1988
Order Number: 270562-002

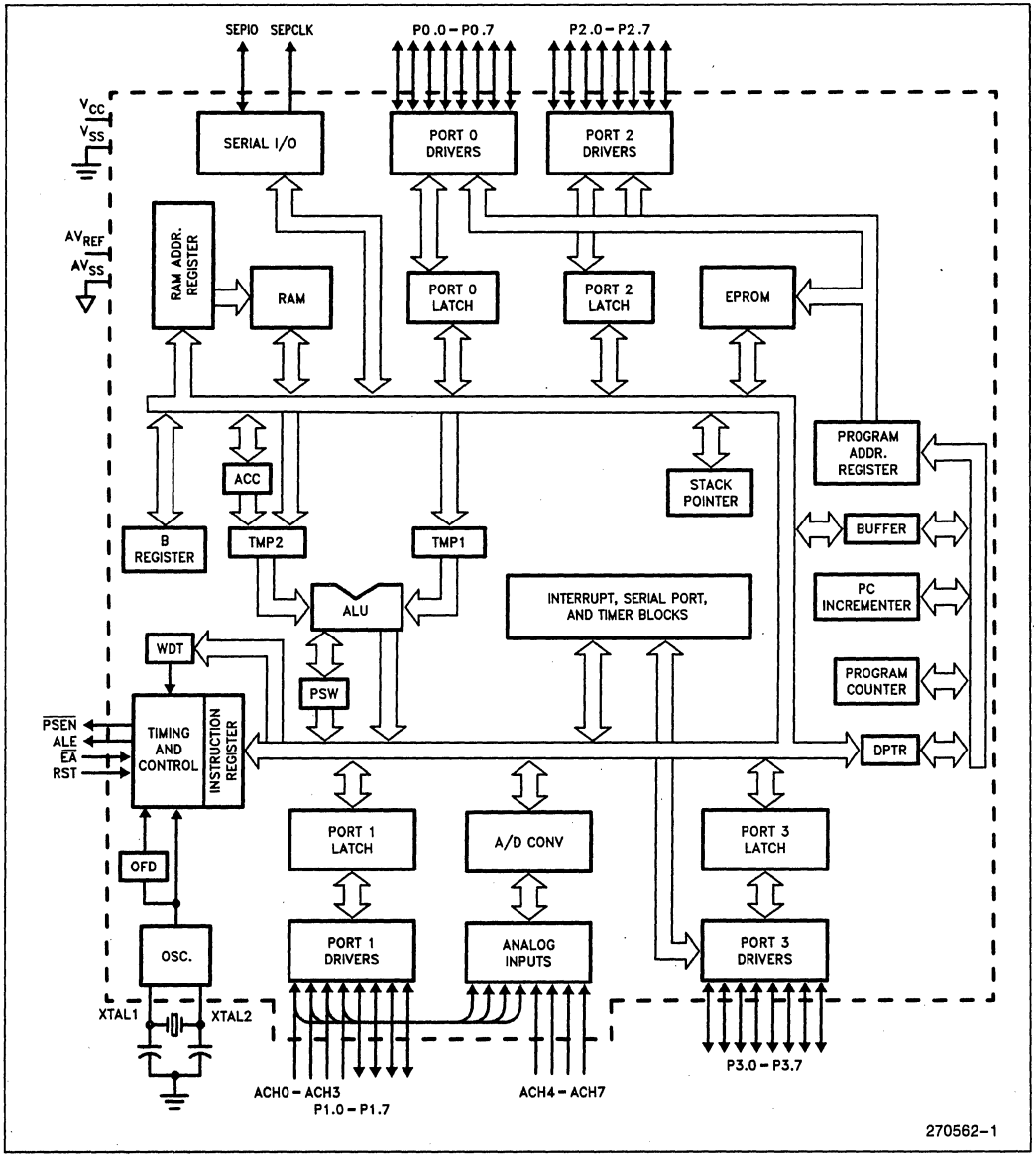


Figure 1. Block Diagram

PACKAGES

Part	Prefix	Package Type
87C51GA/	C	48-Pin Ceramic
87C51GA-1/	P	48-Pin Plastic
87C51GA-2	N	52-Pin PLCC
	J	52-Pin Cerquad

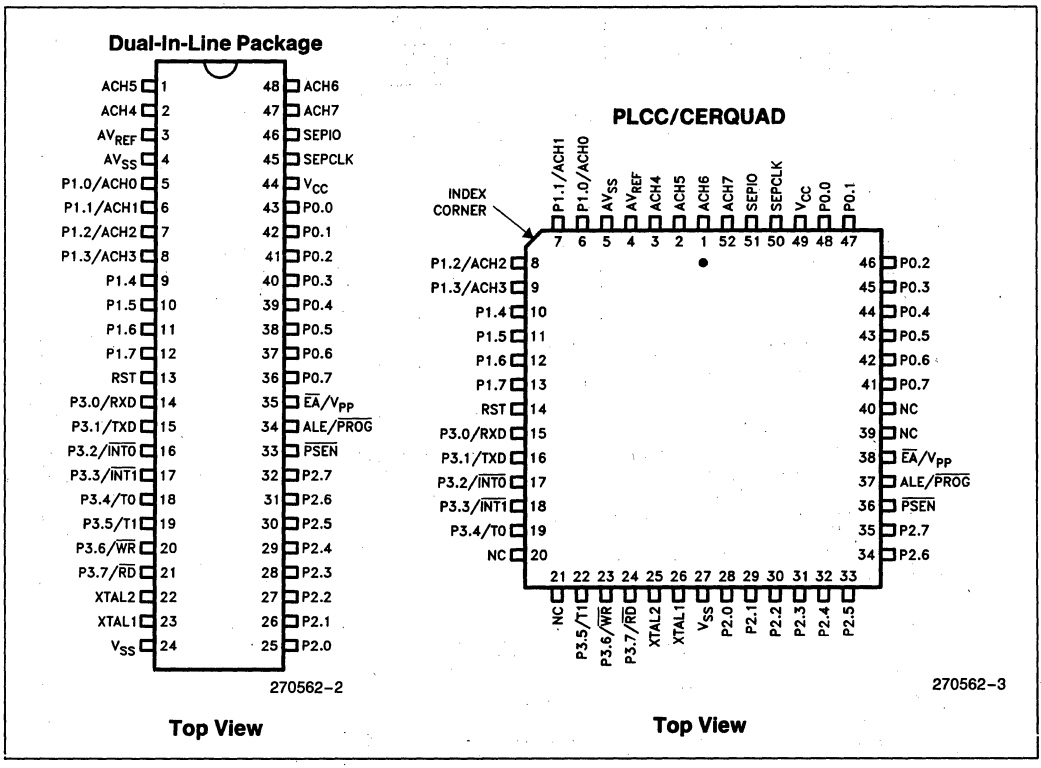


Figure 2. Pin Connections

PIN DESCRIPTION

V_{CC}: Supply voltage during normal, Idle, and Power Down operations.

V_{SS}: Circuit ground.

AV_{REF}: Analog reference voltage.

AV_{SS}: Analog ground.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s.

Port 0 also receives the code bytes during EPROM programming, and outputs the code bytes during program verification. External pullup resistors are required during program verification.

Port 1: Port 1 is an 8-bit I/O port with internal pullups on the 4 high-order bits which can be used for normal I/O. The 4 low-order bits are shared with 4 of the analog inputs and as such, are input only. High-order Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state

can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups. Outputs to the 4 low-order bits have no effect and are ignored. Port 1 inputs are equipped with Schmitt trigger logic with 400 mV of hysteresis and TTL compatible input specifications. Port 1 I/O is explained in the following list:

P1.7—quasi-bidirectional
P1.6—quasi-bidirectional
P1.5—quasi-bidirectional
P1.4—quasi-bidirectional
P1.3—digital input/ACH3—Analog CHannel 3
P1.2—digital input/ACH2—Analog CHannel 2
P1.1—digital input/ACH1—Analog CHannel 1
P1.0—digital input/ACH0—Analog CHannel 0

Port 1 also receives the low-order address bytes during EPROM programming and program verification.

ACH4–ACH7: Analog CHannel 4–7. These are the four high-order analog inputs which have dedicated input pins.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s.

During accesses to External Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives some control signals and the high-order address bits during EPROM programming and program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups. Port 3 inputs are equipped with Schmitt Trigger logic with 400 mV of hysteresis and TTL compatible input specifications.

Port 3 also serves the functions of various special features of the 8051 Family, as listed below:

Pin	Name	Alternate Function
P3.0	RXD	Serial Input Line
P3.1	TXD	Serial Output Line
P3.2	$\overline{INT0}$	External Interrupt 0
P3.3	$\overline{INT1}$	External Interrupt 1
P3.4	T0	Timer 0 External Input
P3.5	T1	Timer 1 External Input
P3.6	$\overline{WR}$	External Data Memory Write Strobe
P3.7	$\overline{RD}$	External Data Memory Read Strobe

SEPIO: Serial Expansion Port I/O bit. This bit is an output for transmission and an input for reception of half-duplex synchronous serial data.

SEPCLK: Serial Expansion Port Clock (Output-only). This clocking signal is an output for transmission and reception of synchronous serial data.

RST: Reset Input. A logic high on this pin resets the device. An internal pulldown resistor permits a power-on reset to be generated using only an external capacitor to V_{CC}.

ALE/$\overline{PROG}$: Address Latch Enable output signal for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during EPROM programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to External Data Memory.

$\overline{PSEN}$: Program Store Enable is the Read strobe to External Program Memory. When the 87C51GA is executing from Internal Program Memory, $\overline{PSEN}$ is inactive (high). When the device is executing code from External Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to External Data Memory.

$\overline{EA}$/V_{PP}: External Access Enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the 87C51GA to fetch code from External Program Memory locations starting at 0000H up to FFFFH. Note, however that if either of the Lock Bits is programmed, the logic level at $\overline{EA}$ is internally latched during reset.

$\overline{EA}$ must be strapped to V_{CC} for internal program execution.

This pin also receives the 12.75V programming supply voltage (V_{PP}) during EPROM programming.

XTAL1: Input to the inverting oscillator amplifier and input to the internal clock generating circuits.

XTAL2: Output from the inverting oscillator amplifier.

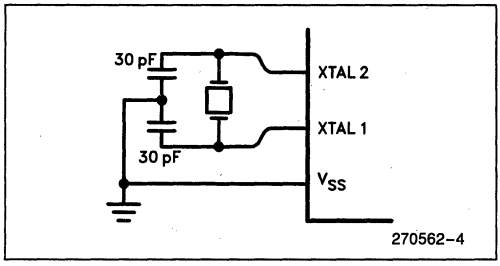


Figure 3. Using the On-Chip Oscillator

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3.

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the data sheet must be observed.

IDLE MODE

In Idle Mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the Special Function Registers remain

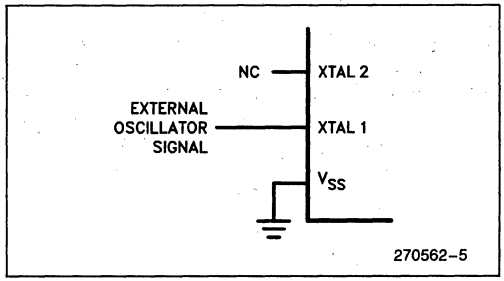


Figure 4. External Clock Drive

unchanged during this mode. The Idle Mode can be terminated by any enabled interrupt or by a hardware reset. Note that the Watchdog Timer is active during Idle Mode. See Design Considerations.

POWER DOWN MODE

In Power Down Mode, the oscillator is stopped and all on-chip functions cease except that the on-chip RAM content is maintained. The mode is invoked by software. The Oscillator Fail Detect circuitry should be disabled before entering Power Down.

The Power Down Mode can be terminated only by a hardware reset. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

A/D Converter

The A/D Converter is an 8-bit successive approximation device with the following features:

8 User Selectable Analog Input Channels
User Selectable Internal Sample and Hold
User Selectable Conversion Speed Control
Nominal Conversion Speed: 22 μs at 12 MHz
Accuracy ± 1 LSB (LSB = 20 mV)
Input Signal Range, Nominally 0V to 5V (AV_{SS} to V_{REF})
Interrupt Driven

Table 1. Status of the External Pins during Idle and Power Down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	Port 0	Port 1	Port 2	Port 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

Like all 8051 Family functions, the A/D converter is controlled by reads and writes to the Special Function Registers. Refer to the *"Hardware Description of the 8XC51GA"* for further information.

Watchdog Timer (WDT)

The Watchdog Timer provides the ability to recover from hardware or software malfunctions by forcing the part into reset. It has the following features:

16-bit Synchronous Counter; Counts Machine Cycles
Asynchronous Reset when Counter = FFFFH (65.5 ms at 12 MHz)
Cleared and Initiated by Reset or Software Clear
Operates in Normal and Idle Mode

Oscillator Fail Detect (OFD)

The Oscillator Fail Detect circuitry triggers a reset if the oscillator frequency is lower than the OFD trigger frequency. It can be disabled by software during Power Down Mode and has the following features.

OFD Trigger Frequency: 20 KHz to 400 KHz
Asynchronous Reset for at Least 4 Machine Cycles
Functions in Normal and Idle Modes
Reactivated by Reset after Software Disable

Serial Expansion Port (SEP)

The Serial Expansion Port is a half-duplex synchronous serial interface with the following features:

Four Clock Frequencies
—XTAL/12
—XTAL/24
—XTAL/48
—XTAL/96

Four Interface Modes
—Rising Edges
—Falling Edges
—High Level
—Low Level
Interrupt Driven

ONCE™ MODE

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems using the 87C51GA without the 87C51GA having to be removed from the circuit. The ONCE Mode is invoked by:

1. Pulling ALE low while the device is in reset and $\overline{PSEN}$ is high;
2. Holding ALE low as RST is deactivated.

While the device is in the ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the 87C51GA is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

DESIGN CONSIDERATIONS

It should be noted that when Idle Mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

During Idle Mode, the Watchdog Timer must periodically be reset under program control to hold off a Watchdog timeout from generating a device reset.

Ambient light is known to affect the internal memory contents during operation. If the 87C51GA application requires the part to be run under ambient lighting, an opaque label should be placed over the window to exclude light.

In this device, ports are reset asynchronously: Port 0 resets to a high impedance (floating) and Ports 1.4–1.7, Port 2 and Port 3 reset to an output high even in the absence of an active internal clock. The 8 analog inputs, ACH0–7, are input-only high-impedance (tri-state) inputs.

ABSOLUTE MAXIMUM RATINGS (1)

Ambient Temperature under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin to V_{SS} ...−0.5V to V_{CC} + 0.5V
Voltage on V_{CC} to V_{SS} .....−0.5V to +6.5V
Maximum I_{OL} per I/O Pin .....15 mA
Power Dissipation.....1.0W*

* Based on package heat transfer limitations, not device power consumption.

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

DC CHARACTERISTICS T_A = 0°C to +70°C; V_{CC} = 5V ±10%, V_{SS} = 0V

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
VT+	High-Going Threshold (Ports 1, 3)	1.1 (3)	2.0	V	
VT−	Low-Going Threshold (Ports 1, 3)	0.6	1.2 (3)	V	
V_{HYS}	Hysteresis (Ports 1, 3)	0.4 (3)		V	
V_{IL}	Input Low Voltage (except $\overline{EA}$)	−0.5	0.2 V_{CC} − 0.1	V	
V_{IL1}	Input Low Voltage ($\overline{EA}$)	−0.5	0.2 V_{CC} − 0.3	V	
V_{IH}	Input High Voltage (except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage (4) (Ports 1, 2, 3)		0.3	V	I_{OL} = 100 μA (1)
			0.45	V	I_{OL} = 1.6 mA (1)
			1.0	V	I_{OL} = 3.5 mA (1)
V_{OL1}	Output Low Voltage (4) (Port 0, ALE, $\overline{PSEN}$)		0.3	V	I_{OL} = 200 μA (1)
			0.45	V	I_{OL} = 3.2 mA (1)
			1.0	V	I_{OL} = 7 mA (1)
V_{OH}	Output High Voltage (Ports 1, 2, 3)	V_{CC} − 0.3		V	I_{OH} = −10 μA
		V_{CC} − 0.7		V	I_{OH} = −30 μA
		V_{CC} − 1.5		V	I_{OH} = −60 μA
V_{OH1}	Output High Voltage (Port 0 in Ext Bus Mode, ALE, $\overline{PSEN}$) (2)	V_{CC} − 0.3		V	I_{OH} = −200 μA
		V_{CC} − 0.7		V	I_{OH} = −3.2 mA
		V_{CC} − 1.5		V	I_{OH} = −6.5 mA
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)		−50	μA	V_{IN} = 0.45V
I_{TL}	Logical 1 to 0 Transition (Ports 1, 2, 3)		−650	μA	V_{IN} = 2V

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

DC CHARACTERISTICS $T_A = 0°C$ to $+70°C$; $V_{CC} = 5V \pm 10\%$, $V_{SS} = 0V$ (Continued)

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
I_{LI}	Input Leakage Current (Ports 0, $\overline{EA}$)		±10	μA	$0 < V_{IN} < V_{CC} - 0.3V$
I_{LI1}	Input Leakage Current (ACH0–7)		±3	μA	$0 < V_{IN} < V_{REF}$
RRST	Reset Pulldown Resistor	50	225	KΩ	
CIO	Pin Capacitance		10	pF	Test Freq. = 1 MHz $T_A = 25°C$
	Power Supply Current				
I_{CC}	Operating, 12 MHz (5)		40	mA	
I_{DL}	Idle Mode, 12 MHz (5)		10	mA	(5)
I_{PD}	Power Down Mode		TBD	μA	
I_{REF}	Reference Voltage = 5.12V		10	mA	

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OLS} of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst case (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. During reset, V_{OH1} for ALE and $\overline{PSEN}$ may fall below the specified value.
3. $VT+_{min}$ and $VT-_{max}$ cannot occur together in the same part as hysteresis is guaranteed to be 400 mV Minimum. See Figure 5.
4. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port —
   Port 0: 26 mA
   Ports 1, 2, and 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification.
   Pins are not guaranteed to sink current greater than the listed test conditions.
5. See Figure 6.
6. See Figures 7 through 10 for I_{CC} test conditions. Minimum V_{CC} for Power Down mode is 2.0V.

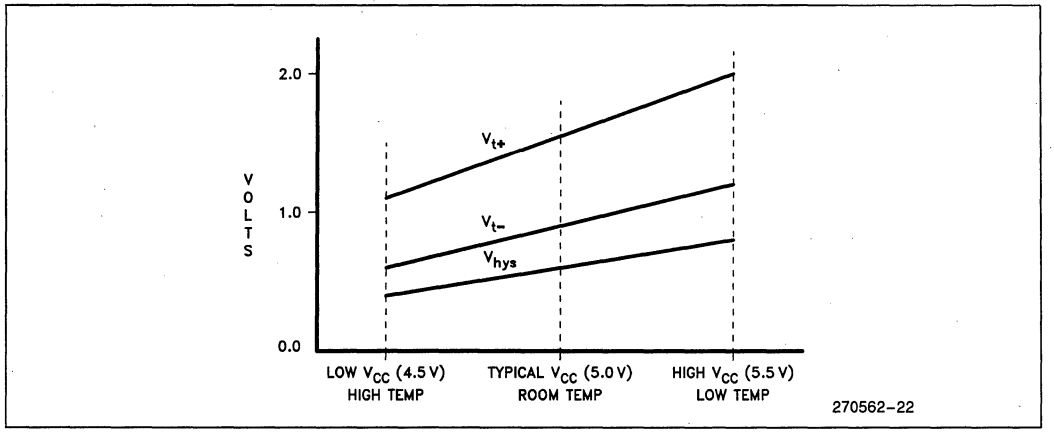
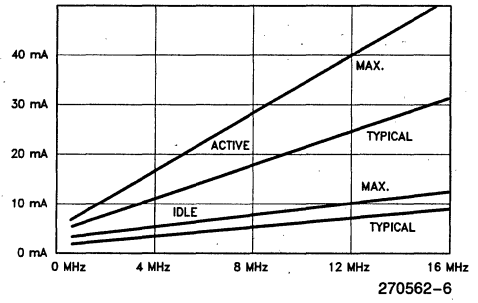


Figure 5. Port 1 and 3 Hysteresis



I_{CC} Max at other frequencies is given by:
Active Mode
I_{CC} MAX = 3.0 × FREQ + 4.1
Idle Mode
I_{CC} MAX = 0.65 × FREQ + 2.1
Where FREQ is in MHz, I_{CC} Max is given in mA.

Figure 6. I_{CC} vs Frequency valid only within frequency specifications of the device under test.

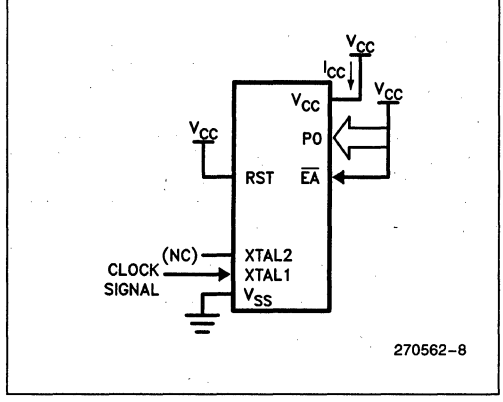


Figure 7. I_{CC} Test Condition, Active Mode. All other pins are disconnected.

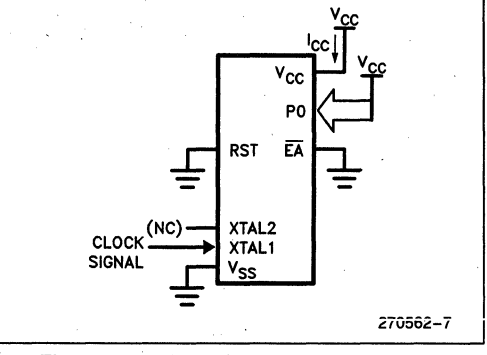


Figure 8. I_{CC} Test Condition, Idle Mode. All other pins are disconnected.

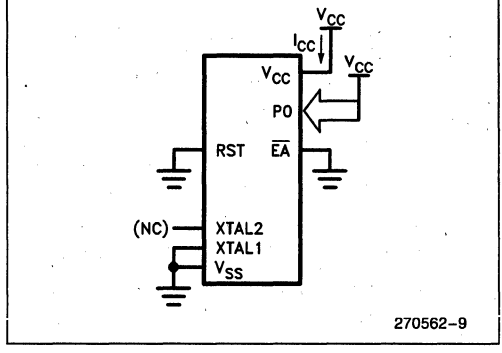


Figure 10. I_{CC} Test Condition, Power Down Mode. V_{CC} = 2.0V to 5.5V

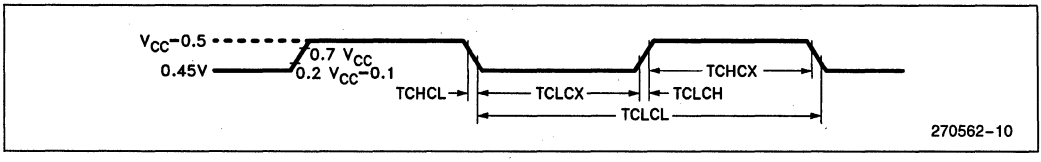


Figure 9. Clock Signal Waveform for I_{CC} tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns

Explanation of the AC Symbols

Each timing symbol has 5 characters. The first character is always a "T" (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address
C: Clock
D: Input data
H: Logic level HIGH
I: Instruction (program memory contents)
L: Logic level LOW, or ALE
P: $\overline{\text{PSEN}}$
Q: Output data
R: $\overline{\text{RD}}$ signal
T: Time
V: Valid
W: $\overline{\text{WR}}$ signal
X: No longer a valid logic level
Z: Float

Example:

TAVLL = Time for Address Valid to ALE Low.
TLLPL = Time for ALE Low to $\overline{\text{PSEN}}$ Low.

AC ELECTRICAL CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±10%, V_{SS} = 0V, load capacitance for port 0, ALE, and $\overline{\text{PSEN}}$ = 100 pF, load capacitance for all other outputs = 80 pF

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

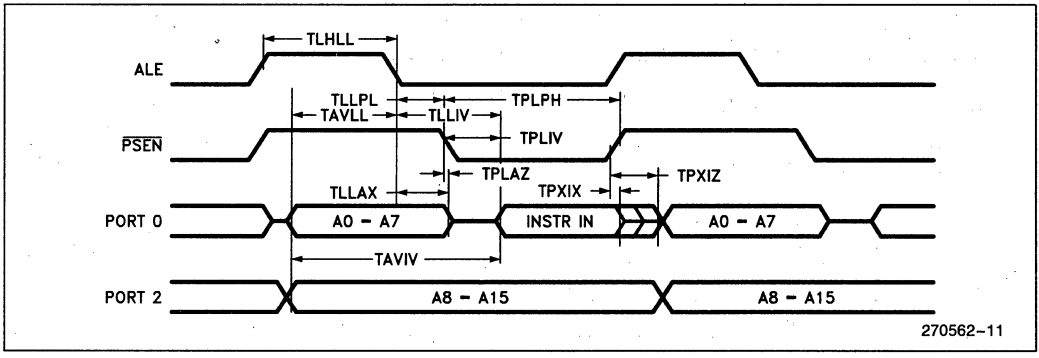
Symbol	Parameter	12 MHz Clock		Variable Clock		Unit
		Min	Max	Min	Max	
1/TCLCL	Oscillator Freqquency 87C51GA 87C52GA-1 87C51GA-2			 3.5 3.5 0.5	 12 16 12	MHz
TLHLL	ALE Pulse Width	127		2 TCLCL − 40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL − 40		ns
TLLAX	Address Hold after ALE Low	53		TCLCL − 30		ns
TLLIV	ALE Low to Valid Instr In		234		4 TCLCL − 100	ns
TLLPL	ALE Low to $\overline{\text{PSEN}}$ Low	53		TCLCL − 30		ns
TPLPH	$\overline{\text{PSEN}}$ Pulse Width	205		3 TCLCL − 45		ns
TPLIV	$\overline{\text{PSEN}}$ Low to Valid Instr In		145		3 TCLCL − 105	ns
TPXIX	Input Instr Hold after $\overline{\text{PSEN}}$	0		0		ns
TPXIZ	Input Instr Float after $\overline{\text{PSEN}}$		59		TCLCL − 25	ns
TAVIV	Address to Valid Instr In		312		5 TCLCL − 105	ns
TPLAZ	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
TRLRH	$\overline{\text{RD}}$ Pulse Width	400		6 TCLCL − 100		ns
TWLWH	$\overline{\text{WR}}$ Pulse Width	400		6 TCLCL − 100		ns
TRLDV	$\overline{\text{RD}}$ Low to Valid Data In		252		5 TCLCL − 165	ns
TRHDX	Data Hold after $\overline{\text{RD}}$	0		0		ns
TRHDZ	Data Float after $\overline{\text{RD}}$		107		2 TCLCL − 60	ns
TLLDV	ALE Low to Valid Data In		517		8 TCLCL − 150	ns
TAVDV	Address to Valid Data In		585		9 TCLCL − 165	ns

ADVANCED INFORMATION—CONTACT INTEL FOR DESIGN-IN INFORMATION

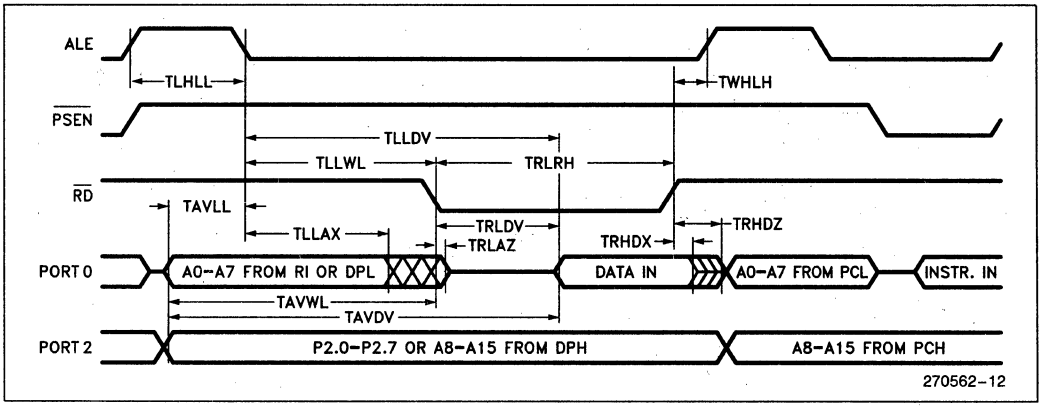
EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS (Continued)

Symbol	Parameter	12MHz Clock		Variable Clock		Unit
		Min	Max	Min	Max	
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3 TCLCL − 50	3 TCLCL + 50	ns
TAVWL	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		4 TCLCL − 130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	33		TCLCL − 50		ns
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL − 50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	TCLCL − 40	TCLCL + 40	ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL − 150		ns

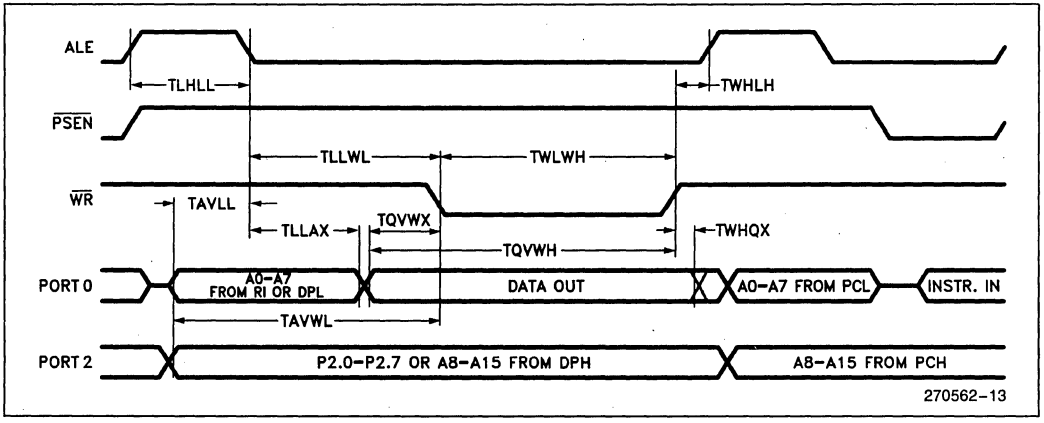
EXTERNAL PROGRAM MEMORY READ CYCLE



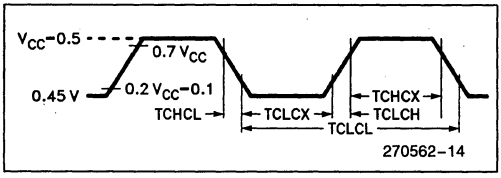
EXTERNAL DATA MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



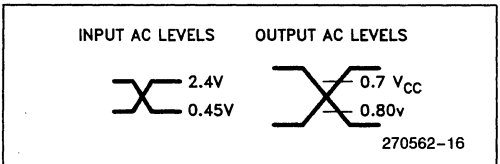
EXTERNAL CLOCK DRIVE WAVEFORM



EXTERNAL CLOCK DRIVE

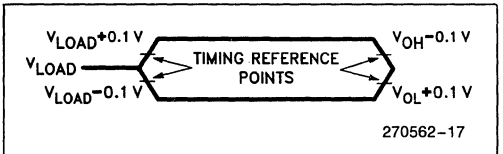
Symbol	Parameter	Min	Max	Unit
1/TCLCL	Oscillator Frequency			
	87C51GA	3.5	12	MHz
	87C51GA-1	3.5	16	
	87C51GA-2	0.5	12	
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

A.C. TESTING INPUT: INPUT, OUTPUT WAVEFORMS



Input timings are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Output timing measurements are made at 0.7 V_{CC} for a logic "1" and 0.8V for a logic "0".

FLOAT WAVEFORM



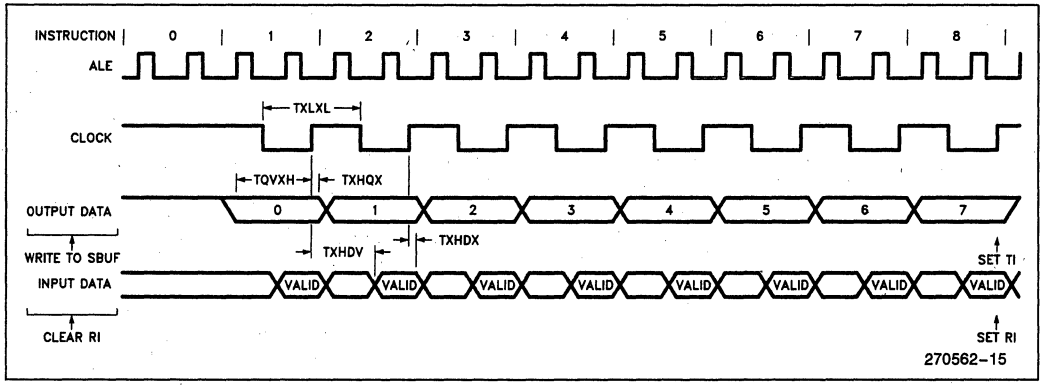
For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

SERIAL TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±10%, V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	12 MHz Osc.		Variable Oscillator		Unit
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12 TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10 TCLCL − 133		ns
TXHQX	Output Data Hold after Clock Rising Edge	50		2 TCLCL − 117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10 TCLCL − 133	ns

SHIFT REGISTER MODE TIMING WAVEFORMS



A TO D CHARACTERISTICS

The absolute conversion accuracy is dependent on the accuracy of V_{REF}. The specifications given below assume adherence to the Operating Conditions section of this data sheet. Testing is done at V_{REF} = 5.12V.

An A/D Glossary of Terms is available at the end of this data sheet.

OPERATING CONDITIONS

V_{CC}, V_{REF} ..... 4.5V to 5.5V
V_{SS}, AV_{SS} 0V
ACH0–7 ..... AV_{SS} to V_{REF}
T_A ..... 0°C to +70°C
F_{OSC} ..... 0.5 MHz to 16.0 MHz
Test Conditions: V_{REF} 5.12V
V_{CC} 5.0V

A/D CONVERTER SPECIFICATIONS

Parameter	Minimum	Typical	Maximum	Unit**	Notes
Resolution	256 8		256 8	Levels Bits	
Absolute Error	0		±1	LSB	
Full Scale Error		−0.5 ± 0.5		LSB	
Zero Offset Error		±0.5		LSB	
Non-Linearity	0		±1	LSB	
Differential Non-Linearity	0		±½	LSB	
Channel-to-Channel Variation	0		±0.4	LSB	
Repeatability		±0.25		LSB	
Temperature Coefficients: Offset Full Scale Differential Non-Linearity		 0.003 0.003 0.003		 LSB/°C LSB/°C LSB/°C	 1 1 1
Off Isolation			−60	dB	1, 2, 3
Feedthrough		−60		dB	1, 2
V_{CC} Power Supply Rejection		−60		dB	1, 2
Input Resistance	1K		5K	Ω	1
D.C. Input Leakage	0		3.0	μA	

NOTES:
* These values are expected for most parts at 25°C
** An "LSB", as used here, has a value of approximately 20 mV.
1. These values are not tested in production and are based on theoretical estimates and laboratory tests.
2. DC to 100 KHz
3. Multiplexer Break-Before-Make Guaranteed.

EPROM CHARACTERISTICS

The 87C51GA is programmed by a modified Quick-Pulse Programming algorithm. It differs from older methods in the value used for V_{PP} (Programming Supply Voltage) and in the width and number of the ALE, $\overline{PROG}$ pulses.

The 87C51GA contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51GA manufactured by Intel.

Table 2 shows the logic levels for reading the signature byte, and for programming the Program Memory, the Encryption Table, and the Lock Bits. The circuit configuration and waveforms for Quick-Pulse Programming are shown in Figures 11 and 12. Figure 13 shows the circuit configuration for normal Program Memory verification.

Table 2. EPROM Programming Modes

Mode	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.7	P2.6	P3.7	P3.6
Read Signature	1	0	1	1	0	0	0	0
Program Code Data	1	0	0*	V_{PP}	1	0	1	1
Verify Code Data	1	0	1	1	0	0	1	1
Pgm Encryption Table	1	0	0*	V_{PP}	1	0	1	0
Pgm Lock Bit 1	1	0	0*	V_{PP}	1	1	1	1
Pgm Lock Bit 2	1	0	0*	V_{PP}	1	1	0	0

NOTES:
"1" = Valid high for that pin
"0" = Valid low for that pin
"V_{PP}" = +12.75V ±0.25V
*ALE/$\overline{PROG}$ receives 25 programming pulses while V_{PP} is held at 12.75V. Each programming pulse is low for 100 μs (±5 μs) and high for a minimum of 10 μs.

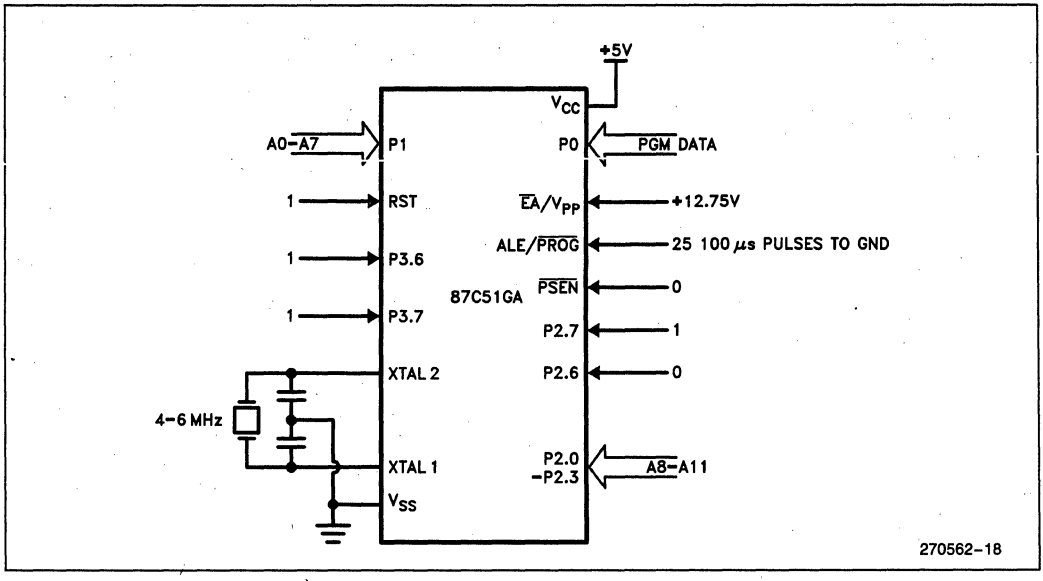


Figure 11. Programming Configuration

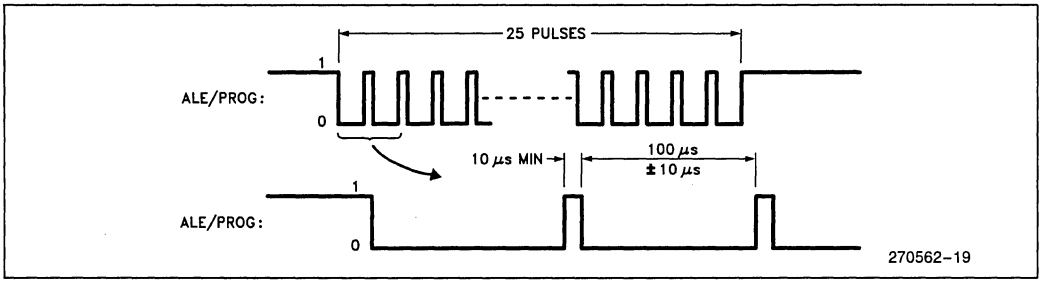


Figure 12. $\overline{\text{PROG}}$ Waveforms

Quick-Pulse Programming™

The setup for Microcontroller Quick-Pulse Programming is shown in Figure 11. Note that the 87C51GA is running with a 4 MHz to 6 MHz oscillator. The reason the oscillator must be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to Ports 1 and 2, as shown in Figure 11. The code byte to be programmed into that location is applied to Port 0. RST, $\overline{\text{PSEN}}$, and pins of Ports 2 and 3 specified in Table 2 are held at the "Program Code Data" levels indicated in Table 2. Then ALE/$\overline{\text{PROG}}$ is pulsed low 25 times as shown in Figure 12.

To program the Encryption Table, repeat the 25-pulse programming sequence for addresses 0 through 1FH, using the "Pgm Encryption Table" levels. Don't forget that after the Encryption Table is programmed, verify cycles will produce only encrypted data.

To program the Lock Bits, repeat the 25-pulse programming sequence using the "Pgm Lock Bit" levels. After one Lock Bit is programmed, further programming of the Code Memory and Encryption Table is disabled. However, the other Lock Bit can still be programmed.

Note that the $\overline{\text{EA}}/V_{PP}$ pin must not be allowed to go above the maximum specified V_{PP} level for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The V_{PP} source should be well regulated and free of glitches and overshoot.

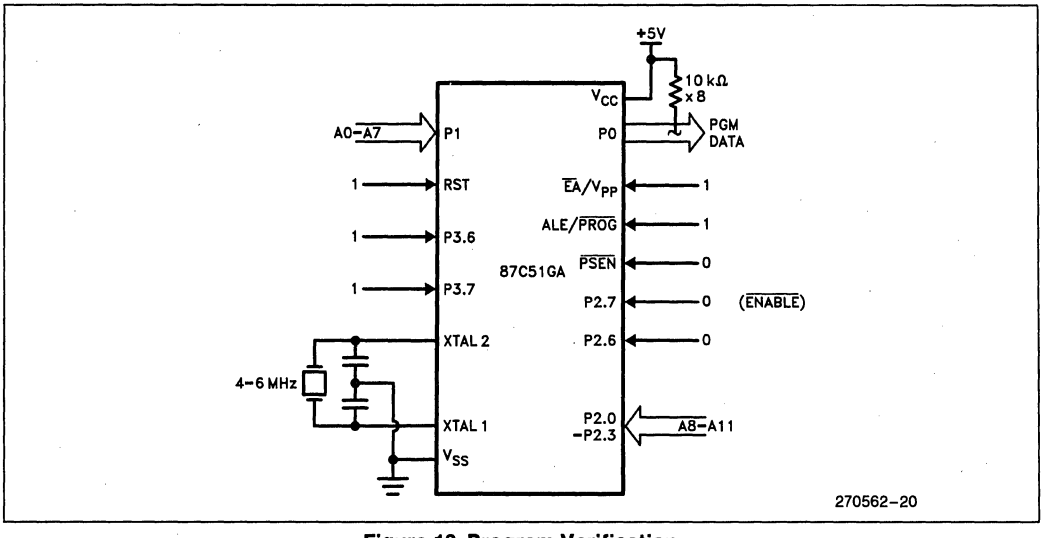


Figure 13. Program Verification

Program Verification

If Lock Bit 2 has not been programmed, the on-chip Program Memory can be read out for program verification. The address of the Program Memory location to be read is applied to Ports 1 and 2 as shown in Figure 13. The other pins are held at the "Verify Code Data" levels indicated in Table 2. The contents of the addressed location will be emitted on Port 0. External pullups are required on Port 0 for this operation. Detailed timing specifications are shown in later sections of this data sheet.

If the Encryption Table has been programmed, the data presented at Port 0 will be the Exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the Encryption Table contents in order to correctly decode the verification data. The Encryption Table itself cannot be read out.

Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 2, and which satisfies the timing specifications, is suitable.

EPROM Program Lock

The two-level Program Lock system consists of two Program Lock bits and a 32 byte Encryption Array which are used to protect the program memory against software piracy.

Encryption Array

Within the EPROM array are 32 bytes of Encryption Array that are initially unprogrammed (all 1's). Every time that a byte is addressed during a verify, 5 address lines are used to select a byte of the Encryption Array. This byte is then exclusive-NOR'ed (XNOR) with the code byte, creating an Encrypted Verify byte. The algorithm, with the array in the unprogrammed state (all 1's), will return the code in it's original, unmodified form.

Program Lock Bits

Also included in the EPROM Program Lock scheme are two Program Lock Bits which are programmed as shown in Table 2.

Table 3 outlines the features of programming the Lock Bits.

Erasing the EPROM also erases the Encryption Array and the Program Lock Bits, returning the part to full functionality.

Table 3. Program Lock Bits and their Features

Program Lock Bits LB1	LB2	Logic Enabled
U	U	No Program Lock features enabled. (Code Verify will still be encrypted by the Encryption Array.)
P	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the EPROM is disabled.
P	P	Same as above, but Verify is also disabled.
U	P	Reserved for Future Definition.

Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are:

(030H) = 89H indicates manufactured by Intel
(031H) = 60H indicates 87C51GA

Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000Å. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. If an application subjects the device to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537Å) to an integrated dose of at least 15 W-sec/cm^2. Exposing the EPROM to an ultraviolet lamp of 12,000 μW/cm^2 rating for 20 to 30 minutes, at a distance of about 1 inch, should be sufficient.

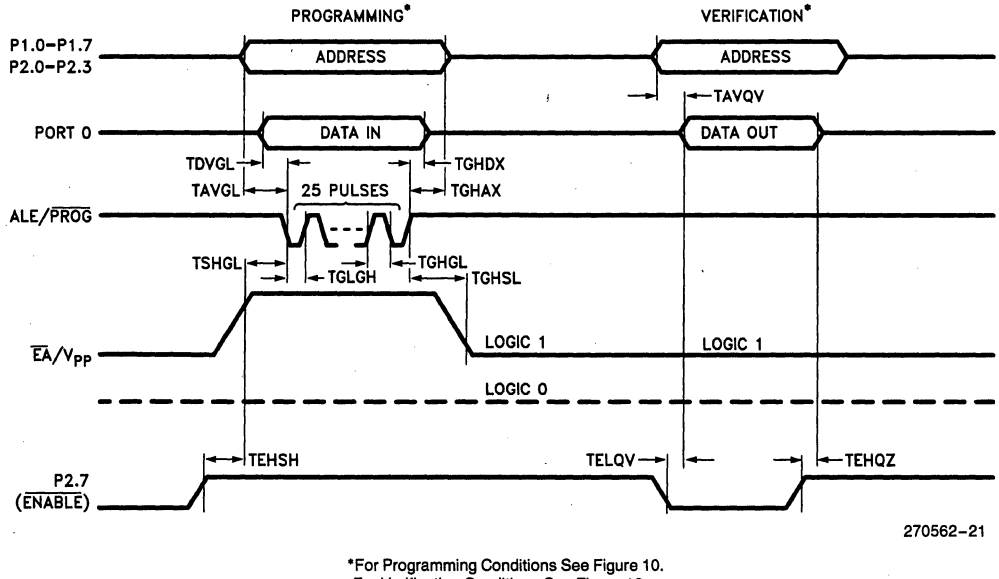
Erasure leaves the array in an all 1s state.

EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS

T_A = 21°C to 27°C, V_{CC} = 5V ±10% V_{SS} = 0V

Symbol	Parameter	Min	Max	Unit
V_{PP}	Programming Supply Voltage	12.5	13.0	V
I_{PP}	Programming Supply Current		50.0	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$ Low	48 TCLCL		
TGHAX	Address Hold after $\overline{PROG}$	48 TCLCL		
TDVGL	Data Setup to $\overline{PROG}$ Low	48 TCLCL		
TGHDX	Data Hold after $\overline{PROG}$	48 TCLCL		
TEHSH	P2.7 ($\overline{ENABLE}$) HIGH to V_{PP}	48 TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
TGHSL	V_{PP} Hold after $\overline{PROG}$	10		µs
TGLGH	$\overline{PROG}$ Width	95	105	µs
TAVQV	Address to Data Valid		48 TCLCL	
TELQV	$\overline{ENABLE}$ Low to Data Valid		48 TCLCL	
TEHQZ	Data Float after $\overline{ENABLE}$	0	48 TCLCL	
TGHGL	$\overline{PROG}$ High to $\overline{PROG}$ Low	10		µs

EPROM PROGRAMMING AND VERIFICATION WAVEFORMS



*For Programming Conditions See Figure 10.
For Verification Conditions See Figure 12.

A/D Glossary of Terms

Absolute Error—The maximum difference between corresponding actual and ideal code transitions. Absolute Error accounts for all deviations of an actual converter from an ideal converter.

Actual Characteristic—The characteristic of an actual converter. The characteristic of a given converter may vary over temperature, supply voltage, and frequency conditions. An actual characteristic rarely has ideal first and last transition locations or ideal code widths. It may even vary over multiple conversions under the same conditions.

Break-Before-Make—The property of a multiplexer which guarantees that a previously selected channel will be deselected before a new channel is selected (e.g. the converter will not short inputs together).

Channel-To-Channel Matching—The difference between corresponding code transitions of actual characteristics taken from different channels under the same temperature, voltage and frequency conditions.

Characteristic—A graph of input voltage versus the resultant output code for an A/D converter. It describes the transfer function of the A/D converter.

Code—The digital value output by the converter.

Code Center—The voltage corresponding to the midpoint between two adjacent code transitions.

Code Transition—The point at which the converter changes from an output code of Q, to a code of Q + 1. The input voltage corresponding to a code transition is defined to be that voltage which is equally likely to produce either of two adjacent codes.

Code Width—The voltage corresponding to the difference between two adjacent code transitions.

Crosstalk—See "Off-Isolation."

D.C. Input Leakage—Leakage current to ground from an analog input pin.

Differential Non-Linearity—The difference between the ideal and actual code widths of the terminal based characteristic.

Feedthrough—Attenuation of a voltage applied on the selected channel of the A/D Converter after the sample window closes.

Full Scale Error—The difference between the expected and actual input voltage corresponding to the full scale code transition.

Ideal Characteristic—A characteristic with its first code transition at V_{IN} = 0.5 LSB, its last code transition at V_{IN} = (V_{REF} − 1.5 LSB) and all code widths equal to one LSB.

Input Resistance—The effective series resistance from the analog input pin to the sample capacitor.

LSB—Least Significant Bit: The voltage corresponding to the full scale voltage divided by 2^n, where n is the number of bits of resolution of the converter. For an 8-bit converter with a reference voltage of 5.12V, one LSB is 20 mV. Note that this is different than digital LSBs since an uncertainty of two LSBs, when referring to an A/D converter, equals 40 mV. (This has been confused with an uncertainty of two digital bits, which would mean four counts, or 80 mV.)

Monotonic—The property of successive approximation converters which guarantees that increasing input voltages produce adjacent codes of increasing value, and that decreasing input voltages produce adjacent codes of decreasing value.

No Missed Codes—For each and every output code, there exists a unique input voltage range which produces that code only.

Non-Linearity—The maximum deviation of code transitions of the terminal based characteristic from the corresponding code transitions of the ideal characteristic.

Off-Isolation—Attenuation of a voltage applied on a deselected channel of the A/D converter. (Also referred to as Crosstalk.)

Repeatability—The difference between corresponding code transitions from different actual characteristics taken from the same converter on the same channel at the same temperature, voltage and frequency conditions.

Resolution—The number of input voltage levels that the converter can unambiguously distinguish between. Also defines the number of useful bits of information which the converter can return.

Sample Delay—The delay from receiving the start conversion signal to when the sample window opens.

Sample Delay Uncertainty—The variation in the sample delay.

Sample Time—The time that the sample window is open.

Sample Time Uncertainty—The variation in the sample time.

Sample Window—Begins when the sample capacitor is attached to a selected channel and ends when the sample capacitor is disconnected from the selected channel.

Successive Approximation—An A/D conversion method which uses a binary search to arrive at the best digital representation of an analog input.

Temperature Coefficients—Change in the stated variable per degree centigrade temperature change. Temperature coefficients are added to the typical values of a specification to see the effect of temperature drift.

Terminal Based Characteristic—An actual characteristic which has been rotated and translated to remove zero offset and full scale error.

V_{CC} Rejection—Attenuation of noise on the V_{CC} line to the A/D converter.

Zero Offset—The difference between the expected and actual input voltage corresponding to the first code transition.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -001 version of the 87C51GA data sheet:

1. Reference to ROM and ROMless versions was reworded.
2. Packages Table was added.
3. Second paragraph to Power Down Mode description was added.
4. A/D nominal conversion speed changed from 27 μs to 22 μs at 12 MHz.
5. Figure 5 for Ports 1 and 3 hysteresis added.
6. Note 2 for DC Characteristics pertaining to the V_{OH1} specification on ALE and $\overline{PSEN}$ was changed.
7. Note 4 on maximum current specifications added to DC Characteristics.
8. The graph for I_{CC} specs was extended on Figure 6 from 12 MHz to 16 MHz and from 3.5 MHz to 0.5 MHz.
9. The following AC Timing specifications were changed:
   TLLAX changed from TCLCL−35 to TCLCL−30.
   TLLPL changed from TCLCL−40 to TCLCL−30.
   TRHDZ changed from 2TCLCL−70 to 2TCLCL−60.
   TQVWX changed from TCLCL−60 to TCLCL−50.
   TQVWH was added.
10. F_{OSC} specifications for Sample and Hold were deleted on A/D Characteristics.
11. Program Memory Lock scheme description was added.
12. Data Sheet Revision Summary added.

83C152

11

HARDWARE DESCRIPTION OF THE 83C152

1.0 INTRODUCTION

The 83C152 Universal Communications Controller is an 8-bit microcontroller designed for the intelligent management of peripheral systems or components. The 83C152 is a derivative of the 80C51BH and retains the same functionality. The 83C152 is fabricated on the same CHMOS III process as the 80C51BH. What makes the 83C152 different is that it has added functions and peripherals to the basic 80C51BH architecture that are supported by new Special Function Registers (SFRs). These enhancements include: a high speed multi-protocol serial communication interface, two channels for DMA transfers, HOLD/HLDA bus control, a fifth I/O port, expanded data memory, and expanded program memory.

In addition to a standard UART, referred to here as Local Serial Channel (LSC), the 83C152 has an on-board multi-protocol communication controller called the Global Serial Channel (GSC). The GSC interface supports SDLC, CSMA/CD, user definable protocols, and a subset of HDLC protocols. The GSC capabilities include: address recognition, collision resolution, CRC generation, flag generation, automatic retransmission, and a hardware based acknowledge feature. This high speed serial channel is capable of implementing the Data Link Layer and the Physical Link Layer as shown in the OSI open systems communication model. This model can be found in the document "Reference Model for Open Systems Interconnection Architecture", ISO/TC97/SC16 N309.

The DMA circuitry consists of two 8-bit DMA channels with 16-bit addressability. The control signals; Read ($\overline{RD}$), Write ($\overline{WR}$), hold and hold acknowledge (HOLD/HLDA) are used to access external memory. The DMA channels are capable of addressing up to 64K bytes (16 bits). The destination or source address can be automatically incremented. The lower 8 bits of the address are multiplexed on the data bus Port 0 and the upper eight bits of address will be on Port 2. Data is transmitted over an 8-bit address/data bus. Up to 64K bytes of data may be transmitted for each DMA activation.

The new I/O port (P4) functions the same as Ports 1-3, found on the 80C51BH.

Internal memory has been doubled in the 83C152. Data memory has been expanded to 256 bytes, and internal program memory has been expanded to 8K bytes.

There are also some specific differences between the 83C152 and the 80C51BH. The first is that the numbering system between the 83C152 and the 80C51BH is slightly different. The 83C152 and the 80C51BH are factory masked ROM devices. The 80C152 and the 80C31BH are ROMless devices which require the use of external program memory. The second difference is that RESET is active low in the 83C152 and active high in the 80C51BH. This is very important to designers who may currently be using the 80C51BH and planning to use the 83C152, or are planning on using both devices on the same board. The third difference is that GF0 and GF1, general purpose flags in PCON, have been renamed GFIEN and XRCLK. GFIEN enables idle flags to be generated in SDLC mode, and XRCLK enables the receiver to be externally clocked. All of the previously unused bits are now being used and interrupt vectors have been added to support the new enhancements. Programmers using old code generated for the 80C51BH will have to examine their programs to ensure that new bits are properly loaded, and that the new interrupt vectors will not interfere with their program.

Throughout the rest of this manual the 80C152 and the 83C152 will be referred to generically as the "C152".

The C152 is based on the 80C51BH architecture and utilizes the same 80C51BH instruction set. Figure 1.1 is a block diagram of the C152. Readers are urged to compare this block diagram with the 80C51BH block diagram. There have been no new instructions added. All the new features and peripherals are supported by an extension of the Special Function Registers (SFRs). Very little of the information pertaining specifically to the 80C51BH core will be discussed in this chapter. The detailed information on such functions as: the instruction set, port operation, timer/counters, etc., can be found in the MCS®-51 Architecture chapter in the Intel Embedded Controller Handbook. Knowledge of the 80C51BH is required to fully understand this manual and the operation of the C152. To gain a basic understanding on the operation of the 80C51BH, the reader should familiarize himself with the entire MCS-51 chapter of the Embedded Controller Handbook.

Another source of information that the reader may find helpful is Intel's LAN Components User's Manual, order number 230814. Inside are descriptions of various protocols, application examples, and application notes dealing with different serial communication environments.

2.0 COMPARISON OF 80C152 AND 80C51BH FEATURES

2.1 Memory Space

A good understanding of the memory space and how it is used in the operation of MCS-51 products is essential. All the enhancements on the C152 are implemented by accessing Special Function Registers (SFRs), added data memory, or added program memory.

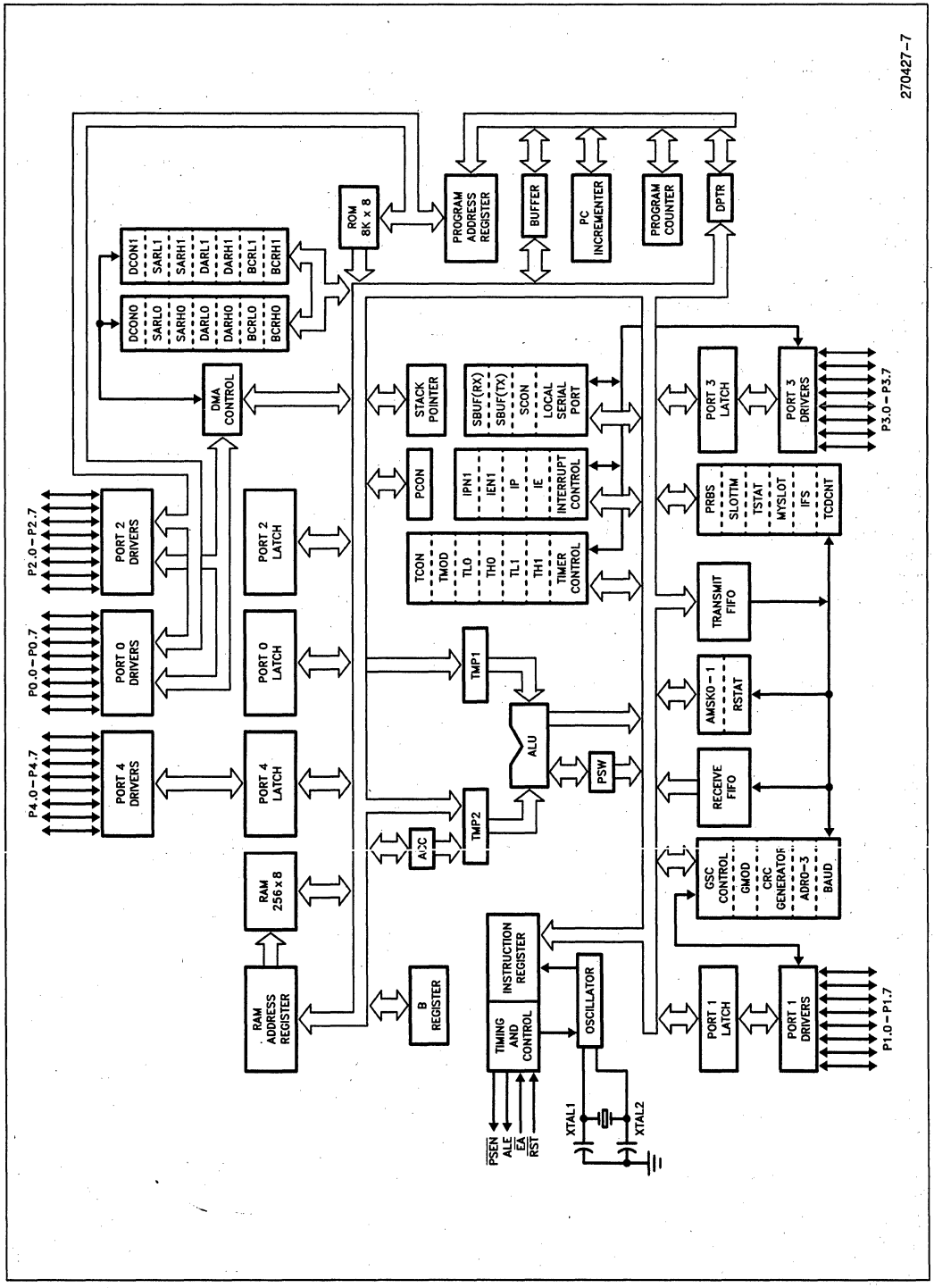


Figure 1.1. Block Diagram

2.1.1 SPECIAL FUNCTION REGISTERS (SFRs)

The following list contains all the SFRs, their names and function. All of the SFRs of the 80C51BH are retained and for a detailed explanation of their operation, please refer to the chapter, "Hardware Description of the 8051 and 8052" that is found in the Embedded Controller Handbook. An overview of the new SFRs is found in Section 2.1.1.1, with a detailed explanation in Section 3.7, Section 4.5, and 6.0.

2.1.1.1 New SFRs

The following descriptions are quick overviews of the new SFRs, and not intended to give a complete understanding of their use. The reader should refer to the detailed explanation in Section 3 for the GSC SFRs, and Section 4 for the DMA SFRs.

ADR 0,1,2,3 - (95H, 0A5H, 0B5H, 0C5H) Contains the four bytes for address matching during GSC operation.

AMSK0 - (0D5H) Selects "don't care" bits to be used with ADR0.

AMSK1 - (0E5H) Selects "don't care" bits to be used with ADR1.

BAUD - (94H) Contains the programmable value for the baud rate generator for the GSC. The baud rate will equal (fosc)/((BAUD+1) $\times$ 8).

BCRL0 - (0E2H) Contains the low byte of a count-down counter that determines when the DMA access for Channel 0 is complete.

BCRH0 - (0E3H) Contains the high byte for count-down counter for Channel 0.

BCRL1 - (0F2H) Same as BCRL0 except for DMA Channel 1.

BCRH1 - (0F3H) Same as BCRH0 except for DMA Channel 1.

BKOFF - (0C4H) An 8-bit count-down timer used with the CSMA/CD resolution algorithm.

DARL0 - (0C2H) Contains the low byte of the destination address for DMA Channel 0.

DARH0 - (0C3H) Contains the high byte of the destination address for DMA Channel 0.

DARL1 - (0D2H) Same as DARL0 except for DMA Channel 1.

DARH1 - (0D3H) Same as DARH0 except for DMA Channel 1.

DCON0 - (92H) Contains the Destination Address Space bit (DAS), Increment Destination Address bit (IDA), Source Address Space bit (SAS), Increment Source Address bit (ISA), DMA Channel Mode bit (DM), Transfer Mode bit (TM), DMA Done bit (DONE), and the GO bit (GO). DCON0 is used to control DMA Channel 0.

DCON1 - (93H) Same as DCON0 except this is for DMA Channel 1.

GMOD - (84H) Contains the Protocol bit (PR), the Preamble Length (PL1,0), CRC Type (CT), Address Length (AL), Mode select (M1,0), and External Transmit Clock (TXC). This register is used for GSC operation only.

IEN1 - (0C8H) Interrupt enable register for DMA and GSC interrupts.

IFS - (0A4H) Determines the number of bit times separating transmitted frames.

IPN1 - (0F8H) Interrupt priority register for DMA and GSC interrupts.

MYSLOT - (0F5H) Contains the Jamming mode bit (DCJ), the Deterministic Collision Resolution Algorithm bit (DCR), and the DCR slot address for the GSC.

P4 - (0C0H) Contains the memory "image" of Port 4.

PRBS - (0E4H) Contains a pseudo-random number to be used in CSMA/CD backoff algorithms. May be read or written to by user software.

RFIFO - (F4H) RFIFO is used to access a 3-byte FIFO that contains the receive data from the GSC.

RSTAT - (0E8H) Contains the Hardware Based Acknowledge Enable bit (HABEN), Global Receive Enable bit (GREN), Receive FIFO Not Empty bit (RFNE), Receive Done bit (RDN), CRC Error bit (CRCE), Alignment Error bit (AE), Receiver Collision/Abort detect bit (RCABT), and the Overrun bit (OVR), used with both DMA and GSC.

SARL0 - (0A2H) Contains the low byte of the source address for DMA transfers.

SARH0 - (0A3H) Contains the high byte of the source address for DMA transfers.

SARL1 - (0B2H) Same as SARL0 but for DMA Channel 1.

SARH1 - (0B3H) Same as SARH1 but for DMA Channel 1.

SLOTTM - (0B4H) Determines the length of the slot time in CSMA/CD.

TCDCNT - (0D4H) Contains the number of collisions in the current frame if using CSMA/CD GSC.

Old(O)/New(N)	Name	Addr	Function
O	A	0E0H	ACCUMULATOR
N	ADR0	095H	GSC MATCH ADDRESS 0
N	ADR1	0A5H	GSC MATCH ADDRESS 1
N	ADR2	0B5H	GSC MATCH ADDRESS 2
N	ADR3	0C5H	GSC MATCH ADDRESS 3
N	AMSK0	0D5H	GSC ADDRESS MASK 0
N	AMSK1	0E5H	GSC ADDRESS MASK 1
O	B	0F0H	B REGISTER
N	BAUD	094H	GSC BAUD RATE
N	BCRL0	0E2H	DMA BYTE COUNT 0 (LOW)
N	BCRH0	0E3H	DMA BYTE COUNT 0 (HIGH)
N	BCRL1	0F2H	DMA BYTE COUNT 1 (LOW)
N	BCRH1	0F3H	DMA BYTE COUNT 1 (HIGH)
N	BKOFF	0C4H	GSC BACKOFF TIMER
N	DARL0	0C2H	DMA DESTINATION ADDR 0 (LOW)
N	DARH0	0C3H	DMA DESTINATION ADDR 0 (HIGH)
N	DARL1	0D2H	DMA DESTINATION ADDR 1 (LOW)
N	DARH1	0D3H	DMA DESTINATION ADDR 1 (HIGH)
N	DCON0	092H	DMA CONTROL 0
N	DCON1	093H	DMA CONTROL 1
O	DPH	083H	DATA POINTER (HIGH)
O	DPL	082H	DATA POINTER (LOW)
N	GMOD	084H	GSC MODE
O	IE	0A8H	INTERRUPT ENABLE REGISTER 0
N	IEN1	0C8H	INTERRUPT ENABLE REGISTER 1
N	IFS	0A4H	GSC INTERFRAME SPACING
O	IP	0B8H	INTERRUPT PRIORITY REGISTER 0
N	IPN1	0F8H	INTERRUPT PRIORITY REGISTER 1
N	MYSLOT	0F5H	GSC SLOT ADDRESS
O	P0	080H	PORT 0
O	P1	090H	PORT 1
O	P2	0A0H	PORT 2
O	P3	0B0H	PORT 3
N	P4	0C0H	PORT 4
N	P5	091H	PORT 5
N	P6	0A1H	PORT 6
O	PCON	087H	POWER CONTROL
N	PRBS	0E4H	GSC PSEUDO RANDOM SEQUENCE
O	PSW	0D0H	PROGRAM STATUS WORD
N	RFIFO	0F4H	GSC RECEIVE BUFFER
N	RSTAT	0E8H	RECEIVE STATUS (DMA & GSC)
N	SARL0	0A2H	DMA SOURCE ADDR 0 (LOW)
N	SARH0	0A3H	DMA SOURCE ADDR 0 (HIGH)
N	SARL1	0B2H	DMA SOURCE ADDR 1 (LOW)
N	SARH1	0B3H	DMA SOURCE ADDR 1 (HIGH)
O	SBUF	099H	LOCAL SERIAL CHANNEL (LSC) BUFFER
O	SCON	098H	LOCAL SERIAL CHANNEL (LSC) CONTROL
N	SLOTTM	0B4H	GSC SLOT TIME
O	SP	081H	STACK POINTER
N	TCDCNT	0D4H	GSC TRANSMIT COLLISION COUNTER
O	TCON	088H	TIMER CONTROL
N	TFIFO	085H	GSC TRANSMIT BUFFER
O	TH0	08CH	TIMER 0 (HIGH)
O	TH1	08DH	TIMER 1 (HIGH)
O	TL0	08AH	TIMER 0 (LOW)
O	TL1	08BH	TIMER 1 (LOW)
O	TMOD	089H	TIMER MODE
N	TSTAT	0D8H	TRANSMIT STATUS (DMA & GSC)

TFIFO - (85H) TFIFO is used to access a 3-byte FIFO that contains the transmission data for the GSC.

TSTAT - (0D8H) Contains the DMA Service bit (DMA), Transmit Enable bit (TEN), Transmit FIFO Not Full bit (TFNF), Transmit Done bit (TDN), Transmit Collision Detect bit (TCDT), Underrun bit (UR), No Acknowledge bit (NOACK), and the Receive Data Line Idle bit (LNI). This register is used with both DMA and GSC.

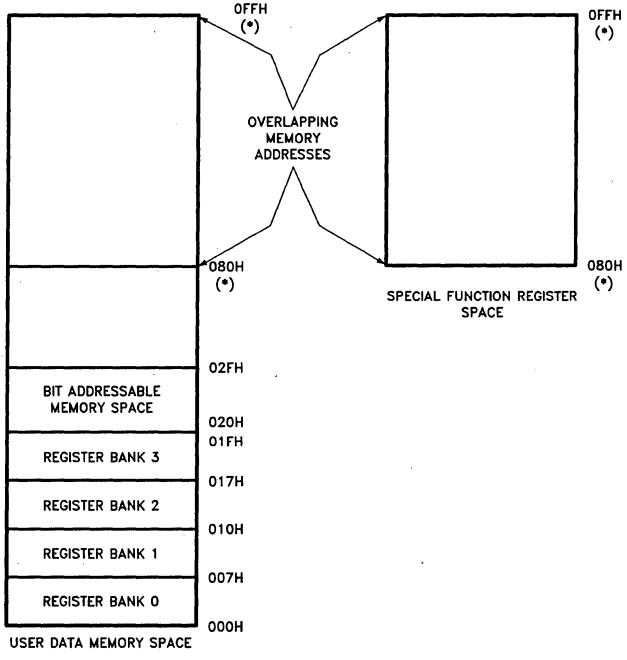
The general purpose flag bits (GF0 and GF1) that exist on the 80C51BH are no longer available on the C152. GF0 has been renamed GFIEN (GSC Flag Idle Enable) and is used to enable idle fill flags. Also GF1 has been renamed XRCLK (External Receive Clock Enable) and is used to enable the receiver to be clocked externally.

2.1.2 DATA MEMORY

Internal data memory consists of 256 bytes as shown in Figure 2.1. The first 128 bytes are addressed exactly like an 80C51BH, using direct addressing.

The addresses of the second 128 bytes of data memory happen to overlap the SFR addresses. The SFRs and their memory locations are shown in Figure 2.2. This means that internal data memory spaces have the same address as the SFR address. However, each type of memory is addressed differently. To access data memory above 80H, indirect addressing or the DMA channels must be used. To access the SFRs, direct addressing is used. When direct addressing is used, the address is the source or destination, e.g. MOV A, 10H, moves the contents of location 10H into the accumulator. When indirect addressing is used, the address of the destination or source exists within another register, e.g. MOV A, @R0. This instruction moves the contents of the memory location addressed by R0 into the accumulator. Directly addressing the locations 80H to 0FFH will access the SFRs. Another form of indirect addressing is with the use of Stack Pointer Operations. If the Stack Pointer contains an address and a PUSH or POP instruction is executed, indirect addressing is actually used. Directly accessing an unused SFR address will give undefined results.

Physically, there are separate SFR memory and data memory spaces allocated on the chip. Since there are separate spaces, the SFRs do not diminish the available data memory space.



***NOTE:**
User data memory above 80H must be addressed indirectly. Using direct addressing above 80H accesses the Special Function Registers.

Figure 2.1. Data Memory Map

External data memory is accessed like an 80C51BH, with "MOVX" instructions. Addresses up to 64K may be accessed when using the Data Pointer (DPTR). When accessing external data memory with the DPTR, the address appears on Port 0 and 2. When using the DPTR, if less than 64K of external data memory is used, the address is emitted on all sixteen pins. This means that when using the DPTR, the pins of Port 2 not used for addresses cannot be used for general purpose I/O. An alternative to using 16-bit addresses with the DPTR is to use R0 or R1 to address the external data memory. When using the registers to address external data memory, the address range is limited to 256 bytes. However, software manipulation of I/O Port 2 pins as normal I/O, allows this 256 bytes restriction to be expanded via bank switching. When using R0 or R1 as data pointers, Port 2 pins that are not used for addressing, can be used as general purpose I/O.

2.1.2.1 Bit Addressable Memory

The C152 has several memory spaces in which the bits are directly addressed by their location. The directly addressable bits and their symbolic names are shown in Figure 2.3A, 2.3B, and 2.3C.

Bit addresses 0 to 7FH reside in on-board user data RAM in byte addresses 20H to 2FH (see Figure 2.3A).

Bit addresses 80H to 0FFH reside in the SFR memory space, but not every SFR is bit addressable, see Figure 2.3B. The addressable bits are scattered throughout the SFRs. The addressable bits occur every eighth SFR address starting at 80H and occupy the entire byte. Most of the bits that are addressable in the SFRs have been given symbolic names. These names will often be referred to in this or other documentation on the C152. Most assemblers also allow the use of the symbolic names when writing in assembly language. These names are shown in Figure 2.3C.

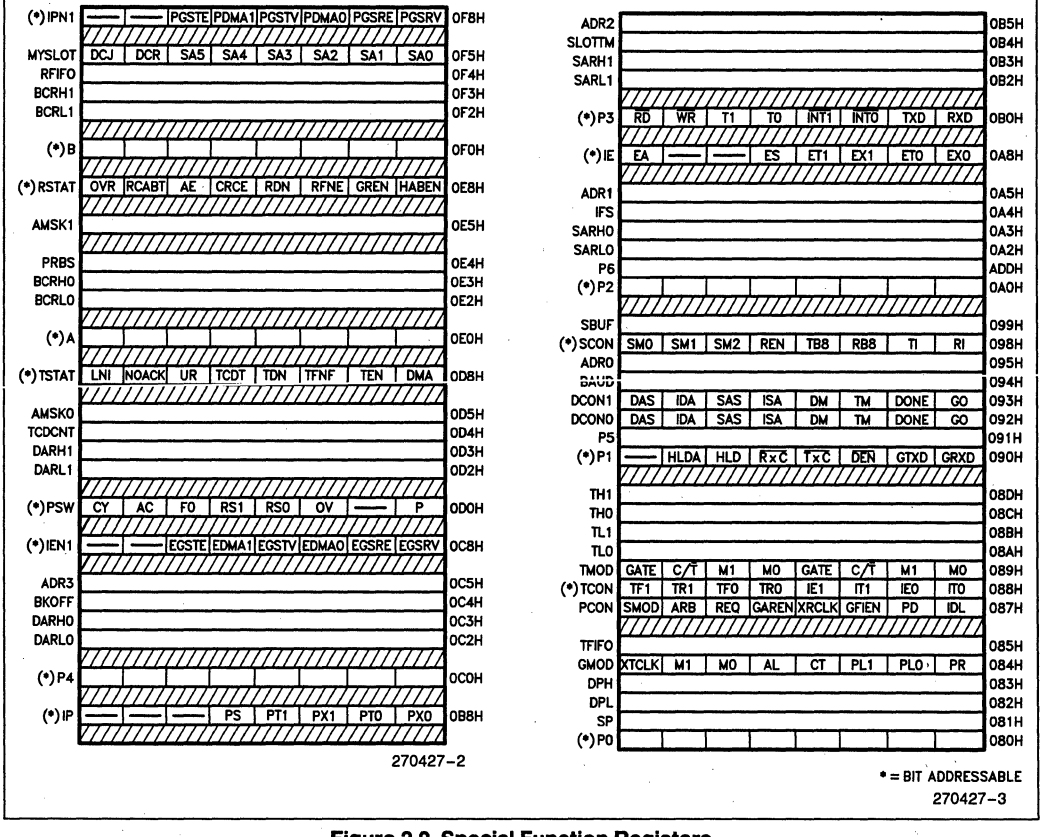


Figure 2.2. Special Function Registers

Data Memory Map (bits):

Byte Address	BIT ADDRESSES (MSB)							(LSB)
020H	07	06	05	04	03	02	01	00
021H	0F	0E	0D	0C	0B	0A	09	08
022H	17	16	15	14	13	12	11	10
023H	1F	1E	1D	1C	1B	1A	19	18
024H	27	26	25	24	23	22	21	20
025H	2F	2E	2D	2C	2B	2A	29	28
026H	37	36	35	34	33	32	31	30
027H	3F	3E	3D	3C	3B	3A	39	38
028H	47	46	45	44	43	42	41	40
029H	4F	4E	4D	4C	4B	4A	49	48
02AH	57	56	55	54	53	52	51	50
02BH	5F	5E	5D	5C	5B	5A	59	58
02CH	67	66	65	64	63	62	61	60
02DH	6F	6E	6D	6C	6B	6A	69	68
02EH	77	76	75	74	73	72	71	70
02FH	7F	7E	7D	7C	7B	7A	79	78

Figure 2.3A. Bit Addresses

Byte Address	BIT ADDRESSES (MSB)							(LSB)	
080H	87	86	85	84	83	82	81	80	(P0)
088H	8F	8E	8D	8C	8B	8A	89	88	(TCON)
090H	97	96	95	94	93	92	91	90	(P1)
098H	9F	9E	9D	9C	9B	9A	99	98	(SCON)
0A0H	A7	A6	A5	A4	A3	A2	A1	A0	(P2)
0A8H	AF	–	–	AC	AB	AA	A9	A8	(IE)
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	(P3)
0B8H	–	–	–	BC	BB	BA	B9	B8	(IP)
0C0H	C7	C6	C5	C4	C3	C2	C1	C0	(P4)
0C8H	–	–	CD	CC	CB	CA	C9	C8	(IEN1)
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	(PSW)
0D8H	DF	DE	DD	DC	DB	DA	D9	D8	(TSTAT)
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	(A)
0E8H	EF	EE	ED	EC	EB	EA	E9	E8	(RSTAT)
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	(B)
0F8H	–	–	FD	FC	FB	FA	F9	F8	(IPN1)

Figure 2.3B. Bit Addresses

Byte Address	SYMBOLIC NAME BIT MAP (MSB)							(LSB)	
080H	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	(P0)
088H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	(TCON)
090H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	(P1)
098H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	(SCON)
0A0H	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	(P2)
0A8H	EA	—	—	ES	ET1	EX1	ET0	EX0	(IE)
0B0H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	(P3)
0B8H	—	—	—	PS	PT1	PX1	PT0	PX0	(IP)
0C0H	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	(P4)
0C8H	—	—	EGSTE	EDMA1	EGSTV	EDMA0	EGSRE	EGSRV	(IEN1)
0D0H	CY	AC	F0	RS1	RS0	OV	—	P	(PSW)
0D8H	LNI	NOACK	UR	TCDT	TDN	TFNF	TEN	DMA	(TSTAT)
0E0H									(A)
0E8H	OVR	RCABT	AE	CRCE	RDN	RFNE	GREN	HABEN	(RSTAT)
0F0H									(B)
0F8H	—	—	PGSTE	PDMA1	PGSTV	PDMA0	PGSRE	PGSRV	(IPN1)

Figure 2.3C. Bit Addresses

2.1.3 PROGRAM MEMORY

The 83C152 contains 8K of ROM program memory, and the 80C152 uses only external program memory. Figure 2.4 shows the program memory locations and where they reside. The user is allowed a maximum of 64K of program memory. In the 83C152 program memory fetches beyond 8K automatically access external program memory. When program memory is externally addressed, all of the Port 2 pins emit the address. Since all of Port 2 is affected by the address, unused address pins cannot be used as normal I/O ports even if less than 64K of memory is being accessed.

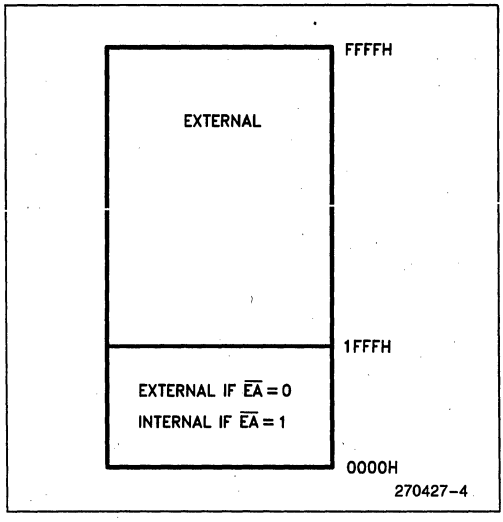


Figure 2.4. Program Memory

2.2 Interrupt Structure

The C152 retains all five interrupts of the 80C51BH. In addition, six new interrupts have been added for a total of 11 available interrupts. Two SFRs have been added to the C152 for control of the new interrupts. These added SFRs are IEN1 (C8H) for enabling the interrupts and IPN1 (F8H) for setting the priority. For an explanation on how the priority of interrupts affects their operation please refer to the MCS-51 Architecture and Hardware Chapters in the Intel Embedded Controller Handbook. A detailed description on how the interrupts function is in the MCS®-51 Architectural Overview.

IEN1 FUNCTIONS			
Symbol	**Position**	**Vector**	**Function**
—	IEN1.7		**RESERVED** and do not exist on chip.
—	IEN1.6		**RESERVED** and do not exist on chip.
EGSTE	IEN1.5	04BH	**GSC TRANSMIT ERROR**—If TSTAT.0 (DMA) is cleared, the interrupt service routine at 4BH is invoked when TSTAT.6 (NOACK) or TSTAT.4 (TCDT) is set and EGSTE is enabled. If TSTAT.0 (DMA) is set, the interrupt service routine will be invoked when the TSTAT.5 (UR) is set and EGSTE is enabled.
EDMA1	IEN1.4	053H	**DMA CHANNEL REQUEST 1**—The interrupt service routine at53H is invoked when DCON1.1 (DONE) is set and EDMA1 is enabled.
EGSTV	IEN1.3	043H	**GSC TRANSMIT VALID**—If TSTAT.0 (DMA) is cleared, the interrupt service routine at 43H is invoked when TSTAT.2 (TFNF) is set and EGSTV is enabled. If TSTAT.0 (DMA) is set, the interrupt service routine will be invoked when TSTAT.3 (TDN) is set and EGSTV is enabled.
EDMA0	IEN1.2	03BH	**DMA CHANNEL REQUEST 0**—The interrupt service routine at 3BH will be invoked when DCON0.1 (DONE) is set and EDMA0 is enabled.
EGSRE	IEN1.1	033H	**GSC RECEIVE ERROR**—The interrupt service routine at 33H will be invoked when RSTAT.4 (CRCE), RSTAT.7 (OVR), RSTAT.6 (RCABT), or RSTAT.5 (AE), is set and EGSRE is enabled. This functions the same whether or not TSTAT.0 (DMA) is set.
EGSRV	IEN1.0	02BH	**GSC RECEIVE VALID**—If TSTAT.0 (DMA) is cleared, the interrupt service routine at 2BH will be invoked when RSTAT.2 (RFNE) is set and EGSRV is enabled. If TSTAT.0 (DMA) is set, the interrupt service routine will be invoked when RSTAT.3 (RDN) is set and EGSRV is enabled.

IPN1 is used the same way the current 80C51BH interrupt priority register (IP) is. By assigning a "1" to the appropriate bit, that interrupt has a higher priority than an interrupt with a "0" assigned to it in the priority register.

The new interrupt priority register (IPN1) contents are:

Symbol	Position	Function
PGSTE	IPN1.5	GSC TRANSMIT ERROR
PDMA1	IPN1.4	DMA CHANNEL REQUEST 1
PGSTV	IPN1.3	GSC TRANSMIT VALID
PDMA0	IPN1.2	DMA CHANNEL REQUEST 0
PGSRE	IPN1.1	GSC RECEIVE ERROR
PGSRV	IPN1.0	GSC RECEIVE VALID

The eleven interrupts are sampled in the following order when assigned the same priority level in the IP and IPN1 registers:

Priority Sequence	Priority Symbolic Address	Priority Symbolic Name	Interrupt Symbolic Address	Interrupt Symbolic Name	Vector Address	
1	IP.0	PX0	IE.0	EX0	03H	(FIRST)
2	IPN1.0	PGSRV	IEN1.0	EGSRV	2BH	
3	IP.1	PT0	IE.1	ET0	0BH	
4	IPN1.1	PGSRE	IEN1.1	EGSRE	33H	
5	IPN1.2	PDMA0	IEN1.2	EDMA0	3BH	
6	IP.2	PX1	IE.2	EX1	13H	
7	IPN1.3	PGSTV	IEN1.3	EGSTV	43H	
8	IPN1.4	PDMA1	IEN1.4	EDMA1	53H	
9	IP.3	PT1	IE.3	ET1	1BH	
10	IPN1.5	PGSTE	IEN1.5	EGSTE	4BH	
11	IP.4	PS	IE.4	ES	23H	(LAST)

2.3 Reset

RESET performs the same operations in both the 80C51BH and the C152 and those conditions that exist at the end of a valid RESET are:

Register	Contents	Register	Contents
ACC	00H	P0–P6	0FFH
ADR0-3	00H	PCON	0XXX0000B
AMSK0	00H	PRBS	00H
AMSK1	00H	PSW	00H
B	00H	RFIFO	INDETERMINATE
BAUD	00H	RSTAT	00000000B
BCRH0	INDETERMINATE	SARH0	INDETERMINATE
BCRH1	INDETERMINATE	SARH1	INDETERMINATE
BCRL0	INDETERMINATE	SARL0	INDETERMINATE
CRL1	INDETERMINATE	SARL1	INDETERMINATE
BKOFF	INDETERMINATE	SBUF	INDETERMINATE
DARH0	INDETERMINATE	SCON	00H
DARH1	INDETERMINATE	SLOTTM	00H
DARL0	INDETERMINATE	SP	07H
DARL1	INDETERMINATE	TCDCNT	INDETERMINATE
DCON0	00H	TCON	00H
DCON1	00H	TFIFO	INDETERMINATE
DPTR	0000H	TH0	00H
GMOD	X0000000B	TH1	00H
IE	0XX00000B	TL0	00H
IEN1	XX000000B	TL1	00H
IFS	00H	TMOD	00H
IP	XXX00000B	TSTAT	XX000100B
IPN1	XX000000B	PC	0000H
MYSLOT	00000000B		

The same conditions apply for both the 80C51BH and C152 for a correct reset pulse or "power-on" reset except that Reset is active low on the C152. Please refer to the 8051/52 Hardware Description Chapter of the Intel Embedded Controller Handbook for an explanation on how to provide a proper power-on reset. Since Reset is active low on the C152, the resistor should be tied to VCC and the capacitor should be tied to VSS.

Because the clocking on part of the GSC circuitry is independent of the processor clock, data may still be transmitted and DEN active for some time after reset is applied. The transmission may continue for a maximum of four machine cycles after reset is first pulled low. Although Reset has to be held low for only three machine cycles to be recognized by the GSC hardware, all of the GSC circuitry may not be reset until four machine cycles have passed. If it is important in the user application that all transmission and DEN becomes inactive at the end of a reset, then Reset will have to be held low for a minimum of four machine cycles.

2.4 Ports 4, 5 and 6

Ports 4, 5 and 6 operation is identical to Ports 1-3 on the 80C51BH. The description of port operation can be found in the 8051/52 Hardware Description Chapter of the Intel Embedded Controller Handbook. Ports 5 and 6 exist only on the "JB" and "JD" version of the C152 and can either function as standard I/O ports or can be configured so that program memory fetches are performed with these two ports. To configure ports 5 and 6 as standard I/O ports, EBEN is tied to a logic low. When in this configuration, ports 5 and 6 operation is identical to that of port 4 except they are not bit addressable. To configure ports 5 and 6 to fetch program memory, EBEN is tied to a logic high. When using ports 5 and 6 to fetch the program memory, the signal EPSEN is used to enable the external memory device instead of PSEN. Regardless of which ports are used to fetch program memory, all data memory fetches occur over ports 0 and 2. The 80C152JB and 80C152JD are available as ROMless devices only. ALE is still used to latch the address in all configurations. Table 2.1 summarizes the control signals and how the ports may be used.

2.5 Timer/Counters

The 80C51BH and C152 have the same pair of 16-bit general purpose timer/counters. The user should refer to the Intel Embedded Controller Handbook which describes the timer/counters and their use. The user should bear in mind, when reading the Intel Embedded Controller Handbook that the C152 does not have the third event timer named Timer 2, which is in the 8052.

2.6 Package

The 83C152 is packaged in a 48 pin DIP and a 68 lead PLCC. This differs from the 40 pin DIP and 44 pin PLCC of the 80C51BH. The larger package is required to accommodate the extra 8 bit I/O port (P4). Figures 2.5A, 2.5B and 2.5C show the packages and the pin names.

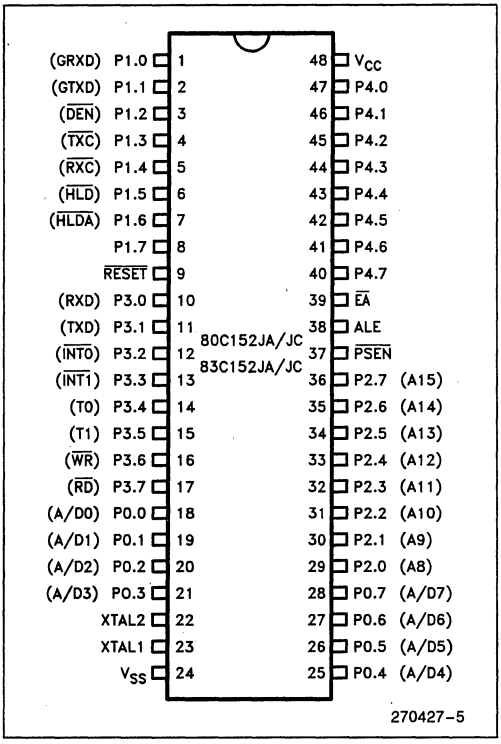


Figure 2.5A. DIP Pin Out

Table 2.1 Program Memory Fetches

EBEN	EA	Program Fetch via	PSEN	EPSEN	Comments
0	0	P0, P2	Active	Inactive	Addresses 0–0FFFFH
0	1	N/A	N/A	N/A	Invalid Combination
1	0	P5, P6	Inactive	Active	Addresses 0–0FFFFH
1	1	P5, P6 P0, P2	Inactive Active	Active Inactive	Addresses 0–1FFFH Addresses ≥ 2000H

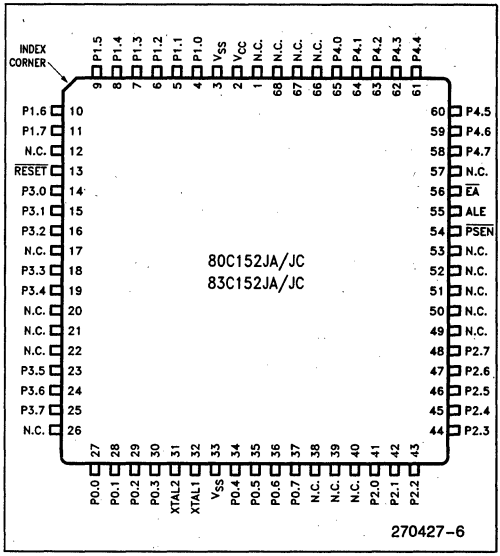


Figure 2.5B. PLCC Pin Out

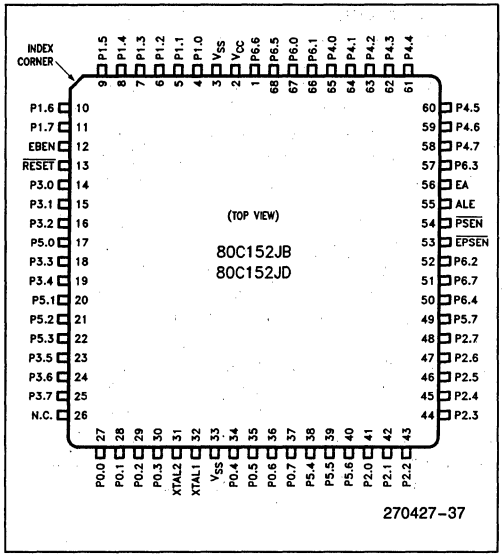


Figure 2.5C. PLCC Pin Out

2.7 Pin Description

The pin description for the 80C51BH also applies to the C152 and is listed below. Changes have been made to the descriptions as they apply to the C152.

PIN DESCRIPTION

Pin #		Description
DIP	PLCC(1)	
48	2	V_{CC}—Supply voltage.
24	3, 33(2)	**VSS**—Circuit ground.
18–21, 25–28	27–30, 34–37	**Port 0**—Port 0 is an 8-bit open drain bi-directional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program memory if EBEN is pulled low. During accesses to external Data Memory, Port 0 always emits the low-order address byte and serves as the multiplexed data bus. In these applications it uses strong internal pullups when emitting 1s. Port 0 also outputs the code bytes during program verification. External pullups are required during program verification.

NOTES:
1. N.C. pins on PLCC package may be connected to internal die and should not be used in customer applications.
2. It is recommended that both Pin 3 and Pin 33 be grounded for PLCC devices.

PIN DESCRIPTION (Continued)

<table>
<tr><th colspan="2">Pin #</th><th colspan="3" rowspan="2">Description</th></tr>
<tr><th>DIP</th><th>PLCC(1)</th></tr>
<tr><td>1–8</td><td>4–11</td><td colspan="3">Port 1—Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.
Port 1 also serves the functions of various special features of the 8XC152, as listed below:</td></tr>
<tr><td></td><td></td><td>Pin</td><td>Name</td><td>Alternate Function</td></tr>
<tr><td></td><td></td><td>P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6</td><td>GRXD
GTXD
$\overline{DEN}$
$\overline{TXC}$
$\overline{RXC}$
$\overline{HLD}$
$\overline{HLDA}$</td><td>GSC data input pin
GSC data output pin
GSC enable signal for an external driver
GSC input pin for external transmit clock
GSC input pin for external receive clock
DMA hold input/output
DMA hold acknowledge input/output</td></tr>
<tr><td>29–36</td><td>41–48</td><td colspan="3">Port 2—Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.
Port 2 emits the high-order address byte during fetches from external Program Memory if EBEN is pulled low. During accesses to external Data Memory that use 16-bit addresses (MOVX @ DPTR and DMA operations), Port 2 emits the high-order address byte. In these applications it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.
Port 2 also receives the high-order address bits during program verification.</td></tr>
<tr><td>10–17</td><td>14–16,
18, 19,
23–25</td><td colspan="3">Port 3—Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.
Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:</td></tr>
<tr><td></td><td></td><td>Pin</td><td>Name</td><td>Alternate Function</td></tr>
<tr><td></td><td></td><td>P3.0
P3.1
P3.2
P3.3
P3.4
P3.5
P3.6
P3.7</td><td>RXD
TXD
$\overline{INT0}$
$\overline{INT1}$
T0
T1
$\overline{WR}$
$\overline{RD}$</td><td>Serial input line
Serial output line
External interrupt 0
External interrupt 1
Timer 0 external input
Timer 1 external input
External Data Memory Write strobe
External Data Memory Read strobe</td></tr>
<tr><td>47–40</td><td>65–58</td><td colspan="3">Port 4—Port 4 is an 8-bit bi-directional I/O port with internal pullups. Port 4 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 4 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups. In addition, Port 4 also receives the low-order address bytes during program verification.</td></tr>
</table>

NOTES:
1. N.C. pins on PLCC package may be connected to internal die and should not be used in customer applications.
2. It is recommended that both Pin 3 and Pin 33 be grounded for PLCC devices.

PIN DESCRIPTION (Continued)

Pin #		Description
DIP	PLCC(1)	
9	13	**RST**—Reset input. A logic low on this pin for three machine cycles while the oscillator is running resets the device. An internal pullup resistor permits a power-on reset to be generated using only an external capacitor to V_{SS}. Although the GSC recognizes the reset after three machine cycles, data may continue to be transmitted for up to 4 machine cycles after Reset is first applied.
38	55	**ALE**—Address Latch Enable output signal for latching the low byte of the address during accesses to external memory. In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory. While in Reset, ALE remains at a constant high level.
37	54	**PSEN**—Program Store Enable is the Read strobe to External Program Memory. When the 8XC152 is executing from external program memory, PSEN is active (low). When the device is executing code from External Program Memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to External Data Memory. While in Reset, PSEN remains at a constant high level.
39	56	**EA**—External Access enable. EA must be externally pulled low in order to enable the 8XC152 to fetch code from External Program Memory locations 0000H to 0FFFH. EA must be connected to V_{CC} for internal program execution.
23	32	**XTAL1**—Input to the inverting oscillator amplifier and input to the internal clock generating circuits.
22	31	**XTAL2**—Output from the oscillator amplifier.
N/A	17, 20 21, 22 38, 39 40, 49	**Port 5**—Port 5 is an 8-bit bi-directional I/O port with internal pullups. Port 5 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 5 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups. Port 5 is also the multiplexed low-order address and data bus during accesses to external program memory if EBEN is pulled high. In this application it uses strong pullups when emitting 1s.
N/A	67, 66 52, 57 50, 68 1, 51	**Port 6**—Port 6 is an 8-bit bi-directional I/O port with internal pullups. Port 6 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 6 pins that are externally pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups. Port 6 emits the high-order address byte during fetches from external Program Memory if EBEN is pulled high. In this application it uses strong pullups when emitting 1s.
N/A	12	**EBEN**—E-Bus Enable input that designates whether program memory fetches take place via Ports 0 and 2 or Ports 5 and 6. Table 2.1 shows how the ports are used in conjunction with EBEN.
	53	**EPSEN**—E-bus Program Store Enable is the Read strobe to external program memory when EBEN is high. Table 2.1 shows when EPSEN is used relative to PSEN depending on the status of EBEN and EA.

NOTES:
1. N.C. pins on PLCC package may be connected to internal die and should not be used in customer applications.
2. It is recommended that both Pin 3 and Pin 33 be grounded for PLCC devices.

2.8 Power Down and Idle

Both of these operations function identically as in the 80C51BH. Application Note 252, "Designing with the 80C51BH" gives an excellent explanation on the use of the reduced power consumption modes. Some of the items not covered in AP-252 are the considerations that are applicable when using the GSC or DMA in conjunction with the power saving modes.

The GSC continues to operate normally in Idle as long as the interrupts are enabled. The interrupts need to be enabled, so that the CPU can service the FIFO's and terminate transmission or reception when appropriate. After servicing the GSC, user software will need to again invoke the Idle command as the CPU does not automatically re-enter the Idle mode after servicing the interrupts.

The GSC does not operate while in Power Down so the steps required prior to entering Power Down become more complicated. The sequence when entering Power Down and the status of the I/O is of major importance in preventing damage to the C152 or other components in the system. Since the only way to exit Power Down is with a Reset, several problem areas become very significant. Some of the problems that merit careful consideration are cases where the Power Down occurs during the middle of a transmission, and the possibility that other stations are not or cannot enter this same mode. The state of the GSC I/O pins becomes critical and the GSC status will need to be saved before power down is entered. There will also need to be some method of identifying to the CPU that the following Reset is probably not a cold start and that other stations on the link may have already been initialized.

The DMA circuitry stops operation in both Idle and Power Down modes. Since operation is stopped in both modes, the process should be similar in each case. Specific steps that need to be taken include: notification to other devices that DMA operation is about to cease for a particular station or network, proper withdrawal from DMA operation, and saving the status of the DMA channels. Again, the status of the I/O pins during Power Down needs careful consideration to avoid damage to the C152 or other components.

Port 4 returns to its input state, which is high level using weak pullup devices.

2.9 Local Serial Channel

The Local Serial Channel (LSC) is the name given to the UART that exists on all MCS-51 devices. The LSC's function and operation is exactly the same as on the 80C51BH. For a description on the use of the LSC, refer to the 8051/52 Hardware Description Chapter in the Intel Embedded Controller Handbook, under Serial Interface.

3.0 GLOBAL SERIAL CHANNEL

3.1 Introduction

The Global Serial Channel (GSC) is a multi-protocol, high performance serial interface targeted for data rates up to 2 MBPS with on-chip clock recovery, and 2.4 MBPS using the external clock options. In applications using the serial channel, the GSC implements the Data Link Layer and Physical Link Layer as described in the ISO reference model for open systems interconnection.

The GSC is designed to meet the requirements of a wide range of serial communications applications and is optimized to implement Carrier-Sense Multi-Access with Collision Detection (CSMA/CD) and Synchronous Data Link Control (SDLC) protocols. The GSC architecture is also designed to provide flexibility in defining non-standard protocols. This provides the ability to retrofit new products into older serial technologies, as well as the development of proprietary interconnect schemes for serial backplane environments.

The versatility of the GSC is demonstrated by the wide range of choices available to the user. The various modes of operation are summarized in Table 3.1. In subsequent sections, each available choice of operation will be explained in detail.

In using Table 3.1, the parameters listed vertically (on the left hand side) represent an option that is selected (X). The parameters listed horizontally (along the top of the table) are all the parameters that could theoretically be selected (Y). The symbol at the junction of both X and Y determines the applicability of the option Y.

Note, that not all combinations are backwards compatible. For example, Manchester encoding requires half duplex, but half duplex does not require Manchester encoding.

Table 3.1

N = NOT AVAILABLE M = MANDATORY O = OPTIONAL P = NORMALLY PREFERRED X = N/A	DATA ENCODING			FLAGS		CRC			DU-PLEX		ACKNOW-LEDGE			ADDRESS RECOG-NITION			BACKOFF			PRE-AMBLE	
	MANCHESTER	NRZ	NRZI	01111110	11/IDLE	NONE	16BITCCIT	32BITAUTO	HALF	FULL	NONE	HARDWARE	USERDEFINED	NONE/ALL	8BIT	16BIT	NORMAL	ALTERNATE	DETERMINISTIC	NONE	8BIT
DATA ENCODING:																					
MANCHESTER(CSMA/CD)	X	N	N	1	P	1	O	O	M	N	O	O	O	O	O	O	O	O	O	N	O
NRZI (SDLC)	N	X	N	P	1	1	O	O	O	O	O	N	P	O	O	O	N	N	N	O	O
NRZ (EXT CLK)	N	N	X	O	O	1	O	O	O	O	O	N	O	O	O	O	O	O	O	O	O
FLAGS:01111110 (SDLC)	N	P	O	X	1	1	O	O	O	O	O	N	P	O	O	O	N	N	N	O	O
11/IDLE	P	N	O	1	X	1	O	O	O	N	O	O	O	O	O	O	O	O	O	1	O
CRC:NONE	1	1	1	1	1	X	N	N	1	N	1	1	1	1	1	1	N	N	N	1	1
16-BIT CCITT	O	O	O	O	O	N	X	N	O	O	O	O	O	O	O	O	O	O	O	O	O
32-BIT AUTODIN II	O	O	O	O	O	N	N	X	O	N	O	O	O	O	O	O	O	O	O	O	O
DUPLEX:HALF	O	O	O	O	O	1	O	O	X	N	O	O	O	O	O	O	O	O	O	O	O
FULL	N	O	O	M	N	N	M	N	N	X	O	N	P	O	O	O	N	N	N	O	O
ACKNOWLEDGEMENT:NONE	O	O	O	O	O	1	O	O	O	O	X	N	N	O	O	O	O	O	O	O	O
HARDWARE	O	N	N	N	O	1	O	O	O	N	N	X	N	O	O	O	N	O	O	N	O
USER DEFINED	O	P	O	O	O	1	O	O	O	P	N	N	X	O	O	O	O	O	O	O	O
ADDRESS RECOGNITION:																					
NONE/ALL	O	O	O	O	O	1	O	O	O	O	O	O	O	X	N	N	O	O	O	O	O
8-BIT	O	O	O	O	O	1	O	O	O	O	O	O	O	N	X	N	O	O	O	O	O
16-BIT	O	O	O	O	O	1	O	O	O	O	O	O	O	N	N	X	O	O	O	O	O
COLLISION RESOLUTION:																					
NORMAL	O	N	O	N	O	N	O	O	M	N	O	O	O	O	O	O	X	N	N	N	O
ALTERNATE	O	N	O	N	O	N	O	O	M	N	O	O	O	O	O	O	N	X	N	N	O
DETERMINISTIC	O	N	O	N	O	N	O	O	M	N	O	O	O	O	O	O	N	N	X	N	O
PREAMBLE:NONE	N	O	O	O	1	1	O	O	O	O	O	N	O	O	O	O	N	N	N	X	N
8-BIT	O	O	O	O	O	1	O	O	O	O	O	O	O	O	O	O	O	O	O	N	X
32-BIT	O	O	O	O	O	1	O	O	O	O	O	O	O	O	O	O	O	O	O	N	N
64-BIT	O	O	O	O	O	1	O	O	O	O	O	O	O	O	O	O	O	O	O	N	N
JAM:D.C.	M	N	N	N	O	N	O	O	M	N	O	O	O	O	O	O	O	O	O	N	O
$\overline{CRC}$	M	N	N	N	O	N	O	O	M	N	O	O	O	O	O	O	O	O	O	N	O
CLOCKING:EXTERNAL	N	M	N	O	O	N	O	O	O	O	O	N	O	O	O	O	N	N	N	O	O
INTERNAL	O	O	N	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
CONTROL: CPU	O	O	O	O	O	1	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
DMA	O	O	O	O	O	1	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
RAW RECEIVE:	1	1	1	1	1	1	1	1	1	N	1	1	1	1	1	1	O	O	O	1	1
RAW TRANSMIT:	1	1	1	1	1	1	1	1	1	N	1	1	1	1	1	1	N	N	N	1	1
CSMA/CD:	O	N	2	1	P	1	O	O	M	N	O	O	O	O	O	O	O	O	O	N	O
SDLC:	N	O	O	P	1	1	O	O	O	O	O	N	O	O	O	O	N	N	N	P	O

Table 3.1 (Continued)

N = NOT AVAILABLE M = MANDATORY O = OPTIONAL P = NORMALLY PREFERRED X = N/A	PRE-AMBLE		JAM		CLOCK		CONTROL					
	32 BIT	64 BIT	DC	CRC/	EXTERNAL	INTERNAL	CPU	DMA	RAW RECEIVE	RAW TRANSMIT	CSMA/CD	SDLC
DATA ENCODING:												
MANCHESTER	O	O	O	O	N	M	O	O	O	O	M	N
NRZI	O	O	N	N	N	M	O	O	O	O	N	M
NRZ	O	O	O	O	M	N	O	O	O	O	O	O
FLAGS:01111110	O	O	N	N	O	O	O	O	O	1	1	P
11/IDLE	O	O	O	O	O	O	O	O	O	1	P	1
CRC:NONE	1	1	N	N	1	1	1	1	1	1	1	1
16-BIT CCITT	O	O	O	O	O	O	O	O	1	1	O	O
32-BIT AUTODIN II	O	O	O	O	O	O	O	O	1	1	O	O
DUPLEX:HALF	O	O	O	O	O	O	O	O	O	O	O	O
FULL	O	O	N	N	O	O	O	O	N	N	N	P
ACKNOWLEDGEMENT:NONE	O	O	O	O	O	O	O	O	O	O	O	O
HARDWARE	O	O	O	O	N	O	O	O	N	N	O	N
USER DEFINED	O	O	O	O	O	O	O	O	O	O	O	1
ADDRESS RECOGNITION:												
NONE	O	O	O	O	O	O	O	O	O	O	O	O
8-BIT	O	O	O	O	O	O	O	O	1	1	O	O
16-BIT	O	O	O	O	O	O	O	O	1	1	O	O
COLLISION RESOLUTION:												
NORMAL	O	O	O	O	N	O	O	O	O	N	M	N
ALTERNATE	O	O	O	O	N	O	O	O	O	N	M	N
DETERMINISTIC	O	O	O	O	N	O	O	O	O	N	M	N
PREAMBLE:NONE	N	N	N	N	O	O	O	O	O	O	N	P
8-BIT	N	N	O	O	O	O	O	O	1	1	O	O
32-BIT	X	N	O	O	O	O	O	O	1	1	O	O
64-BIT	N	X	O	O	O	O	O	O	1	1	O	O
JAM:D.C.	O	O	X	N	2	O	O	O	O	N	M	N
$\overline{\text{CRC}}$	O	O	N	X	2	O	O	O	O	N	M	N
CLOCKING:EXTERNAL	O	O	N	N	X	N	O	O	O	O	2	O
INTERNAL	O	O	O	O	N	X	O	O	O	O	O	O
CONTROL:CPU	O	O	O	O	O	O	X	N	O	O	O	O
DMA	O	O	O	O	O	O	N	X	O	O	O	O
RAW RECEIVE:	1	1	O	O	1	1	1	1	X	N	1	1
RAW TRANSMIT:	1	1	N	N	1	1	1	1	N	X	1	1
CSMA/CD:	O	O	O	O	2	O	O	O	O	O	X	N
SDLC:	O	O	N	N	O	O	O	O	O	O	N	X

Note 1: Programmable in Raw transmit or receive mode.

Note 2: When CSMA/CD is enabled, an external clock can be used on the transmitter, but not the receiver. Since the receiver monitors the link for Manchester violations, external hardware would be required to reformat the data from NRZ to Manchester on the transmitter. These hardware requirements go beyond the expectations of this table for implementation. For that reason it was assumed that the external clock cannot be used at all with CSMA/CD protocol, although it is actually possible to do so.

Almost all the options available from Table 3.1 can be implemented with the proper software to perform the functions that are necessary for the options selected. In Table 3.1, a judgment has been made by the authors on which options are practical and which are not. What this means is that in Table 3.1, an "N" should be interpreted as meaning that the option is either not practical when implemented with user software or that it cannot be done. An "O" is used when that function is one of several that can be implemented with the GSC without additional user software.

The GSC is targeted to operate at bit rates up to 2.4 MBps using the external clock options and up to 2 MBps using the internal baud rate generator, internal data formatting and on-chip clock recovery. The baud rate generator allows most standard rates to be achieved. These standards include the proposed IEEE802.3 LAN standard (1.0MBps) and the T1 standard (1.544MBps). The baud rate is derived from the crystal frequency. This makes crystal selection important when determining the frequency and accuracy of the baud rate.

3.2 CSMA/CD Operation

3.2.1 CSMA/CD OVERVIEW

CSMA/CD operates by sensing the transmission line for a carrier, which indicates link activity. At the end of link activity, a station must wait a period of time, called the deference period, before transmission may begin. The deference period is also known as the interframe space. The interframe space is explained in Section 3.2.3.

With this type of operation, there is always the possibility of a collision occurring after the deference period due to line delays. If a collision is detected after transmission is started, a jamming mechanism is used to ensure that all stations monitoring the line are aware of the collision. A resolution algorithm is then executed to resolve the contention. There are three different modes of collision resolution made available to the user on the C152. Re-transmission is attempted when a resolution algorithm indicates that a station's opportunity has arrived.

Normally, in CSMA/CD, re-transmission slot assignments are intended to be random. This method gives all stations an equal opportunity to utilize the serial communication link but also leaves the possibility of another collision due to two stations having the same slot assignment. There is an option on the C152 which allows all the stations to have their slot assignments previously determined by user software. This pre-assignment of slots is called the deterministic resolution mode. This method allows resolution after the first collision and ensures the access of the link to each station during the resolution. Deterministic resolution can be advantageous when the link is being heavily used and collisions are frequently occurring and in real time applications where determinism is required. Deterministic resolution may also be desirable if it is known beforehand that a certain station's communication needs to be prioritized over those of other stations if it is involved in a collision.

3.2.2 CSMA/CD FRAME FORMAT

The frame format in CSMA/CD consists of a preamble, Beginning of Frame flag (BOF), address field, information field, CRC, and End of Frame flag (EOF) as shown in Figure 3.1.

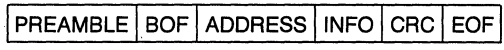


Figure 3.1 Typical CSMA/CD Frame

PREAMBLE - The preamble is a series of alternating 1s and 0s. The length of the preamble is programmable to be 0, 8, 32, or 64 bits. The purpose of the preamble is to allow all the receivers to synchronize to the same clock edges and identifies to the other stations on-line that there is activity indicating the link is being used. For these reasons zero preamble length is not compatible with standard CSMA/CD, protocols. When using CSMA/CD, the BOF is considered part of the preamble compared to SDLC, where the BOF is not part of the preamble. This means that if zero preamble length were to be used in CSMA/CD mode, no BOF would be generated. It is strongly recommended that zero preamble length never be used in CSMA/CD mode. If the preamble contains two consecutive 0s, the preamble is considered invalid. If the C152 detects an invalid preamble, the frame is ignored.

BOF - In CSMA/CD the Beginning-Of-Frame is a part of the preamble and consists of two sequential 1s. The purpose of the BOF is to identify the end of the preamble and indicate to the receiver(s) that the address will immediately follow.

ADDRESS - The address field is used to identify which messages are intended for which stations. The user must assign addresses to each destination and source. How the addresses are assigned, how they are maintained, and how each transmitter is made aware of which addresses are available is an issue that is left to the user. Some suggestions are discussed in Section 3.5.5. Generally, each address is unique to each station but there are special cases where this is not true. In these special cases, a message is intended for more than one station. These multi-targeted messages are called broadcast or multicast-group addresses. A broadcast address consisting of all 1s will always be received by all stations. A multicast-group address usually is indicated by using a 1as the first address bit. The user can choose to mask off all or selective bits of the address so that the GSC receives all messages or multicast-group messages. The address length is programmable to be 8 or 16 bits. An address consisting of all 1s will always be received by the GSC on the C152. The address bits are always passed from the GSC to the CPU. With user software, the address can be extended beyond 16 bits, but the automatic address recognition will only work on a maximum of 16 bits. User software will have to resolve any remaining address bits.

INFO - This is the information field and contains the data that one device on the link wishes to transmit to another device. It can be of any length the user wishes but needs to be in multiples of 8 bits. This is because multiples of 8 bits are used to transfer data into or out of the GSC FIFOs. The information field is delineated from the rest of the components of the frame by the preceding address field and the following CRC. The receiver determines the position of the end of the information field by passing the bytes through a temporary storage space. When the EOF is received the bytes in temporary storage are the CRC, and the last bit received previous to the CRC constitute the end of the information field.

CRC - The Cyclic Redundancy Check (CRC) is an error checking algorithm commonly used in serial communications. The C152 offers two types of CRC algorithms, a 16-bit and a 32-bit. The 16-bit algorithm is normally used in the SDLC mode and will be described in the SDLC section. In CSMA/CD applications either algorithm can be used but IEEE 802.3 uses a 32-bit CRC. The generation polynomial the C152 uses with the 32-bit CRC is:

$$G(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

The CRC generator, as shown in Figure 3.2, operates by taking each bit as it is received and XOR'ing it with bit 31 of the current CRC. This result is then placed in temporary storage. The result of XOR'ing bit 31 with the received bit is then XOR'd with bits 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22, 25 as the CRC is shifted right one position. When the CRC is shifted right, the temporary storage space holding the result of XOR'ing bit 31 and the incoming bit is shifted into position 0. The whole process is then repeated with the next incoming or outgoing bit.

The user has no access to the CRC generator or the bits which constitute the CRC while in CSMA/CD. On transmission, the CRC is automatically appended to the data being sent, and on reception, the CRC bits are not normally loaded into the receive FIFO. Instead, they are automatically stripped. The only indication the user has for the status of the CRC is a pass/fail flag. The pass/fail flag only operates during reception. A CRC is considered as passing when the the CRC generator has 11000111 00000100 11011010 01111011B as a remainder after all of the data, including the CRC checksum, from the transmitting station has been cycled through the CRC generator. The preamble, BOF and EOF are not included as part of the CRC algorithm. An interrupt is available that will interrupt the CPU if the CRC of the receiver is invalid. The user can enable the CRC to be passed to the CPU by placing the receiver in the raw receive mode.

This method of calculating the CRC is compatible with IEEE 802.3.

EOF - The End Of Frame indicates when the transmission is completed. The end flag in CSMA/CD consists of an idle condition. An idle condition is assumed when there is no transitions and the link remains high for 2 or more bit times.

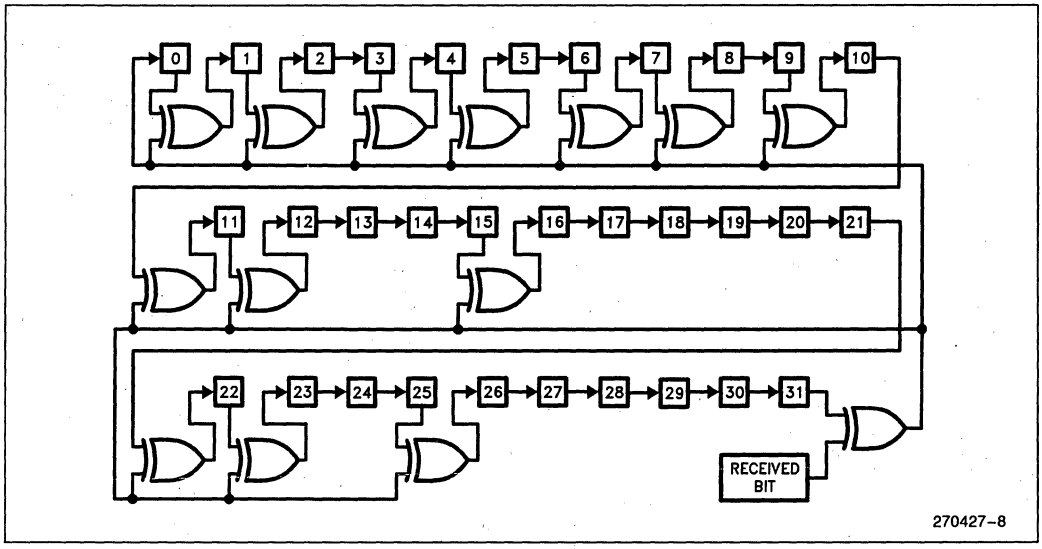


Figure 3.2. CRC Generator

3.2.3 INTERFRAME SPACE

The interframe space is the amount of time that transmission is delayed after the link is sensed as being idle and is used to separate transmitted frames. In alternate backoff mode, the interframe space may also be included in the determination of when retransmissions may actually begin. The C152 allows programmable interframe spaces of even numbers of bit times from 2 to 256.

The period of the interframe space is determined by the contents of IFS. IFS is an SFR that is programmable from 0 to 254. The interframe space is measured in bit times. The value in IFS multiplied by the bit time equals the interframe space unless IFS equals 0. If IFS does equal 0, then the interframe space will equal 256 bit times. One of the considerations when loading the IFS is that only even numbers (LSB must be 0) can be used because only the 7 most significant bits are loaded into IFS. The LSB is controlled by the GSC and determines which half of the IFS is currently being used. In some modes, the interframe space timer is re-triggered if activity is detected during the first half of the period. The GSC determines which half of the interframe space is currently being used by examining the LSB. A one indicates the first half and zero indicates the second half of the IFS.

After reset IFS is 0, which delays the first transmission for both SDLC and CSMA/CD by 256 bit times (after reset, a bit time equals 8 oscillator clock periods).

In most applications, the period of the interframe space will be equal to or greater than the amount of time needed to turn-around the received frame. The turn-around period is the amount of time that is needed by user software to complete the handling of a received frame and be prepared to receive the next frame. An interframe space smaller than the required turn-around period could be used, but would allow some frames to be missed.

When a GSC transmitter has a new message to send, it will first sense the link. If activity is detected, transmission will be deferred to allow the frame in progress to complete. When link activity ceases, the station continues deferring for one interframe space period.

As mentioned earlier, the interframe space is used during the collision resolution period as well as during normal transmission. The backoff method selected affects how the deference period is handled during normal transmission. If normal backoff mode is selected, the interframe space timer is reset if activity occurs during approximately the first half of the interframe space. If alternate backoff or deterministic backoff is selected, the timer is not reset. In all cases when the interframe space timer expires, transmission may begin, regardless if there is activity on the link or not. Although the C152 resets the interframe space timer if activity is detected during the first one-half of the interframe space, this is not necessarily true of all CSMA/CD systems. (IEEE 802.3 recommends that the interframe space be reset if activity is detected during the first two-thirds or less of the interframe space.)

3.2.4 CSMA/CD DATA ENCODING

Manchester encoding/decoding is automatically selected when the user software selects CSMA/CD transmission mode (See Figure 3.3). In Manchester encoding the value of the bit is determined by the transition in the middle of the bit time, a positive transition is decoded as a 1 and a negative transition is decoded as a 0.

If the external 1X clock feature is chosen the transmission mode is always NRZ (see Section 3.5.11). Using CSMA/CD with the external clock option is not supported because the data needs reformatting from NRZ to Manchester for the receiver to be able to detect code violations and collisions.

3.2.5 COLLISION DETECTION

The GSC hardware detects collisions by detecting Manchester waveform violations at its GRXD pin. Three kinds of waveform violations are detected: a missing 0-to-1 transition where one was expected, a 1-to-0 transition where none was expected, and a waveform that stays low (or high) for too short a time.

Jitter Tolerance

A valid Manchester waveform must have a transition at the midpoint of any bit cell, and may have a transition at the edge of any bit cell. Therefore, transitions will nominally be separated by either 1/2 bit-time or 1 bit-time.

The GSC samples the GRXD pin at the rate of 8 x the bit rate. The sequence of samples for the received bit sequence 001 would nominally be:

```
samples: 11110000:11110000:00001111:
bit value:      0    :    0    :    1    :
        :<-bit cell->:<-bit cell->:<-bit cell->:
```

The sampling system allows a jitter tolerance of ±1 sample for transitions that are 1/2 bit-time apart, and ±2 samples for transitions that are 1 bit-time apart.

Narrow Pulses

A valid Manchester waveform must stay high or low for at least a half bit-time, nominally 4 sample-times. Jitter tolerance allows a waveform which stays high or low for 3 sample-times to also be considered valid. A sample sequence which shows a second transition only 1 or 2 sample-times after the previous transition is considered to be the result of a collision. Thus, sample sequences such as 0000110000 and 111101111 are interpreted as collisions.

The GSC hardware recognizes the collision to have occurred within 3/8 to 1/2 bit-time following the second transition.

Missing 0-to-1 Transition

A 0-to-1 transition is expected to occur at the center of any bit cell that begins with 0. If the previous 1-to-0 transition occurred at the bit cell edge, a jitter tolerance of ±1 sample is allowed. Sample sequences such as 1111:00001111 and 1111:000001111 are valid, where ":" indicates a bit cell edge. Sequences of the form 1111:000000XXX are interpreted as collisions.

For these kinds of sequences, the GSC recognizes the collision to have occurred within 1 to 1 1/8 bit-times after the previous 1-to-0 transition.

If the previous 1-to-0 transition occurred at the center of the previous bit cell, a jitter tolerance of ±2 samples is allowed. Thus, sample sequences such as 11110000:00001111 and 111100000:000001111 are valid. Sequences of the form 111100000:000000XXX are interpreted as collisions.

For these kinds of sequences, the GSC recognizes the collision to have occurred within 1 5/8 to 1 3/4 bit-times after the previous 1-to-0 transition.

Unexpected 1-to-0 Transition

If the line is at a logic 1 during the first half of a bit cell, then it is expected to make a 1-to-0 transition at the midpoint of the bit cell. If the transition is missed, it is assumed that this bit cell is the first half of an EOF flag

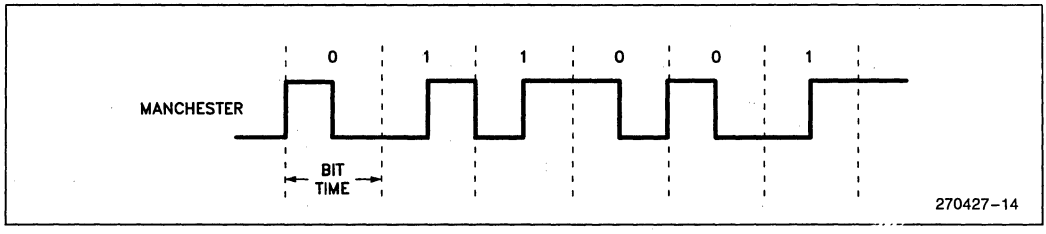


Figure 3.3. Manchester Encoding

(line idle for two bit-times). One bit-time later (which marks the midpoint of the next bit cell), if there is still no 1-to-0 transition, a valid EOF is assumed and the line idle bit (LNI in TSTAT) gets set.

However, if the assumed EOF flag is interrupted by a 1-to-0 transition in the bit-time following the first missing transition, a collision is assumed. In that case the GSC hardware recognizes the collision to have occurred within 1/2 to 5/8 bit-time after the unexpected transition.

3.2.6 RESOLUTION OF COLLISIONS

How the GSC responds to a detected collision depends on what it was doing at the time the collision was detected. What it might be doing is either transmitting or receiving a frame, or it might be inactive.

GSC Inactive

The collision is detected whether the GSC is active or not. If the GSC is neither transmitting nor receiving at the time the collision is detected, it takes no action unless user software has selected the Deterministic Collision Resolution (DCR) algorithm. If DCR has been selected, the GSC will participate in the resolution algorithm.

GSC Receiving

If the GSC is already in the process of receiving a frame at the time the collision is detected, its response depends on whether the first byte of the frame has been transferred into RFIFO yet or not. If that hasn't occurred, the GSC simply aborts the reception, but takes no other action unless DCR has been selected. If DCR has been selected, the GSC participates in the resolution algorithm.

If the reception has already progressed to the point where a byte has been transferred to RFIFO by the time the collision is detected, the receiver is disabled (GREN = 0), and the Receive Error Interrupt flag RCABT is set. If DCR has been selected, the GSC participates in the resolution algorithm.

Incoming bits take 1/2 bit time to get from the GRXD pin to the bit decoder. The bit decoder strips off the preamble/BOF bits, and the first bit after BOF is shifted into a serial strip buffer. The length of the strip buffer is equal to the number of bits in the selected CRC. It is within this buffer that address recognition takes place. If the address is recognized as one for which reception should proceed, then when the first address bit exits the strip buffer it is shifted into an 8-bit shift register. When the shift register is full, its content is transferred to RFIFO. That is the event that determines whether a collision sets RCABT or not.

GSC Transmitting

If the GSC is in the process of transmitting a frame at the time the collision is detected, it will in every case execute its jam/backoff procedure. Its reponse beyond that depends on whether the first byte of the frame has been transferred from TFIFO to the output shift register yet or not. That transfer takes place at the beginning of the first bit of the BOF; that is, 2 bit-times before the end of the preamble/BOF sequence.

If the transfer from TFIFO hasn't occurred yet, the GSC hardware will try again to gain access to the line after its backoff time has expired. Up to 8 automatic restarts can be attempted. If the 8th restart is interrupted by yet another collision, the transmitter is disabled (TEN = 0) and the Transmit Error Interrupt flag TCDT is set.

If the transfer from TFIFO occurs before a collision is detected, the transmitter is disabled (TEN = 0) and the TCDT flag is set.

The response of the GSC to detected collisions is summarized in Figure 3.4.

What the GSC was doing	Response
nothing	None, unless DCR = 1. If DCR = 1, begin DCR countdown.
Receiving a Frame, first byte not in RFIFO yet.	None, unless DCR = 1. If DCR = 1, begin DCR countdown.
Receiving a Frame, first byte already in RFIFO.	Set RCABT, clear GREN. If DCR = 1, begin DCR countdown.
Transmitting a Frame, first byte still in TFIFO	Execute jam/backoff. Restart if collision count ≤8.
Transmitting a Frame, first byte already taken from TFIFO	Execute jam/backoff. Set TCDT, clear TEN.

Figure 3-4. Response to a Detected Collision. References to DCR and the DCR Countdown Have to Do with the Deterministic Collision Resolution Algorithm.

Jam

The jam signal is generated by any 8XC152 that is involved in transmitting a frame at the time a collision is detected at its GRXD pin. This is to ensure that if one transmitting station detects a collision, all the other stations on the network will also detect a collision.

If a transmitting 8XC152 detects a collision during the preamble/BOF part of the frame that it is trying to transmit, it will complete the preamble/BOF and then begin the jam signal in the first bit time after BOF. If the collision is detected later in the frame, the jam signal will begin in the next bit time after the collision was detected.

The jam signal lasts for the same number of bit times as the selected CRC length—either 16- or 32-bit times.

The 8XC152 provides two types of jam signals that can be selected by user software. If the node is DC-coupled to the network, the DC jam can be selected. In this case the GTXD pin is pulled to a logic 0 for the duration of the jam. If the node is AC-coupled to the network, then AC jam must be selected. In this case the GSC takes the CRC it has calculated thus far in the transmission, inverts each bit, and transmits the inverted CRC. The selection of DC or AC jam is made by setting or clearing the DCJ bit, which resides in the SFR named MYSLOT.

When the jam signal is completed, the 8XC152 goes into an idle state. Presumeably, other stations on the network are also generating their own jam signals, after which they too go into an idle state. When the 8XC152 detects the idle state at its own GRXD pin, the backoff sequence begins.

Backoff

There are three software selectable collision resolution algorithms in the 8XC152. The selection is made by writing values to 3 bits:

DCR	M1	M0	Algorithm
0	0	0	Normal Random
0	1	1	Alternate Random
1	1	1	Deterministic

M1 and M0 reside in GMOD, and DCR is in MYSLOT.

In the Normal Random algorithm, the GSC backs off for a random number of slot times and then decides whether to restart the transmission. The backoff time begins as soon as a line idle condition is detected.

The Alternate Random algorithm is the same as the Normal Random except the backoff time doesn't start until an IFS has transpired.

In the Deterministic algorithm, the GSC backs off to await its pre-determined turn.

Random Backoff

In either of the random algorithms, the first thing that happens after a collision is detected is that a 1 gets shifted into the TCDCNT (Transmit Collision Detect Count) register, from the right.

Thus if the software cleared TCDCNT before telling the GSC to transmit, then TCDCNT keeps track of how many times the transmission had to be aborted because of collisions:

TCDCNT =	00000000	first attempt
	00000001	first collision
	00000011	second collision
	00000111	third collision
	00001111	fourth collision
	. . . . .	
	11111111	eighth collision

After TCDCNT gets a 1 shifted into it, the logical AND of TCDCNT and PRBS is loaded into a countdown timer named BKOFF. PRBS is the name of an SFR which contains the output of a pseudo-random binary sequence generator. Its function is to provide a random number for use in the backoff algorithm.

Thus on the first collision BKOFF gets loaded randomly with either 00000000 or 00000001. If there is a second collision it gets loaded with the random selection of 00000000, 00000001, 00000010, or 00000011. On the third collision there will be a random selection among 8 possible numbers. On the fourth, among 16, etc. Figure 3.5 shows the logical arrangement of PRBS, TCDCNT, and BKOFF.

BKOFF starts counting down from its preload value, counting slot times. At any time, the current value in BKOFF can be read by the CPU, but CPU writes to BKOFF have no effect. While BKOFF is counting down, if its current value is not 0, transmission is disabled. The output signal "BKOFF = 0" is asserted when BKOFF reaches 0, and is used to re-enable transmission.

At that time transmission can proceed, subject of course to IFS enforcement, unless:

- shifting a 1 into TCDCNT from the right caused a 1 to shift out from the MSB of TCDCNT, or
- the collision was detected after TFIFO had been accessed by the transmit hardware.

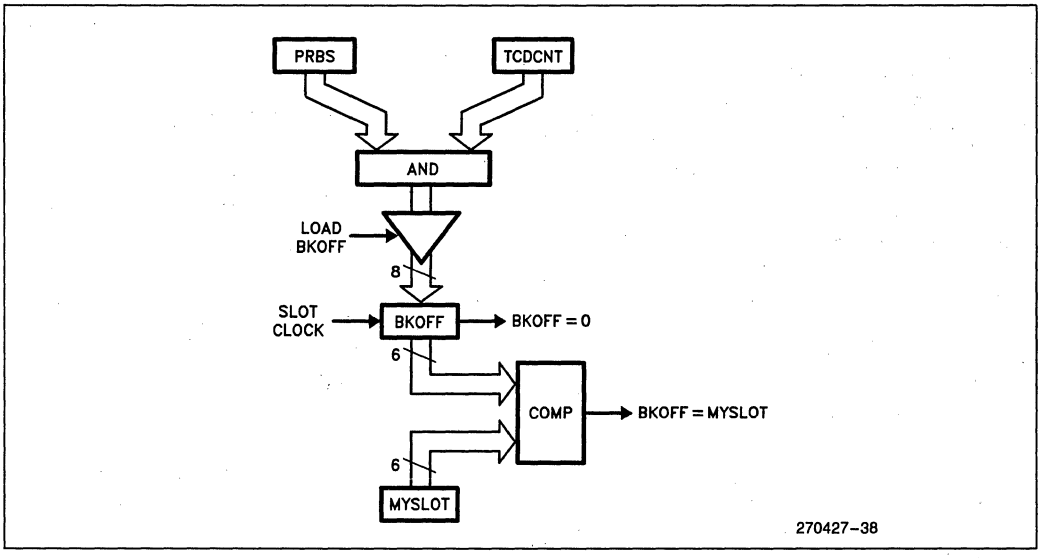


Figure 3.5. Backoff Timer Logic

In either of these cases, the transmitter is disabled (TEN = 0) and the Transmit Error flag TCDT is set. The automatic restart is canceled.

Where the Normal and Alternate Random backoff algorithms differ is that in Normal Random backoff the BKOFF timer starts counting down as soon as a line idle condition is detected, whereas in Alternate Random backoff the BKOFF timer doesn't start counting down till the IFS expires.

The Alternate Random mode was designed for networks in which the slot time is less than the IFS. If the randomly assigned backoff time for a given transmitter happens to be 0, then it is free to transmit as soon as the IFS ends. If the slot time is shorter than the IFS, Normal Random mode would nearly guarantee that if there's a first collision there will be a second collision. The situation is avoided in Alternate Random mode, since the BKOFF countdown doesn't start till the IFS is over.

The unit of count to the BKOFF timer is the slot time. The slot time is measured in bit-times, and is determined by a CPU write to the register SLOTTM. The slot time clock is a 1-byte downcounter which starts its countdown from the value written to SLOTTM. It is decremented each bit time when a backoff is in progress, and when it gets to 1 it generates one tick in the slot time clock. The next state after 1 is the reload value which was written to SLOTTM. If 0 is the value written to SLOTTM, the slot time clock will equal 256 bit times.

A CPU write to SLOTTM accesses the reload register. A CPU read of SLOTTM accesses the downcounter. In most protocols, the slot period must be equal to or greater than the longest round trip propagation time plus the jam time.

Deterministic Backoff

In the Deterministic backoff mode, the GSC is assigned (in software) a slot number. The slot assignment is written to the low 6 bits of the register MYSLOT. This same register also contains, in the 2 high bit positions, the control bits DCJ and DCR.

Slot assignments therefore can run from 0 to 63. It will turn out that the higher the slot assignment, the sooner the GSC will get to restart its transmission in the event of a collision.

The highest slot assignment in the network is written by each station's software into its TCDCNT register. Normally the highest slot assignment is just the total number of stations that are going to participate in the backoff algorithm.

In deterministic backoff mode a collision will not cause a 1 to be shifted into TCDCNT. TCDCNT will still be ANDed with PRBS and the result loaded into BKOFF. In order to insure that all stations have the same value loaded into BKOFF, which determines the first slot number to occur, the PRBS should be loaded with 0FFH; the PRBS will maintain this value until either the 8XC152 is reset or the user writes some other value into PRBS. After BKOFF is loaded it begins counting down slot times as soon as the IFS ends. Slot times are defined by the user, the same way as before, by loading SLOTTM with the number of bit times per slot.

When BKOFF equals the slot assignment (as defined in MYSLOT), the signal "BKOFF = MYSLOT" in Figure 3.5 is asserted for one slot time, during which the GSC can restart its transmission.

While BKOFF is counting down, if any activity is detected at the GRXD pin, the countdown is frozen until the activity ends, a line idle condition is detected, and an IFS transpires. Then the countdown resumes from where it left off.

If a collision is detected at the GRXD pin while BKOFF is counting down, the collision resolution algorithm is restarted from the beginning.

In effect, the GSC "owns" its assigned slot number, but with one exception. Nobody owns slot number 0. Therefore if the GSC is assigned slot number 0, then when BKOFF = 0, this station and any other station that has something to say at this time will have an equal chance to take the line.

3.2.7 HARDWARE BASED ACKNOWLEDGE

Hardware Based Acknowledge (HBA) is a data link packet acknowledging scheme that the user software can enable with CSMA/CD protocol. It is not an option with SDLC protocol however.

In general HBA can give improved system response time and increased effective transmission rates over acknowledge schemes implemented in higher layers of the network architecture. Another benefit is the possibility of early release of the transmit buffer as soon as the acknowledge is received.

The acknowledge consists of a preamble followed by an idle condition. A receiving station with HABEN enabled will send an acknowledge only if the incoming address is unique to the receiving station and if the frame is determined to be correct with no errors. For the acknowledge to be sent, ten must be set. For the transmitting station to recognize the acknowledge GREN must be set. A zero as the LSB of the address indicates that the address is unique and not a group or broadcast address. Errors can be caused by collisions, incorrect CRC, misalignment, or FIFO overflow. The receiver sends the acknowledge as soon as the line is sensed to be idle. The user must program the interframe space and the preamble length such that the acknowledge is completed before IFS expires. This is normally done by programming IFS larger than the preamble.

A transmitting station with HABEN enabled expects an acknowledge. It must receive one prior to the end of the interframe space, or else an error is assumed and the NOACK bit is set. Setting of the TDN bit is also delayed until the end of the interframe space. Collisions detected during the interframe space will also cause NOACK to be set.

The user software may enable the interrupt so that the CPU is notified when TDN is set. If the GSC is serviced by DMA, the user must time out one interframe space and then check the NOACK bit or the TDN bit.

3.3 SDLC Operation

3.3.1 SDLC OVERVIEW

SDLC is a communication protocol developed by IBM and widely used in industry. It is based on a primary/secondary architecture and requires that each secondary station have a unique address. The secondary stations can only communicate to the primary station, and then, only when the primary station allows communication to take place. This eliminates the possibility of contention on the serial line caused by the secondary station's trying to transmit simultaneously.

In the C152, SDLC can be configured to work in either full or half duplex. When adhering to strict SDLC protocol, full duplex is required. Full duplex is selected whenever a 16-bit CRC is selected. At the end of a valid reset the 16-bit CRC is selected. To select half duplex with a 16-bit CRC, the receiver must be turned off by user software before transmission. The receiver is turned off by clearing the GREN bit (RSTAT.1). The receiver needs to be turned off because the address that is transmitted is the address of the secondary station's receiver. If not turned off, the receiver could mistake the outgoing message as being intended for itself. When 32-bit CRCs are used, half duplex is the only method available for transmission.

3.3.2 SDLC Frame Format

The format of an SDLC frame is shown in Figure 3.6. The frame consists of a Beginning of Frame flag, Address field, Control Field, Information field (optional), a CRC, and the End of Frame flag.

Figure 3.6. Typical SDLC Frame

BOF - The begin of frame flag for SDLC is 01111110. It is only one of two possible combinations that have six consecutive ones in SDLC. The other possibility is an abort character which consists of eight or more consecutive ones. This is because SDLC utilizes a process called bit stuffing. Bit stuffing is the insertion of a 0 as the next bit every time a sequence of five consecutive 1s is detected. The receiver automatically removes a 0 after every consecutive group of five ones. This removal of the 0 bit is referred to as bit stripping. Bit stuffing is discussed in Section 3.3.4. All the procedures required for bit stuffing and bit stripping are automatically handled by the GSC.

In standard SDLC protocol the BOF signals the start of a frame and is limited to 8 bits in length. Since there is no preamble in SDLC the BOF is considered an entire separate field and marks the beginning of the frame. The BOF also serves as the clock synchronization mechanism and the reference point for determining the position of the address and control fields.

ADDRESS - The address field is used to identify which stations the message is intended for. Each secondary station must have a unique address. The primary station must then be made aware of which addresses are assigned to each station. The address length is specified as 8-bits in standard SDLC protocols but it is expandable to 16-bits in the C152. User software can further expand the number of address bits, but the automatic address recognition feature works on a maximum of 16-bits.

In SDLC the addresses are normally unique for each station. However, there are several classes of messages that are intended for more than one station. These messages are called broadcast and group addressed frames. An address consisting of all 1s will always be automatically received by the GSC, this is defined as the broadcast address in SDLC. A group address is an address that is common to more than one station. The GSC provides address masking bits to provide the capability of receiving group addresses.

If desired, the user software can mask off all the bits of the address. This type of masking puts the GSC in a promiscuous mode so that all addresses are received.

CONTROL - The control field is used for initialization of the system, identifying the sequence of a frame, to identify if the message is complete, to tell secondary stations if a response is expected, and acknowledgement of previously sent frames. The user software is responsible for insertion of the control field as the GSC hardware has no provisions for the management of this field. The interpretation and formation of the control field must also be handled by user software. The information following the control field is typically used for information transfer, error reporting, and various other functions. These functions are accomplished by the format of the control field. There are three formats available. The types of formats are Informational, Supervisory, or Unnumbered. Figure 3.7 shows the various format types and how to identify them.

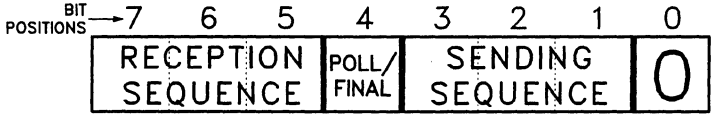
Since the user software is responsible for the implementation of the control field, what follows is a simple explanation on the control field and its functions. For a complete understanding and proper implementation of SDLC, the user should refer to the IBM document, GA27-3093-2, IBM Synchronous Data Link Control General Information. Within that document, is another list of IBM documents which go into detail on the SDLC protocol and its use.

The control field is eight bits wide and the format is determined by bits 0 and 1. If bit 0 is a zero, then the frame is an informational frame. If bit 0 is a one and bit 1 a zero, then it is a supervisory frame, and if bit 0 is a one and bit 1 a one then the frame is an unnumbered frame.

In an informational frame bits 3,2,1 contain the sequence count of the frame being sent.

Bit 4 is the P/F (Poll/Final) bit. If bit 4 equals 1 and originates from the primary, then the secondary station is expected to initiate a transmission. If bit 4 equals 1 and originates from a secondary station, then the frame is the final frame in a transmission.

Bits 7,6,5 contain the sequence count a station expects on the next transmission to it. The sequence count can vary from 000B to 111B. The count then starts over again at 000B after the value 111B is incremented. The acknowledgement is recognized by the receiving station when it decodes bits 7,6,5 of an incoming frame. The station sending the transmission is acknowledging the frames received up to the count represented in bits 7,6,5 (sequence count-1). With this method, up to seven sequential frames may be transmitted prior to an acknowledgement being received. If eight frames were allowed to pass before an acknowledgement, the sequence count would roll over and this would negate the purpose of the sequence numbers.



270427-15

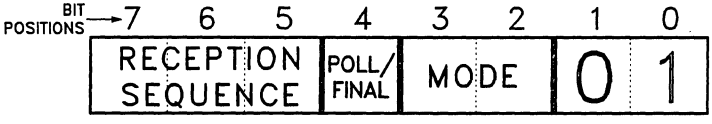
RECEPTION SEQUENCE - The sequence expected in the SENDING SEQUENCE portion of the control byte in the next received frame. This also confirms correct reception of up to seven frames prior to the sequence given.

POLL/FINAL - Identifies the frame as being a polling request from the master station or the last in a series of frames from the master or secondary.

SENDING SEQUENCE - Identifies the sequence of the frame being transmitted.

0 - If bit 0 = 0 the frame is identified as a informational format type.

INFORMATION FORMAT



270427-16

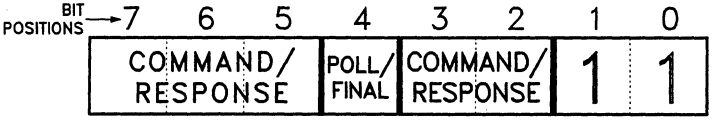
RECEPTION SEQUENCE - Expected sequence of frame for next reception.

POLL/FINAL - Identifies frame as being a polling request from the master station or the last in a series of frames from the master or secondary.

MODE - Identifies whether receiver is ready (00), not ready (10) or a frame was rejected (01). The rejected frame is identified by the reception sequence.

0,1 - If bits 1,0 = 0,1 the frame is identified as a supervisory format type.

SUPERVISORY FORMAT



270427-17

COMMAND/RESPONSE - Identifies the type of command or response.

POLL/FINAL - Identifies frame as being a polling request from the master station or the last in a series of frames from the master or secondary.

1,1 - If bits 1,0 = 1,1 the frame is identified as an unnumbered format type.

NONSEQUENCED FORMAT

270427-18

Figure 3.7. SDLC Control Field

Following the informational control field comes the information to be transferred.

In the supervisory format (bits 1,0 = 0,1) bits 3,2 determine which mode is being used.

When the mode is 00 it indicates that the receive line of the station that sent the supervisory frame is enabled and ready to accept frames.

When the mode is 01, it indicates that previously a received frame was rejected. The value in the receive count identifies which frame(s) need to be retransmitted.

When the mode is 10, the sending station is indicating that its receiver is not ready to accept frames.

Mode 11 is an illegal mode in SDLC protocol.

Bits 7,6,5 represent the value of the sequence the station expects when the next transfer occurs for that station. There is no information following the control field when the supervisory format is used.

In the unnumbered format (bits 1,0 = 1,1) bits 7, 6, 5, 3, 2 (notice bit 4 is missing) indicate commands from the primary to secondary stations or requests of secondary stations to the primary.

The standard commands are:

BITS 7	6	5	3	2	Command
0	0	0	0	0	Unnumbered Information (UI)
0	0	0	0	1	Set initialization mode (SIM)
0	1	0	0	0	Disconnect (DISC)
0	0	1	0	0	Response optional (UP)
1	1	0	0	1	Function descriptor in information field (CFGR)
1	0	1	1	1	Identification in information field. (XID)
1	1	1	0	0	Test pattern in information field. (TEST)

The standard responses are:

BITS 7	6	5	3	2	Command
0	0	0	0	0	Unnumbered information (UI)
0	0	0	0	1	Request for initialization (RIM)
0	0	0	1	1	Station in disconnected mode (DM)
1	0	0	0	1	Invalid frame received (FRMR)
0	1	1	0	0	Unnumbered acknowledgement (UA)
1	1	1	1	1	Signal loss of input (BCN)
1	1	0	0	1	Function descriptor in information field (CFGR)
0	1	0	0	0	Station wants to disconnect (RD)
1	0	1	1	1	Identification in information field (XID)
1	1	1	0	0	Test pattern in information field (TEST)

In an unnumbered frame, information of variable length may follow the control field if UI is used, or information of fixed length may follow if FRMR is used.

As stated earlier, the user software is responsible for the proper management of the control field. This portion of the frame is passed to or from the GSC FIFOs as basic informational type data.

INFO - This is the information field and contains the data that one device on the link wishes to transmit to another device. It can be of any length the user wishes, but must be a multiple of 8 bits. It is possible that some frames may contain no information field. The information field is identified to the receiving stations by the preceding control field and the following CRC. The GSC determines where the last of the information field is by passing the bits through the CRC generator. When the last bit or EOF is received the bits that remain constitute the CRC.

CRC - The Cyclic Redundancy Check (CRC) is an error checking sequence commonly used in serial communications. The C152 offers two types of CRC algorithms, a 16-bit and a 32-bit. The 32-bit algorithm is normally used in CSMA/CD applications and is described in section 3.2.2. In most SDLC applications a 16-bit CRC is used and the hardware configuration that supports 16-bit CRC is shown in Figure 3.8. The generating polynomial that the CRC generator uses with the 16-bit CRC is:

$$G(X) = X^{16} + X^{12} + X^{5} + 1$$

The way the CRC operates is that as a bit is received it is XOR'd with bit 15 of the current CRC and placed in temporary storage. The result of XOR'ing bit 15 with the received bit is then XOR'd with bit 4 and bit 11 as the CRC is shifted one position to the right. The bit in temporary storage is shifted into position 0.

The required CRC length for SDLC is 16 bits. The CRC is automatically stripped from the frame and not passed on to the CPU. The last 16 bits are then run though the CRC generator to insure that the correct remainder is left. The remainder that is checked for is 001110100001111B (1D0F Hex). If there is a mismatch, an error is generated. The user software has the option of enabling this interrupt so the CPU is notified.

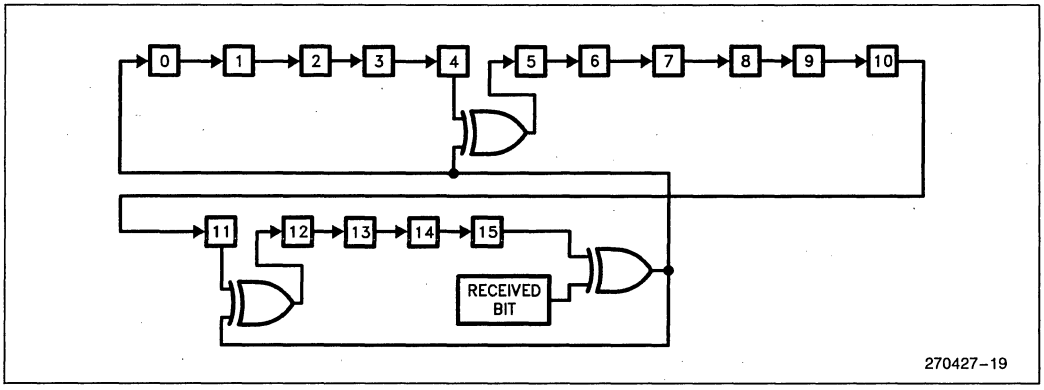


Figure 3.8. 16-Bit CRC

EOF - The End Of Frame (EOF) indicates when the transmission is complete. The EOF is identified by the end flag. An end flag consists of the bit pattern 01111110. The EOF can also serve as the BOF for the next frame.

3.3.3 DATA ENCODING

The transmission of data in SDLC mode is done via NRZI encoding as shown in Figure 3.9. NRZI encoding transmits data by changing the state of the output whenever a 0 is being transmitted. Whenever a 1 is transmitted the state of the output remains the same as the previous bit and remains valid for the entire bit time. When SDLC mode is selected it automatically enables the NRZI encoding on the transmit line and NRZI decoding on the receive line.

3.3.4 BIT STUFFING/STRIPPING

In SDLC mode one of the primary rules of the protocol is that in any normal data transmission, there will never be an occurrence of more than 5 consecutive 1s. The GSC takes care of this housekeeping chore by automatically inserting a 0 after every occurrence of 5 consecutive 1s and the receiver automatically removes a zero after receiving 5 consecutive 1s. All the necessary steps required for implementing bit stuffing and stripping are incorporated into the GSC hardware. This makes the operation transparent to the user. About the only time this operation becomes apparent to the user, is if the actual data on the transmission medium is being monitored by a device that is not aware of the automatic insertion of 0s. The bit stuffing/stripping guarantees that there will be at least one transition every 6 bit times while the line is active.

3.3.5 SENDING ABORT CHARACTER

An abort character is one of the exceptions to the rule that disallows more than 5 consecutive 1s. The abort character consists of any occurrence of seven or more consecutive ones. The simplest way for the C152 to send an abort character is to clear the TEN bit. This causes the output to be disabled which, in turn, forces it to a constant high state. The delay necessary to insure that the link is high for seven bit times is a task that needs to be handled by user software. Other methods of sending an abort character are using the IFS register or using the Raw Transmit mode. Using IFS still entails clearing the TEN bit, but TEN can be immediately re-enabled. The next message will not begin until the IFS expires. The IFS begins timing out as soon as $\overline{DEN}$ goes high which identifies the end of transmission. This also requires that IFS contain a value equal to or greater than 8. This method may have the undesirable effect that $\overline{DEN}$ goes high and disables the external drivers. The other alternative is to switch to Raw Transmit mode. Then, writing 0FFH to TFIFO would generate a high output for 8 bit times. This method would leave $\overline{DEN}$ active during the transmission of the abort character.

When the receiver detects seven or more consecutive 1s and data has been loaded into the receive FIFO, the RCABT flag is set in RSTAT and that frame is ignored. If no data has been loaded into the receive FIFO, there are no abort flags set and that frame is just ignored. A retransmitted frame may immediately follow an abort character, provided the proper flags are used.

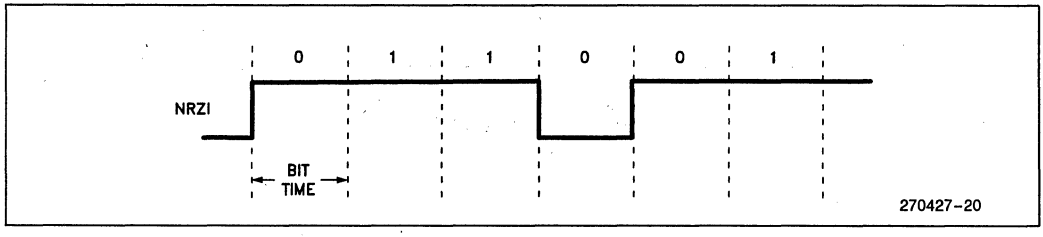


Figure 3.9. NRZI Encoding

3.3.6 LINE IDLE

If 15 or more consecutive 1s are detected by the receiver the Line Idle bit (LNI) in TSTAT is set. The seven 1s from the abort character may be included when sensing for a line idle condition. The same methods used for sending the Abort character can be used for creating the Idle condition. However, the values would need to be changed to reflect 15 bit times, instead of seven bit times.

3.3.7 ACKNOWLEDGEMENT

Acknowledgment in SDLC is an implied acknowledge and is contained in the control field. Part of the control frame is the sequence number of the next expected frame. This sequence number is called the Receive Count. In transmitting the Receive Count, the receiver is in fact acknowledging all the previous frames prior to the count that was transmitted. This allows for the transmission of up to seven frames before an acknowledge is required back to the transmitter. The limitation of seven frames is necessary because the Receive Count in the control field is limited to three binary digits. This means that if an eighth transmission occurred this would cause the next Receive Count to repeat the first count that still is waiting for an acknowledge. This would defeat the purpose of the acknowledgement. The processing and general maintenance of the sequence count must be done by the user software. The Hardware Based Acknowledge option that is provided in the C152 is not compatible with standard SDLC protocol.

3.3.8 PRIMARY/SECONDARY STATIONS

All SDLC networks are based upon a primary/secondary station relationship. There can be only one primary station in a network and all the other stations are considered secondary. All communication is between the primary and secondary station. Secondary station to secondary station direct communication is prohibited. If there is a need for secondary to secondary communication, the user software will have to make allowances for the master to act as an intermediary. Secondary stations are allowed use of the serial line only when the master permits them. This is done by the master polling the secondary stations to see if they have a need to access the serial line. This should prevent any collisions from occurring, provided each secondary station has its own unique address. This arrangement also partially determines the types of networks supported. Normal SDLC networks consist of point-to-point, multi-drop, or ring configurations and the C152 supports all of these. However, some SDLC processors support an automatic one bit delay at each node that is not supported by the C152. In a "Loop Mode" configuration, is is necessary that the transmission be delayed from the reception of the frames from the upstream station before passing the message to the downstream station. This delay is necessary so that a station can decode its own address before the message is passed on. The various networks are shown in Figure 3.10.

3.3.9 HDLC/SDLC COMPARISON

HDLC (High level Data Link Control) is a standard adopted by the International Standards Organization (ISO). The HDLC standard is defined in the ISO document #ISO 6159 - HDLC unbalanced classes of procedures. IBM developed the SDLC protocol as a subset of HDLC. SDLC conforms to HDLC protocol requirements, but is more restrictive. SDLC contains a more precise definition on the modes of operation.

Some of the major differences between SDLC and HDLC are:

SDLC	HDLC
Unbalanced (primary/ secondary)	Balanced (peer to peer)
Modulo 8 (no extensions allowed, up to 7 outstanding frames before acknowledge is required)	Modulo 128 (up to 127 outstanding frames before acknowledge is required)
8-bit addressing only	Extended addressing
Byte aligned data	Variable size of data

The C152 does not support HDLC implementation requiring data alignment other than byte alignment. The user will find that many of the protocol parameters are programmable in the C152 which allows easy implementation of proprietary or standard HDLC network. User software needs to implement the control field functions.

3.4 User Defined Protocols

The explanation on the implementation of user defined protocols would go beyond the scope of this manual, but examining Table 3.1 should give the reader a consolidated list of most of the possibilities. In this manual, any deviation from the documents that cover the implementation of CSMA/CD or SDLC are considered user defined protocols. Examples of this would be the use of SDLC with the 32-bit CRC selected or CSMA/CD with hardware based acknowledge.

3.5 Using the GSC

3.5.1 LINE DISCIPLINE

Line discipline is how the management of the transfer of data over the physical medium is controlled. Two types of line discipline will be discussed in this section: full duplex and half duplex.

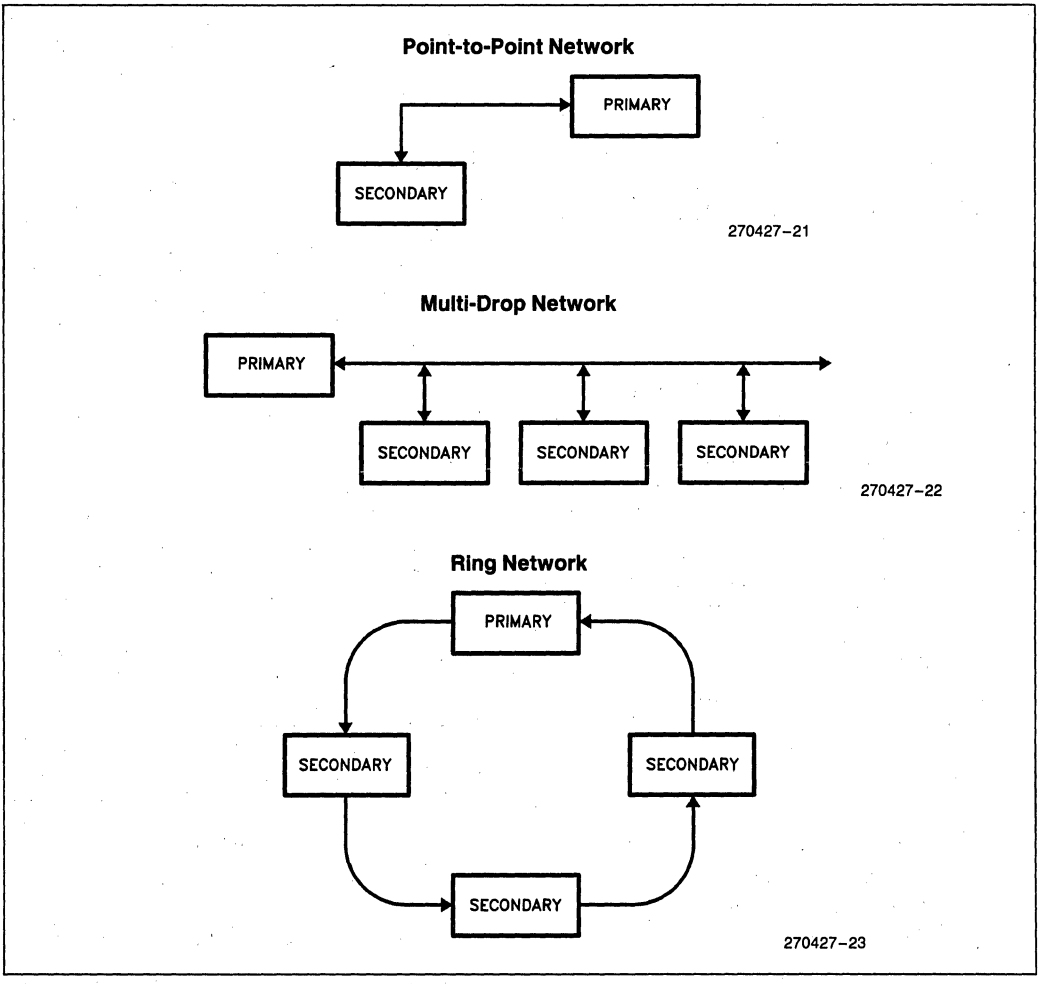


Figure 3.10. SDLC Networks

Full duplex is the simultaneous transmission and reception of data. Full duplex uses anywhere from two to four wires. At least one wire is needed for transmission and one wire for reception. Usually there will also be a ground reference on each signal if the distance from station to station is relatively long. Full-duplex operation in the C152 requires that both the receive and the transmit portion of the GSC are functioning at the same time. Since both the transmitter and receiver are operating, two CRC generators are also needed. The C152 handles this problem by having one 32-bit CRC generator and one 16-bit CRC generator. When supporting full-duplex operation, the 32-bit CRC generator is modified to work as a 16-bit CRC generator. Whenever the 16-bit CRC is selected, the GSC automatically enters the full duplex mode. Half duplex with a 16-bit CRC is discussed in the following paragraph.

Half duplex is the alternate transmission and reception of data over a single common wire. Only one or two wires are needed in half-duplex systems. One wire is needed for the signal and if the distance to be covered is long there will also be a wire for the ground reference. In half-duplex mode, only the receiver or transmitter can operate at one time. When the receiver or transmitter operates is determined by user software, but typically the receiver will always be enabled unless the GSC is transmitting. Whenever half duplex is being used the software must insure that only the receiver or transmitter is enabled at any given time. This is particularly important when using SDLC, so that the receiver will not recognize its own address when the transmitter is operating. Half-duplex operation in the C152 is supported with either 16-bit or 32-bit CRCs. Whenever a 32-bit CRC is selected, only half-duplex operation can be supported by the GSC. It is possible to simulate full-duplex operation with a 32-bit CRC, but this would require that the CRC be performed with software. Calculating the CRC with the CPU would greatly reduce the data rates that could be used with the GSC. Whenever a 16-bit CRC is selected, full-duplex operation is automatically chosen and the GSC must be reconfigured if half-duplex operation is preferred.

3.5.2 PLANNING FOR NETWORK CHANGES AND EXPANSIONS

A complete explanation on how to plan for network expansion will not be covered in this manual as there are far too many possibilities that would need to be discussed. But there are several areas that will have major impact when allowing for changes in the system. In cases where there will never be any changes allowed, expansion plans become a mute issue. However, it is strongly suggested that there always be some allowance for future modifications.

Some of the general areas that will impact the overall scheme on how to incorporate future changes to the system are:

1) Communication of the change to all the stations or the primary station.

2) Maximum distance for communication. This will affect the drivers used and the slot time.

3) More stations may be on the line at one time. This may impact the interframe space or the collision resolution used.

4) If using CSMA/CD without deterministic resolution, any increase in network size will have a negative impact on the average throughput of the network and lower the efficiency. The user will have to give careful consideration when deciding how large a system can ultimately be and still maintain adequate performance.

3.5.3 DMA SERVICING OF GSC CHANNELS

There are two sources that can be used to control the GSC. The first is CPU control and the second is DMA control.

CPU control is used when user software takes care of the tasks such as: loading the TFIFO, reading the RFIFO, checking the status flags, and general tracking of the transmission process. As the number of tasks grow and higher data transfer rates are used, the overhead required by the CPU becomes the dominant consumption of time. Eventually, a point is reached where the CPU is spending 100% of its time responding to the needs of the GSC. An alternative is to have the DMA channels control the GSC.

A detailed explanation on the general use of the DMA channels is covered in Section 4. In this section only those details required for the use of the DMA channels with the GSC will be covered.

The DMA channels can be configured by user software so that the GSC data transfers are serviced by the DMA controller. Since there are two DMA channels, one channel can be used to service the receiver, and one channel can be used to service the transmitter. In using the DMA channels, the CPU is relieved of much of the time required to do the basic servicing of the GSC buffers. The types of servicing that the DMA channels can provide are: loading of the transmit FIFO, removing data from the receive FIFO, notification of the CPU when the transmission or reception has ended, and response to certain error conditions. When using the

DMA channels the source or destination of the data intended for serial transmission can be internal data memory, external data memory, or any of the SFRs.

The only tasks required after initialization of the DMA and GSC registers are enabling the proper interrupts and informing the DMA controller when to start. After the DMA channels are started all that is required of the CPU is to respond to error conditions or wait until the end of transmission.

Initialization of the DMA channels requires setting up the control, source, and destination address registers. On the DMA channel servicing the receiver, the control register needs to be loaded as follows: DCONn.2 = 0, this sets the transfer mode so that response is to GSC interrupts and put the DMA control in alternate cycle mode; DCONn.3 = 1, this enables the demand mode; DCONn.4 = 0, this clears the automatic increment option for the source address; and DCONn.5 = 1, this defines the source as SFR. The DMA channel servicing the receiver also needs its source address register to contain the address of RFIFO (SARHN = XXH, SARLN = 0F4H). On the DMA channel servicing the transmitter, the control register needs to be loaded as follows: DCONn.2 = 0; DCONn.3 = 1; DCONn.6 = 0, this clears the automatic increment option for the destination address; and DCONn.7 = 1, this sets the destination as SFR. The DMA channel serving the transmitter also requires that its destination address register contains the address of TFIFO (DARHN = XXH, DARLN = 85H). Assuming that DCON0 would be serving the receiver and DCON1 the transmitter, DCON0 would be loaded with XX1010X0B and DCON1 would be loaded with 10XX10X0B. The contents of SARH0 and DARH1 do not have any impact when using internal SFRs as the source or destination.

When using the DMA channels to service the GSC, the byte count registers will also need to be initialized.

The Done flag for the DMA channel servicing the receiver should be used if fixed packet lengths only are being transmitted or to insure that memory is not overwritten by long received data packets. Overwriting of data can occur when using a smaller buffer than the packet size. In these cases the servicing of the DMA and/or GSC would be in response to the DMA Done flag when the byte count reaches zero.

In some cases the buffer size is not the limiting factor and the packet lengths will be unknown. In these cases it would be desirable to eliminate the function of the Done flag. To effectively disable the Done flag for the DMA channel servicing the receiver, the byte count should be set to some number larger than any packet that will be received, up to 64K. If not using the Done flag, then GSC servicing would be driven by the receive Done (RDN) flag and/or interrupt. RDN is set when the EOF is detected. When using the RDN flag, RFNE should also be checked to insure that all the data has been emptied out of the receive FIFO.

The byte count register is used for all transmissions and this means that all packets going out will have to be of the same length or the length of the packet to be sent will have to be known prior to the start of transmission. When using the DMA channels to service the GSC transmitter, there is no practical way to disable the Done flag. This is because the transmit done flag (TDN) is set when the transmit FIFO is empty and the last message bit has been transmitted. But, when using the DMA channel to service the transmitter, loads to the TFIFO continue to occur until the byte count reaches 0. This makes it impossible to use TDN as a flag to stop the DMA transfers to TFIFO. It is possible to examine some other registers or conditions, such as the current byte count, to determine when to stop the DMA transfers to TFIFO, but this is not recommended as a way to service the DMA and GSC when transmitting because frequent reading of the DMA registers will cause the effective DMA transfer rate to slow down.

When using the DMA channels, initialization of the GSC would be exactly the same as normal except that TSTAT.0 = 1 (DMA), this informs the GSC that the DMA channels are going to be used to service the GSC. Although only TSTAT is written to, both the receiver and transmitter use this same DMA bit.

The interrupts EGSTE (IEN1.5), GSC transmit error; EGSTV (IEN1.3), GSC transmit valid; EGSRE (IEN1.1), GSC receive error; and EGSRV (IEN1.0), GSC receive valid; need to be enabled. The DMA interrupts are normally not used when servicing the GSC with the DMA channels. To ensure that the DMA interrupts are not responded to is a function of the user software and should be checked by the software to make sure they are not enabled. Priority for these interrupts can also be set at this time. Whether to use high or low priority needs to be decided by the user. When responding to the GSC interrupts, if a buffer is being used to store the GSC information, then the DMA registers used for the buffer will probably need updating.

After this initialization, all that needs to be done when the GSC is actually going to be used is: load the byte count, set-up the source addresses for the DMA channel servicing the transmitter, set-up the destination addresses for the DMA channel servicing the receiver, and start the DMA transfer. The GSC enable bits should be set first and then the GO bits for the DMA. This initiates the data transfers.

This simplifies the maintenance of the GSC and can make the implementation of an external buffer for packetized information automatic.

An external buffer can be used as the source of data for transmission, or the destination of data from the receiver. In this arrangement, the message size is limited to the RAM size or 64K, whichever is smaller. By using an external buffer, the data can be accessed by other devices which may want access to the serial data. The amount of time required for the external data moves will also decrease. Under CPU control, a "MOVX" command would take 24 oscillator periods to complete. Under DMA control, external to internal, or internal to external, data moves take only 12 oscillator periods.

3.5.4 BAUD RATE

The GSC baud rate is determined by the contents of the SFR, BAUD, or the external clock. The formula used to determine the baud rate when using the internal clock is:

(fosc)/((BAUD+1)*8)

For example if a 12 MHz oscillator is used the baud rate can vary from:

12,000,000/((0+1)*8) = 1.5 MBPS

to:

12,000,000/((255+1)*8) = 5.859 KBPS

There are certain requirements that the external clock will need to meet. These requirements are specified in the data sheet. For a description of the use of the GSC with external clock please read Section 3.5.11.

3.5.5 INITIALIZATION

Initialization can be broken down into two major components, 1) initialization of the component so that its serial port is capable of proper communication; and 2) initialization of the system or a station so that intelligible communication can take place.

Most of the initialization of the component has already been discussed in the previous sections. Those items not covered are the parameters required for the component to effectively communicate with other components. These types of issues are common to both system and component initialization and will be covered in the following text.

Initialization of the system can be broken down into several steps. First, are the assumptions of each network station.

The first assumption is that the type of data encoding to be used is predetermined for the system and that each station will adhere to the same basic rules defining that encoding. The second assumption is that the basic protocol and line discipline is predetermined and known. This means that all stations are using CSMA/CD or SDLC or whatever, and that all stations are either full or half duplex. The third assumption is that the baud rate is preset for the whole system. Although the baud rate could probably be determined by the microprocessor just by monitoring the link, it will make it much simpler if the baud rate is known in advance.

One of the first things that will be required during system initialization is the assignment of unique addresses for each station. In a two-station only environment this is not necessary and can be ignored. However, keep in mind, that all systems should be constructed for easy future expansions. Therefore, even in only a two station system, addresses should be assigned. There are three basic ways in which addresses can be assigned. The first, and most common is preassigned addresses that are loaded into the station by the user. This could be done with a DIP-switch, through a keyboard. The second method of assigning addresses is to randomly assign an address and then check for its uniqueness throughout the system, and the third method is to make an inquiry to the system for the assignment of a unique address. Once the method of address assignment is determined, the method should become part of the specifications for the system to which all additions will have to adhere. This, then, is the final assumption.

The negotiation process may not be clear for some readers. The following two procedures are given as a guideline for dynamic address assignment.

In the first procedure, a station assumes a random address and then checks for its uniqueness throughout the system. As a station is initialized into the system it sends out a message containing its assumed address. The format of the message should be such that any station decoding the address recognizes it as a request for initialization. If that address is already used, the receiving station returns a message, with its own address stating that the address in question is already taken. The initializing station then picks another address. When the initializing station sends its inquiry for the address check, a timer is also started. If the timer expires before the inquiry is responded to, then that station assumes the address chosen is okay.

In the second procedure, an initializing station asks for an address assignment from the system. This requires that some station on the link take care of the task of maintaining a record of which addresses are used. This station will be called station-1. When the initializing station, called station-2, gets on the link, it sends out a message with a broadcast address. The format of the message should be such that all other stations on the link recognize it as a request for address assignment. Part of the message from station-2 is a random number generated by the station requesting the address. Station-2 then examines all received messages for this random number. The random number could be the address of the received message or could be within the information section of a broadcast frame. All the stations, except station-1, on the link should ignore the initialization request. Station-1, upon receiving the initialization request, assigns an address and returns it to station-2. Station-1 will be required to format the message in such a manner so that all stations on the link recognize it as a response to initialization. This means that all stations except station-2 ignore the return message.

3.5.6 TEST MODES

There are two test modes associated with the GSC that are made available to the user. The test modes are named Raw Receive and Raw Transmit. The test modes are selected by the proper setting of the two mode bits in GMOD (M0 = GMOD.5, M1 = GMOD.6). If M1,M0 = 0,1 then Raw Transmit is selected. If M1,M0 = 1,0 then Raw Receive is enabled.

In Raw Transmit, the transmit output is internally connected to the Receiver input. This is intended to be used as a local loop-back test mode, so that all data written to the transmitter will be returned by the receiver. Raw Transmit can also be used to transmit user data. If Raw Transmit is used in this way the data is emitted with no preamble, flag, address, CRC, and no bit insertion. The data is still encoded with whatever format is selected, Manchester with CSMA/CD, NRZI with SDLC or as NRZ if external clocks are used. The receiver still operates as normal and in this mode most of the receive functions can be tested.

In Raw Receive, the transmitter should be externally connected to the receiver. To do this a port pin should be used to enable an external device to connect the two pins together. In Raw Receive mode the receiver acts as normal except that all bytes following the BOF are loaded into the receive FIFO, including the CRC. Also address recognition is not active but needs to be performed in software. If SDLC is selected as the protocol, zero-bit deletion is still enabled. The transmitter still operates as normal and in this mode most of the transmitter functions and an external transceiver can be tested. This is also the only way that the CRC can be read by the CPU, but the CRC error bit will not be set.

3.5.7 EXTERNAL DRIVER INTERFACE

A signal is provided from the C152 to enable transmitter drivers for the serial link. This is provided for systems that require more than what the GSC ports are capable of delivering. The voltage and currents that the GSC is capable of providing are the same levels as those for normal port operation. The signal used to enable the external drivers is $\overline{DEN}$. No similar signal is needed for the receiver.

3.5.8 JITTER (RECEIVE)

Data jitter is the difference between the actual transmitted waveform and the exact calculated value(s). In NRZI, data jitter would be how much the actual waveform exceeds or falls short of one calculated bit time. A bit time equals 1/baud rate. If using Manchester encoding, there can be two transitions during one bit time as shown in Figure 3.11. This causes a second parameter to be considered when trying to figure out the complete data jitter amount. This other parameter is the half-bit jitter. The half-bit jitter is comprised of the difference in time that the half-bit transition actually occurs and the calculated value. Jitter is important because if the transition occurs too soon it is considered noise, and if the transition occurs too late, then either the bit is missed or a collision is assumed.

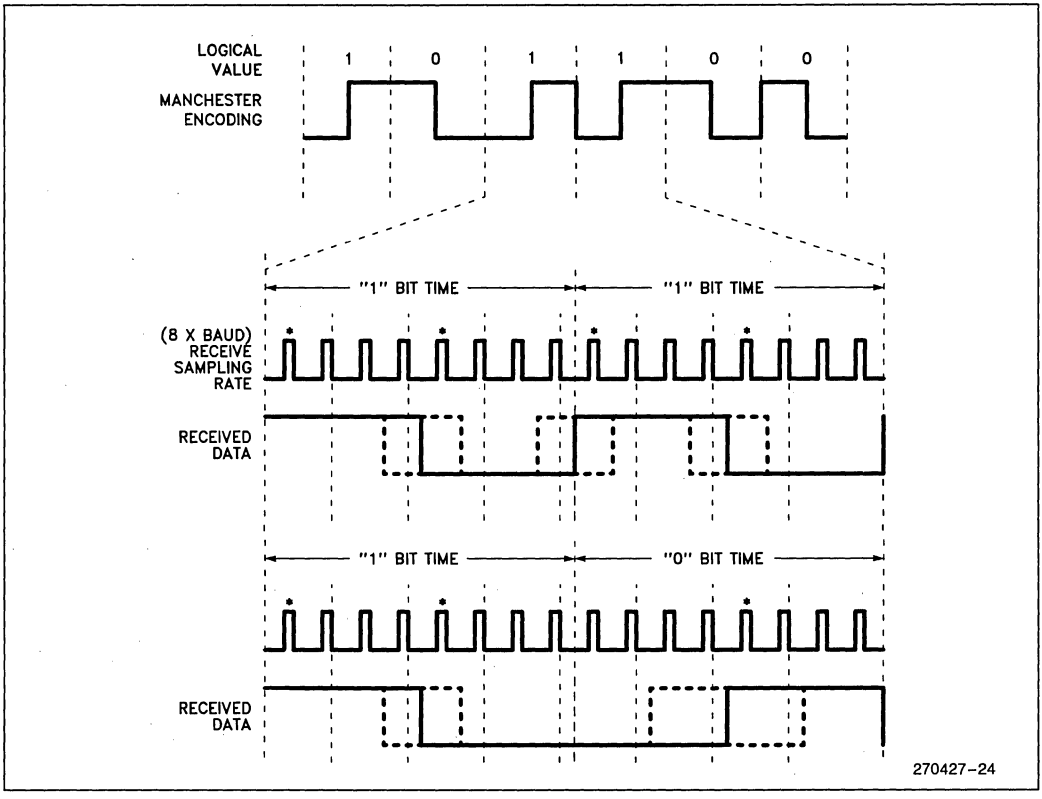


Figure 3.11. Jitter

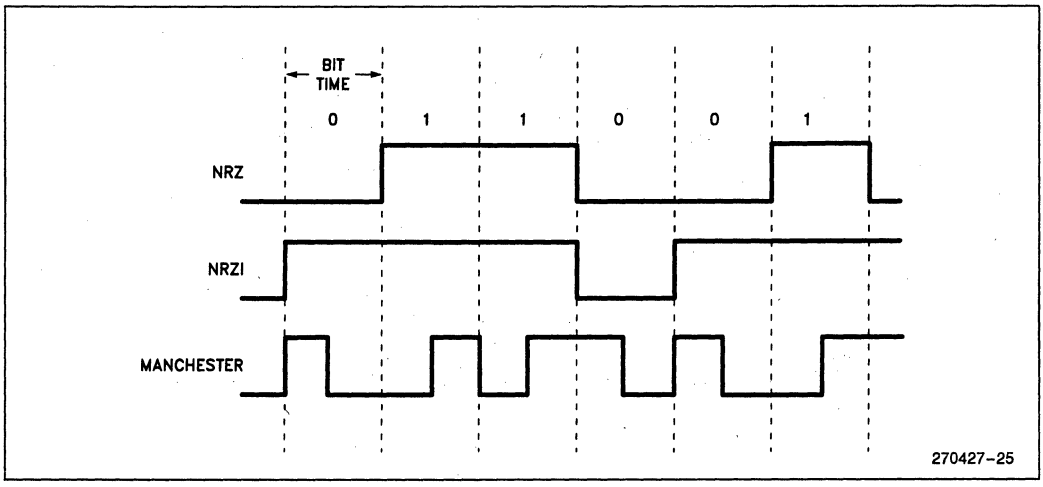


Figure 3.12. Transmit Waveforms

3.5.9 Transmit Waveforms

The GSC is capable of three types of data encoding, Manchester, NRZI, and NRZ. Figure 3.12 shows examples of all three types of data encoding.

3.5.10 Receiver Clock Recovery

The receiver is always monitored at eight times the baud rate frequency, except when an external clock is used. When using an external clock the receiver is loaded during the clock cycle.

In CSMA/CD mode the receiver synchronizes to the transmitted data during the preamble. If a pulse is detected as being too short it is assumed to be noise or a collision. If a pulse is too long it is assumed to be a collision or an idle condition.

In SDLC the synchronization takes place during the BOF flag. In addition, pulses less than four sample periods are ignored, and assumed to be noise. This sets a lower limit on the pulse size of received zeros.

In CSMA/CD the preamble consists of alternating 1s and 0s. Consequently, the preamble looks like the waveform in Figure 3.13A and 3.13B.

3.5.11 External Clocking

To select external clocking, the user is given three choices. External clocking can be used with the transmitter, with the receiver, or with both. To select external clocking for the transmitter, XTCLK (GMOD.7) has to be set to a 1. To select external clocking for the receiver, XRCLK (PCON.3) has to be set to a 1. Setting both bits to 1 forces external clocking for the receiver and transmitter.

The external transmit clock is applied to pin 4 ($\overline{TXC}$), P1.3. The external receive clock is applied to pin 5 ($\overline{RXC}$), P1.4. To enable the external clock function on the port pin, that pin has to be set to a 1 in the appropriate SFR, P1.

Whenever the external clock option is used, the format of the transmitted and received data is restricted to NRZ encoding and the protocol is restricted to SDLC. With external clock, the bit stuffing/stripping is still active with SDLC protocol.

3.6 GSC Operation

3.6.1 Determining Line Discipline

In normal operation the GSC uses full or half duplex operation. When using a 32-bit CRC (GMOD.3 = 1), operation can only be half duplex. If using a 16-bit CRC (GMOD.3 = 0), full duplex is selected by default. When using a 16-bit CRC the receiver can be turned off while transmitting (RSTAT.1 = 0), and the transmitter can be turned off during reception (TSTAT.1 = 0). This simulates half-duplex operation when using a 16-bit CRC.

Normally, HDLC uses a 16-bit CRC, so half duplex is determined by turning off the receiver or transmitter. This is so that the receiver will not detect its own ad-

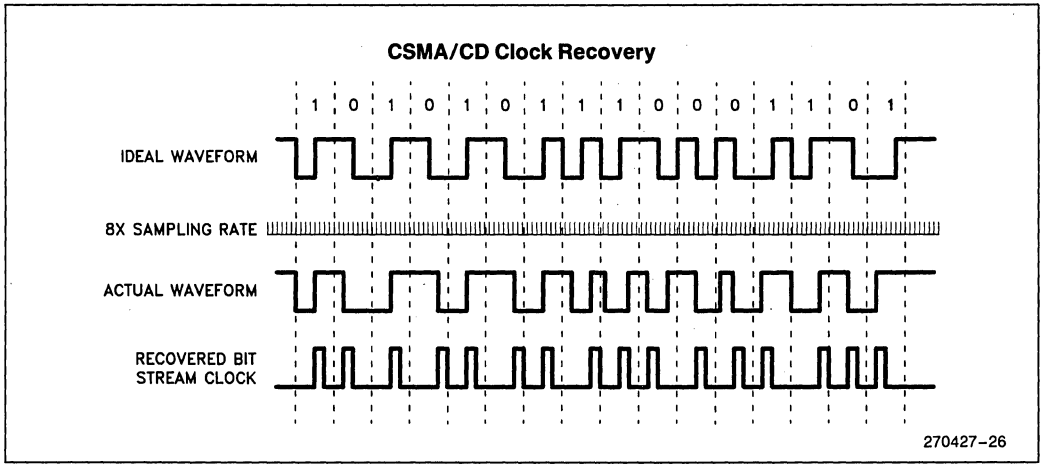


Figure 3.13A. Clock Recovery

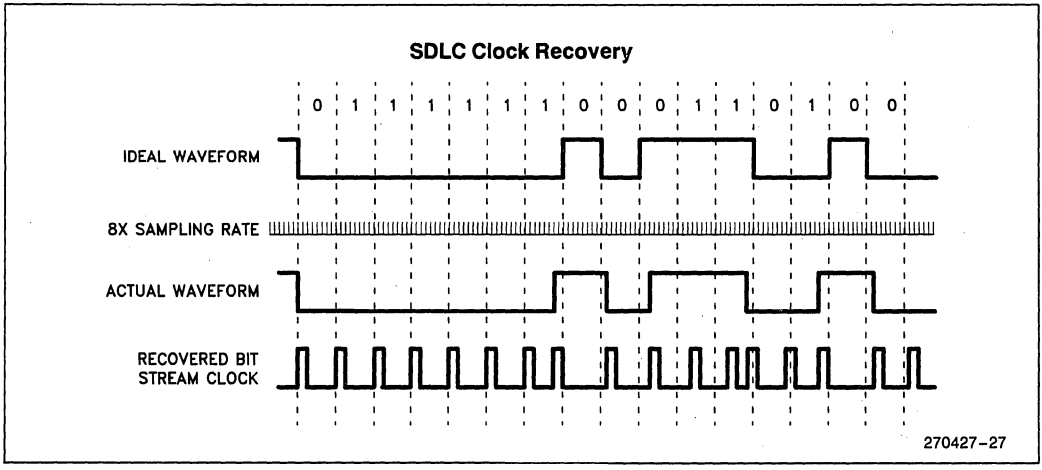


Figure 3.13B. Clock Recovery

dress as transmission takes place. This also needs to be done when using CSMA/CD with a 16-bit CRC for the same reason.

3.6.2 CPU/DMA CONTROL OF THE GSC

The data for transmission or reception can be handled by either the CPU (TSTAT.0 = 0) or DMA controller (TSTAT.0 = 1). This allows the user two sets of flags to control the FIFO. Associated with these flags are interrupts, which may be enabled by the user software. Either one or both sets of flags may be used at the same time.

In CPU control mode the flags (RFNE,TFNF) are generated by the condition of the receive or transmit FIFO's. After loading a byte into the transmit FIFO, there is a one machine cycle latency until the TFNF flag is updated. Because of this latency, the status of TFNF should not be checked immediately following the instruction to load the transmit FIFO. If using the interrupts to service the transmit FIFO,the one machine cycle of latency must be considered if the TFNF flag is checked prior to leaving the subroutine.

When using the CPU for control, transmission normally is initiated by setting the TEN bit (TSTAT.1) and then writing to TFIFO. TEN must be set before loading the transmit FIFO, as setting TEN clears the transmit FIFO. TCDCNT should also be checked by user software and cleared if a collision occurred on a prior transmission.

To enable the receiver, GREN (RSTAT.1) is set. After GREN is set, the GSC begins to look for a valid BOF. After detecting a valid BOF the GSC attempts to match the received address byte(s) against the address match registers. When a match occurs the frame is loaded into the GSC. Due to the CRC strip hardware, there is a 40 or 24 bit time delay following the BOF until the first data byte is loaded into RFIFO if the 32 or 16 bit CRC is chosen. If the end of frame is detected before data is loaded into the receive FIFO, the receiver ignores that frame.

If the receiver detects a collision during reception in CSMA/CD mode and if any bytes have been loaded into the receive FIFO, the RCABT flag is set. The GSC hardware then halts reception and resets GREN. The user software needs to filter any collision fragment data which may have been received. If the collision occurred prior to the data being loaded into RFIFO the CPU is not notified and the receiver is left enabled. At the end of a reception the RDN bit is set and GREN is cleared. In HABEN mode this causes an acknowledgement to be transmitted if the frame did not have a broadcast or multi-cast address. The user software can enable the interrupt for RDN to determine when a frame is completed.

In DMA mode the interrupts are generated by the internal "transmit/receive done" (TDN,RDN) conditions. When the CPU responds to TDN or RDN, checks are performed to see if the transmit underrun error has occurred. The underrun condition is only checked when using the DMA channels.

Upon power up the CPU mode is initialized. General DMA control is covered in Section 4.0. DMA control of the GSC is covered in Section 3.5.4. If DMA is to be used for serving the GSC, it must be configured into the serial channel demand mode and the DMA bit in TSTAT has to be set.

3.6.3 COLLISIONS AND BACKOFF

The actions that are taken by the GSC if a collision occurs while transmitting depend on where the collision occurs. If a collision occurs in CSMA/CD mode following the preamble and BOF flag, the TCDT flag is set and the transmit hardware completes a jam. When this type of collision occurs, there will be no automatic retry at transmission. After the jam, control is returned to the CPU and user software must then initiate whatever actions are necessary for a proper recovery. The possibility that data might have been loaded into or from the GSC deserves special consideration. If these fragments of a message have been passed on to other devices, user software may have to perform some extensive error handling or notification. Before starting a new message, the transmit and receive FIFOs will need to be cleared. If DMA servicing is being used the pointers must also be reinitialized. It should be noted that a collision should never occur after the BOF flag in a well designed system, since the system slot time will likely be less than the preamble length. The occurrence of such a situation is normally due to a station on the link that is not adhering to proper CSMA/CD protocol or is not using the same timings as the rest of the network.

A collision occurring during the preamble or BOF flag is the normal type of collision that is expected. When this type of collision occurs the GSC automatically handles the retransmission attempts for as many as eight tries. If on the eighth attempt a collision occurs, the transmitter is disabled, although the jam and backoff are performed. If enabled, the CPU is then interrupted. The user software should then determine what action to take. The possibilities range from just reporting the error and aborting transmission to reinitializing the serial channel registers and attempt retransmission.

If less than eight attempts are desired TCDCNT can be loaded with some value which will reduce the number of collisions possible before TCDCNT overflows. The value loaded should consist of all 1s as the least significant bits, e.g. 7, 0FH, 3FH. A solid block of 1s is suggested because TCDCNT is used as a mask when generating the random slot number assignment. The TCDCNT register operates by shifting the contents one bit position to the left as each collision is detected. As each shift occurs a 1 is loaded into the LSB. When TCDCNT overflows, GSC operation stops and the CPU is notified by the setting of the TCDT bit which can flag an interrupt.

The amount of time that the GSC has before it must be ready to retransmit after a collision is determined by the mode which is selected. The mode is determined M0 (GMOD.5) and M1 (GMOD.6). If M0 and M1 equal 0,0 (normal backoff) then the minimum period before retransmission will be either the interframe space or the backoff period, whichever is longer. If M0 and M1 equal 1,1 (alternate backoff) then the minimum period before retransmission will be the interframe space plus the backoff period. Both of these are shown in Figure 3.4. Alternate backoff must be enabled if using deterministic resolution. If the GSC is not ready to retransmit by the time its assigned slot becomes available, the slot time is lost and the station must wait until the collision resolution time period has passed.

Instead of waiting for the collision resolution to pass, the transmission could be aborted. The decision to abort is usually dependent on the number of stations on the link and how many collisions have already occurred. The number of collisions can be obtained by examining the register, TCDCNT. The abort is normally implemented by clearing TEN. The new transmission begins by setting TEN and loading TFIFO. The minimum amount of time available to initiate a retransmission would be one interframe space period after the line is sensed as being idle.

As the number of stations approach 256 the probability of a successful transmission decreases rapidly. If there are more than 256 stations involved in the collision there would be no resolution since at least two of the stations will always have the same backoff interval selected.

All the stations monitor the link as long as that station is active, even if not attempting to transmit. This is to ensure that each station always defers the minimum amount of time before attempting a transmission and so that addresses are recognized. However, the collision detect circuitry operates slightly differently.

In normal back-off mode, a transmitting station always monitors the link while transmitting. If a collision is detected one or more of the transmitting stations apply the jam signal and all transmitting stations enter the back-off algorithm. The receiving stations also constantly monitor for a collision but do not take part in the resolution phase. This allows a station to try to transmit in the middle of a resolution period. This in turn may or may not cause another collision. If the new station trying to transmit on the link does so during an unused slot time then there will probably not be a collision. If trying to transmit during a used slot time, then there will probably be a collision. The actions the receiver does take when detecting a collision is to just stop receiving data if data has not been loaded into RFIFO or to stop reception, clear receiver enable (REN) and set the receiver abort flag (RCABT - RSTAT.6).

If deterministic resolution is used, the transmitting stations go through pretty much the same process as in normal back-off, except that the slots are predetermined. All the receivers go through the back-off algorithm and may only transmit during their assigned slot.

3.6.4 SUCCESSFUL ENDING OF TRANSMISSIONS AND RECEPTIONS

In both CSMA/CD and SDLC modes, the TDN bit is set and TEN cleared at the end of a successful transmission. The end of the transmission occurs when the TFIFO is empty and the last byte has been transmitted. In CSMA/CD the user should clear the TCDCNT register after successful transmission.

At the end of a successful reception, the RDN bit is set and GREN is cleared. The end of reception occurs when the EOF flag is detected by the GSC hardware.

3.7 Register Descriptions

ADR0,1,2,3 (95H, 0A5H, 0B5H, 0C5H) - Address Match Registers 0,1,2,3 - Contains the address match values which determines which data will be accepted as valid. In 8 bit addressing mode, a match with any of the four registers will trigger acceptance. In 16 bit addressing mode a match with ADR1:ADR0 or ADR3:ADR2 will be accepted. Addressing mode is determined in GMOD (AL).

AMSK0,1 (0D5H, 0E5H) - Address Match Mask 0,1 - Identifies which bits in ADR0,1 are "don't care" bits. Writing a one to a bit in AMSK0,1 masks out that corresponding bit in ADDR0,1.

BAUD (94H) - GSC Baud Rate Generator - Contains the value of the programmable baud rate. The data rate will equal (frequency of the oscillator)/((BAUD + 1) × (8)). Writing to BAUD actually stores the value in a reload register. The reload register contents are copied into the BAUD register when the Baud register decrements to 00H. Reading BAUD yields the current timer value. A read during GSC operation will give a value that may not be current because the timer could decrement between the time it is read by the CPU and by the time the value is loaded into its destination.

BKOFF (0C4H) - Backoff Timer - The backoff timer is an eight bit count-down timer with a clock period equal to one slot time. The backoff time is used in the CSMA/CD collision resolution algorithm. The user software may read the timer but the value may be invalid as the timer is clocked asynchronously to the CPU. Writing to 0C4H will have no effect.

GMOD (84H)

7	6	5	4	3	2	1	0
XTCLK	M1	M0	AL	CT	PL1	PL0	PR

Figure 3.14. GMOD

GMOD.0 (PR) - Protocol - If set, SDLC protocols with NRZI encoding and SDLC flags are used. If cleared, CSMA/CD link access with Manchester encoding is used. The user software is responsible for setting or clearing this flag.

GMOD.1,2 (PL0,1) - Preamble length

PL1	PL0	LENGTH (BITS)
0	0	0
0	1	8
1	0	32
1	1	64

The length includes the two bit Begin Of Frame (BOF) flag in CSMA/CD but does not include the SDLC flag. In SDLC mode, the BOF is an SDLC flag, otherwise it is two consecutive ones. Zero length is not compatible in CSMA/CD mode. The user software is responsible for setting or clearing these bits.

GMOD.3 (CT) - CRC Type - If set, 32 bit AUTODIN-II-32 is used. If cleared, 16 bit CRC-CCITT is used. The user software is responsible for setting or clearing this flag.

GMOD.4 (AL) - Address Length - If set, 16 bit addressing is used. If cleared, 8 bit addressing is used. In 8 bit mode a match with any of the 4 address registers will be accepted (ADR0, ADR1, ADR2, ADR3). "Don't Care" bits may be masked in ADR0 and ADR1 with AMSK0 and AMSK1. In 16 bit mode, addresses are matched against "ADR1:ADR0" or "ADR3:ADR2". Again, "Don't Care" bits in ADR1:ADR0 can be masked in AMSK1:AMSK0. A received address of all ones will always be recognized in any mode. The user software is responsible for setting or clearing this flag.

GMOD.5,6 (M0,M1) - Mode Select - Two test modes, an optional "alternate backoff" mode, or normal backoff can be enabled with these two bits. The user software is responsible for setting or clearing the mode bits.

M1	M0	Mode
0	0	Normal
0	1	Raw Transmit
1	0	Raw Receive
1	1	Alternate Backoff

In raw receive mode, the receiver operates as normal except that all the bytes following the BOF are loaded into the receive FIFO, including the CRC. The transmitter operates as normal.

In raw transmit mode the transmit output is internally connected to the receiver input. The internal connection is not at the actual port pin, but inside the port latch. All data transmitted is done without a preamble, flag or zero bit insertion, and without appending a CRC. The receiver operates as normal. Zero bit deletion is performed.

In alternate backoff mode the standard backoff process is modified so the the backoff is delayed until the end of the IFS. This should help to prevent collisions constantly happening because the IFS time is usually larger than the slot time.

GMOD.7 (XTCLK) - External Transmit Clock - If set an external 1X clock is used for the transmitter. If cleared the internal baud rate generator provides the transmit clock. The input clock is applied to P1.3 ($\overline{T \times C}$). The user software is responsible for setting or clearing this flag. External receive clock is enabled by setting PCON.3.

IFS (0A4H) - Interframe Spacing - Determines the number of bit times separating transmitted frames in CSMA/CD. A bit time is equal to 1/baud rate. Only even interframe space periods can be used. The number written into this register is divided by two and loaded in the most significant seven bits. Complete interframe space is obtained by counting this seven bit number down to zero twice. A user software read of this register will give a value where the seven most significant bits gives the current count value and the least significant bit shows a one for the first count-down and a zero for the second count. The value read may not be valid as the timer is clocked in periods not necessarily associated with the CPU read of IFS. Loading this register with zero results in 256 bit times.

MYSLOT (0F5H) - Slot Address Register

7	6	5	4	3	2	1	0
DCJ	DCR	SA5	SA4	SA3	SA2	SA1	SA0
SAn = SLOT ADDRESS (BITS 5 – 0)							

Figure 3.15. MYSLOT

MYSLOT.0, 1, 2, 3, 4, 5 - Slot Address - The six address bits choose 1 of 64 slot addresses. Address 63 has the highest priority and address 1 has the lowest. A value of zero will prevent a station from transmitting during the collision resolution period by waiting until all the possible slot times have elapsed. The user software normally initializes this address in the operating software.

MYSLOT.6 (DCR) - Deterministic Collision Resolution Algorithm - When set, the alternate collision resolution algorithm is selected. Retriggering of the IFS on reappearance of the carrier is also disabled. When using this feature Alternate Backoff Mode must be selected and several other registers must be initialized. User software must initialize TCDCNT with the maximum number of slots that are most appropriate for a particular application. The PRBS register must be set to all ones. This disables the PRBS by freezing it's contents at 0FFH. The backoff timer is used to count down the number of slots based on the slot timer value setting the period of one slot. The user software is responsible for setting or clearing this flag.

MYSLOT.7 (DCJ) - D.C. Jam - When set selects D.C. type jam, when clear, selects A.C. type jam. The user software is responsible for setting or clearing this flag.

PCON (087H)

7	6	5	4	3	2	1	0
SMOD	ARB	REQ	GAREN	XRCLK	GFIEN	PD	IDL

PCON contains bits for power control, LSC control, DMA control, and GSC control. The bits used for the GSC are PCON.2, PCON.3, and PCON.4.

PCON.2 (GFIEN) - GSC Flag Idle Enable - Setting GFIEN to a 1 caused idle flags to be generated between transmitted frames in SDLC mode. SDLC idle flags consist of 01111110 flags creating the sequence 01111110011111110 . . . . . 011111110. A possible side effect of enabling GFIEN is that the maximum possible latency from writing to TFIFO until the first bit is transmitted increased from approximately 2 bit-times to around 8 bit-times. GFIEN has no effect with CSMA/CD.

PCON.3 (XRCLK) - GSC External Receive Clock Enable - Writing a 1 to XRCLK enables an external clock to be applied to pin 5 (Port 1.4). The external clock is used to determine when bits are loaded into the receiver.

PCON.4 (GAREN) - GSC Auxiliary Receiver Enable Bit - This bit needs to be set to a 1 to enable the reception of back-to-back SDLC frames. A back-to-back SDLC frame is when the EOF and BOF is shared between two sequential frames intended for the same station on the link. If GAREN contains a 0 then the receiver will be disabled upon reception of the EOF and by the time user software re-enables the receiver the first bit(s) may have already passed, in the case of back-to-back frames. Setting GAREN to a 1, prevents the receiver from being disabled by the EOF but GREN will be cleared and can be checked by user software to determine that an EOF has been received. GAREN has no effect if the GSC is in CSMA/CD mode.

PRBS (0E4H) - Pseudo-Random Binary Sequence - This register contains a pseudo-random number to be used in the CSMA/CD backoff algorithm. The number is generated by using a feedback shift register clocked by the CPU phase clocks. Writing all ones to the PRBS will freeze the value at all ones. Writing any other value to it will restart the PRBS generator. The PRBS is initialized to all zero's during RESET. A read of location 0E4H will not necessarily give the seed used in the backoff algorithm because the PRBS counters are clocked by internal CPU phase clocks. This means the contents of the PRBS may have been altered between the time when the seed was generated and before a READ has been internally executed.

RFIFO (0F4H) - Receive FIFO - RFIFO is a 3 byte buffer that is loaded each time the GSC receiver has a byte of data. Associated with RFIFO is a pointer that is automatically updated with each read of the FIFO. A read of RFIFO fetches the oldest data in the FIFO.

RSTAT (0E8H) - Receive Status Register

7	6	5	4	3	2	1	0
OR	RCABT	AE	CRCE	RDN	RFNE	GREN	HABEN

Figure 3.16. RSTAT

RSTAT.0 (HABEN) - Hardware Based Acknowledge Enable - If set, enables the hardware based acknowledge feature. The user software is responsible for setting or clearing this flag.

RSTAT.1 (GREN) - Receiver Enable - When set, the receiver is enabled to accept incoming frames. This also clears RDN, CRCE, AE, RCABT, and the receive FIFO. It is cleared by the receiver at the end of a reception or if any errors occurred. The user software is responsible for setting this flag and the GSC or user software can clear it. The status of GREN has no effect on whether the receiver detects a collision in CSMA/CD mode as the receiver input circuitry always monitors the receive pin.

RSTAT.2 (RFNE) - Receive FIFO Not Empty - If set, indicates that the receive FIFO contains data. The receive FIFO is a three byte buffer into which the receive data is loaded. A CPU read of the FIFO retrieves the oldest data and automatically updates the FIFO pointers. Setting GREN to a one will clear the receive FIFO. The status of this flag is controlled by the GSC. It is cleared if user empties receive FIFO.

RSTAT.3 (RDN) - Receive Done - If set, indicates the successful completion of a receiver operation. Will not be set if a CRC, alignment, abort, or FIFO overrun error occurred. The status of this flag is controlled by the GSC.

RSTAT.4 (CRCE) - CRC Error - If set, indicates that a properly aligned frame was received with a mismatched CRC. The status of this flag is controlled by the GSC.

RSTAT.5 (AE) - Alignment Error - If set, indicates that the line went idle when the receiver shift register was not full and the resulting CRC was bad in the CSMA/CD mode. If a correct CRC was valid then AE is not set. In SDLC mode, AE indicates that a non-byte-aligned flag was received. The status of this flag is controlled by the GSC.

RSTAT.6 (RCABT) - Receiver Collision/Abort Detect - If set, indicates that a collision was detected after data had been loaded into the receive FIFO in CSMA/CD mode. In SDLC mode, RCABT indicates that 7 consecutive ones were detected prior to the end flag but after data has been loaded into the receive FIFO. The status of this flag is controlled by the GSC.

RSTAT.7 (OVR) - Overrun - If set, indicates that the receive FIFO was full and new shift register data was written into it. The setting of this flag is controlled by the GSC and it is cleared by user software.

SLOTTM (0BH) - Slot Time - Determines the length of the slot time used in CSMA/CD. A slot time equals (256 − SLOTTM) × (1 / baud rate). A read of SLOTTM will give the value of the slot time timer but the value may be invalid as the timer is clocked asynchronously to the CPU. Loading SLOTTM with 0 results in 256 bit times.

TCDCNT (0D4H) - Transmit Collision Detect Count - Contains the number of collisions that have occurred if probabilistic CSMA/CD is used. The user software must clear this register before transmitting a new frame so that the GSC backoff hardware can accurately distinguish a new frame from a retransmit attempt.

In deterministic backoff mode, TCDCNT is used to hold the maximum number of slots.

TFIFO (85H) - GSC Transmit FIFO - TFIFO is a 3 byte buffer with an associated pointer that is automatically updated for each write by user software. Writing a byte to TFIFO loads the data into the next available location in the transmit FIFO. Setting TEN clears the transmit FIFO so the transmit FIFO should not be written to prior to setting TEN. If TEN is already set transmission begins as soon as data is written to TFIFO.

TSTAT (0D8) - Transmit Status Register

7	6	5	4	3	2	1	0
LNI	NOACK	UR	TCDT	TDN	TFNF	TEN	DMA

Figure 3.17. TSTAT

TSTAT.0 (DMA) - DMA Select - If set, indicates that DMA channels are used to service the GSC FIFO's and GSC interrupts occur on TDN and RDN, and also enables UR to become set. If cleared, indicates that the GSC is operating in its normal mode and interrupts occur on TFNF and RFNE. For more information on DMA servicing please refer to the DMA section on DMA serial demand mode (4.2.2.3). The user software is responsible for setting or clearing this flag.

TSTAT.1 (TEN) - Transmit Enable - When set causes TDN, UR, TCDT, and NOACK flag to be reset and the TFIFO cleared. The transmitter will clear TEN after a successful transmission, a collision during the data, CRC, or end flag. The user software is responsible for setting but the GSC or user software may clear this flag. If cleared during a transmission the GSC transmit pin goes to a steady state high level. This is the method used to send an abort character in SDLC. Also $\overline{DEN}$ is forced to a high level. The end of transmission occurs whenever the TFIFO is emptied.

TSTAT.2 (TFNF) - Transmit FIFO not full - When set, indicates that new data may be written into the transmit FIFO. The transmit FIFO is a three byte buffer that loads the transmit shift register with data. The status of this flag is controlled by the GSC.

TSTAT.3 (TDN) - Transmit Done - When set, indicates the successful completion of a frame transmission. If HABEN is set, TDN will not be set until the end of the IFS following the transmitted message, so that the acknowledge can be checked. If an acknowledge is expected and not received, TDN is not set. An acknowledge is not expected following a broadcast or multi-cast packet. The status of this flag is controlled by the GSC.

TSTAT.4 (TCDT) - Transmit Collision Detect - If set, indicates that the transmitter halted due to a collision. It is set if a collision occurs during the data or CRC or if there are more than eight collisions. The status of this flag is controlled by the GSC.

TSTAT.5 (UR) - Underrun - If set, indicates that in DMA mode the last bit was shifted out of the transmit register and that the DMA byte count did not equal zero. When an underrun occurs, the transmitter halts without sending the CRC or the end flag. The status of this flag is controlled by the GSC.

TSTAT.6 (NOACK) - No Acknowledge - If set, indicates that no acknowledge was received for the previous frame. Will be set only if HABEN is set and no acknowledge is received prior to the end of the IFS. NOACK is not set following a broadcast or a multicast packet. The status of this flag is controlled by the GSC.

TSTAT.7 (LNI) - Line Idle - If set, indicates the receive line is idle. In SDLC protocol it is set if 15 consecutive ones are received. In CSMA/CD protocol, line idle is set if no transitions occur on GR×D for approximately 1.6 bit times after a required transition. LNI is cleared after a transition on GR×D. The status of this flag is controlled by the GSC.

3.8 Serial Backplane vs. Network Environment

The C152 GSC port is intended to fulfill the needs of both serial backplane environment and the serial communication network environment. The serial backplane is where typically, only processor to processor communications take place within a self contained box. The communication usually only encompasses those items which are necessary to accomplish the dedicated task for the box. In these types of applications there may not be a need for line drivers as the distance between the transmitter and receiver is relatively short. The network environment; however, usually requires transmission of data over large distances and requires drivers and/or repeaters to ensure the data is received on both ends.

4.0 DMA Operation

The C152 contains DMA (Direct Memory Accessing) logic to perform high speed data transfers between any two of

Internal Data RAM
Internal SFRs
External Data RAM

If external RAM is involved, the Port 2 and Port 0 pins are used as the address/data bus, and $\overline{RD}$ and $\overline{WR}$ signals are generated as required.

Hardware is also implemented to generate a Hold Request signal and await a Hold Acknowledge response before commencing a DMA that involves external RAM.

Alternatively, the Hold/Hold Acknowledge hardware can be programmed to accept a Hold Request signal from an external device and generate a Hold Acknowledge signal in response, to indicate to the requesting device that the C152 will not commence a DMA to or from external RAM while the Hold Request is active.

4.1 DMA with the 80C152

The C152 contains two identical general purpose 8-bit DMA channels with 16-bit addressability: DMA0 and DMA1. DMA transfers can be executed by either channel independent of the other, but only by one channel at a time. During the time that a DMA transfer is being executed, program execution is suspended. A DMA transfer takes one machine cycle (12 oscillator

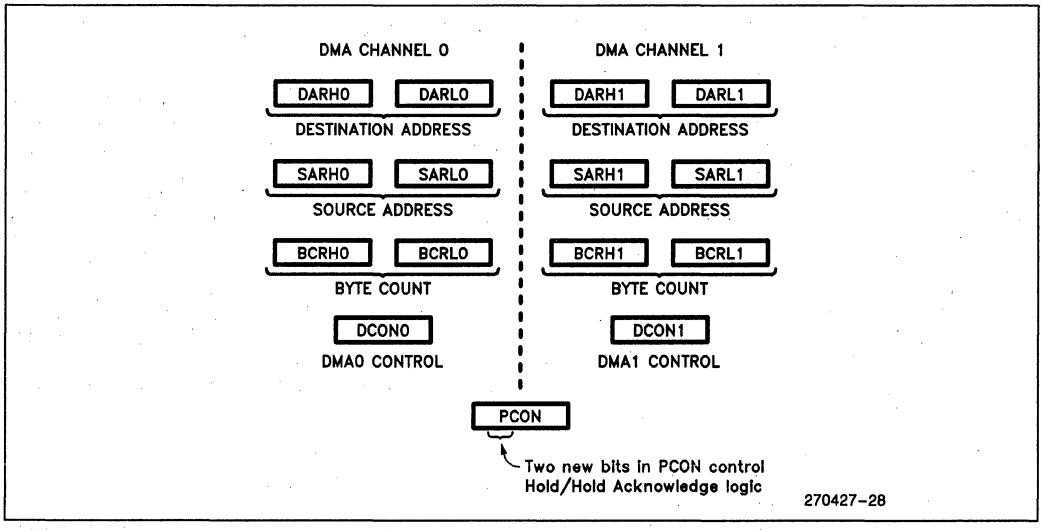


Figure 4.1. DMA Registers

periods) per byte transferred, except when the destination and source are both in External Data RAM. In that case the transfer takes two machine cycles per byte. The term DMA Cycle will be used to mean the transfer of a single data byte, whether it takes 1 or 2 machine cycles.

Associated with each channel are seven SFRs, shown in Figure 4.1. SARLn and SARHn holds the low and high bytes of the source address. Taken together they form a 16-bit Source Address Register. DARLn and DARHn hold the low and high bytes of the destination address, and together form the Destination Address Register. BCRLn and BCRHn hold the low and high bytes of the number of bytes to be transferred, and together form the Byte Count Register. DCONn contains control and flag bits.

Two bits in DCONn are used to specify the physical destination of the data transfer. These bits are DAS (Destination Address Space) and IDA (Increment Destination Address). If DAS = 0, the destination is in data memory external to the C152. If DAS = 1, the destination is internal to the C152. If DAS = 1 and IDA = 0, the internal destination is a Special Function Register (SFR). If DAS = 1 and IDA = 1, the internal destination is in the 256-byte data RAM.

In any case, if IDA = 1, the destination address is automatically incremented after each byte transfer. If IDA = 0, it is not.

Two other bits in DCONn specify the physical source of the data to be transferred. These are SAS (Source Address Space) and ISA (Increment Source Address). If SAS = 0, the source is in data memory external to the C152. If SAS = 1, the source is internal. If SAS = 1 and ISA = 0, the internal source is an SFR. If SAS = 1 and ISA = 1, the internal source is in the 256-byte data RAM.

In any case, if ISA = 1, the source address is automatically incremented after each byte transfer. If ISA = 0, it is not.

The functions of these four control bits are summarized below:

DAS	IDA	Destination	Auto-Increment
0	0	External RAM	no
0	1	External RAM	yes
1	0	SFR	no
1	1	Internal RAM	yes
SAS	**ISA**	**Source**	**Auto-Increment**
0	0	External RAM	no
0	1	External RAM	yes
1	0	SFR	no
1	1	Internal RAM	yes

There are four modes in which the DMA channel can operate. These are selected by the bits DM and TM (Demand Mode and Transfer Mode) in DCONn:

DM	TM	Operating Mode
0	0	Alternate Cycles Mode
0	1	Burst Mode
1	0	Serial Port Demand Mode
1	1	External Demand Mode

The operating modes are described below.

4.1.1 ALTERNATE CYCLE MODE

In Alternate Cycles Mode the DMA is initiated by setting the GO bit in DCONn. Following the instruction that set the GO bit, one more instruction is executed, and then the first data byte is transferred from the source address to the destination address. Then another instruction is executed, and then another byte of data is transferred, and so on in this manner.

Each time a data byte is transferred, BCRn (Byte Count Register for DMA Channel n) is decremented. When it reaches 0000H, on-chip hardware clears the GO bit and sets the DONE bit, and the DMA ceases. The DONE bit flags an interrupt.

4.1.2 BURST MODE

Burst Mode differs from Alternate Cycles mode only in that once the data transfer has begun, program execution is entirely suspended until BCRn reaches 0000H, indicating that all data bytes that were to be transferred have been transferred. The interrupt control hardware remains active during the DMA, so interrupt flags may get set, but since program execution is suspended, the interrupts will not be serviced while the DMA is in progress.

4.1.3 SERIAL PORT DEMAND MODE

In this mode the DMA can be used to service the Local Serial Channel (LSC) or the Global Serial Channel (GSC).

In Serial Port Demand Mode the DMA is initiated by any of the following conditions, if the GO bit is set:

Source Address = SBUF .AND. RI = 1
Destination Address = SBUF .AND. TI = 1
Source Address = RFIFO .AND. RFNE = 1
Destination Address = TFIFO .AND. TFNF = 1

Each time one of the above conditions is met, one DMA Cycle is executed; that is, one data byte is transferred from the source address to the destination address. On-chip hardware then clears the flag (RI, TI, RFNE, or TFNF) that initiated the DMA, and decrements BCRn. Note that since the flag that initiated the DMA is cleared, it will not generate an interrupt unless DMA servicing is held off or the byte count equals 0. DMA servicing may be held off when alternate cycle is being used or by the status of the HOLD/HLDA logic. In these situations the interrupt for the LSC may occur before the DMA can clear the RI or TI flag. This is because the LSC is serviced according to the status of RI and TI, whether or not the DMA channels are being used for the transferring of data. The GSC does not use RFNE or TFNF flags when using the DMA channels so these do not need to be disabled. When using the DMA channels to service the LSC it is recommended that the interrupts (RI and TI) be disabled. If the decremented BCRn is 0000H, on-chip hardware then clears the GO bit and sets the DONE bit. The DONE bit flags an interrupt.

4.1.4 EXTERNAL DEMAND MODE

In External Demand Mode the DMA is initiated by one of the External Interrupt pins, provided the GO bit is set. $\overline{INT0}$ initiates a Channel 0 DMA, and $\overline{INT1}$ initiates a Channel 1 DMA.

If the external interrupt is configured to be transition-activated, then each 1-to-0 transition at the interrupt pin sets the corresponding external interrupt flag, and generates one DMA Cycle. Then, BCRn is decremented. No more DMA Cycles take place until another 1-to-0 transition is seen at the external interrupt pin. If the decremented BCRn = 0000H, on-chip hardware clears the GO bit and sets the DONE bit. If the external interrupt is enabled, it will be serviced.

If the external interrupt is configured to be level-activated, then DMA Cycles commence when the interrupt pin is pulled low, and continue for as long as the pin is held low and BCRn is not 0000H. If BCRn reaches 0 while the interrupt pin is still low, the GO bit is cleared, the DONE bit is set, and the DMA ceases. If the external interrupt is enabled, it will be serviced.

If the interrupt pin is pulled up before BCRn reaches 0000H, then the DMA ceases, but the GO bit is still 1 and the DONE bit is still 0. An external interrupt is not generated in this case, since in level-activated mode, pulling the pin to a logical 1 clears the interrupt flag. If the interrupt pin is then pulled low again, DMA transfers will continue from where they were previously stopped.

The timing for the DMA Cycle in the transition-activated mode, or for the first DMA Cycle in the level-activated mode is as follows: If the 1-to-0 transition is

detected before the final machine cycle of the instruction in progress, then the DMA commences as soon as the instruction in progress is completed. Otherwise, one more instruction will be executed before the DMA starts. No instruction is executed during any DMA Cycle.

4.2 Timing Diagrams

Timing diagrams for single-byte DMA transfers are shown in Figures 4.2 through 4.5 for four kinds of DMA Cycles: internal memory to internal memory, internal memory to external memory, external memory to internal memory, and external memory to external memory. In each case we assume the C152 is executing out of external program memory. If the C152 is executing out of internal program memory, then $\overline{PSEN}$ is inactive, and the Port 0 and Port 2 pins emit P0 and P2 SFR data. If External Data Memory is involved, the Port 0 and Port 2 pins are used as the address/data bus, and $\overline{RD}$ and/or $\overline{WR}$ signals are generated as needed, in the same manner as in the execution of a MOVX @DPTR instruction.

4.3 Hold/Hold Acknowledge

Two operating modes of Hold/Hold Acknowledge logic are available, and either or neither may be invoked by software. In one mode, the C152 generates a Hold Request signal and awaits a Hold Acknowledge response before commencing a DMA that involves external RAM. This is called the Requester Mode.

In the other mode, the C152 accepts a Hold Request signal from an external device and generates a Hold Acknowledge signal in response, to indicate to the requesting device that the C152 will not commence a DMA to or from external RAM while the Hold Request is active. This is called the Arbiter mode.

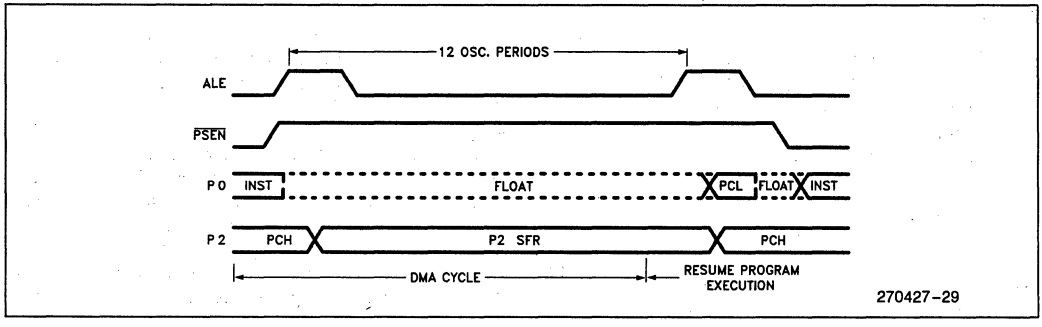


Figure 4.2. DMA Transfer from Internal Memory to Internal Memory

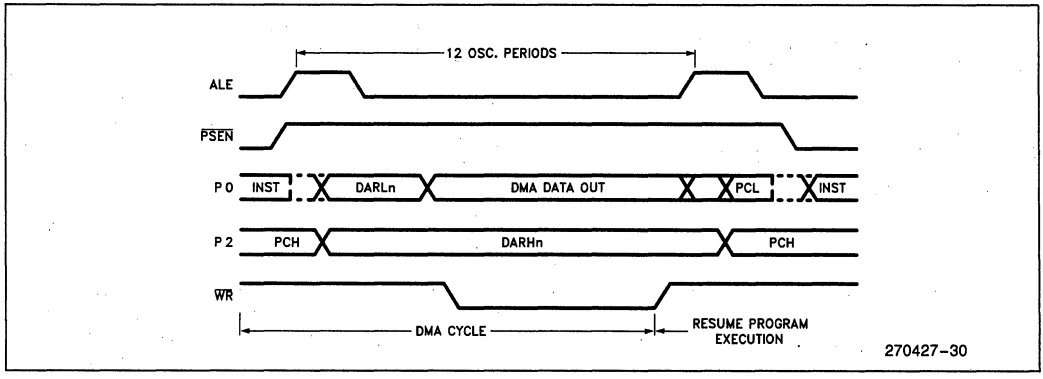


Figure 4.3. DMA Transfer from Internal Memory to External Memory

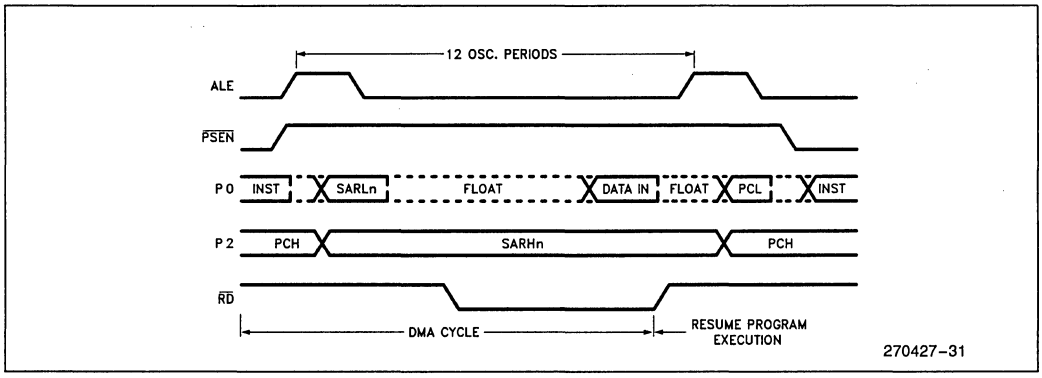


Figure 4.4. DMA Transfer from External Memory to Internal Memory

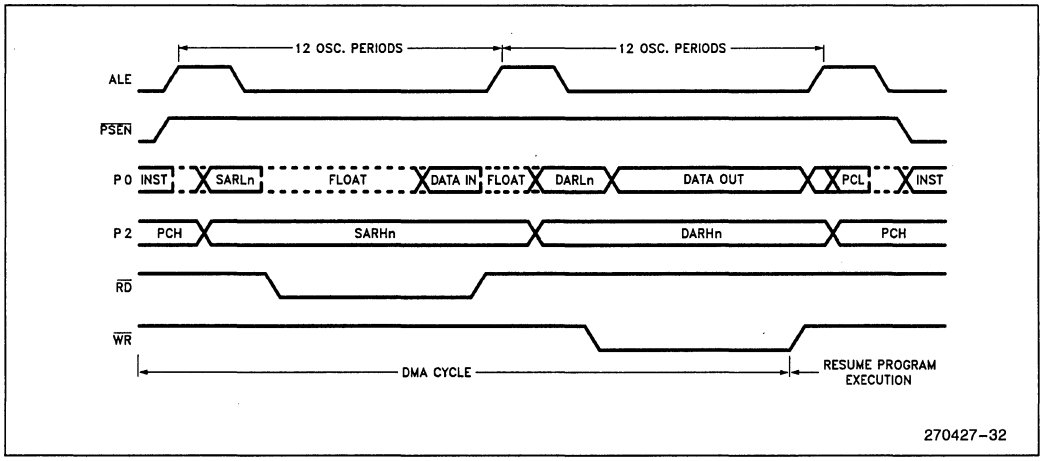


Figure 4.5. DMA Transfer from External Memory to External Memory

4.3.1 REQUESTER MODE

The Requester Mode is selected by setting the control bit REQ, which resides in PCON. In that mode, when the C152 wants to do a DMA to External Data Memory, it first generates a Hold Request signal, $\overline{\text{HLD}}$, and waits for a Hold Acknowledge signal, $\overline{\text{HLDA}}$, before commencing the DMA operation. Note that program execution continues while $\overline{\text{HLDA}}$ is awaited. The DMA is not begun until a logical 0 is detected at the $\overline{\text{HLDA}}$ pin. Then, once the DMA has begun, it goes to completion regardless of the logic level at $\overline{\text{HLDA}}$.

The protocol is activated only for DMAs (not for program fetches or MOVX operations), and only for DMAs to or from External Data Memory. If the data destination and source are both internal to the C152, the $\overline{\text{HLD}}$/$\overline{\text{HLDA}}$ protocol is not used.

The $\overline{\text{HLD}}$ output is an alternate function of port pin P1.5, and the $\overline{\text{HLDA}}$ input is an alternate function of port pin P1.6.

4.3.2 ARBITER MODE

For DMAs that are to be driven by some device other than the C152, a different version of the Hold/Hold Acknowledge protocol is available. In this version, the device which is to drive the DMA sends a Hold Request signal, $\overline{\text{HLD}}$, to the C152. If the C152 is currently performing a DMA to or from External Data Memory, it will complete this DMA before responding to the Hold Request. When the C152 responds to the Hold Request, it does so by activating a Hold Acknowledge signal, $\overline{\text{HLDA}}$. This indicates that the C152 will not commence a new DMA to or from External Data Memory while $\overline{\text{HLD}}$ remains active.

Note that in the Arbiter Mode the C152 does not suspend program execution at all, even if it is executing from external program memory. It does not surrender use of its own bus.

The Hold Request input, $\overline{\text{HLD}}$, is at P1.5. The Hold Acknowledge output, $\overline{\text{HLDA}}$, is at P1.6. This

version of the Hold/Hold Acknowledge feature is selected by setting the control bit ARB in PCON.

The functions of the ARB and REQ bits in PCON, then, are

ARB	REQ	Hold/Hold Acknowledge Logic
0	0	Disabled
0	1	C152 generates $\overline{HLD}$, detects $\overline{HLDA}$
1	0	C152 detects $\overline{HLD}$, generates $\overline{HLDA}$
1	1	Invalid

4.3.3 USING THE HOLD/HOLD ACKNOWLEDGE

The $\overline{HOLD}/\overline{HOLDA}$ logic only affects DMA operation with external RAM and doesn't affect other operations with external RAM, such as MOVX instruction.

Figure 4.6 shows a system in which two 83C152s are sharing a global RAM. In this system, both CPUs are executing from internal ROM. Neither CPU uses the bus except to access the shared RAM, and such accesses are done only through DMA operations, not by MOVX instructions.

One CPU is programmed to be the Arbiter and the other, to be the Requester. The ALE Switch selects which CPU's ALE signal will be directed to the address latch. The Arbiter's ALE is selected if $\overline{HLDA}$ is high, and the Requester's ALE is selected if $\overline{HLDA}$ is low.

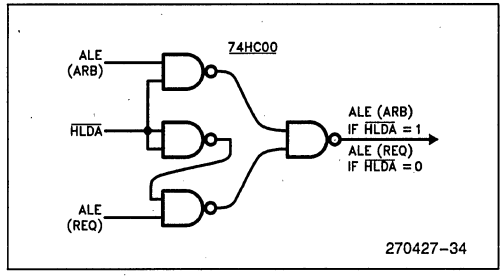


Figure 4.7. ALE Switch Select

The ALE Switch logic can be implemented by a single 74HC00, as shown in Figure 4.7.

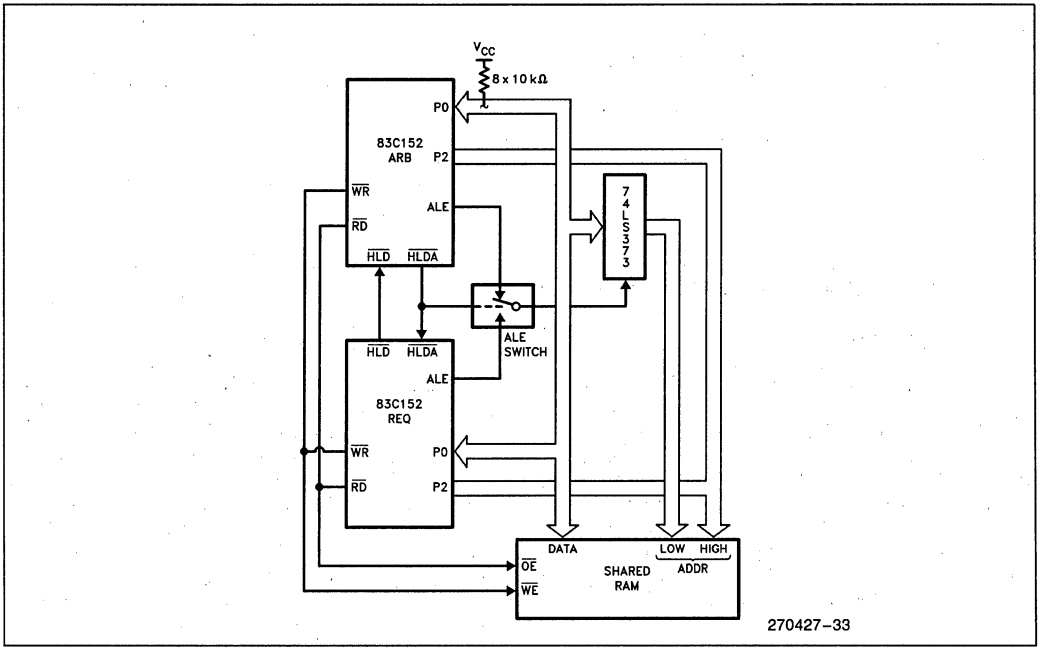


Figure 4.6. Two 83C152s Sharing External RAM

4.3.4 INTERNAL LOGIC OF THE ARBITER

The internal logic of the arbiter is shown in Figure 4.8. In operation an input low at $\overline{HLD}$ sets Q2 if the arbiter's internal signal DMXRQ is low. DMXRQ is the arbiter's "DMA to XRAM Request". Setting Q2 activates $\overline{HLDA}$ through Q3. Q2 being set also disables any DMAs to XRAM that the arbiter might decide to do during the requester's DMA.

Figure 4.9 shows the minimum response time, 4 to 7 CPU oscillator periods, between a transition at the $\overline{HLD}$ input and the response at $\overline{HLDA}$.

When the arbiter wants to DMA the XRAM, it first activates DMXRQ. This signal prevents Q2 from being set if it is not already set. An output low from Q2 enables the arbiter to carry out its DMA to XRAM, and maintains an output high at $\overline{HLDA}$. When the arbiter completes its DMA, the signal DMXRQ goes to O, which enables Q2 to accept signals from the $\overline{HLD}$ input again.

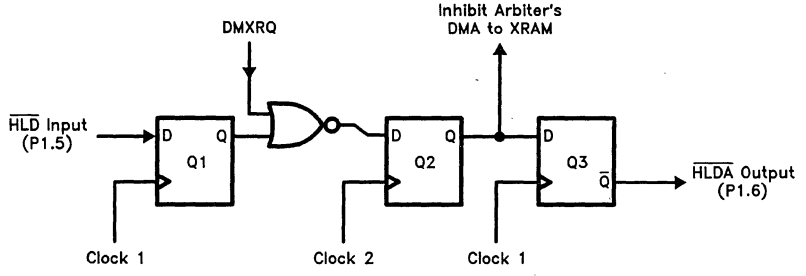


Figure 4.8. Internal Logic of the Arbiter

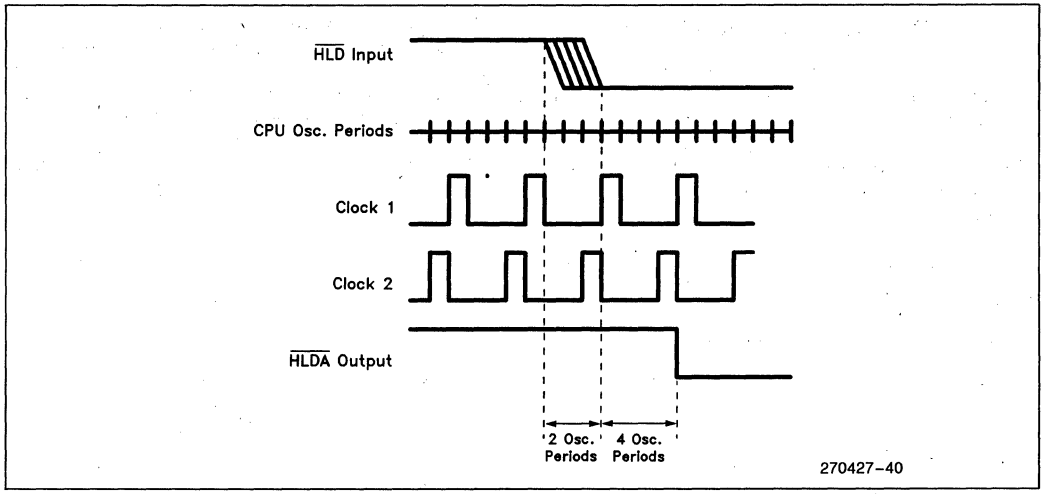


Figure 4.9. Minimum $\overline{HLD}$/$\overline{HLDA}$ Response Time

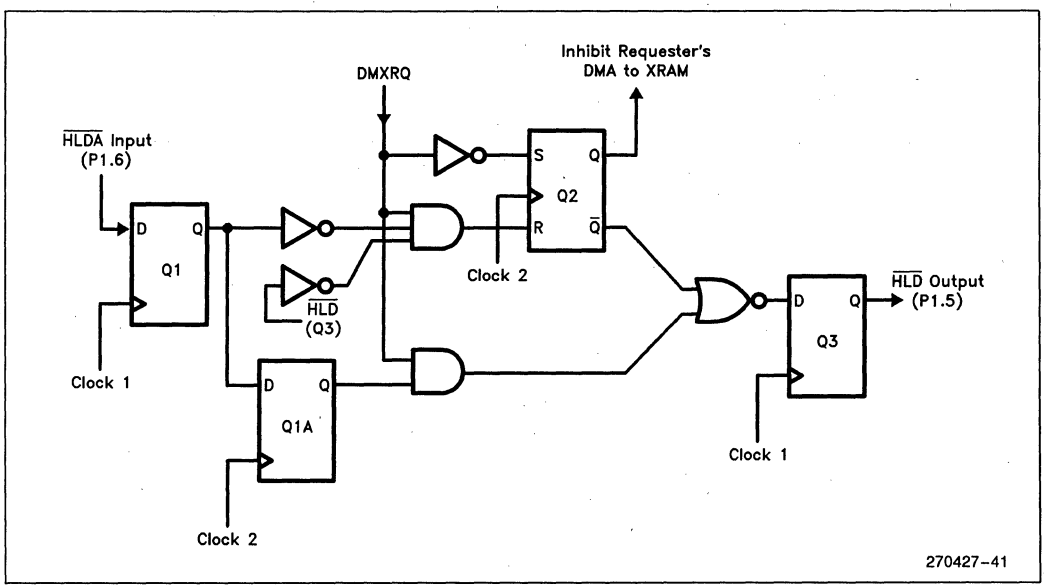


Figure 4.10. Internal Logic of the Requester (Clock 1 and Clock 2 are Shown in Figure 4.9)

4.3.5 Internal Logic of the Requester

The internal logic of the requester is shown in Figure 4.10. Initially, the requester's internal signal DMXRQ (DMA to XRAM Request) is at 0, so Q2 is set and the $\overline{HLD}$ output is high. As long as Q2 stays set, the requester is inhibited from starting any DMA to XRAM.

When the requester wants to DMA the XRAM, it first activates DMXRQ. This signal enables Q2 to be cleared (but doesn't clear it), and, if $\overline{HLDA}$ is high, also activates the $\overline{HLD}$ output.

A 1-to-0 transition from $\overline{HLDA}$ can now clear Q2, which will enable the requester to commence its DMA to XRAM. Q2 being low also maintains an output low at $\overline{HLD}$. When the DMA is completed, DMXRQ goes to 0, which sets Q2 and de-activates $\overline{HLD}$.

Only DMXRQ going to 0 can set Q2. That means once Q2 gets cleared, enabling the requester's DMA to proceed, the arbiter has no way to stop the requester's DMA in progress. At this point, de-activating $\overline{HLDA}$ will have no effect on the requester's use of the bus. Only the requester itself can stop the DMA in progress, and when it does, it de-activates both DMXRQ and $\overline{HLD}$.

If the DMA is in alternate cycles mode, then each time a DMA cycle is completed DMXRQ goes to 0, thus de-activating $\overline{HLD}$. Once $\overline{HLD}$ has been de-activated, it can't be re-asserted till after $\overline{HLDA}$ has been seen to go high (through flip-flop Q1A). Thus every time the DMA is suspended to allow an instruction cycle to proceed, the requester gives up the bus and must renew the request and receive another acknowledge before another DMA cycle to XRAM can proceed. Obviously in this case, the "alternate cycles" mode may consist of single DMA cycles separated by any number of instruction cycles, depending on how long it takes the requester to regain the bus.

A channel 1 DMA in progress will always be overridden by a DMA request of any kind from channel 0. If a channel 1 DMA to XRAM is in progress and is overridden by a channel 0 DMA which does not require the bus, DMXRQ will go to 0 during the channel 0 DMA, thus de-activating $\overline{HLD}$. Again, the requester must renew its request for the bus, and must receive a new 1-to-0 transition in $\overline{HLDA}$ before channel 1 can continue its DMA to XRAM.

4.4 DMA Arbitration

The DMA Arbitration described in this section is not arbitration between two devices wanting to access a shared RAM, but on-chip arbitration between the two DMA channels on the 8XC152.

The 8XC152 provides two DMA channels, either of which may be called into operation at any time in response to real time conditions in the application circuit. Since a DMA cycle always uses the 8XC152's internal bus, and there's only one internal bus, only one DMA channel can be serviced during a single DMA cycle. Executing program instructions also requires the internal bus, so program execution will also be suspended in order for a DMA to take place.

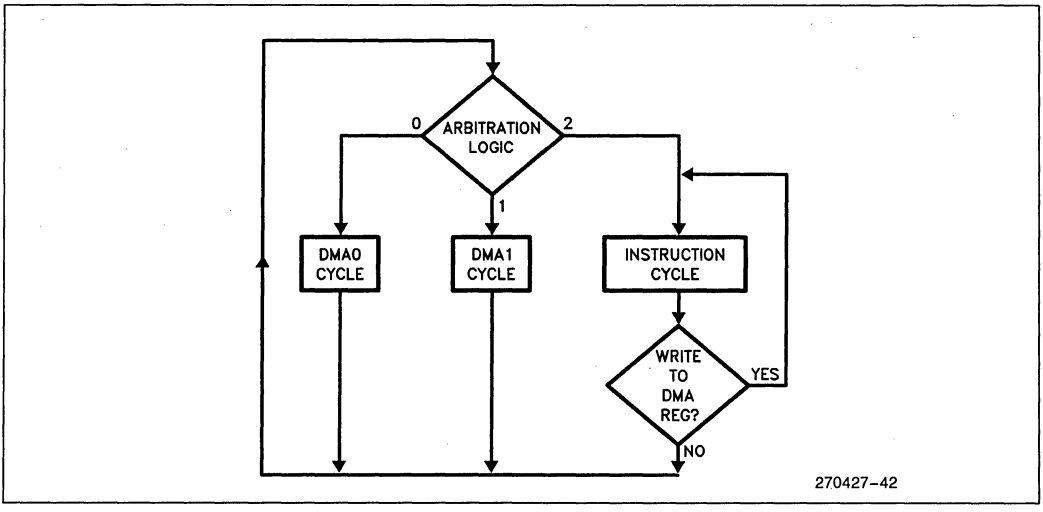


Figure 4.11. Internal Bus Usage

Figure 4.11 shows the three tasks to which the internal bus of the 8XC152 can be dedicated. In this figure, Instruction Cycle means the complete execution of a single instruction, whether it takes 1, 2 or 4 machine cycles. DMA Cycle means the transfer of a single data byte from source to destination, whether it takes 1 or 2 machine cycles. Each time a DMA Cycle or an Instruction Cycle is executed, on-chip arbitration logic determines which type of cycle is to be executed next.

Note that when an instruction is executed, if the instruction wrote to a DMA register (defined in Figure 4.1 but excluding PCON), then another instruction is executed without further arbitration. Therefore, a single write or a series of writes to DMA registers will prevent a DMA from taking place, and will continue to prevent a DMA from taking place until at least one instruction is executed which does not write to any DMA register.

The logic that determines whether the next cycle will be a DMA0 cycle, a DMA1 cycle, or an Instruction Cycle is shown in Figure 4.12 as a pseudo-HLL function. The statements in Figure 4.12 are executed sequentially unless an "if" condition is satisfied, in which case the corresponding "return" is executed and the remainder of the function is not. The return value of 0, 1, or 2 is passed to the arbitration logic block in Figure 4.11 to determine which exit path from the block is used.

The return value is based on the condition of the GO bit for each channel, and on the value returned by another function, named mode__logic ( ). The algorithm for mode__logic ( ) is the same for both channels. The function is shown in Figure 4.13 as a pseudo-HLL function, mode__logic (n), where n = 0 when the function is invoked for DMA channel 0, and n = 1 when it's invoked for DMA channel 1. The value returned by this function is either 0 or 1, and will be passed on to the DMA arbitration logic in Figure 4.12.

Note that the arbitration logic as shown in Figure 4.12 always gives precedence to channel 0 over channel 1. If GO0 is set and mode__logic (0) returns a 1, then a DMA0 cycle is called without further reference to the situation in channel 1. That is not to say a DMA1 Cycle will be interrupted once it has begun. Once a cycle has begun, be it an Instruction Cycle or a DMA Cycle, it will be completed without interruption.

The statements in mode__logic (n), Figure 4.13, are executed sequentially until an "if" condition, based on the DMA mode programmed into DCONn, is satisfied. For example, if the channel is configured to Burst mode, then the first if-condition is satisfied, so the "return 1" expression is executed and the remainder of the function is not.

```
arbitration_logic:

    if (GO0 = 1 .AND. mode_logic(0) = 1) return 0;

    if (GO1 = 1 .AND. mode_logic(1) = 1) return 1;

    else return 2;

    end arbitration_logic;
```

Figure 4.12. DMA Arbitration Logic

```
mode_logic(n):
    if (DCONn indicates burst_mode) return 1;
    if (DCONn indicates extern_demand_mode)
        {
        if (demand_flag = 1) return 1;
        else return 0;
        }
    if (DCONn indicates SP_demand_mode)
        {
        if (SARn = SBUF .AND. RI = 1) return 1;
        if (DARn = SBUF .AND. TI = 1) return 1;
        if (SARn = RFIFO .AND. RFNE = 1) return 1;
        if (DARn = TFIFO .AND. TFNF = 1 .AND.
         previous_cycle = instruction_cycle) return 1;
        else return 0;
        }
    if (DCONn indicates alt_cycles_mode)
        {
        if (DCONm indicates .NOT. alt_cycles_mode
            .OR. GOm = 0)
            {
            if (previous_cycle = instruction_cycle)
                return 1;
            else return 0;
            }
        if (previous_cycle = instruction_cycle
            .AND. previous_dma_cycle = .NOT. DMAn)
            return 1;
        }
    return 0;
    end mode_logic(n);
```

Figure 4.13. DMA Mode Logic

If the channel is configured to External Demand mode, then the first if-condition is not satisfied but the second one is. In that case the block of statements following that if-condition and delimited by {...} is executed: if the demand flag (IEO for channel 0 and IE1 for channel 1) is set, the "return 1" expression is executed and the remainder of the function is not. If the demand flag is not set, the "return 0" expression is executed and the remainder of the function is not.

If the channel is configured to Serial Port Demand mode, the source and destination addresses, SARn and DARn, have to be checked to see which Serial Port buffer is being addressed, and whether its demand flag is set.

SARn refers to the 16-bit source address for "this channel." Note that the condition

SARn = SBUF

cannot be true unless the SAS and ISA bits in DCONn are configured to select SFR space. If SARn is numerically equal to the address of SBUF (99H), and SAS and ISA are configured to select internal RAM rather than SFR space, then SARn refers to location 99H in the "upper 128" of internal RAM, not to SBUF.

If the test for SARn = SBUF *is* true, and if the flag RI is set, mode_logic (n) returns as 1 and the remainder of the function is not executed. Otherwise, execution proceeds to the next if-condition, testing DARn against SBUF and T1 against 1.

The same considerations regarding SAS and ISA in the SARn test are now applied to DAS and IDA in the DARn test. If SFR space isn't selected, no Serial Port buffer is being addressed.

Note that if DMA channel n is configured to Alternate Cycles mode, the logic must examine the other DCON register, DCONm, to determine if the other channel is also configured to Alternate Cycles mode and whether its GO bit is set. In Figure 4.13, the symbol DCONn refers to the DCON register for "this channel," and DCONm refers to "the other channel."

A careful examination of the logic in Figure 4.13 will reveal some idiosyncracies that the user should be aware of. First, the logic allows sequential DMA cycles to be generated to service RFIFO, but not to service TFIFO. This idiosyncracy is due to internal timing conflicts, and results in each individual DMA cycle to TFIFO having to be immediately preceded by an Instruction cycle. The logic disallows that there be two DMAs to TFIFO in a row.

If the user is unaware of this idiosyncracy, it can cause problems in situations where one DMA channel is servicing TFIFO and the other is configured to a completely different mode of operation.

For example, consider the situation where channel 0 is configured to service TFIFO and channel 1 is configured to Alternate Cycles mode. Then DMAs to TFIFO will always override the alternate cycles of channel 1. If TFIFO needs more than 1 byte it will receive them in precendence over channel 1, but each DMA to TFIFO must be preceded by an Instruction cycle. The sequence of cycles might be:

DMA1 cycle
Instruction cycle
DMA1 cycle, during which TFNF gets set
Instruction cycle
DMA0 cycle
Instruction cycle
DMA0 cycle, as a result of which TFNF gets cleared
Instruction cycle
DMA1 cycle
Instruction cycle
DMA1 cycle
Instruction cycle
. . .

The requirement that a DMA to TFIFO be preceded by an Instruction cycle can result in the normal precedence of channel 0 over channel 1 being thwarted. Consider for example the situation where channel 0 is configured to service TFIFO, and is in the process of doing so, and channel 1 decides it wants to do a Burst mode DMA. The sequence of events might be:

Instruction cycle (sets GO bit in DCON1)
Instruction cycle (during which TFNF gets set)
DMA0 cycle
DMA1 cycle
DMA1 cycle
DMA1 cycle
. . .
DMA1 cycle (completes channel 1 burst)
Instruction cycle
DMA0 cycle
Instruction cycle
. . .

This sequence begins with two Instruction cycles. The first one accesses a DMA register (DCON1), and therefore is followed by another Instruction cycle, which presumably does not access a DMA register. After the second Instruction cycle both channels are ready to generate DMA cycles, and channel 0 of course takes precedence. After the DMA0 cycle, channel 0 must wait for an Instruction cycle before it can access TFIFO again. Channel 1, being in Burst mode, doesn't have that restriction, and is therefore granted a DMA1 cycle. After the first DMA1 cycle, channel 0 is still waiting for an Instruction cycle and channel 1 still does not have that restriction. There follows another DMA1 cycle.

The result is that in this particular case channel 0 has to wait until channel 1 completes its Burst mode DMA, and then has to wait for an Instruction cycle to be generated, before it can continue its own DMA to TFIFO. The delay in servicing TFIFO can cause an Underflow condition in the GSC transmission.

The delay will not occur if channel 1 is configured to Alternate Cycles mode, since channel 0 would then see the Instruction cycles it needs to complete its logic requirements for asserting its request.

4.4.1 DMA Arbitration with Hold/Hold Ack

The Hold/Hold Acknowledge feature is invoked by setting either the ARB or REQ bit in PCON. Their effect is to add the requirements of the Hold/Hold Ack protocol to mode_logic ( ). This amounts to replacing every expression "return 1" in Figure 4.13 with the expression "return hld_hlda_logic ( )", where hld_hlda_logic ( ) is a function which returns 1 if the Hold/Hold Ack protocol is satisfied, and returns 0 otherwise. A suitable definition for hld_hlda_logic ( ) is shown in Figure 4.14.

4.5 Summary of DMA Control Bits

DCONn	DAS	IDA	SAS	ISA	DM	TM	DONE	GO

DAS specifies the Destination Address Space. If DAS = 0, the destination is in External Data Memory. If DAS = 1 and IDA = 0, the destination is a Special Function Register (SFR). If DAS = 1 and IDA = 1, the destination is in Internal Data RAM.

IDA (Increment Destination Address) If IDA = 1, the destination address is automatically incremented after each byte transfer. If IDA = 0, it is not.

SAS specifies the Source Address Space. If SAS = 0, the source is in External Data Memory. If SAS = 1 and ISA = 0, the source is an SFR. If SAS = 1 and ISA = 1, the source is Internal Data RAM.

ISA (Increment Source Address) If ISA = 1, the source address is automatically incremented after each byte transfer. If ISA = 0, it is not.

DM (Demand Mode) If DM = 1, the DMA Channel operates in Demand Mode. In Demand Mode the DMA is initiated either by an external signal or by a Serial Port flag, depending on the value of the TM bit. If DM = 0, the DMA is requested by setting the GO bit in software.

TM (Transfer Mode) If DM = 1 then TM selects whether a DMA is initiated by an external signal (TM = 1) or by a Serial Port flag (TM = 0). If DM = 0 then TM selects whether the data transfers are to be in bursts (TM = 1) or in alternate cycles (TM = 0).

DONE indicates the completion of a DMA operation and flags an interrupt. It is set to 1 by on-chip hardware when BCRn = 0, and is cleared to 0 by on-chip hardware when the interrupt is vectored to. It can also be set or cleared by software.

```
hold_holda( ):
    if (ARB = 0 .AND. REQ = 0) return 1;
    if SARn = XRAM .OR. DARn = XRAM)
        {
        if (ARB = 1 .AND. HLDA = 1) return 1;
        if (REQ = 1 .AND. HLDA = 0) return 1;
        else return 0;
        }
    return 1;
    end hold_holda( );
```

Figure 4.14. Hold/Hold Acknowledge Logic as a Pseudo-HLL Function

GO is the enable bit for the DMA Channel itself. The DMA Channel is inactive if GO = 0.

PCON	SMOD	ARB	REQ	GAREN	XRCLK	GFIEN	PDN	IDL

ARB enables the DMA logic to detect $\overline{HLD}$ and generate $\overline{HLDA}$. After it has activated $\overline{HLDA}$, the C152 will not begin a new DMA to or from External Data Memory as long as $\overline{HLD}$ is seen to be active. This logic is disabled when ARB = 0, and enabled when ARB = 1.

REQ enables the DMA logic to generate $\overline{HLD}$ and detect $\overline{HLDA}$ before performing a DMA to or from External Data Memory. After it has activated $\overline{HLD}$, the C152 will not begin the DMA until $\overline{HLDA}$ is seen to be active. This logic is disabled when REQ = 0, and enabled when REQ = 1.

5.0 INTERRUPT STRUCTURE

The 8XC152 retains all five interrupts of the 80C51BH. Six new interrupts are added in the 8XC152, to support its GSC and the DMA features. They are as listed below, and the flags that generate them are shown in Figure 5.1.

GSCRV — GSC Receive Valid
GSCRE — GSC Receive Error
GSCTV — GSC Transmit Valid
GSCTE — GSC Transmit Error
DMA0 — DMA Channel 0 Done
DMA1 — DMA Channel 1 Done

As shown in Figure 5.1, the Receive Valid interrupt can be signaled either by the RFNE flag (Receive FIFO Not Empty), or by the RDN flag (Receive Done). Which one of these flags causes the interrupt depends on the setting of the DMA bit in the SFR named TSTAT.

DMA = 0 means the DMA hardware is not configured to service the GSC, so the CPU will service it in software in response to the Receive FIFO not being empty. In that case, RFNE generates the Receive Valid interrupt.

DMA = 1 means the DMA hardware is configured to service the GSC, in which case the CPU need not be interrupted till the receive is complete. In that case, RDN generates the Receive Valid interrupt.

Similarly the Transmit Valid interrupt can be signaled either by the TFNF flag (Transmit FIFO Not Full), or by the TDN flag (Transmit Done), depending on whether the DMA bit is 0 or 1.

Note that setting the DMA bit does not itself configure the DMA channels to service the GSC. That job must be done by software writes to the DMA registers. The DMA bit only selects whether the GSCRV and GSCTV interrupts are flagged by a FIFO needing service or by an "operation done" signal.

The Receive and Transmit Error interrupt flags are generated by the logical OR of a number of error conditions, which are described in Section 3.6.5.

Each interrupt is assigned a fixed location in Program Memory, and the interrupt causes the CPU to jump to that location. All the interrupt flags are sampled at S5P2 of every machine cycle, and then the samples are sequentially polled during the next machine cycle. If more than one interrupt of the same priority is active, the one that is highest in the polling sequence is serviced first. The interrupts and their fixed locations in Program Memory are listed below in the order of their polling sequence.

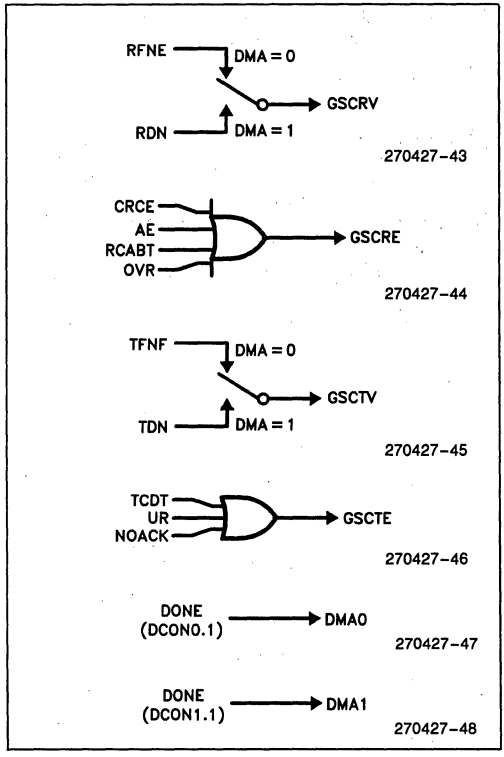


Figure 5.1. Six New Interrupts in the 8XC152

Interrupt	Location	Name
IE0	0003H	External Interrupt 0
GSCRV	002BH	GSC Receive Valid
TF0	000BH	Timer 0 Overflow
GSCRE	0033H	GSC Receive Error
DMA0	003BH	DMA Channel 0 Done
IE1	0013H	External Interrupt 1
GSCTV	0043H	GSC Transmit Valid
DMA1	0053H	DMA Channel 1 Done
TF1	001BH	Timer 1 Overflow
GSCTE	004BH	GSC Transmit Error
TI+RI	0023H	UART Transmit/Receive

Note that the locations of the basic 8051 interrupts are the same as in the rest of the MCS-51 Family. And relative to each other they retain their same positions in the polling sequence.

The locations of the new interrupts all follow the locations of the basic 8051 interrupts in Program Memory, but they are interleaved with them in the polling sequence.

To support the new interrupts a second Interrupt Enable register and a second Interrupt Priority register are implemented in bit-addressable SFR space. The two Interrupt Enable registers in the 8XC152 are as follows:

	7	6	5	4	3	2	1	0
IE:	EA	—	—	ES	ET1	EX1	ET0	EX0

Address of IE in SFR space = 0A8H (bit-addressable)

	7	6	5	4	3	2	1	0
IEN1:	—	—	EGSTE	EDMA1	EGSTV	EDMA0	EGSRE	EGSRV

Address pF IE1 in SFR space = 0C8H (bit-addressable)

The bits in IE are unchanged from the standard 8051 IE register. The bits in IEN1 are as follows:

EGSTE = 1 Enable GSC Transmit Error Interrupt
= 0 Disable
EDMA1 = 1 Enable DMA Channel 1 Done Interrupt
= 0 Disable
EGSTV = 1 Enable GSC Transmit Valid Interrupt
= 0 Disable
EDMA0 = 1 Enable DMA Channel 0 Done Interrupt
= 0 Disable
EGSRE = 1 Enable GSC Receive Error Interrupt
= 0 Disable
EGSRV = 1 Enable GSC Receive Valid Interrupt
= 0 Disable

The two Interrupt Priority registers in the 8XC152 are as follows:

	7	6	5	4	3	2	1	0
IP:	—	—	—	PS	PT1	PX1	PT0	PX0

Address of IP in SFR space = 0B8H (bit-addressable)

	7	6	5	4	3	2	1	0
IPN1:	—	—	PGSTE	PDMA1	PGSTV	PDMA0	PGSRE	PGSRV

Address of IPN1 in SFR space = 0F8H (bit-addressable)

The bits in IP are unchanged from the standard 8051 IP register. The bits in IPN1 are as follows:

PGSTE = 1 GSC Transmit Error Interrupt Priority to High
= 0 Priority to Low
PDMA1 = 1 DMA Channel 1 Done Interrupt Priority to High
= 0 Priority to Low
PGSTV = 1 GSC Transmit Valid Interrupt Priority to High
= 0 Priority to Low
PDMA0 = 1 DMA Channel 0 Done Interrupt Priority to High
= 0 Priority to Low
PGSRE = 1 GSC Receive Error Interrupt Priority to High
= 0 Priority to Low
PGSRV = 1 GSC Receive Valid Interrupt Priority to High
= 0 Priority to Low

Note that these registers all have unimplemented bits ("—"). If these bits are read, they will return unpredictable values. If they are written to, the value written goes nowhere.

It is recommended that user software should never write 1s to unimplemented bits in MCS-51 devices. Future versions of the device may have new bits installed in these locations. If so, their reset value will be 0. Old software that writes 1s to newly implemented bits may unexpectedly invoke new features.

The MCS-51 interrupt structure provides hardware support for only two priority levels, High and Low. With as many interrupt sources as the 8XC152 has, it may be helpful to know how to augment the priority structure in software. Any number of priority levels can be implemented in software by saving and redefining the interrupt enable registers within the interrupt service routines. The technique is described in the "MCS-51" Architectural Overview" chapter in this handbook.

5.1 GSC Transmitter Error Conditions

The GSC Transmitter section reports three kinds of error conditions:

TCDT — Transmitter Collision Detector

UR — Underrun in Transmit FIFO

NOACK — No Acknowledge

These bits reside in the TSTAT register. User software can read them, but only the GSC hardware can write to them. The GSC hardware will set them in response to the various error conditions that they represent. When user software sets the TEN bit, the GSC hardware will at that time clear these flags. This is the only way these flags can be cleared.

The logical OR of these three bits flags the GSC Transmit Error interrupt (GSCTE) and clears the TEN bit, as shown in Figure 5.2. Thus any detected error condition aborts the transmission. No CRC bits are transmitted. In SDLC mode, no EOF flag is generated. In CSMA/CD mode, an EOF is generated by default, since the GTXD pin is pulled to a logic 1 and held there.

The TCDT bit can get set only if the GSC is configured to CSMA/CD mode. In that case, the GSC hardware sets TCDT when a collision is detected during a transmission, and the collision was detected after TFIFO has been accessed. Also, the GSC hardware sets TCDT when a detected collision causes the TCDCNT register to overflow.

The UR bit can get set only if the DMA bit in TSTAT is set. The DMA bit being set informs the GSC hardware that TFIFO is being serviced by DMA. In that case, if the GSC goes to fetch another byte from TFIFO and finds it empty, and the byte count register of the DMA channel servicing TFIFO is not zero, it sets the UR bit.

If the DMA hardware is not being used to service TFIFO, the UR bit cannot get set. If the DMA bit is 0, then when the GSC finds TFIFO empty, it assumes that the transmission of data is complete and the transmission of CRC bits can begin.

The NOACK bit is functional only in CSMA/CD mode, and only when the HABEN bit in RSTAT is set. The HABEN bit turns on the Hardware Based Acknowledge feature, as described in Section 3.2.6. If this feature is not invoked, the NOACK bit will stay at 0.

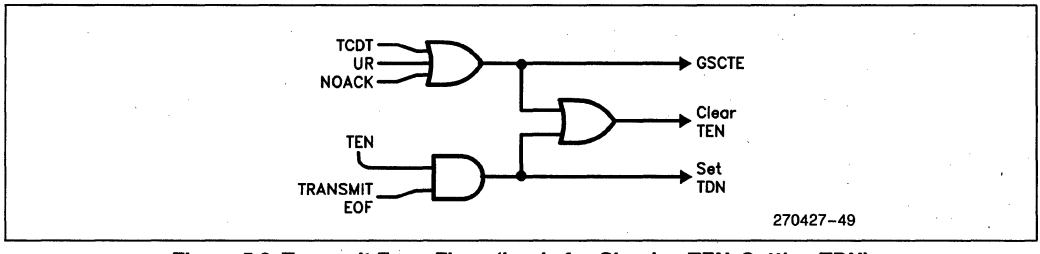


Figure 5.2. Transmit Error Flags (Logic for Clearing TEN, Setting TDN)

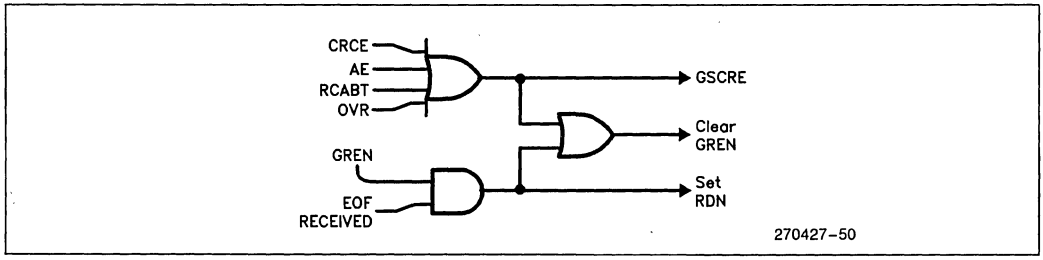


Figure 5.3. Receive Error Flag (Logic for Clearing GREN, setting RDN)

If the NOACK bit gets set, it means the GSC has completed a transmission, and was expecting to receive a hardware based acknowledge from the receiver of the message, but did not receive the acknowledge, or at least did not receive it cleanly. There are three ways the NOACK bit can get set:

1. The acknowledge signal (an unattached preamble) was not received before the IFS was completed.
2. A collision was detected during the IFS.
3. The line was active during the last bit-time of the IFS.

The first condition is an obvious reason for setting the NOACK bit, since that's what the hardware based acknowledge is for. The other two ways the NOACK bit can get set are to guard against the possibility that the transmitting station might mistake an unrelated transmission or transmission fragment for an acknowledge signal.

5.2 GSC Receiver Error Conditions

The GSC Receiver section reports four kinds of error conditions:

CRCE — CRC Error

AE — Alignment Error

RCABT — Receive Abort

OVR — Overrun in Receive FIFO

These bits reside in the RSTAT register. User software can read them, but only the GSC hardware can write to them. The GSC hardware will set them in response to the various error conditions that they represent. When user software sets the GREN bit, the GSC hardware will at that time clear these flags. This is the only way these flags can be cleared.

The logical OR of these four bits flags the GSC Receive Error interrupt (GSCRE) and clears the GREN bit, as shown in Figure 5.3. Note in this figure that any error condition will prevent RDN from being set.

A CRC Error means the CRC generator did not come to its correct value after calculating the CRC of the message plus received CRC. An Alignment Error means the number of bits received between the BOF and EOF was not a multiple of 8.

In SDLC mode, the CRCE bit gets set at the end of any frame in which there is a CRC Error, and the AE bit gets set at the end of any frame in which there is an Alignment Error.

In CSMA/CD mode, if there is no CRC Error, neither CRCE nor AE will get set. If there is a CRC Error and no Alignment Error, the CRCE bit will get set, but not the AE bit. If there is both a CRC Error and an Alignment Error, the AE bit will get set, but not the CRCE bit. Thus in CSMA/CD mode, the CRCE and AE bits are mutually exclusive.

The Receive Abort flag, RCABT, gets set if an incoming frame was interrupted after received data had already passed to the Receive FIFO. In SDLC mode, this can happen if a line idle condition is detected before an EOF flag is. In CSMA/CD mode, it can happen if there is a collision. In either case, the CPU will have to re-initialize whatever pointers and counters it might have been using.

The Overrun Error flag, OVR, gets set if the GSC Receiver is ready to push a newly received byte onto the Receive FIFO, but the FIFO is full.

Up to 7 "dribble bits" can be received after the EOF without causing an error condition.

6.0 GLOSSARY

ADR0,1,2,3 (95H, 0A5H, 0B5H, 0C5H) - Address Match Registers 0,1,2,3 - The contents of these SFRs are compared against the address bits from the serial data on the GSC. If the address matches the SFR, then the C152 accepts that frame. If in 8 bit addressing mode, a match with any of the four registers will trigger acceptance. In 16 bit addressing mode, a match with ADR1:ADR0 or ADR3:ADR2 will be accepted. Address length is determined by GMOD (AL).

AE - Alignment Error, see RSTAT.
AL - Address Length, see GMOD.

AMSK0,1 (0D5H, 0E5H) - Address Match Mask 0,1 - Identifies which bits in ADR0,1 are "don't care" bits. Setting a bit to 1 in AMSK0,1 identifies the corresponding bit in ADDR0,1 as not to be examined when comparing addresses.

BAUD - (94H) Contains the programmable value for the baud rate generator for the GSC. The baud rate will equal (fosc)/((BAUD + 1) × 8).

BCRL0,1 (0E2H, 0F2H) - Byte Count Register Low 0,1 - Contains the lower byte of the byte count. Used during DMA transfers to identify to the DMA channels when the transfer is complete.

BCRH0,1 (0E3H, 0F3H) - Byte Count Register High 0,1 - Contains the upper byte of the byte count.

BKOFF (0C4H) - Backoff Timer - The backoff timer is an eight bit count-down timer with a clock period equal to one slot time. The backoff time is used in the CSMA/CD collision resolution algorithm.

BOF - Beginning of Frame flag - A term commonly used when dealing with packetized data. Signifies the beginning of a frame.

CRC - Cyclic Redundancy Check - An error checking routine that mathematically manipulates a value dependent on the incoming data. The purpose is to identify when a frame has been received in error.

CRCE - CRC Error, see RSTAT.

CSMA/CD - Stands for Carrier Sense, Multiple Access, with Collision Detection.

CT - CRC Type, see GMOD.

DARL0/1 (0C2H, 0D2H) - Destination Address Register Low 0/1 - Contains the lower byte of the destinations' address when performing DMA transfers.

DARH0/1 (0C3H, 0D3H) - Destination Address Register Low 0/1 - Contains the upper byte of the destinations' address when performing DMA transfers.

DAS - Destination Address Space, see DCON.

DCJ - D.C. Jam, see MYSLOT.

DCON0/1 (092H,093H)

7	6	5	4	3	2	1	0
DAS	IDA	SAS	ISA	DM	TM	DONE	GO

The DCON registers control the operation of the DMA channels by determining the source of data to be transferred, the destination of the data to be transfer, and the various modes of operation.

DCON.0 (GO) - Enables DMA Transfer - When set it enables a DMA channel. If block mode is set then DMA transfer starts as soon as possible under CPU control. If demand mode is set then DMA transfer starts when a demand is asserted and recognized.

DCON.1 (DONE) - DMA Transfer is Complete - When set the DMA transfer is complete. It is set when BCR equals 0 and is automatically reset when the DMA vectors to its interrupt routine. If DMA interrupt is disabled and the user software executes a jump on the DONE bit, then the user software must also reset the done bit. If DONE is not set, then the DMA transfer is not complete.

DCON.2 (TM) - Transfer Mode - When set, DMA burst transfers are used if the DMA channel is configured in block mode or external interrupts are used to initiate a transfer if in Demand Mode. When TM is cleared, Alternate Cycle Transfers are used if DMA is in the Block Mode, or Local Serial channel/GSC interrupts are used to initiate a transfer if in Demand Mode.

DCON.3 (DM) - DMA Channel Mode - When set, Demand Mode is used and when cleared, Block Mode is used.

DCON.4 (ISA) - Increment Source Address - When set, the source address registers are automatically incremented during each transfer. When cleared, the source address registers are not incremented.

DCON.5 (SAS) - Source Address Space - When set, the source of data for the DMA transfers is internal data memory if autoincrement is also set. If autoincrement is not set but SAS is, then the source for data will be one of the Special Function Registers. When SAS is cleared, the source for data is external data memory.

DCON.6 (IDA) - Increment Destination Address Space - When set, destination address registers are incremented once after each byte is transferred. When cleared, the destination address registers are not automatically incremented.

DCON.7 (DAS) - Destination Address Space - When set, destination of data to be transferred is internal data memory if autoincrement mode is also set. If autoincrement is not set the destinationwill be one of the Special Function Registers. When DAS is cleared then the destination is external data memory.

DCR - Deterministic Resolution, see MYSLOT.

$\overline{\text{DEN}}$ - An alternate function of one of the port 1 pins (P1.2). Its purpose is to enable external drivers when the GSC is transmitting data. This function is always active when using the GSC and if P1.2 is programmed to a 1.

DM - DMA Mode, see DCON0.

DMA - Direct Memory Access mode, see TSTAT.

DONE - DMA done bit, see DCON0.

DPH - Data Pointer High, an SFR that contains the high order byte of a general purpose pointer called the data pointer (DPTR).

DPL - Data Pointer Low, an SFR that contains the low order byte of the data pointer.

EDMA0 - Enable DMA Channel 0 interrupt, see IEN1.

EDMA1 - Enable DMA Channel 1 interrupt, see IEN1.

EGSRE - Enable GSC Receive Error interrupt, see IEN1.

EGSRV - Enable GSC Receive Valid interrupt, see IEN1.

EGSTE - Enable GSC Transmit Error interrupt, see IEN1.

EGSTV - Enable GSC Transmit Valid interrupt, see IEN1.

EOF - A general term used in serial communications. EOF stands for End Of Frame and signifies when the last bits of data are transmitted when using packetized data.

ES - Enable LSC Service interrupt, see IE.

ET0 - Enable Timer 0 interrupt, see IE.

ET1 - Enable Timer 1 interrupt, see IE.

EX0 - Enable External interrupt 0, see IE.

EX1 - Enable External interrupt 1, see IE.

GMOD (84H)

7	6	5	4	3	2	1	0
XTCLK	M1	M0	AL	CT	PL1	PL0	PR

The bits in this SFR, perform most of the configuration on the type of data transfers to be used with the GSC. Determines the mode, address length, preamble length, protocol select, and enables the external clocking of the transmit data.

GMOD.0 (PR) - Protocol - If set, SDLC protocols with NRZI encoding, zero bit insertion, and SDLC flags are used. If cleared, CSMA/CD link access with Manchester encoding is used.

GMOD.1,2 (PL0,1) - Preamble length

PL1	PL0	LENGTH (BITS)
0	0	0
0	1	8
1	0	32
1	1	64

The length includes the two bit Begin Of frame (BOF) flag in CSMA/CD but does not include the SDLC flag. In SDLC mode, the BOF is an SDLC flag, otherwise it is two consecutive ones. Zero length is not compatible in CSMA/CD mode.

GMOD.3 (CT) - CRC Type - If set, 32-bit AUTODIN-II-32 is used. If cleared, 16-bit CRC-CCITT is used.

GMOD.4 (AL) - Address Length - If set, 16-bit addressing is used. If cleared, 8-bit addressing is used. In 8-bit mode, a match with any of the 4 address registers will allow that frame to be accepted (ADR0, ADR1, ADR2, ADR3). "Don't Care" bits may be masked in ADR0 and ADR1 with AMSK0 and AMSK1. In 16-bit mode, addresses are matched against "ADR1:ADR0" or "ADR3:ADR2". Again, "Don't Care" bits in ADR1:ADR0 can be masked in AMSK1:AMSK0. A received address of all ones will always be recognized in any mode.

GMOD.5, 6 (M0,M1) - Mode Select - Two test modes, an optional "alternate backoff" mode, or normal backoff can be enabled with these two bits.

M1	M0	Mode
0	0	Normal
0	1	Raw Transmit
1	0	Raw Receive
1	1	Alternate Backoff

GMOD.7 (XTCLK) - External Transmit Clock - If set an external 1X clock is used for the transmitter. If cleared the internal baud rate generator provides the

transmit clock. The input clock is applied to P1.3 ($\overline{TxC}$). The user software is responsible for setting or clearing this flag. External receive clock is enabled by setting PCON.3.

GO - DMA Go bit, see DCON0.

GRxD - GSC Receive Data input, an alternate function of one of the port 1 pins (P1.0). This pin is used as the receive input for the GSC. P1.0 must be programmed to a 1 for this function to operate.

GSC - Global Serial Channel - A high-level, multi-protocol, serial communication controller added to the 80C51BH core to accomplish high-speed transfers of packetized serial data.

GTxD - GSC Transmit Data output, an alternate function of one of the port 1 pins (P1.1). This pin is used as the transmit output for the GSC. P1.1 must be programmed to a 1 for this function to operate.

HBAEN - Hardware Based Acknowledge Enable, see RSTAT.

HLDA - Hold Acknowledge, an alternate function of one of the port 1 pins (P1.6). This pin is used to perform the "HOLD ACKNOWLEDGE" function for DMA transfers. HLDA can be an input or an output, depending on the configuration of the DMA channels. P1.6 must be programmed to a 1 for this function to operate.

HOLD - Hold, an alternate function of one of the port 1 pins (P1.5). This pin is used to perform the "HOLD" function for DMA transfers. HOLD can be an input or an output, depending on the configuration of the DMA channels. P1.5 must be programmed to a 1 for this function to operate.

IDA - Increment Destination Address, see DCON0.

IE (0A8H)

7	6	5	4	3	2	1	0
EA			ES	ET1	EX1	ET0	EX0

Interrupt Enable SFR, used to individually enable the Timer and Local Serial Channel interrupts. Also contains the global enable bit which must be set to a 1 to enable any interrupt to be automatically recognized by the CPU.

IE.0 (EX0) - Enables the external interrupt $\overline{INT0}$ on P3.2.

IE.1 (ET0) - Enables the Timer 0 interrupt.

IE.2 (EX1) - Enables the external interrupt $\overline{INT1}$ on P3.3.

IE.3 (ET1) - Enables the Timer 1 interrupt.

IE.4 (ES) - Enables the Local Serial Channel interrupt.

IE.7 (EA) - The global interrupt enable bit. This bit must be set to a 1 for any other interrupt to be enabled.

IEN1 - (0C8H)

7	6	5	4	3	2	1	0
		EGSTE	EDMA1	EGSTV	EDMA0	EGSRE	EGSRV

Interrupt enable register for DMA and GSC interrupts. A 1 in any bit position enables that interrupt.

IEN1.0 (EGSRV) - Enables the GSC valid receive interrupt.

IEN1.1 (EGSRE) - Enables the GSC receive error interrupt.

IEN1.2 (EDMA0) - Enables the DMA done interrupt for Channel 0.

IEN1.3 (EGSTV) - Enables the GSC valid transmit interrupt.

IEN1.4 (EDMA1) - Enables the DMA done interrupt for Channel 1.

IEN1.5 (EGSTE) - Enables the GSC transmit error interrupt

IFS - (0A4H) Interframe Space, determines the number of bit times separating transmitted frames.

IP (0B8H)

7	6	5	4	3	2	1	0
			PS	PT1	PX1	PT0	PX0

Allows the user software two levels of prioritization to be assigned to each of the interrupts in IE. A 1 assigns the corresponding interrupt in IE a higher interrupt than an interrupt with a corresponding 0.

IP.0 (PX0) - Assigns the priority of external interrupt, $\overline{INT0}$.

IP.1 (PT0) - Assigns the priority of Timer 0 interrupt, T0.

IP.2 (PX1) - Assigns the priority of external interrupt, $\overline{INT1}$.

IP.3 (PT1) - Assigns the priority of Timer 1 interrupt, T1.

IP.4 (PS) - Assigns the priority of the LSC interrupt, SBUF.

IPN1 - (0F8H)

7	6	5	4	3	2	1	0
		PGSTE	PDMA1	PGSTV	PDMA0	PGSRE	PGSRV

Allows the user software two levels of prioritization to be assigned to each of the interrupts in IEN1. A 1 assigns the corresponding interrupt in IEN1 a higher interrupt than an interrupt with a corresponding 0.

IPN1.0 (PGSRV) - Assigns the priority of GSC receive valid interrupt.

IPN1.1 (PGSRE) - Assigns the priority of GSC error receive interrupt.

IPN1.2 (PDMA0) - Assigns the priority of DMA done interrupt for Channel 0.

IPN1.3 (PGSTV) - Assigns the priority of GSC transmit valid interrupt.

IPN1.4 (PDMA1) - Assigns the priority of DMA done interrupt for Channel 1.

IPN1.5 (PGSTE) - Assigns the priority of GSC transmit error interrupt.

ISA - Increment Source Address, see DCON0.

LNI - Line Idle, see TSTAT.

LSC - Local Serial Channel - The asynchronous serial port found on all MCS-51 devices. Uses start/stop bits and can transfer only 1 byte at a time.

M0 - One of two GSC mode bits, see TMOD.

M1 - One of two GSC mode bits, see TMOD

MYSLOT - (0F5H)

7	6	5	4	3	2	1	0
DCJ	DCR	SA5	SA4	SA3	SA2	SA1	SA0

Determines which type of Jam is used, which backoff algorithm is used, and the DCR slot address for the GSC.

MYSLOT.0,1,2,3,4,5 (SA0,1,2,3,4,5) - These bits determine which slot address is assigned to the C152 when using deterministic backoff during CSMA/CD operations on the GSC. Maximum slots available is 63. An address of 00H prevents that station from participating in the backoff process.

MYSLOT.6 (DCR) - Determines which collision resolution algorithm is used. If set to a 1, then the deterministic backoff is used. If cleared, then a random slot assignment is used.

MYSLOT.7 (DCJ) - Determines the type of Jam used during CSMA/CD operation when a collision occurs. If set to a 1 then a low D.C. level is used as the jam signal. If cleared, then $\overline{CRC}$ is used as the jam signal. The jam is applied for a length of time equal to the CRC length.

NOACK - No Acknowledgment error bit, see TSTAT.

NRZI - Non-Return to Zero inverted, a type of data encoding where a 0 is represented by a change in the level of the serial link. A 1 is represented by no change.

OVR - Overrun error bit, see RSTAT.

PR - Protocol select bit, see GMOD. PCON (87H)

7	6	5	4	3	2	1	0
SMOD	ARB	REQ	GAREN	XRCLK	GFIEN	PD	IDL

PCON.0 (IDL) - Idle bit, used to place the C152 into the idle power saving mode.

PCON.1 (PD) - Power Down bit, used to place the C152 into the power down power saving mode.

PCON.2 (GFIEN) - GSC Flag Idle Enable bit, when set, enables idle flags (01111110) to be generated between transmitted frames in SDLC mode.

PCON.3 (XRCLK) - External Receive Clock bit, used to enable an external clock to be used for only the receiver portion of the GSC.

PCON.4 (GAREN) - GSC Auxiliary Receive Enable bit, used to enable the GSC to receive back-to-back SDLC frames. This bit has no effect in CSMA/CD mode.

PCON.5 (REQ) - Requester mode bit, set to a 1 when C152 is to be operated as the requester station during DMA transfers.

PCON.6 (ARB) - Arbiter mode bit, set to a 1 when C152 is to be operated as the arbiter during DMA transfers.

PCON.7 (SMOD) - LSC mode bit, used to double the baud rate on the LSC.

PDMA0 - Priority bit for DMA Channel 0 interrupt, see IPN1.

PDMA1 - Priority bit for DMA Channel 1 interrupt, see IPN1.

PGSRE - Priority bit for GSC Receive Error interrupt, see IPN1.

PGSRV - Priority bit for GSC Receive Valid interrupt, see IPN1.

PGSTE - Priority bit for GSC Transmit Error interrupt, see IPN1.

PGSTV - Priority bit for GSC Transmit Valid interrupt, see IPN1.

PL0 - One of two bits that determines the Preamble Length, see GMOD.

PL1 - One of two bits that determines the Preamble Length, see GMOD.

PRBS - (0E4H) Pseudo-Random Binary Sequence, generates the pseudo-random number to be used in CSMA/CD backoff algorithms.

PS - Priority bit for the LSC service interrupt, see IP.

PT0 - Priority bit for Timer 0 interrupt, see IP.

PT1 - Priority bit for Timer 1 interrupt, see IP.

PX0 - Priority bit for External interrupt 0, see IP.

PX1 - Priority bit for External interrupt 1, see IP.

RCABT - GSC Receiver Abort error bit, see RSTAT.

RDN - GSC Receiver Done bit, see RSTAT.

GREN - GSC Receiver Enable bit, see RSTAT.

RFNE - GSC Receive FIFO Not Empty bit, see RSTAT.

RI - LSC Receive Interrupt bit, see SCON.

RFIFO - (F4H) RFIFO is a 3-byte FIFO that contains the receive data from the GSC.

RSTAT (0E8H) - Receive Status Register

7	6	5	4	3	2	1	0
OVR	RCABT	AE	CRCE	RDN	RFNE	GREN	HABEN

RSTAT.0 (HBAEN) - Hardware Based Acknowledge Enable - If set, enables the hardware based acknowledge feature.

RSTAT.1 (GREN) - Receiver Enable - When set, the receiver is enabled to accept incoming frames. This also clears RDN, CRCE, AE, RCABT and the receive FIFO. It is cleared by the receiver at the end of a reception or if any errors occurred. The status of GREN has no effect on whether the receiver detects a collision in CSMA/CD mode as the receiver input circuitry always monitors the receive pin.

RSTAT.2 (RFNE) - Receive FIFO Not Empty - If set, indicates that the receive FIFO contains data. The receive FIFO is a three byte buffer into which the receive data is loaded. A CPU read of the FIFO retrieves the oldest data and automatically updates the FIFO pointers. Setting GREN to a one will clear the receive FIFO. The status of this flag is controlled by the GSC. This bit is cleared if user S/W empties receive FIFO.

RSTAT.3 (RDN) - Receive Done - If set, indicates the successful completion of a receiver operation. Will not be set if a CRC, alignment, abort, or FIFO overrun error occurred.

RSTAT.4 (CRCE) - CRC Error - If set, indicates that a properly aligned frame was received with a mismatched CRC.

RSTAT.5 (AE) - Alignment Error - If set, indicates that the line went idle when the receiver shift register was not full and the resulting CRC was bad in the CSMA/CD mode. If a correct CRC was valid then AE is not set. In SDLC mode, AE indicates that a non-byte-aligned flag was received.

RSTAT.6 (RCABT) - Receiver Collision/Abort Detect - If set, indicates that a collision was detected after data had been loaded into the receive FIFO in CSMA/CD mode. In SDLC mode, RCABT indicates that 7 consecutive ones were detected prior to the end flag but after data has been loaded into the receive FIFO.

RSTAT.7 (OVR) - Overrun - If set, indicates that the receive FIFO was full and new shift register data was written into it. It is cleared by user S/W.

SARH0 (0A3H) - Source Address Register High 0, contains the high byte of the source address for DMA Channel 0.

SARH1 (0B3H) - Source Address Register High 1, contains the high byte of the source address for DMA Channel 1.

SARL0 (0A2H) - Source Address Register Low 0, contains the low byte of the source address for DMA Channel 0.

SARL1 (0B2H) - Source Address Register Low 1, contains the low byte of the source address for DMA Channel 1.

SAS - Source Address Space bit, see DCON0.

SBUF (099H) - Serial Buffer, both the receive and transmit SFR location for the LSC.

SCON (098H)

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SCON.0 (RI) - Receive Interrupt flag.

SCON.1 (TI) - Transmit Interrupt flag.

SCON.2 (RB8) - Receive Bit 8, contains the ninth bit that was received in Modes 2 and 3 or the stop bit in Mode 1 if SM20. Not used in Mode 0.

SCON.3 (TB8) - Transmit Bit 8, the ninth bit to be transmitted in Modes 2 and 3.

SCON.4 (REN) - Receiver Enable, enables reception for the LSC.

SCON.5 (SM2) - Enables the multiprocessor communication feature in Modes 2 and 3 for the LSC.

SCON.6 (SM1) - LSC mode specifier.

SCON.7 (SM2) - LSC mode specifier.

SDLC - Stands for Synchronous Data Link Communication and is a protocol developed by IBM.

SLOTTM - (0B4H) Determines the length of the slot time in CSMA/CD.

SP (081H) - Stack Pointer, an eight bit pointer register used during a PUSH, POP, CALL, RET, or RETI.

TCDCNT - (0D4H) Contains the number of collisions in the current frame if using probabilistic CSMA/CD and contains the maximum number of slots in the deterministic mode.

TCDT - Transmit Collision Detect, see TSTAT.

TCON (088H)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TCON.0 (IT0) - Interrupt 0 mode control bit.

TCON.1 (IE0) - External interrupt 0 edge flag.

TCON.2 (IT1) - Interrupt 1 mode control bit.

TCON.3 (IE1) - External interrupt 1 edge flag.

TCON.4 (TR0) - Timer 0 run control bit.

CON.5 (TF0) - Timer 0 overflow flag.

TCON.6 (TR1) - Timer 1 run control bit.

TCON.7 (TF1) - Timer 1 overflow flag.

TDN - Transmit Done flag, see TSTAT.

TEN - Transmit Enable bit, see TSTAT.

TFNF - Transmit FIFO Not Full flag, see TSTAT.

TFIFO - (85H) TFIFO is a 3-byte FIFO that contains the transmission data for the GSC.

TH0 (08CH) - Timer 0 High byte, contains the high byte for timer/counter 0.

TH1 (08DH) - Timer 1 High byte, contains the high byte for timer/counter 1.

TI - Transmit Interrupt, see SCON.

TL0 (08AH) - Timer 0 Low byte, contains the low byte for timer/counter 0.

TL1 (08BH) - Timer 1 Low byte, contains the low byte for timer/counter 1.

TM - Transfer Mode, see, DCON0.

TMOD (089H)

7	6	5	4	3	2	1	0
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

TMOD.0 (M0) - Mode selector bit for Timer 0.

TMOD.1 (M1) - Mode selector bit for Timer 0.

TMOD.2 (C/$\overline{T}$) - Timer/Counter selector bit for Timer 0.

TMOD.3 (GATE) - Gating Mode bit for Timer 0.

TMOD.4 (M0) - Mode selector bit for Timer 1.

TMOD.5 (M1) - Mode selector bit for Timer 1.

TMOD.6 (C/$\overline{T}$) - Timer/Counter selector bit for Timer 1.

TMOD.7 (GATE) - Gating Mode bit for Timer 1.

TSTAT (0D8) - Transmit Status Register

7	6	5	4	3	2	1	0
LNI	NOACK	UR	TCDT	TDN	TFNF	TEN	DMA

TSTAT.0 (DMA) - DMA Select - If set, indicates that DMA channels are used to service the GSC FIFO's and GSC interrupts occur on TDN and RDN, and also enables UR to become set. If cleared, indicates that the GSC is operating in it normal mode and interrupts occur on TFNE and RFNE.For more information on DMA servicing please refer to the DMA section on DMA serial demand mode (4.2.2.3).

TSTAT.1 (TEN) - Transmit Enable - When set causes TDN, UR, TCDT, and NOACK flags to be reset and the TFIFO cleared. The transmitter will clear TEN after a successful transmission, a collision during the data, CRC, or end flag. If cleared during a transmission the GSC transmit pin goes to a steady state high level. This is the method used to send an abort character in SDLC. Also $\overline{DEN}$ is forced to a high level. The end of transmission occurs whenever the TFIFO is emptied.

TSTAT.2 (TFNF) - Transmit FIFO not full - When set, indicates that new data may be written into the transmit FIFO. The transmit FIFO is a three byte buffer that loads the transmit shift register with data.

TSTAT.3 (TDN) - Transmit Done - When set, indicates the successful completion of a frame transmission. If HBAEN is set, TDN will not be set until the end of the IFS following the transmitted message, so that the acknowledge can be checked. If an acknowledge is expected and not received, TDN is not set. An acknowledge is not expected following a broadcast or multi-cast packet.

TSTAT.4 (TCDT) - Transmit Collision Detect - If set, indicates that the transmitter halted due to a collision. It is set if a collision occurs during the data or CRC or if there are more than eight collisions.

TSTAT.5 (UR) - Underrun - If set, indicates that in DMA mode the last bit was shifted out of the transmit register and that the DMA byte count did not equal zero. When an underrun occurs, the transmitter halts without sending the CRC or the end flag.

TSTAT.6 (NOACK) - No Acknowledge - If set, indicates that no acknowledge was received for the previous frame. Will be set only if HBAEN is set and no acknowledge is received prior to the end of the IFS. NOACK is not set following a broadcast or a multicast packet.

TSTAT.7 (LNI) - Line Idle - If set, indicates the receive line is idle. In SDLC protocol it is set if 15 consecutive ones are received. In CSMA/CD protocol, line idle is set if no transitions occur on GR$\times$D for 1.6 bit times after a required transition. LNI is cleared after a transition on GR$\times$D.

TxC - External Clock input for GSC transmitter.

UR - Underrun flag, see TSTAT.

XRCLK - External GSC Receive Clock Enable bit, see PCON.

XTCLK - External GSC Transmit Clock Enable bit, see GMOD.

8XC152JA/JB/JC/JD
UNIVERSAL COMMUNICATION CONTROLLER
8-BIT MICROCOMPUTER

- **8K Factory Mask Programmable ROM Available**

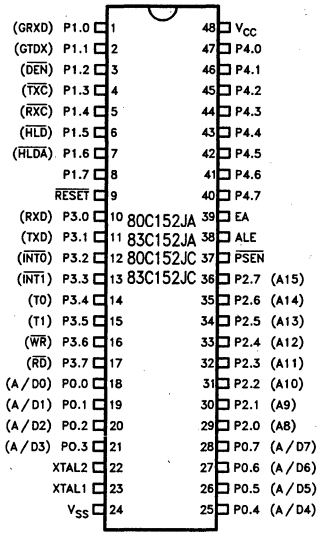
- **Superset of 80C51 Architecture**
- **Multi-Protocol Serial Communication I/O Port (2.048 Mbps/2.4 Mbps Max)**
  - **— SDLC/HDLC Only**
  - **— CSMA/CD and SDLC/HDLC**
  - **— User Definable Protocols**
- **Full Duplex/Half Duplex**
- **MCS®-51 Compatible UART**
- **16.5 MHz Maximum Clock Frequency**
- **Multiple Power Conservation Modes**
- **64KB Program Memory Addressing**
- **64KB Data Memory Addressing**
- **256 Bytes On-Chip RAM**
- **Dual On-Chip DMA Channels**
- **Hold/Hold Acknowledge**
- **Two General Purpose Timer/Counters**
- **5 or 7 I/O Ports**
- **56 Special Function Registers**
- **11 Interrupt Sources**
- **Available in 48 Pin Dual-in-Line Package and 68 Pin Surface Mount PLCC Package**

  (See Packaging Spec. Order #231369)

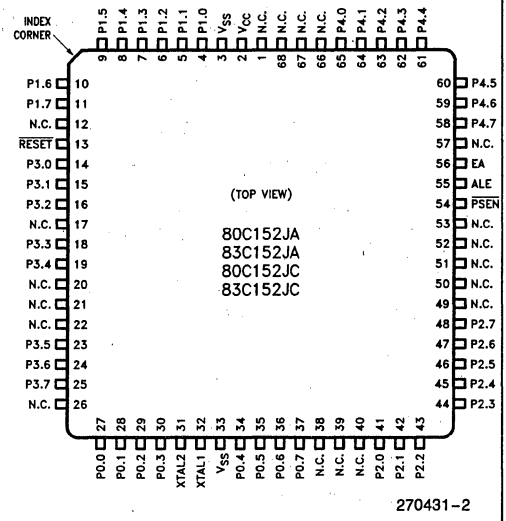
The 80C152, which is based on the MCS®-51 CPU, is a highly integrated single-chip 8-bit microcontroller designed for cost-sensitive, high-speed, serial communications. It is well suited for implementing Integrated Services Digital Networks (ISDN), emerging Local Area Networks, and user defined serial backplane applications. In addition to the multi-protocol communication capability, the 80C152 offers traditional microcontroller features for peripheral I/O interface and control.

Silicon implementations are much more cost effective than multi-wire cables found in board level parallel-to-serial and serial-to-parallel converters. The 83C152 contains, in silicon, all the features needed for the serial-to-parallel conversion. Other 83C152 benefits include: 1) better noise immunity through differential signaling or fiber optic connections, 2) data integrity utilizing the standard, designed in CRC checks, and 3) better modularity of hardware and software designs. All of these—cost, network parameter and real estate improvements—apply to 83C152 serial links between boards or systems and 83C152 serial links on a single board.

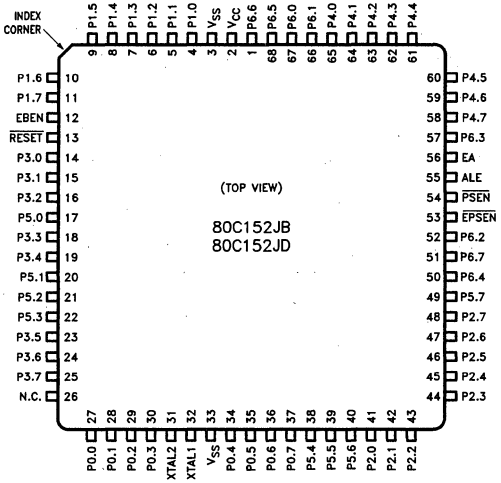
September 1988
Order Number: 270431-002



270431-1



270431-2



270431-3

Figure 1. Connection Diagrams

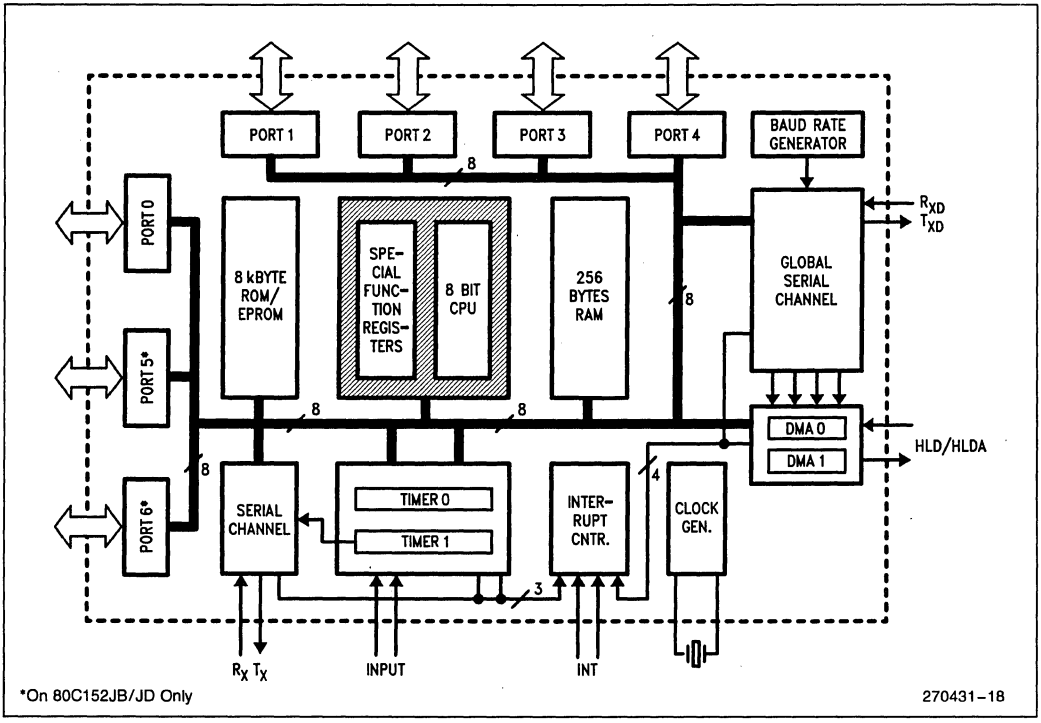


Figure 2. Block Diagram

80C152JB/JD General Description

The 80C152JB/JD is a ROMless extension of the 80C152 Universal Communication controller. The 80C152JB has the same five 8-bit I/O ports of the 80C152, plus an additional two 8-bit I/O ports, Port 5 and Port 6. The 80C152JB/JD also has two additional control pins, EBEN (**E**PROM **B**us **EN**able), and $\overline{\text{EPSEN}}$ (**E**PROM bus **P**rogram **S**tore **EN**able).

EBEN selects the functionality of Port 5 and Port 6. When EBEN is low, these ports are strictly I/O, similar to Port 4. The SFR location for Port 5 is 91H and Port 6 is 0A1H. This means Port 5 and Port 6 are not bit addressable. With EBEN low, all program memory fetches take place via Port 0 and Port 2. (The 80C152 is a ROMless only product). When EBEN is high, Port 5 and Port 6 form an address/data bus called the E-Bus (EPROM-Bus) for program memory operations.

$\overline{\text{EPSEN}}$ is used in conjunction with Port 5 and Port 6 program memory operations. $\overline{\text{EPSEN}}$ functions like $\overline{\text{PSEN}}$ during program memory operation, but supports Port 5 and Port 6. $\overline{\text{EPSEN}}$ is the read strobe to external program memory for Port 5 and Port 6. $\overline{\text{EPSEN}}$ is activated twice during each machine cycle unless an external data memory operation occurs on Port(s) 0 and Port 2. When external data memory is accessed the second activation of $\overline{\text{EPSEN}}$ is skipped, which is the same as when using $\overline{\text{PSEN}}$. Note that data memory fetches cannot be made through Ports 5 and 6.

When EBEN is high and EA is low, all program memory operations take place via Ports 5 and 6. The high byte of the address goes out on Port 6, and the low byte is output on Port 5. ALE is still used to latch the address on Port 5. Next, the op code is read on Port 5. The timing is the same as when using Ports 0 and 2 for external program memory operations.

Table 1. Program Memory Fetches

EBEN	$\overline{\text{EA}}$	Program Fetch via	$\overline{\text{PSEN}}$	$\overline{\text{EPSEN}}$	Comments
0	0	P0, P2	Active	Inactive	Addresses 0–0FFFFH
0	1	N/A	N/A	N/A	Invalid Combination
1	0	P5, P6	Inactive	Active	Addresses 0–0FFFFH
1	1	P5, P6 P0, P2	Inactive Active	Active Inactive	Addresses 0–1FFFH Addresses ≥ 2000H

Table 2. 8XC152 Product Differences

ROMless Version	CSMA/CD and HDLC/SDLC	HDLC/SDLC Only	ROM Version Available	PLCC and DIP	PLCC Only	5 I/O Ports	7 I/0 Ports
80C152JA	*		*(83C152JA)	*		*	
80C152JB	*				*		*
80C152JC		*	*(83C152JC)	*		*	
80C152JD		*			*		*

NOTES:
* = options available
0 standard frequency range 3.5 MHz to 12 MHz
0 "–1" frequency range 3.5 MHz to 16.5 MHz

<table>
<tr><th colspan="2">Pin #</th><th colspan="3">Pin Description</th></tr>
<tr><td>DIP</td><td>PLCC(1)</td><td colspan="3"></td></tr>
<tr><td>48</td><td>2</td><td colspan="3">V_{CC}—Supply voltage.</td></tr>
<tr><td>24</td><td>3,33(2)</td><td colspan="3">V_{SS}—Circuit ground.</td></tr>
<tr><td>18-21,
25-28</td><td>27-30,
34-37</td><td colspan="3">Port 0—Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.
Port 0 is also the multiplexed low-order address and data bus during accesses to external program memory if EBEN is pulled low. During accesses to external Data Memory, Port 0 always emits the low-order address byte and serves as the multiplexed data bus. In these applications it uses strong internal pullups when emitting 1s.
Port 0 also outputs the code bytes during program verification. External pullups are required during program verification.</td></tr>
<tr><td>1-8</td><td>4-11</td><td colspan="3">Port 1—Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.
Port 1 also serves the functions of various special features of the 8XC152, as listed below:
<table>
<tr><th>Pin</th><th>Name</th><th>Alternate Function</th></tr>
<tr><td>P1.0</td><td>GRXD</td><td>GSC data input pin</td></tr>
<tr><td>P1.1</td><td>GTXD</td><td>GSC data output pin</td></tr>
<tr><td>P1.2</td><td>$\overline{DEN}$</td><td>GSC enable signal for an external driver</td></tr>
<tr><td>P1.3</td><td>$\overline{TXC}$</td><td>GSC input pin for external transmit clock</td></tr>
<tr><td>P1.4</td><td>$\overline{RXC}$</td><td>GSC input pin for external receive clock</td></tr>
<tr><td>P1.5</td><td>$\overline{HLD}$</td><td>DMA hold input/output</td></tr>
<tr><td>P1.6</td><td>$\overline{HLDA}$</td><td>DMA hold acknowledge input/output</td></tr>
</table></td></tr>
<tr><td>29-36</td><td>41-48</td><td colspan="3">Port 2—Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.
Port 2 emits the high-order address byte during fetches from external Program Memory if EBEN is pulled low. During accesses to external Data Memory that use 16-bit addresses (MOVX @ DPTR and DMA operations), Port 2 emits the high-order address byte. In these applications it uses strong internal pullups when emitting 1s.
During accesses to external Data Memory that use 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.
Port 2 also receives the high-order address bits during program verification.</td></tr>
<tr><td>10- 17</td><td>14-16,
18, 19,
23-25</td><td colspan="3">Port 3—Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.
Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:
<table>
<tr><th>Pin</th><th>Name</th><th>Alternate Function</th></tr>
<tr><td>P3.0</td><td>RXD</td><td>Serial input line</td></tr>
<tr><td>P3.1</td><td>TXD</td><td>Serial output line</td></tr>
<tr><td>P3.2</td><td>$\overline{INT0}$</td><td>External Interrupt 0</td></tr>
<tr><td>P3.3</td><td>$\overline{INT1}$</td><td>External Interrupt 1</td></tr>
<tr><td>P3.4</td><td>T0</td><td>Timer 0 external input</td></tr>
<tr><td>P3.5</td><td>T1</td><td>Timer 1 external input</td></tr>
<tr><td>P3.6</td><td>$\overline{WR}$</td><td>External Data Memory Write strobe</td></tr>
<tr><td>P3.7</td><td>$\overline{RD}$</td><td>External Data Memory Read strobe</td></tr>
</table></td></tr>
</table>

NOTES:
1. N.C. pins on PLCC package may be connected to internal die and should not be used in customer applications.
2. It is recommended that both Pin 3 and Pin 33 be grounded for PLCC devices.

Pin Description (Continued)

Pin #		Pin Description
47-40	65-58	**Port 4**—Port 4 is an 8-bit bidirectional I/O port with internal pullups. Port 4 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 4 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups. In addition, Port 4 also receives the low-order address bytes during program verification.
9	13	**$\overline{RST}$**—Reset input. A logic low on this pin for three machine cycles while the oscillator is running resets the device. An internal pullup resistor permits a power-on reset to be generated using only an external capacitor to V_{SS}. Although the GSC recognizes the reset after three machine cycles, data may continue to be transmitted for up to 4 machine cycles after Reset is first applied.
38	55	**ALE**—Address Latch Enable output signal for latching the low byte of the address during accesses to external memory. In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory. While in Reset, ALE remains at a constant high level.
37	54	**$\overline{PSEN}$**—Program Store Enable is the Read strobe to External Program Memory. When the 8XC152 is executing from external program memory, $\overline{PSEN}$ is active (low). When the device is executing code from External Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to External Data Memory. While in Reset, $\overline{PSEN}$ remains at a constant high level.
39	56	**$\overline{EA}$**—External Access enable. $\overline{EA}$ must be externally pulled low in order to enable the 8XC152 to fetch code from External Program Memory locations 0000H to 0FFFH. $\overline{EA}$ must be connected to V_{CC} for internal program execution.
23	32	**XTAL1**—Input to the inverting oscillator amplifier and input to the internal clock generating circuits.
22	31	**XTAL2**—Output from the inverting oscillator amplifier.
N/A	17, 20 21, 22 38, 39 40, 49	**Port 5**—Port 5 is an 8-bit bidirectional I/O port with internal pullups. Port 5 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 5 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups. Port 5 is also the multiplexed low-order address and data bus during accesses to external program memory if EBEN is pulled high. In this application it uses strong pullups when emitting 1s.
N/A	67, 66 52, 57 50, 68 1, 51	**Port 6**—Port 6 is an 8-bit bidirectional I/O port with internal pullups. Port 6 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 6 pins that are externally pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups. Port 6 emits the high-order address byte during fetches from external Program Memory if EBEN is pulled high. In this application it uses strong pullups when emitting 1s.
N/A	12	**EBEN**—E-Bus Enable input that designates whether program memory fetches take place via Ports 0 and 2 or Ports 5 and 6. Table 1 shows how the ports are used in conjunction with EBEN.
N/A	53	**$\overline{EPSEN}$**—E-bus Program Store Enable is the Read strobe to external program memory when EBEN is high. Table 2 shows when $\overline{EPSEN}$ is used relative to $\overline{PSEN}$ depending on the status of EBEN and $\overline{EA}$.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3.

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

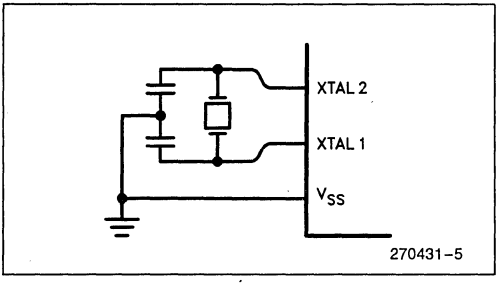


Figure 3. Using the On-Chip Oscillator

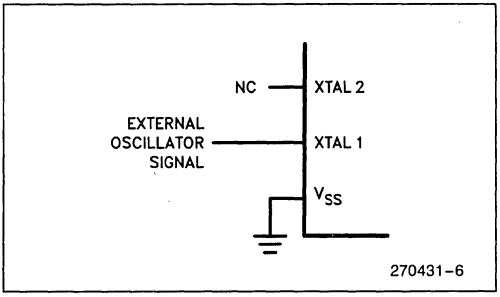


Figure 4. External Clock Drive

IDLE MODE

In Idle Mode, the CPU puts itself to sleep while most of the on-chip peripherals remain active. The major peripherals that do not remain active during Idle, are the DMA channels. The Idle Mode is invoked by software. The content of the on-chip RAM and all the Special Function Registers remain unchanged during this mode. The Idle Mode can be terminated by any enabled interrupt or by a hardware reset.

POWER DOWN MODE

In Power Down Mode, the oscillator is stopped and all on-chip functions cease except that the on-chip RAM contents are maintained. The mode Power Down is invoked by software. The Power Down Mode can be terminated only by a hardware reset.

Table 3. Status of the External Pins During Idle and Power Down Modes

80C152JA/83C152JA/80C152JC/83C152JC

Mode	Program Memory	ALE	$\overline{PSEN}$	Port 0	Port 1	Port 2	Port 3	Port 4
Idle	Internal	1	1	Data	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data	Data
Power Down	Internal	0	0	Data	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data	Data

80C152JB/80C152JD

Mode	Instruction Bus	ALE	$\overline{PSEN}$	$\overline{EPSEN}$	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6
Idle	P0, P2	1	1	1	Float	Data	Address	Data	Data	0FFH	0FFH
Idle	P5, P6	1	1	1	Data	Data	Data	Data	Data	0FFH	Address
Power Down	P0, P2	0	0	1	Float	Data	Data	Data	Data	0FFH	0FFH
Power Down	P5, P6	0	1	0	Data	Data	Data	Data	Data	0FFH	0FFH

NOTE:
For more detailed information on the reduced power modes refer to the Embedded Controller Handbook, and Application Note AP-252, "Designing with the 80C51BH."

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any pin to V_{SS} ..−0.5V to (V_{CC} + 0.5V)
Voltage on V_{CC} to VSS.....−0.5V to +6.5V
Power Dissipation .....1.0W(9)

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V)

Symbol	Parameter	Min	Typ (Note 3)	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (All Except $\overline{EA}$, EBEN)	−0.5		$0.2V_{CC}-0.1$	V	
V_{IL1}	Input Low Voltage ($\overline{EA}$, EBEN)	−0.5		$0.2V_{CC}-0.3$	V	
V_{IH}	Input High Voltage (Except XTAL1, $\overline{RST}$)	$0.2V_{CC}+0.9$		$V_{CC}+0.5$	V	
V_{IH1}	Input High Voltage (XTAL1, $\overline{RST}$)	$0.7V_{CC}$		$V_{CC}+0.5$	V	
V_{OL}	Output Low Voltage (Ports 1, 2, 3, 4, 5, 6)			0.45	V	I_{OL} = 1.6 mA (Note 4)
V_{OL1}	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$, $\overline{EPSEN}$)			0.45	V	I_{OL} = 3.2 mA (Note 4)
V_{OH}	Output High Voltage (Ports 1, 2, 3, 4, 5, 6 COMM9 ALE, $\overline{PSEN}$, $\overline{EPSEN}$)	2.4			V	I_{OH} = −60 μA V_{CC} = 5V ±10%
		$0.9V_{CC}$			V	I_{OH} = −10 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	2.4			V	I_{OH} = −400 μA V_{CC} = 5V ±10%
		$0.9V_{CC}$			V	I_{OH} = −40 μA (Note 5)
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3, 4, 5, 6)			−50	μA	V_{IN} = 0.45V
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3, 4, 5, 6)			−650	μA	V_{IN} = 2V
I_{LI}	Input Leakage (Port 0, $\overline{EA}$)			±10	μA	$0.45<V_{IN}<V_{CC}$
RRST	Reset Pullup Resistor	40			kΩ	
I_{IH}	Logical 1 Input Current (EBEN)			+60	μA	
I_{CC}	Power Supply Current :					
	Active (16.5 MHz)		31	41.1	mA	(Note 6)
	Idle (16.5 MHz)		8	15.4	mA	(Note 6)
	Power Down Mode		10		μA	V_{CC} = 2.0V to 5.5V

NOTES:
3. "Typicals" are based on samples taken from early manufacturing lots and are not guaranteed. The measurements were made with V_{CC} = 5V at room temperature.
4. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
5. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the $0.9V_{CC}$ specification when the address bits are stabilizing.
6. I_{CC} is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 5 ns, $V_{IL} = V_{SS} + 0.5V$, $V_{IH} = V_{CC} - 0.5V$; XTAL2 N.C.; Port 0 pins connected to V_{CC}. "Operating" current is measured with $\overline{EA}$ connected to V_{CC} and $\overline{RST}$ connected to V_{SS}. "Idle" current is measured with $\overline{EA}$ connected to V_{SS}, $\overline{RST}$ connected to V_{CC} and GSC inactive.
7. The specifications relating to external data memory characteristics are also applicable to DMA operations.
8. TQVWX should not be confused with TQVWX as specified for 80C51BH. On 80C152, TQVWX is measured from data valid to rising edge of $\overline{WR}$. On 80C51BH, TQVWX is measured from data valid to falling edge of $\overline{WR}$. See timing diagrams.
9. This value is based on the maximum allowable die temperature and the thermal resistance of the package.
10. All specifications relating to external program memory characteristics are applicable to:
    $\overline{EPSEN}$ for $\overline{PSEN}$
    Port 5 for Port 0
    Port 6 for Port 2
    when EBEN is at a Logical 1 on the 80C152JB/JD.

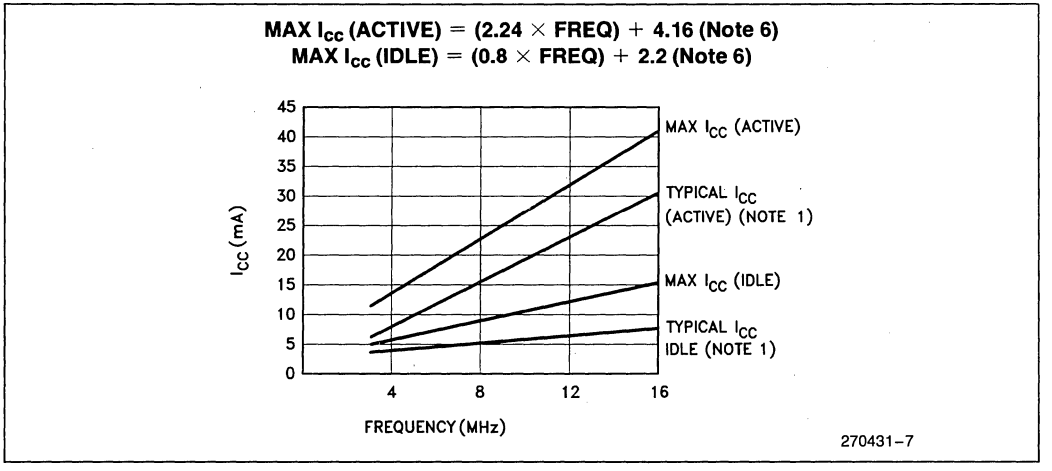


Figure 5. I_{CC} vs Frequency

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address.
C: Clock
D: Input data.
H: Logic level HIGH.
I: Instruction (program memory contents).
L: Logic level LOW, or ALE.
P: $\overline{PSEN}$.
Q: Output data.
R: $\overline{READ}$ signal.
T: Time.
V: Valid.
W: $\overline{WRITE}$ signal.
X: No longer a valid logic level.
Z: Float.

For example,

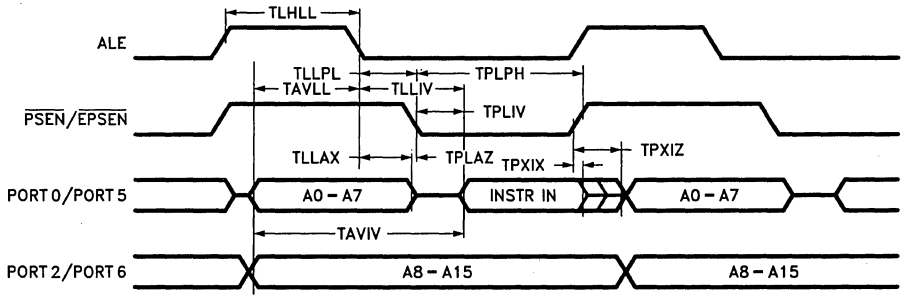
TAVLL = Time for Address Valid to ALE Low.
TLLPL = Time for ALE Low to $\overline{PSEN}$ Low.

A.C. CHARACTERISTICS (T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF; Load Capacitance for All Other Outputs = 80 pF)

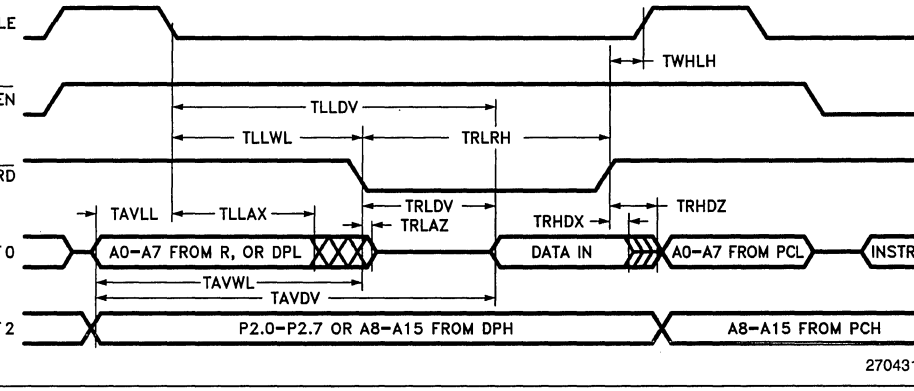
EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS (Note 7, 10)

Symbol	Parameter	16.5 MHz		Variable Oscillator		Unit
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency 80C152JA/JC 83C152JA/JC 80C152JB/JD			3.5	12	MHz
	80C152JA/JC-1 83C152JA/JC-1 80C152JB/JD-1			3.5	16.5	MHz
TLHLL	ALE Pulse Width	81		2TCLCL-40		ns
TAVLL	Address Valid to ALE Low	5		TCLCL-55		ns
TLLAX	Address Hold After ALE Low	25		TCLCL-35		ns
TLLIV	ALE Low to Valid Instruction In		142		4TCLCL-100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	20		TCLCL-40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	137		3TCLCL-45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instruction In		77		3TCLCL-105	ns
TPXIX	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instruction Float After $\overline{PSEN}$		35		TCLCL-25	ns
TAVIV	Address to Valid Instruction In		198		5TCLCL-105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	263		6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	263		6TCLCL-100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		138		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		51		2TCLCL-70	ns
TLLDV	ALE Low to Valid Data In		335		8TCLCL-150	ns
TAVDV	Address to Valid Data In		380		9TCLCL-165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	132	232	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	112		4TCLCL-130		ns
TQVWX(8)	Data Valid to $\overline{WR}$ Transition	196		6TCLCL-167		ns
TWHQX	Data Hold After $\overline{WR}$	10		TCLCL-50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	20	100	TCLCL-40	TCLCL+40	ns

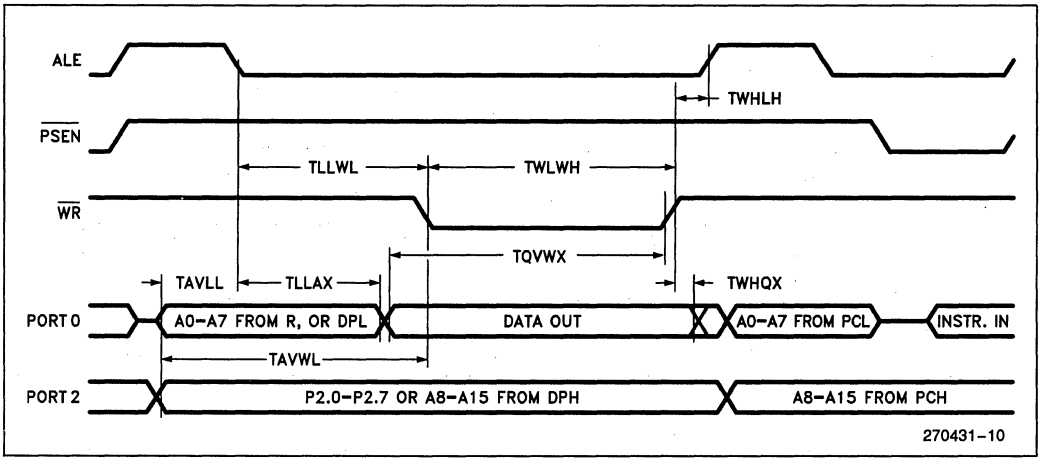
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY READ CYCLE



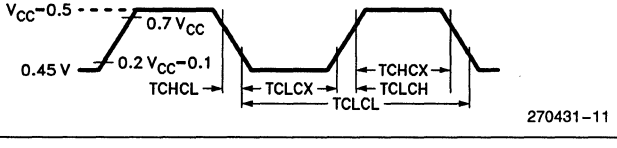
EXTERNAL DATA MEMORY WRITE CYCLE



EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	3.5	16.5	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

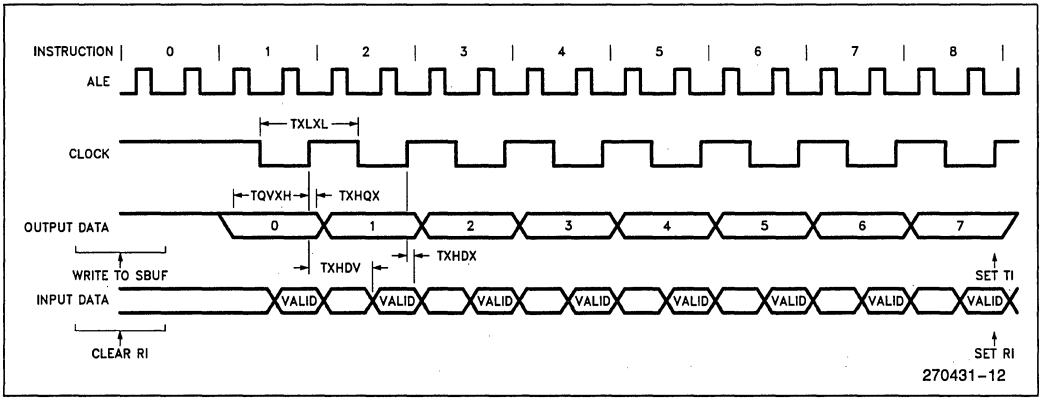
EXTERNAL CLOCK DRIVE WAVEFORM



LOCAL SERIAL CHANNEL TIMING—SHIFT REGISTER MODE

Symbol	Parameter	16.5 MHz		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	727		12TCLCL		ns
TQVXH	Output Data Setup to Clock Rising Edge	473		10TCLCL-133		ns
TXHQX	Output Data Hold After Clock Rising Edge	4		2TCLCL-117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		473		10TCLCL-133	ns

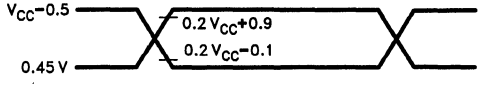
SHIFT REGISTER MODE TIMING WAVEFORMS



270431-12

A.C. TESTING:

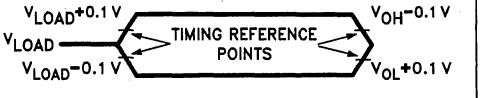
INPUT, OUTPUT WAVEFORMS



270431-13

AC Inputs During Testing are Driven at $V_{CC}-0.5$ for a Logic "1" and 0.45V for a Logic "0". Timing Measurements are made at V_{IH} Min for a Logic "1" and V_{IL} Max for a Logic "0".

FLOAT WAVEFORM



270431-14

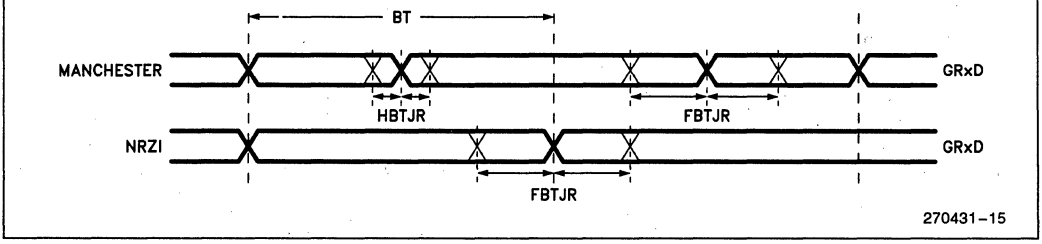
For Timing Purposes a Port Pin is no Longer Floating when a 100 mV change from Load Voltage Occurs, and Begins to Float when a 100 mV change from the Loaded V_{OH}/V_{OL} Level occurs $I_{OL}/I_{OH} \geq \pm 20$ mA.

GLOBAL SERIAL PORT TIMINGS—Internal Baud Rate Generator

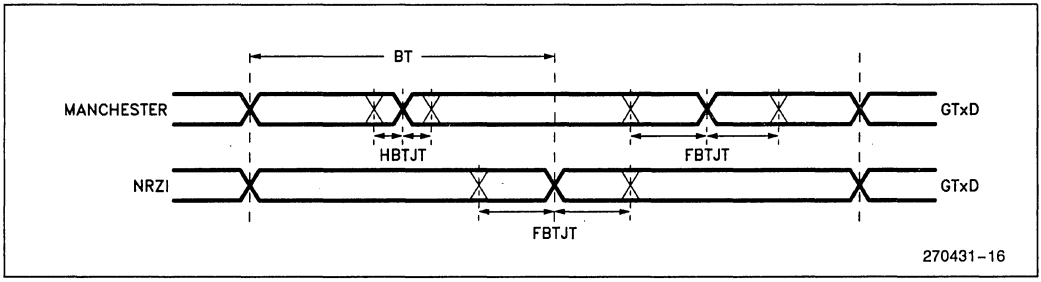
Symbol	Parameter	16.5 MHz (BAUD = 0)		Variable Oscillator		Unit
		Min	Max	Min	Max	
HBTJR	Allowable jitter on the Receiver for ½ bit time (Manchester encoding only)		0.0375		(0.125 × (BAUD+1)× 8TCLCL) −25 ns	μs
FBTJR	Allowable jitter on the Receiver for one full bit time (NRZI and Manchester)		0.10		(0.25 × (BAUD+1)× 8TCLCL) −25 ns	μs
HBTJT	Jitter of data from Transmitter for ½ bit time (Manchester encoding only)		±10		±10	ns
FBTJT	Jitter of data from Transmitter for one full bit time (NRZI and Manchester)		±10		±10	ns
DRTR	Data rise time for Receiver(11)		20		20	ns
DFTR	Data fall time for Receiver(12)		20		20	ns

NOTES:
11. Same as TCLCH, use External Clock Drive Waveform.
12. Same as TCHCL, use External Clock Drive Waveform.

GSC RECEIVER TIMINGS (INTERNAL BAUD RATE GENERATOR)



GSC TRANSMIT TIMINGS (INTERNAL BAUD RATE GENERATOR)

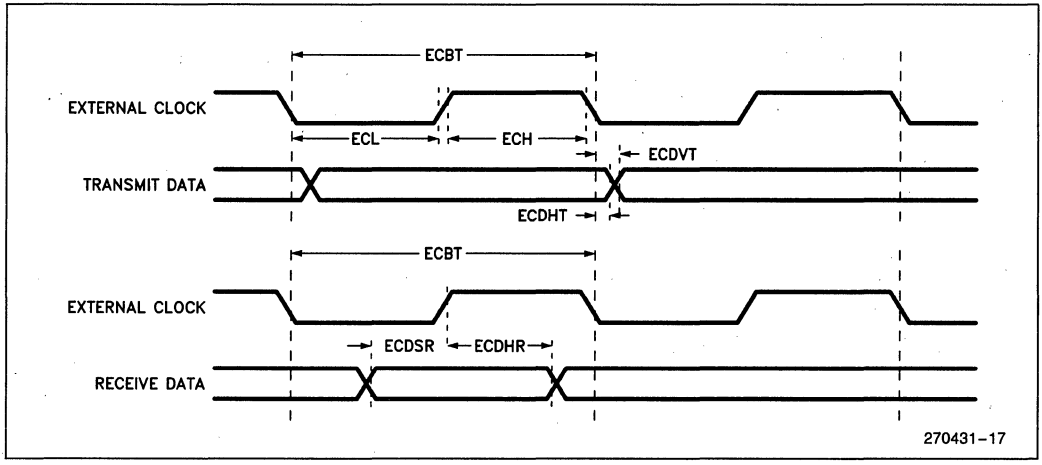


GLOBAL SERIAL PORT TIMINGS—External Clock

Symbol	Parameter	16.5 MHz		Variable Oscillator		Unit
		Min	Max	Min	Max	
1/ECBT	GSC Frequency with an External Clock	0.009	2.4	0.009	$F_{OSC} \times 0.145$	MHz
ECH	External Clock High	170		2TCLCL + 45 ns		ns
ECL(13)	External Clock Low	170		2TCLCL + 45 ns		ns
ECRT	External Clock Rise Time(11)		20		20	ns
ECFT	External Clock Fall Time(12)		20		20	ns
ECDVT	External Clock to Data Valid Out - Transmit (to External Clock Negative Edge)		150		150	ns
ECDHT	External Clock Data Hold - Transmit (to External Clock Negative Edge)	0		0		ns
ECDSR	External Clock Data Set-up - Receiver (to External Clock Positive Edge)	45		45		ns
ECDHR	External Clock to Data Hold - Receiver (to External Clock Positive Edge)	50		50		ns

NOTE:
13. When using the same external clock to drive both the receiver and transmitter, the minimum ECL spec effectively becomes 195 ns at all frequencies (assuming 0 ns propagation delay) because ECDVT (150 ns) plus ECDSR (45 ns) requirements must also be met (150 + 45 = 195 ns). The 195 ns requirement would also increase to include the maximum propagation delay between receivers and transmitters.

GSC TIMINGS (EXTERNAL CLOCK)



DESIGN NOTES

Within the 8XC152 there exists a race condition that may set both the RDN and AE bits at the end of a valid reception. This will not cause a problem in the application as long as the following steps are followed:

—Never give the receive error interrupt a higher priority than the valid reception interrupt

—Do not leave the valid reception interrupt service routine when AE is set by using a RETI instruction until AE is cleared. To clear AE set the GREN bit, this enables the receiver. If the user desires that the receiver remain disabled, clear GREN after setting it before leaving the interrupt service routine.

—If the AE bit is checked by user software in response to a valid reception interrupt, the status of AE should be considered invalid.

The race condition is dependent upon both the temperature that the device is currently operating at and the processing the device received during the wafer fabrication.

DATA SHEET REVISION SUMMARY

The following represents the key differences between this datasheet and the "-001" version of the 80C152/83C152 data sheet. Please review this summary carefully.

1. Status of data sheet changed from "ADVANCED" to "PRELIMINARY".
2. 80C152JC, 83C152JC, and 80C152JD were added.
3. Added AE/RDN design note.
4. This revision summary was added.
5. Note #13 was added (Effective ECL spec at higher clock rates).
6. Table #2 changed to Table #3 (Status of pins during Idle/Power Down).
7. Current Table #2 was added (JA vs. JB vs. JC vs. JD matrix).
8. Transmit jitter spec changed from ±35 ns and ±70 ns to ±10 ns.

8XC152JA/JB/JC/JD
UNIVERSAL COMMUNICATION CONTROLLER
8-BIT MICROCOMPUTER

EXPRESS

- **Extended Temperature Range**
- **Burn-In**
- **3.5 to 12 MHz V_{CC} = 5V ±10%**

The Intel EXPRESS system offers enhancements to the operational specifications of the 8XC152 microcontroller. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in and an extended temperature range.

With the commercial standard temperature range, operational characteristics are guaranteed over the temperature range of 0°C to +70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of −40°C to +85°C.

The optional burn-in is dynamic for a minimum time of 160 hours at 125°C with V_{CC} = 6.9V ±0.25V, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

September 1988
Order Number: 270540-002

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets. Maximum oscillator frequency for express products is 12 MHz.

D.C. CHARACTERISTICS $T_A = -40°C$ to $+85°C$; $V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage (Except $\overline{EA}$, EBEN)	−0.5	$0.2V_{CC} - 0.15$	V	
V_{IL1}	$\overline{EA}$, EBEN	−0.5	$0.2V_{CC} - 0.35$	V	
V_{IH}	Input High Voltage (Except XTAL1, $\overline{RST}$)	$0.2V_{CC} + 1.0$	$V_{CC} + 0.5$	V	
V_{IH1}	Input High Voltage to XTAL1, $\overline{RST}$	$0.7V_{CC} + 0.1$	$V_{CC} + 0.5$	V	
I_{IL}	Logical 0 Input Current (Port 1, 2, 3, 4, 5, 6)		−75	μA	$V_{in} = 0.45V$
I_{TL}	Logical 1 to 0 transition Current (Ports 1, 2, 3, 4, 5, 6)		−750	μA	$V_{in} = 2.0V$

Table 1. Prefix Identification

Prefix	Package Type	Temperature Range	Burn-In
P	Plastic	Commercial	No
C	Ceramic	Commercial	No
N	PLCC	Commercial	No
LP	Plastic	Extended	Yes
LC	Ceramic	Extended	Yes
LN	PLCC	Extended	Yes

NOTE:
- Commercial temperature range is 0°C to 70°C. Extended temperature range is −40°C to +85°C.
- Burn-in is dynamic for a minimum time of 160 hours at 125°C, $V_{CC} = 6.9V \pm 0.25V$, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).

Examples:

P80C152JA indicates 80C152JA in a plastic package and specified for commercial temperature range, without burn-in.
LC83C152JA indicates 83C152JA in a ceramic package and specified for extended temperature range with burn-in.

DATA SHEET REVISION SUMMARY

The following represents the key differences between this datasheet and the "-001" version of the express 80C152/83C152 datasheet. Please review this summary carefully.

1. Status of datasheet changed from "ADVANCED" to "PRELIMINARY".
2. 80C152JC, 83C152JC, and 80C152JD were added.
3. This revision summary was added.

HARDWARE DESCRIPTION OF THE 8XC451

INTRODUCTION

The 8XC451 is an expanded I/O, 8-bit control-oriented microcontroller based on the MCS®-51 architecture. The 8XC451 contains all of the features of the 80C51 with three extra I/O ports, one of which is a bi-directional bus-type port with I/O strobes and flags. The 8XC451 features include:

- One Extra Strobed Bus Port
- Two Extra I/O Ports
- 56 Programmable I/O Lines
- 4 Kbytes Mask-Programmable ROM
- 128 x 8-Bit RAM
- Full Duplex Serial Channel
- 64K Program Memory Space
- 64K Data Memory Space
- Power Control Modes
- Boolean Processor

The 8XC451 uses the standard 8051 instruction set and is compatible with the existing MCS-51 family of products. The 83C451 is the factory masked ROM device; the 80C451 is the ROMless device; and the 87C451 is the EPROM device. It is assumed that the reader is familiar with the 8051 architecture. For more detailed information on the 8051, consult the "Architectural Overview Chapter" and the "Hardware Description of the 8051, 8052, and 80C51" chapter in the *Embedded Controller Handbook.*

PORT 4

This port is an 8-bit bidirectional I/O port with internal pullups similar to Port 1. Port 4 has SFR address C0H assigned. Port 4 is bit addressable and is functionally identical to Port 1.

P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0

Port 4 Data Address = 0C0H Reset Value = 11111111B

PORT 5

This port is an 8-bit bidirectional I/O port with internal pullups similar to Port 1. Port 5 has SFR address C8H assigned. Port 5 is bit addressable and is functionally identical to Port 1.

P5.7	P5.6	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0

Port 5 Data Address = 0C8H Reset Value = 11111111B

PORT 6

Port 6 is a special 8-bit bidirectional I/O port with internal pullups. This port can be used as a standard I/O port, or in strobed modes of operation in conjunction with four newly added special control lines: $\overline{ODS}$, $\overline{IDS}$, AFLAG, and BFLAG (see Figure 1). Port 6 operating modes are controlled by the port 6 control status register (CSR). Port 6 and the CSR are addressed at the special function register addresses shown in Table 1. The following four control pins are used in conjunction with port 6.

$\overline{ODS}$: Handshake input for port 6 strobed output data. $\overline{ODS}$ can be programmed to control the port 6 output drivers and the output buffer full flag (OBF), or to clear only the OBF flag bit in the CSR (output-always mode). $\overline{ODS}$ is active low for output driver control. The OBF flag can be programmed to be cleared on the negative or positive edge of $\overline{ODS}$.

$\overline{IDS}$: Handshake input for port 6 strobed input data. $\overline{IDS}$ is used to control the port 6 input latch and input buffer full flag (IBF) bit in the CSR. The input data latch can be programmed to be transparent when the $\overline{IDS}$ is low and latched on the positive transition of $\overline{IDS}$, or to latch only on the positive transition of $\overline{IDS}$. Correspondingly, the IBF flag is set on the negative or positive transition of $\overline{IDS}$.

AFLAG: A bidirectional I/O pin. AFLAG can be programmed to be an output, set high or low under program control, or to output the state of the output buffer full flag. AFLAG can also be programmed to be an input which selects whether the contents of the output buffer, or the contents of the port 6 control status register will be output on port 6. This feature grants complete port 6 status to external devices.

BFLAG: A bidirectional I/O pin. BFLAG can be programmed to be an output, set high or low under program control, or to output the state of the input buffer full flag. BFLAG can also be programmed to input an enable signal for port 6. When BFLAG is used as an enable input, port 6 output drivers are in the high impedance state, and the input latch does not respond to the $\overline{IDS}$ strobe when BFLAG is high. Both features are enabled when BFLAG is low. This feature facilitates the use of the 8XC451 in bussed multiprocessor systems.

P6.7	P6.6	P6.5	P6.4	P6.3	P6.2	P6.1	P6.0

Port 6 Data Address = 0D8H Reset Value = 11111111B

PORT 6 CONTROL STATUS REGISTER

The control status register (CSR) establishes the mode of operation for port 6 and indicates the current status of the Port 6 I/O registers. All control status register bits can be read and written by the CPU, except bits 0 and 1, which are read only. Reset writes ones to bits 2 through 7, and writes zeros to bits 0 and 1.

CSR.0—Input Buffer Full Flag (IBF) (Read Only)

The IBF bit is set to a logic 1 when Port 6 data is loaded into the input buffer under control of $\overline{IDS}$. This can occur on the negative or positive edge of $\overline{IDS}$, as determined by CSR.2. IBF is cleared when the CPU reads the input buffer register.

CSR.1—Output Buffer Full Flag (OBF) (Read Only)

The OBF flag is set to a logic 1 when the CPU writes to the Port 6 output data buffer. OBF is cleared by the positive or negative edge of $\overline{ODS}$, as determined by CSR.3.

CSR.2—$\overline{IDS}$ Mode Select (IDSM)

When CSR.2 = 0, a low-to-high transition on the $\overline{IDS}$ pin sets the IBF flag. The Port 6 input buffer is loaded on the $\overline{IDS}$ positive edge. When CSR.2 = 1, a high-to-low transition on the $\overline{IDS}$ pin sets the IBF flag. The Port 6 input buffer is transparent when $\overline{IDS}$ is low, and latched when $\overline{IDS}$ is high.

CSR.3—Output Buffer Full Flag Clear Mode (OBFC)

When CSR.3 = 1, the positive edge of the $\overline{ODS}$ input clears the OBF flag. When CSR.3 = 0, the negative edge of the $\overline{ODS}$ input clears the OBF flag.

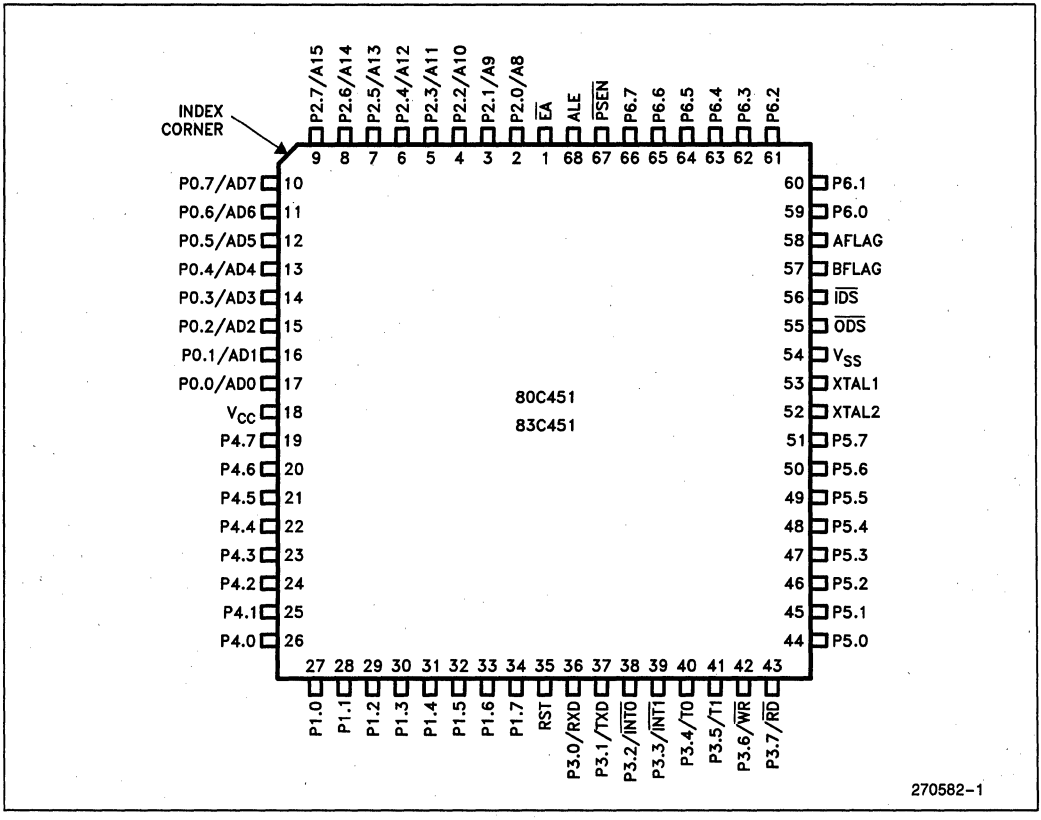


Figure 1. Pin Connections (PLCC)

Port 6 Control Status Register

Symbol	Position	Function
MB1, MB0	CSR.7, 6	BFLAG Mode Select: 0,0 = Logic 0 Out; 0,1 = Logic 1 Out* 1,0 = IBF Output; 1,1 = $\overline{PE}$ Input BFLAG: 0 = Select; 1 = Disable I/O
MA1, MA0	CSR.5, 4	AFLAG Mode Select: 0,0 = Logic 0 Out; 0,1 = Logic 1 Out 1,0 = OBF Output*; 1,1 = SEL Input Status AFLAG: 0 = Data; 1 = Control/
OBFC	CSR.3	Output Buffer Flag Clear Mode: 0 = Negative Edge of $\overline{ODS}$ 1 = Positive Edge of $\overline{ODS}$
IDSM	CSR.2	Input Data Strobe Mode: 0 = Positive Edge of $\overline{IDS}$ 1 = Low Level of $\overline{IDS}$
OBF	CSR.1	Output Buffer Full Flag: 0 = Output Data Buffer Empty 1 = Output Data Buffer Full
IBF	CSR.0	Input Buffer Full Flag: 0 = Input Data Buffer Empty 1 = Input Data Buffer Full

***NOTE**
Output-always mode: MB1 = 0, MA1 = 1, and MA0 = 0. In this mode, Port 6 is always enabled for output. $\overline{ODS}$ only clears the OBF flag.

CSR.4, CSR.5—AFLAG Mode Select (MA0, MA1)

Bits 4 and 5 select the mode of operation for the AFLAG pin, as follows:

MA1	MA0	AFLAG Function
0	0	Logic 0 Output
0	1	Logic 1 Output
1	0	OBF Flag Output (CSR.1)
1	1	Select (SEL) Input Mode

The select (SEL) input mode is used to determine whether the Port 6 data register or the control status register is output on Port 6. When the select feature is enabled, the AFLAG input controls the source of Port 6 output data. A logic 0 on AFLAG input selects the Port 6 data register, and a logic 1 on AFLAG input selects the control status register.

CSR.6, CSR.7—BFLAG Mode Select (MB0, MB1)

Bits 6 and 7 select the mode of operation for the BFLAG pin, as follows:

MB1	MB0	BFLAG Function
0	0	Logic 0 Output
0	1	Logic 1 Output
1	0	IBF Flag Output (CSR.0)
1	1	Port Enable ($\overline{PE}$)

In the port enable mode, $\overline{IDS}$ and $\overline{ODS}$ inputs are disabled when BFLAG input is high. When the BFLAG input is low, the port is enabled for I/O.

MB1	MB0	MA1	MA0	OBFC	IDSM	OBF	IBF

Port 6 Control Status Address = 0E8H
Reset Value = 11111100B Bit Addressable

Upon reset, Port 6 is configured in the bus interface input mode to prevent disruption of data on the bus of a host processor. It can also be configured as a standard I/O port by tying the control pins $\overline{IDS}$, $\overline{ODS}$, AFLAG and BFLAG to V_{SS} before reset. AFLAG and BFLAG can be used as simple outputs in the standard I/O port mode by tying only $\overline{IDS}$ and $\overline{ODS}$ to V_{SS}. Grounding $\overline{IDS}$ and $\overline{ODS}$ prevents any edges from occuring on these signals thereby preventing automatic data transfers and setting and resetting of flags.

SPECIAL FUNCTION REGISTERS

A map of the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have no effect.

User software should not write 1's to these unimplemented locations, since they may be used in future 8051 products to invoke new features. In that case the reset or inactive values of the new bits will always be 0, and their active values will be 1.

Table 1. Special Function Register Memory Map and Values after Reset

F8									FF
F0	*B 00000000								F7
E8	P6 CSR 11111100								EF
E0	*ACC 00000000								E7
D8	P6 Data 11111111								DF
D0	* PSW 00000000								D7
C8	P5 11111111								CF
C0	P4 11111111								C7
B8	*IP XXX00000								BF
B0	*P3 11111111								B7
A8	* IE 0XX00000								AF
A0	* P2 11111111								A7
98	* SCON 00000000	* SBUF XXXXXXXX							9F
90	* P1 11111111								97
88	* TCON 00000000	* TMOD 00000000	* TL0 00000000	* TL1 00000000	* TH0 00000000	* TH1 00000000			8F
80	* P0 11111111	* SP 00000111	* DPL 00000000	* DPH 00000000				* PCON ** 00XX0000	87

NOTES:
* = Found in the 8051 core (See 8051 Hardware Description for explanations of these SFRs)
** = See description of PCON SFR. Bit PCON.4 is not affected by reset.
X = Undefined

80C451/80C451-1/80C451-2 CHMOS SINGLE-CHIP 8-BIT CPU WITH RAM AND EXPANDED I/O

83C451/83C451-1/83C451-2 CHMOS SINGLE-CHIP 8-BIT CPU WITH 4K BYTES FACTORY MASK-PROGRAMMABLE ROM

8XC451 — 3.5 to 12 MHz, V_{CC} = 5V ±20%
8XC451-1 — 3.5 to 16 MHz, V_{CC} = 5V ±20%
8XC451-2 — 0.5 to 12 MHz, V_{CC} = 5V ±20%

- **Idle and Power Down Modes**
- **128 x 8-Bit RAM**
- **56 Programmable I/O Lines**
- **One Extra Strobed Bus Port**
- **Two 16-Bit Timer/Counters**
- **64K Program Memory Space**
- **TTL- and CMOS-Compatible Logic**
- **High Performance CHMOS Process**
- **Boolean Processor**
- **5 Interrupt Sources**
- **Two Extra Standard Ports**
- **Full Duplex Serial Channel**
- **64K Data Memory Space**
- **68-Pin PLCC Package**

The 8051 family of CHMOS products are fabricated on Intel's CHMOS III process and are functionally compatible with the standard 8051 family of HMOS and EPROM products. CHMOS III is a technology which combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. This combination expands the effectiveness of the powerful 8051 architecture and instruction set.

The 80C451/83C451 are similar to the 80C51 in features with three additional I/O ports, one of which is a special strobed port. The 83C451 contains 4K bytes of ROM. Both devices contain 128 bytes of RAM; 56 I/O lines; two 16-bit timer/counters; a five source two-level interrupt structure; a full-duplex serial port; on-chip oscillator and clock circuitry; and two software selectable modes of reduced activity for further power reduction — Idle and Power Down.

The extremely low operating power, along with the two reduced power modes, Idle and Power Down, make this part very suitable for low power applications. The Idle mode freezes the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

September 1988
Order Number: 270561-002

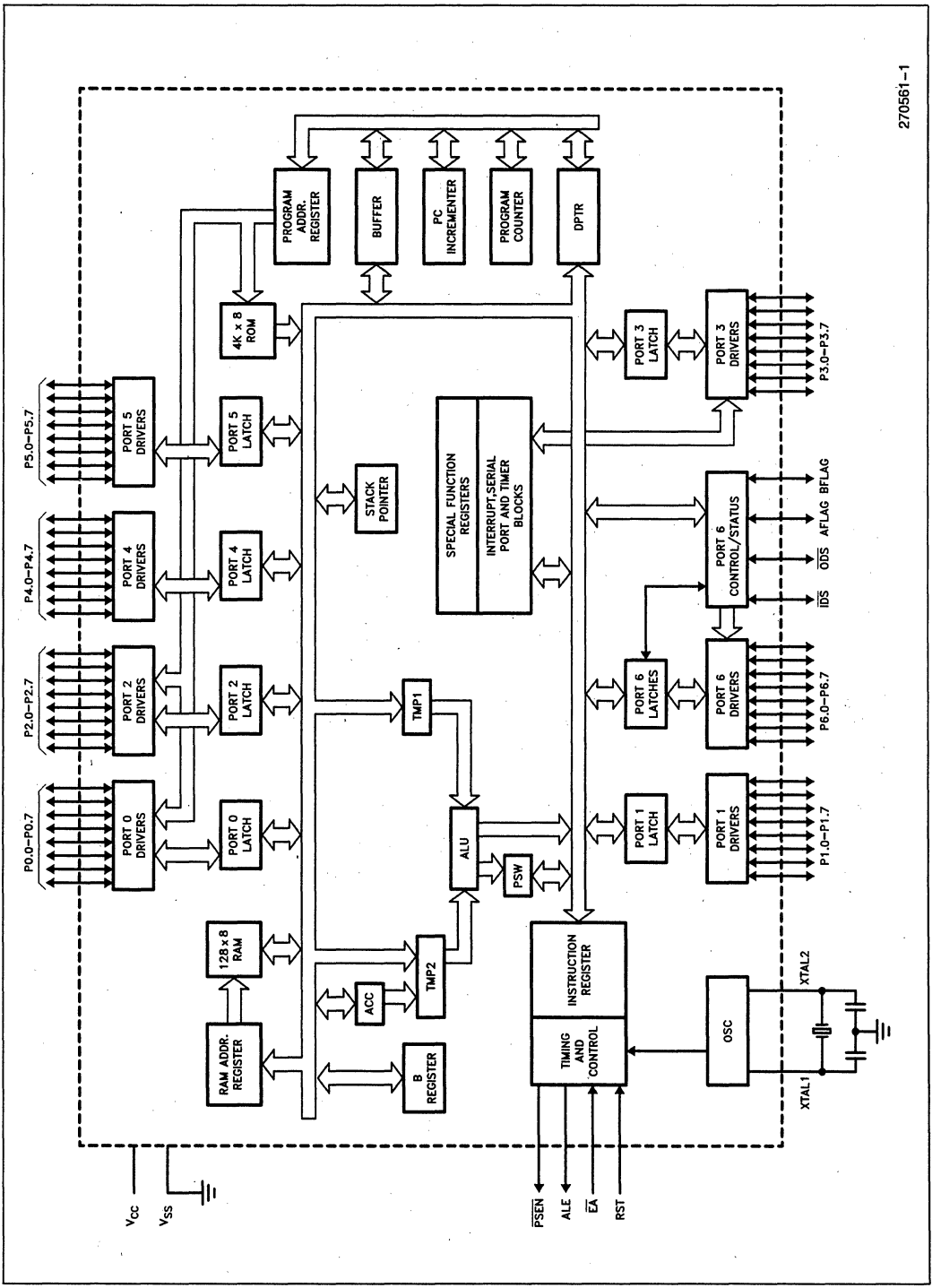


Figure 1. Block Diagram

PACKAGES

Part	Prefix	Package Type
83C451 80C451	N	68 Pin PLCC

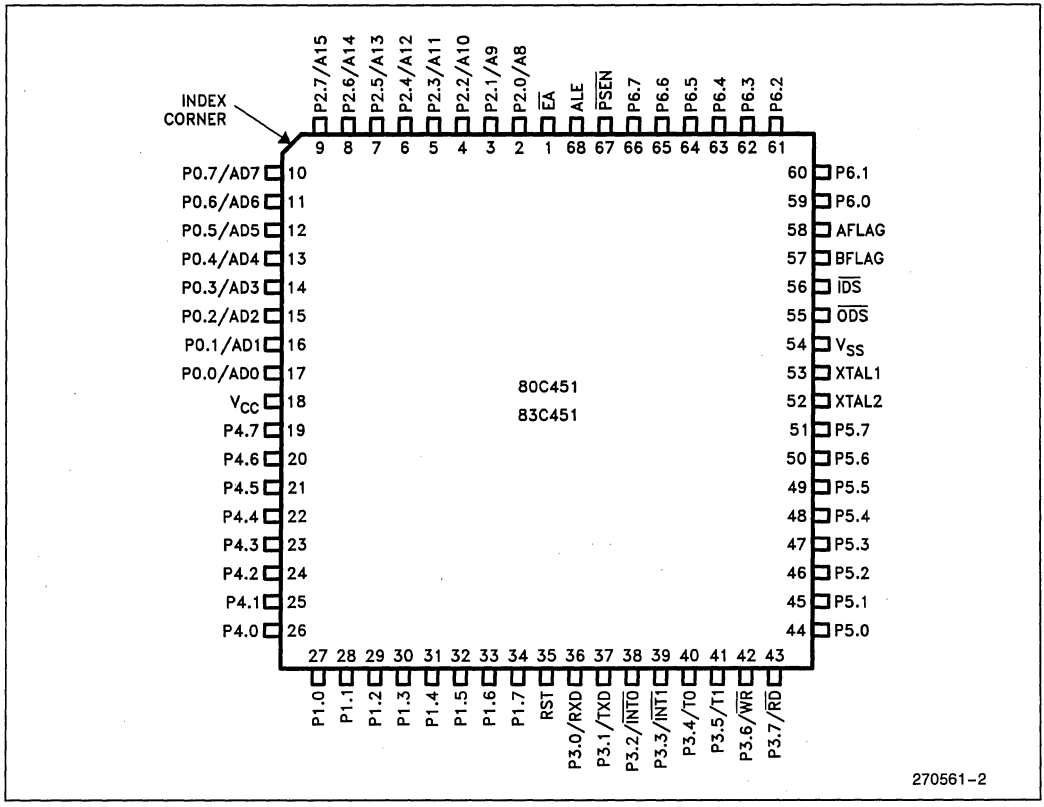


Figure 2. Pin Connections (PLCC)

PIN DESCRIPTION

V_{CC}: Supply voltage during normal, Idle, and Power Down operations.

V_{SS}: Circuit ground.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s. Port 0 also outputs the code bytes during program verification in the 83C451. External pullups are required during program verification.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX@DPTR). In this application it uses strong internal pullups when emitting 1s.

During accesses to external Data Memory that use 8-bit addresses (MOVX@Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives some control signals and the high-order address bits during program verification.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the 8051 Family, as listed below:

Pin	Name	Alternate Function
P3.0	RXD	Serial input line
P3.1	TXD	Serial output line
P3.2	$\overline{INT0}$	External interrupt 0
P3.3	$\overline{INT1}$	External interrupt 1
P3.4	T0	Timer 0 external input
P3.5	T1	Timer 1 external input
P3.6	$\overline{WR}$	External Data Memory write strobe
P3.7	$\overline{RD}$	External Data Memory read strobe

Port 4: Port 4 is an 8-bit bidirectional I/O port with internal pullups. Port 4 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 4 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 5: Port 5 is an 8-bit bidirectional I/O port with internal pullups. Port 5 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 5 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 6: Port 6 is a special 8-bit bidirectional I/O port with internal pullups. This port can be used as a standard I/O port, or in strobed modes of operation in conjunction with the following four special control lines.

Port 6 Control Lines

$\overline{ODS}$:	Output data strobe
$\overline{IDS}$:	Input data strobe
AFLAG:	A bidirectional I/O pin with internal pullups
BFLAG:	A bidirectional I/O pin with internal pullups

RST: Reset input. A logic high on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits a power-on reset to be generated using only an external capacitor to V_{CC}.

ALE: Address Latch Enable output signal for latching the low byte of the address during accesses to external memory.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$: Program Store Enable is the Read strobe to External Program Memory. When the 83C451 is executing from Internal Program Memory, $\overline{PSEN}$ is inactive (high). When the device is executing code from External Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to External Data Memory.

$\overline{EA}$: External Access enable. $\overline{EA}$ must be strapped to V_{SS} in order to enable the 8XC451 to fetch code from External Program Memory locations 0000H to 0FFFH.

$\overline{EA}$ must be strapped to V_{CC} for Internal Program execution.

XTAL1: Input to the inverting oscillator amplifier and input to the internal clock generating circuits.

XTAL2: Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3.

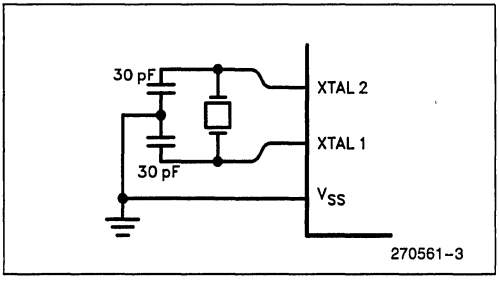


Figure 3. Using the On-Chip Oscillator

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

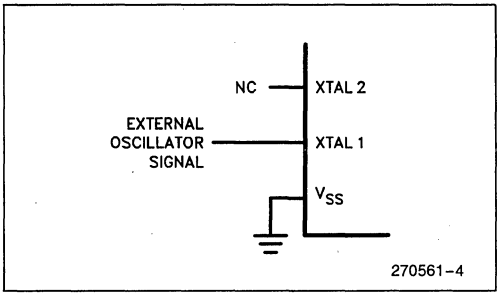


Figure 4. External Clock Drive

IDLE MODE

In Idle Mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the Special Function Registers remain unchanged during this mode. The Idle Mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when Idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

POWER DOWN MODE

In Power Down Mode, the oscillator is stopped and all on-chip functions cease except that the on-chip RAM content is maintained. The mode is invoked by software. The Power Down Mode can be terminated only by a hardware reset.

Table 1. Status of the External Pins during Idle and Power Down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	Port 0	Port 2	Ports 1, 3, 4, 5, 6
Idle	Internal	1	1	Data	Data	Data
Idle	External	1	1	Float	Address	Data
Power Down	Internal	0	0	Data	Data	Data
Power Down	External	0	0	Float	Data	Data

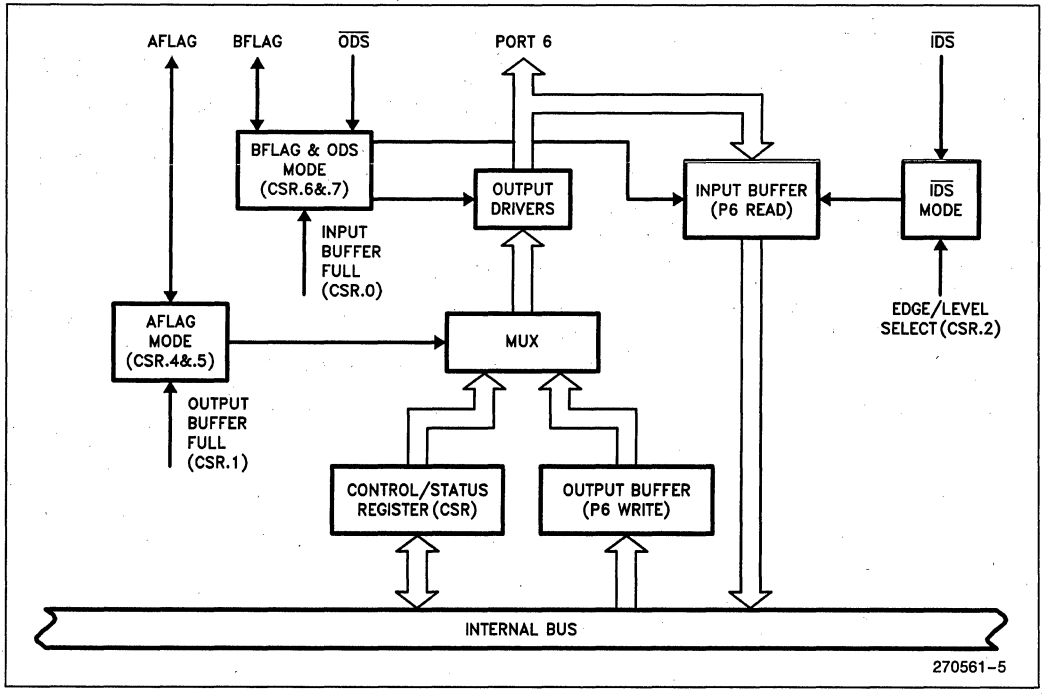


Figure 5. Port 6 Block Diagram

ABSOLUTE MAXIMUM RATINGS(1)*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin to V_{SS} ...−0.5V to V_{CC} + 0.5V
Voltage on V_{CC} to V_{SS} .....−0.5V to +6.5V
Maximum I_{OL} per I/O Pin .....15 mA
Power Dissipation.....1.0W
This value is based on the maximum allowable die temperature and the thermal resistance of the package.

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

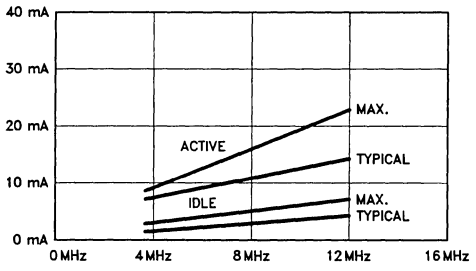
D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5V ±20%; V_{SS} = 0V

Symbol	Parameter	Limits		Unit	Test Conditions
		Min	Max		
V_{IL}	Input Low Voltage, except $\overline{EA}$	−0.5	0.2 V_{CC} − 0.1	V	
V_{IL1}	Input Low Voltage, $\overline{EA}$	−0.5	0.2 V_{CC} − 0.3	V	
V_{IH}	Input High Voltage (except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage (Ports 1,2,3,4,5,6, AFLAG, BFLAG)		0.3	V	I_{OL} = 100 μA(1)
			0.45	V	I_{OL} = 1.6 mA (1)
			1.0	V	I_{OL} = 3.5 mA(1,4)
V_{OL1}	Output Low Voltage (Port 0, ALE, $\overline{PSEN}$)		0.3	V	I_{OL} = 200 μA(1)
			0.45	V	I_{OL} = 3.2 mA(1)
			1.0	V	I_{OL} = 7 mA(1,4)
V_{OH}	Output High Voltage (Port 1,2,3,4,5,6, AFLAG, BFLAG)	V_{CC} − 0.3		V	I_{OH} = −10 μA
		V_{CC} − 0.7		V	I_{OH} = −30 μA
		V_{CC} − 1.5		V	I_{OH} = −60 μA
V_{OH1}	Output High Voltage (Port 0 in ext bus mode, ALE, $\overline{PSEN}$)	V_{CC} − 0.3		V	I_{OH} = −200 μA
		V_{CC} − 0.7		V	I_{OH} = −3.2 mA(4)
		V_{CC} − 1.5		V	I_{OH} = −6.5 mA(4)
I_{IL}	Logical 0 Input Current (Ports 1,2,3,4,5,6, AFLAG, BFLAG)		−50	μA	V_{IN} = 0.45V
I_{TL}	Logical 1 to 0 Transition (Ports 1,2,3,4,5,6, AFLAG, BFLAG)		−650	μA	V_{IN} = 2V

D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5V ±20%; V_{SS} = 0V (Continued)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$, $\overline{IDS}$, $\overline{ODS}$)			±10	μA	$0 < V_{IN} < V_{CC} - 0.3V$
RRST	Reset Pulldown Resistor	50		150	kΩ	
CIO	Pin Capacitance			10	pF	Test Freq = 1 MHz T_A = 25°C
I_{CC}	Power Supply Current					
	Active Mode, 12 MHz(4)		11	22	mA	
	Idle Mode, 12 MHz(4)		1.7	5	mA	(Note 3)
	Power Down Mode		5	50	μA	

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst case (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9 V_{CC} specification when the address bits are stabilizing.
3. See Figures 6 through 9 for I_{CC} test conditions.
4. Care should be taken to assure that the total power dissipation is held within the package limits.



I_{CC} MAX at other frequencies is given by:
Active Mode
I_{CC} MAX = 1.6 x FREQ + 2.6
Idle Mode
I_{CC} MAX = 0.36 x FREQ + 1.2
Where FREQ is in MHz, I_{CC} MAX is given in mA.

Figure 6. I_{CC} vs Frequency. Valid only within frequency specifications of the device under test.

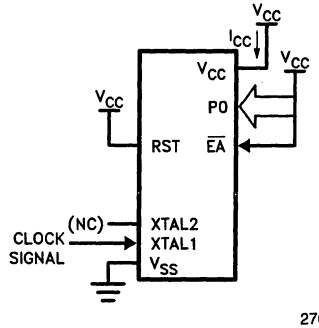


Figure 7. I_{CC} Test Condition, Active Mode. All other pins are disconnected.

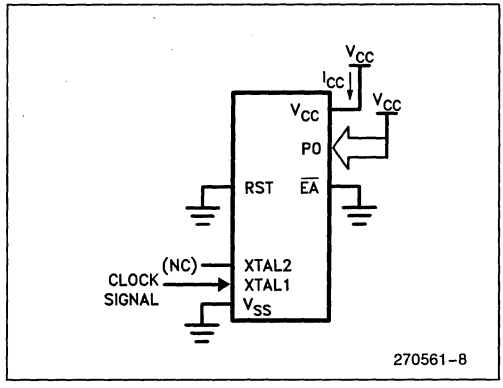


Figure 8. I_{CC} Test Condition, Idle Mode. All other pins are disconnected.

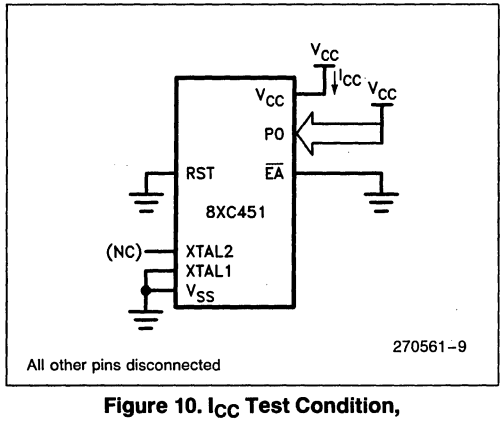


Figure 10. I_{CC} Test Condition, Power Down Mode. All other pins are disconnected.

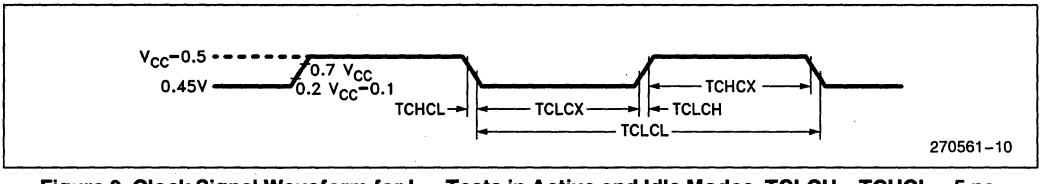


Figure 9. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.

Explanation of the AC Symbols

Each timing symbol has 5 characters. The first character is always a "T" (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address
C: Clock
D: Input data
H: Logic level HIGH
I: Instruction (program memory contents)
L: Logic level LOW, or ALE
P: $\overline{\text{PSEN}}$
Q: Output data
R: $\overline{\text{RD}}$ signal
T: Time
V: Valid
W: $\overline{\text{WR}}$ signal
X: No longer a valid logic level
Z: Float

Example:

TAVLL = Time for Address Valid to ALE Low.
TLLPL = Time for ALE Low to $\overline{\text{PSEN}}$ Low.

A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±20%, V_{SS} = 0V, load capacitance for Port 0, ALE, and $\overline{\text{PSEN}}$ = 100 pF, load capacitance for all other outputs = 80 pF.

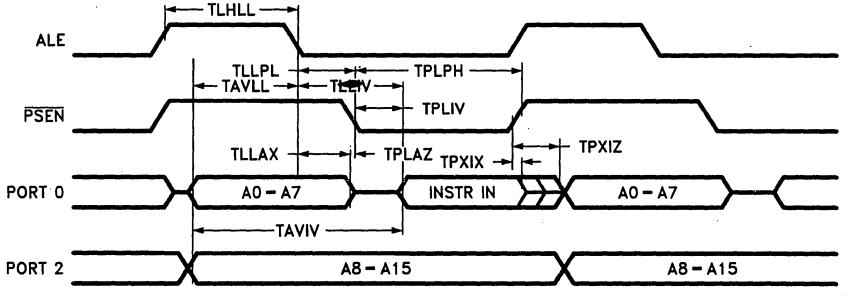
Symbol	Parameter	12 MHz Osc		Variable Osc		Unit
		Min	Max	Min	Max	
1/TCLCL	Oscillator Freq. 8XC451 8XC451-1 8XC451-2			3.5 3.5 0.5	12 16 12	MHz
TLHLL	ALE Pulse Width	127		2TCLCL − 40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL − 40		ns
TLLAX	Address Hold after ALE Low	48		TCLCL − 35		ns
TLLIV	ALE Low to Valid Instr In		234		4TCLCL − 100	ns
TLLPL	ALE Low to $\overline{\text{PSEN}}$ Low	43		TCLCL − 40		ns
TPLPH	$\overline{\text{PSEN}}$ Pulse Width	205		3TCLCL − 45		ns
TPLIV	$\overline{\text{PSEN}}$ Low to Valid Instr In		145		3TCLCL − 105	ns
TPXIX	Input Instr Hold after $\overline{\text{PSEN}}$	0		0		ns
TPXIZ	Input Instr Float after $\overline{\text{PSEN}}$		59		TCLCL − 25	ns
TAVIV	Address to Valid Instr In		312		5TCLCL − 105	ns
TPLAZ	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
TRLRH	$\overline{\text{RD}}$ Pulse Width	400		6TCLCL − 100		ns
TWLWH	$\overline{\text{WR}}$ Pulse Width	400		6TCLCL − 100		ns
TRLDV	$\overline{\text{RD}}$ Low to Valid Data In		252		5TCLCL − 165	ns
TRHDX	Data Hold after $\overline{\text{RD}}$	0		0		ns
TRHDZ	Data Float after $\overline{\text{RD}}$		97		2TCLCL − 70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL − 150	ns
TAVDV	Address to Valid Data In		585		9TCLCL − 165	ns
TLLWL	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	3TCLCL − 50	3TCLCL + 50	ns

A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±20%, V_{SS} = 0V, load capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF, load capacitance for all other outputs = 80 pF. (Continued)

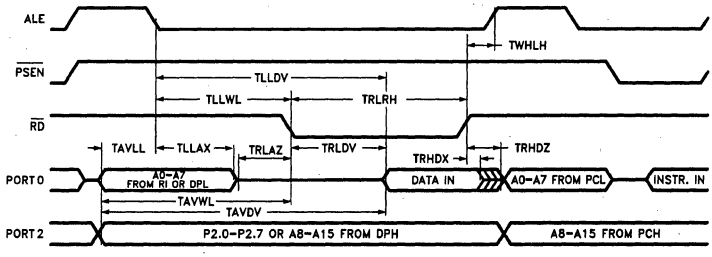
Symbol	Parameter	12 MHz Clock		Variable Clock		Unit
		Min	Max	Min	Max	
TAVWL	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL − 130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	23		TCLCL − 60		ns
TWHQX	Data Hold after $\overline{WR}$	33		TCLCL − 50		ns
TQVWH	Data Valid to $\overline{WR}$ High	433		7TCLCL − 70		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	102	TCLCL − 40	TCLCL + 40	ns
PORT 6 INPUT (Input Rise and Fall Times 5 ns)						
TILIH	$\overline{IDS}$ Width	270		3TCLCL + 20		ns
TDVIH	Data Setup to $\overline{IDS}$ High	0		0		ns
TIHDX	Data Hold after $\overline{IDS}$	30		30		ns
TFLIL	$\overline{PE}$ to $\overline{IDS}$	25		25		ns
TIVFV	$\overline{IDS}$ to BFLAG (IBF) Delay		130		130	ns
PORT 6 OUTPUT						
TOLOH	$\overline{ODS}$ Width	270		3TCLCL + 20		ns
TFVDV	SEL to Data Out Delay		85		85	ns
TOLDV	$\overline{ODS}$ to Data Out Delay		80		80	ns
TOHDZ	$\overline{ODS}$ to Data Float Delay		35		35	ns
TOVFV	$\overline{ODS}$ to AFLAG (OBF) Delay		100		100	ns
TFLDV	$\overline{PE}$ to Data Out Delay		120		120	ns
TOHFH	$\overline{ODS}$ High to AFLAG (SEL) Delay	100		100		ns

EXTERNAL PROGRAM MEMORY READ CYCLE



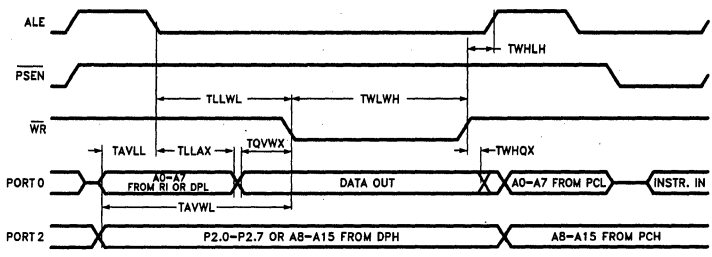
270561-11

EXTERNAL DATA MEMORY READ CYCLE



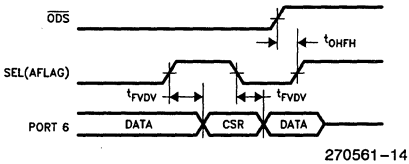
270561-12

EXTERNAL DATA MEMORY WRITE CYCLE



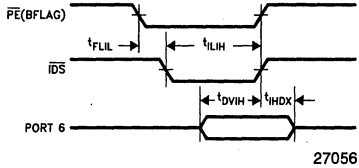
270561-13

PORT 6 SELECT MODE WAVEFORMS



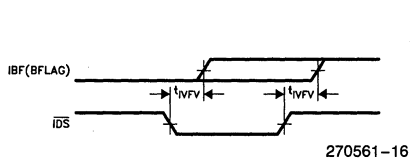
270561-14

PORT 6 INPUT WAVEFORMS



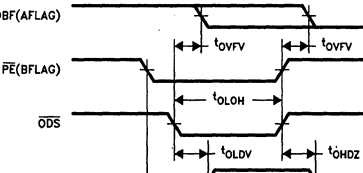
270561-15

IBF FLAG OUTPUT WAVEFORMS



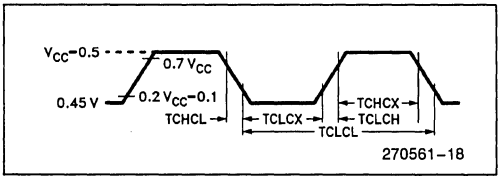
270561-16

PORT 6 OUTPUT WAVEFORMS



270561-17

EXTERNAL CLOCK DRIVE WAVEFORM



EXTERNAL CLOCK DRIVE

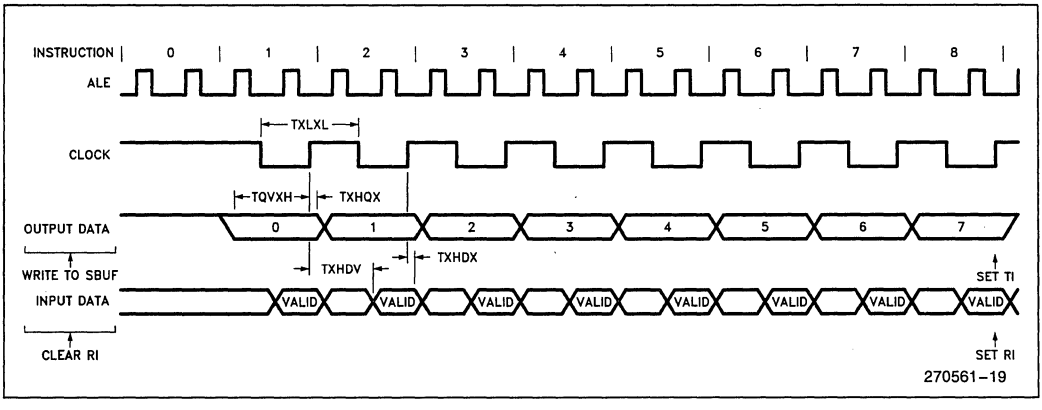
Symbol	Parameter	Min	Max	Unit
1/TCLCL	Oscillator Frequency			
	80C451	3.5	12	MHz
	80C451-1	3.5	16	MHz
	80C451-2	0.5	12	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

SERIAL TIMING—SHIFT REGISTER MODE

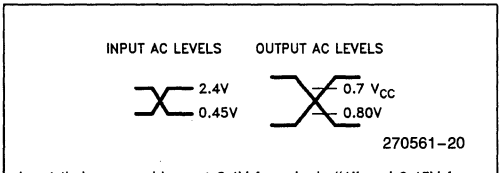
Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±20%; V_{SS} = 0V; load capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc.		Variable Oscillator		Unit
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	1.0		12TCLCL		μs
TQVXH	Output Data Setup to Clock Rising Edge	700		10TCLCL − 133		ns
TXHQX	Output Data Hold after Clock Rising Edge	50		2TCLCL − 117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		700		10TCLCL − 133	ns

SHIFT REGISTER MODE TIMING WAVEFORMS

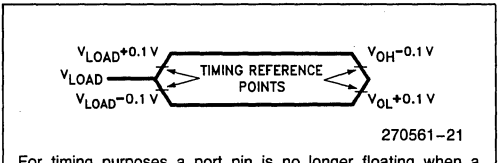


A.C. TESTING INPUT, OUTPUT WAVEFORMS



Input timings are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Output timing measurements are made at 0.7 V_{CC} for a logic "1" and 0.8V for a logic "0".

FLOAT WAVEFORM



For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -001 version of the 83C451/80C451 data sheet:

1. Package table was added.
2. Typical values for ICC were added.
3. Note 5 was added to explain the test conditions for typical values.
4. Timing spec TQVWH was added.
5. Data sheet revision summary was added.

UPI-452 CHMOS Programmable I/O Processor

13

UPI-452
CHMOS PROGRAMMABLE I/O PROCESSOR

83C452 - 8K × 8 Mask Programmable Internal ROM

80C452 - External ROM/EPROM

- 83C452/80C452:3.5 to 14 MHz Clock Rate
- Software Compatible with the MCS-51 Family
- 128-Byte Bi-Directional FIFO Slave Interface
- Two DMA Channels
- 256 × 8-Bit Internal RAM
- 34 Additional Special Function Registers
- 40 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Boolean Processor
- Bit Addressable RAM
- 8 Interrupt Sources
- Programmable Full Duplex Serial Channel
- 64K Program Memory Space
- 64K Data Memory Space
- 68-Pin PGA and PLCC
  (See Packaging Spec., Order: #231369)

The Intel UPI-452 (Universal Peripheral Interface) is a 68 pin CHMOS Slave I/O Processor with a sophisticated bi-directional FIFO buffer interface on the slave bus and a two channel DMA processor on-chip. The UPI-452 is the newest member of Intel's UPI family of products. It is a general-purpose slave I/O Processor that allows the designer to grow a customized interface solution.

The UPI-452 contains a complete 80C51 with twice the on-chip data and program memory. The sophisticated slave FIFO module acts as a buffer between the UPI-452 internal CPU and the external host CPU. To both the external host and the internal CPU, the FIFO module looks like a bi-directional bottomless buffer that can both read and write data. The FIFO manages the transfer of data independent of the UPI-452 core CPU and generates an interrupt or DMA request to either CPU, host or internal, as a FIFO service request.

The FIFO consists of two channels:the Input FIFO and the Output FIFO. The division of the FIFO module array, 128 bytes, between Input channel and Output channel is programmable by the user. Each FIFO byte has an additional logical ninth bit to distinguish between a data byte and a Data Stream Command byte. Additionally, Immediate Commands allow direct, interrupt driven, bi-directional communication between the UPI-452 internal CPU and external host CPU, bypassing the FIFO.

The on-chip DMA processor allows high speed data transfers from one writeable memory space to another. As many as 64K bytes can be transferred in a single DMA operation. Three distinct memory spaces may be used in DMA operations; Internal Data Memory, External Data Memory, and the Special Function Registers (including the FIFO IN, FIFO OUT, and Serial Channel Special Functions Registers).

October 1988
Order Number: 231428-005

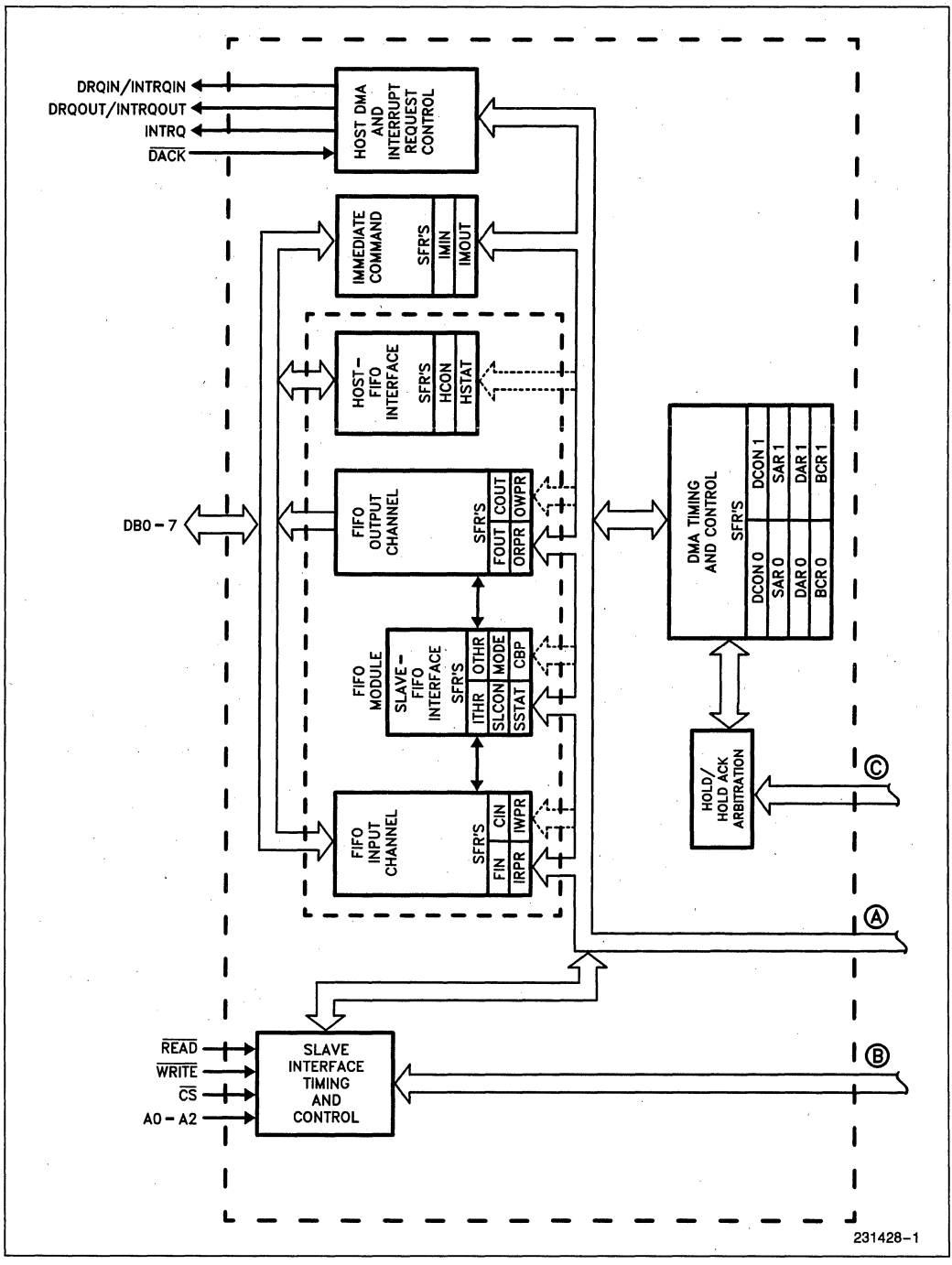


Figure 1. Architectural Block Diagram

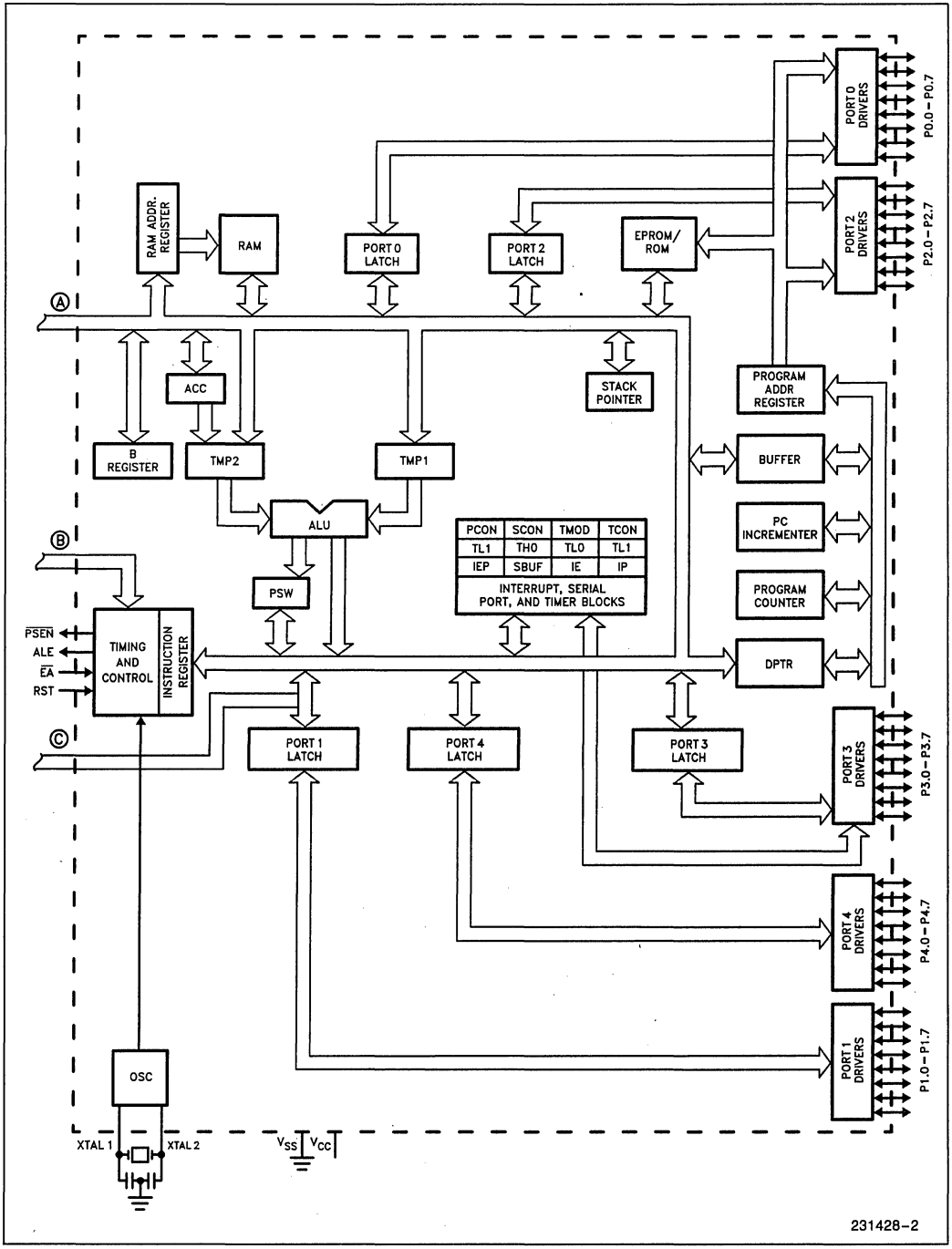


Figure 1. Architectural Block Diagram (Continued)

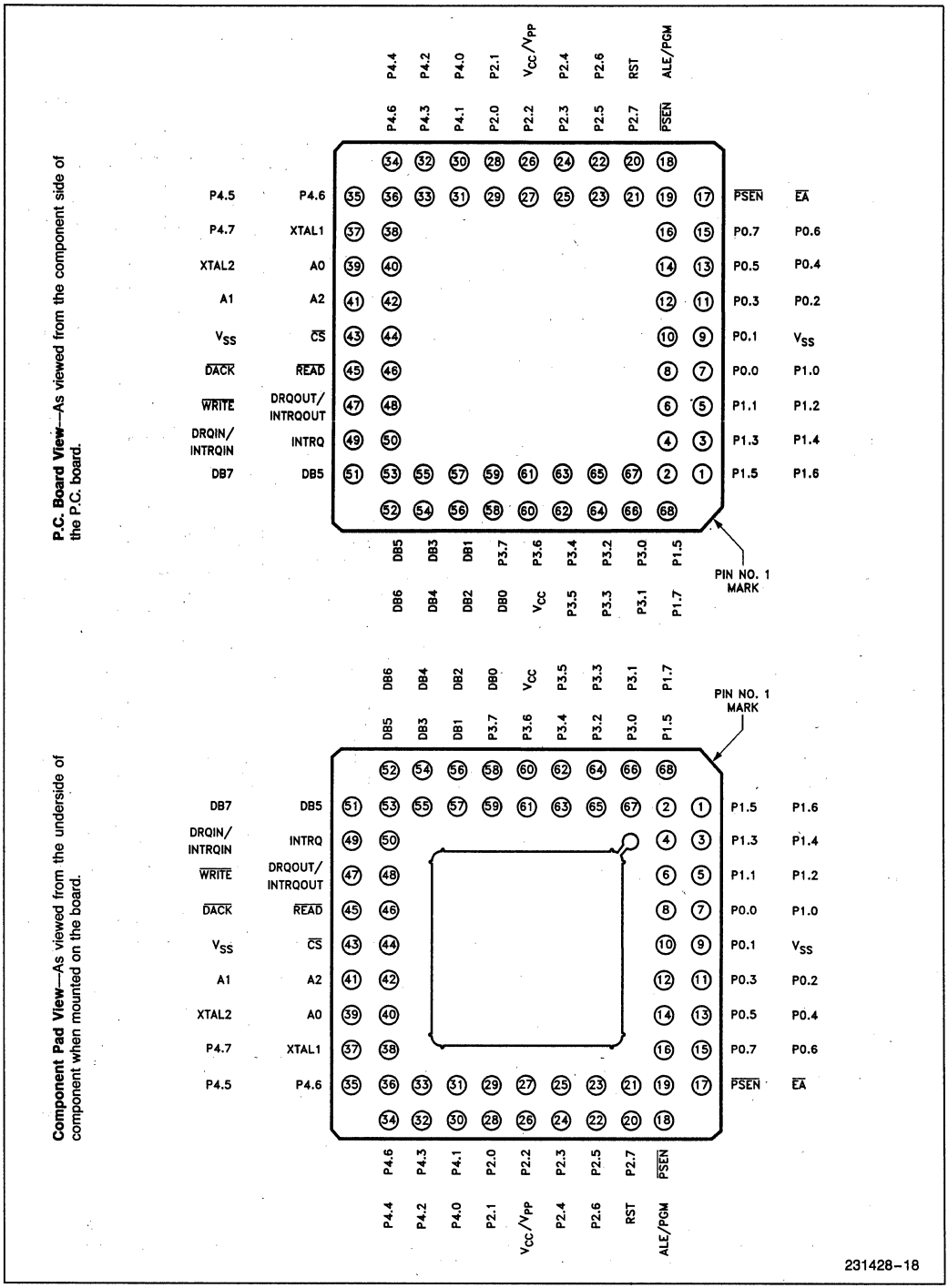
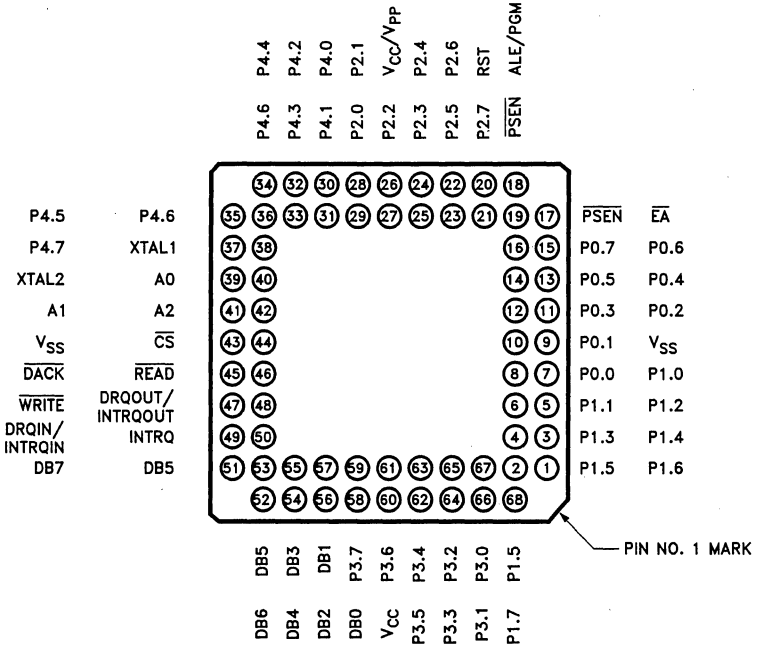
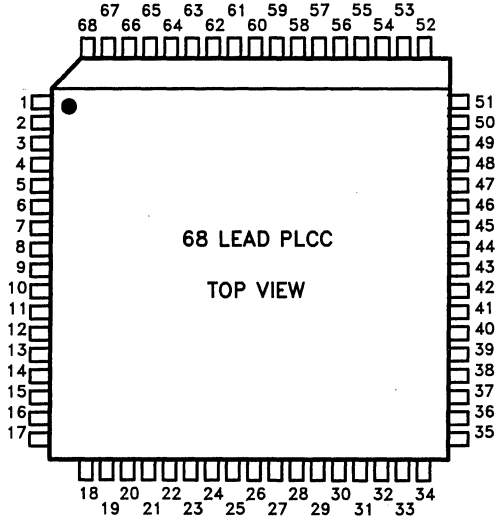


Figure 2. UPI-452 68-Pin PGA Pinout Diagram

P.C. Board View—As Viewed from the Component Side of the P.C. Board (Underside of Socket)





231428-32

Figure 2A. UPI 452 68-Pin PLCC Pinout Diagram

UPI MICROCONTROLLER FAMILY

The UPI-452 joins the current members of the UPI microcontroller family. UPI's are derivatives of the MCS™ family of microcontrollers. Because of their on-chip system bus interface, UPI's are designed to be system bus "slaves", while their microcontroller counterparts are intended as system bus "masters".

These UPI Microcontrollers are fully supported by Intel's development tools (ICE, ASM and PLM).

Packaging

The 80C452/83C452 is available in either a 68-pin PGA (Pin Grid Array) or 68-pin PLCC package.

UPI Family (Slave Configuration)	MCS Family (Master Configuration)	Speed	RAM (Bytes)	ROM (Bytes)
80C452	80C51	12 MHz	256	—
83C452	80C51	12 MHz	256	8K
80C452-1	80C51	14 MHz	256	—
83C452-1	80C51	14 MHz	256	8K

UPI-452 PIN DESCRIPTIONS

Symbol	Pin #	Type	Name and Function
V_{SS}	9/43	I	Circuit Ground.
V_{CC}	60	I	+5V power supply during normal and idle mode operation. It is also the standby power pin for power down mode.
XTAL1	38	I	Input to the oscillator's high gain amplifier. A crystal or external source can be used.
XTAL2	39	O	Output from the high gain amplifier.
Port 0 (AD0–AD7) P0.0 .1 .2 .3 .4 .5 .6 P0.7	 8 10 11 12 13 14 15 16	I/O	Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 can sink eight LS TTL inputs. It is also the multiplexed low-order address and data local expansion bus during accesses to external memory.

UPI-452 PIN DESCRIPTIONS (Continued)

Symbol	Pin #	Type	Name and Function
Port 1 (A0–A7) (HLD, HLDA) P1.0 .1 .2 .3 .4 .5 .6 P1.7	 7 6 5 4 3 2 1 68	I/O	Port 1 is an 8-bit quasi-bi-directional I/O port. Port 1 can sink four LS TTL inputs. The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise, the port pin is stuck at 0. Pins P1.5 and P1.6 are multiplexed with $\overline{HLD}$ and $\overline{HLDA}$ respectively whose functions are defined as below: Port Pin — Alternate Function P1.5 — $\overline{HLD}$ —Local bus hold input/output signal P1.6 — $\overline{HLDA}$ —Local bus hold acknowledge input
Port 2 (A8–A15) P2.0 .1 .2 .3 .4 .5 .6 .7	 29 28 27 25 24 23 22 21	I/O	Port 2 is an 8-bit quasi-bi-directional I/O port. It also emits the high-order 8 bits of address when accessing local expansion bus external memory. Port 2 can sink four LS TTL inputs.
Port 3 P3.0 .1 .2 .3 .4 .5 .6 P3.7	 67 66 65 64 63 62 61 59	I/O	Port 3 is an 8-bit quasi-bi-directional I/O port. It is also multiplexed with the interrupt, timer, local serial channel, RD/ and WR/ functions that are used by various options. The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise, the port pin is stuck at 0. Port 3 can sink four LS TTL inputs. The alternate functions assigned to the pins of Port 3 are as follows: Port Pin — Alternate Function P3.0 — RxD — Serial input port P3.1 — TxD — Serial output port P3.2 — INT0 — Interrupt 0 Input P3.3 — INT1 — Interrupt 1 Input P3.4 — T0 — Input to counter 0 P3.5 — T1 — Input to counter 1 P3.6 — WR/ — The write control signal latches the data from Port 0 outputs into the External Data Memory on the local bus. P3.7 — RD/ — The read control signal latches the data from Port 0 outputs on the local bus.

UPI-452 PIN DESCRIPTIONS (Continued)

Symbol	Pin #	Type	Name and Function
Port 4 P4.0 .1 .2 .3 .4 .5 .6 .7	 30 32 33 34 35 36 37	I/O	Port 4 is an 8-bit quasi-bi-directional I/O port. Port 4 can sink/source four TTL inputs.
RST	20	I	A high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown resistor permits Power-on reset using only a capacitor connected to V_{CC}. This pin does not receive the power down voltage as is the case for HMOS MCS-51 family members. This function has been transferred to the V_{CC} pin.
ALE	18	O	Provides Address Latch Enable output used for latching the address into external memory during normal operation. ALE can sink/source eight LS TTL inputs.
$\overline{PSEN}$	19	O	The Program Store Enable output is a control signal that enables the external Program Memory to the bus during normal fetch operation. $\overline{PSEN}$ can sink/source eight LS TTL inputs.
$\overline{EA}$	17	I	When held at TTL high level, the UPI-452 executes instructions from the internal ROM when the PC is less than 8192 (8K, 2000H). When held at a TTL low level, the UPI-452 fetches all instructions from external Program Memory.
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7	58 57 56 55 54 53 52 51	I/O	Host Bus Interface is an 8-bit bi-directional bus. It is used to transfer data and commands between the UPI-452 and the host processor. This bus can sink/source eight LS TTL inputs.
$\overline{CS}$	44	I	This pin is the Chip Select of the UPI-452.
A0 A1 A2	40 41 42	I	These three address lines are used to interface with the host system. They define the UPI-452 operations. The interface is compatible with the Intel microprocessors and the MULTIBUS.
$\overline{READ}$	46	I	This pin is the read strobe from the host CPU. Activating this pin causes the UPI-452 to place the contents of the Output FIFO (either a command or data) or the Host Status/Control Special Function Register on the Slave Data Bus.
$\overline{WRITE}$	47	I	This pin is the write strobe from the host. Activating this pin will cause the value on the Slave Data Bus to be written into the register specified by A0–A2.
DRQIN/ INTRQIN	49	O	This pin requests an input transfer from the host system whenever the Input Channel requires data.
DRQOUT/ INTRQOUT	48	O	This output pin requests an output transfer whenever the Output Channel requires service. If the external host to UPI-452 DMA is enabled, and a Data Stream Command is at the Output FIFO, DRQOUT is deactivated and INTRQ is activated (see 'GENERAL PURPOSE DMA CHANNELS' section).

UPI-452 PIN DESCRIPTIONS (Continued)

Symbol	Pin #	Type	Name and Function
INTRQ	50	O	This output pin is used to interrupt the host processor when an Immediate Command Out or an error condition is encountered. It is also used to interrupt the host processor when the FIFO requests service if the DMA is disabled and INTRQIN and INTRQOUT are not used.
$\overline{DACK}$	45	I	This pin is the DMA acknowledge for the host bus interface Input and Output Channels. When activated, a write command will cause the data on the Slave Data Bus to be written as data to the Input Channel (to the Input FIFO). A read command will cause the Output Channel to output data (from the Output FIFO) on to the Slave Data Bus. This pin should be driven high (+5V) in systems which do not have a DMA controller (see Address Decoding).
V_{CC}	26	I	+5V power supply during operation.

ARCHITECTURAL OVERVIEW

Introduction

The UPI-452 slave microcontroller incorporates an 80C51 with double the program and data memory, a slave interface which allows it to be connected directly to the host system bus as a peripheral, a FIFO buffer module, a two channel DMA processor, and a fifth I/O port (Figure 3). The UPI-452 retains all of the 80C51 architecture, and is fully compatible with the MCS-51 instruction set.

The Special Function Register (SFR) interface concept introduced in the MCS-51 family of microcontrollers has been expanded in the UPI-452. To the 20 Special Function Registers of the MCS-51, the UPI-452 adds 34 more. These additional Special Function Registers, like those of the MCS-51, provide access to the UPI-452 functional elements including the FIFO, DMA and added interrupt capabilities. Several of the 80C51 core Special Function Registers have also been expanded to support added features of the UPI-452.

This data sheet describes the unique features of the UPI-452. Refer to the 80C51 data sheet for a description of the UPI-452's core CPU functional blocks including;

— Timers/Counters
— I/O Ports
— Interrupt timing and control (other than FIFO and DMA interrupts)
— Serial Channel
— Local Expansion Bus
— Program/Data Memory structure
— Power-Saving Modes of Operation
— CHMOS Features
— Instruction Set

Figure 3 contains a conceptual block diagram of the UPI-452. Figure 4 provides a functional block diagram.

FIFO Buffer Interface

A unique feature of the UPI-452 is the incorporation of a 128 byte FIFO array at the host-slave interface. The FIFO allows asynchronous bi-directional transfers between the host CPU and the internal CPU.

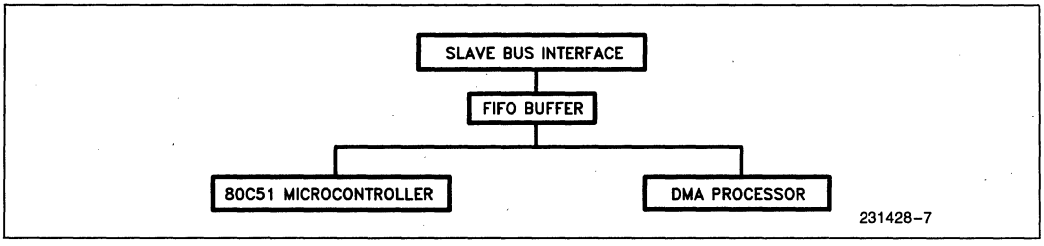


Figure 3. UPI-452 Conceptual Block Diagram

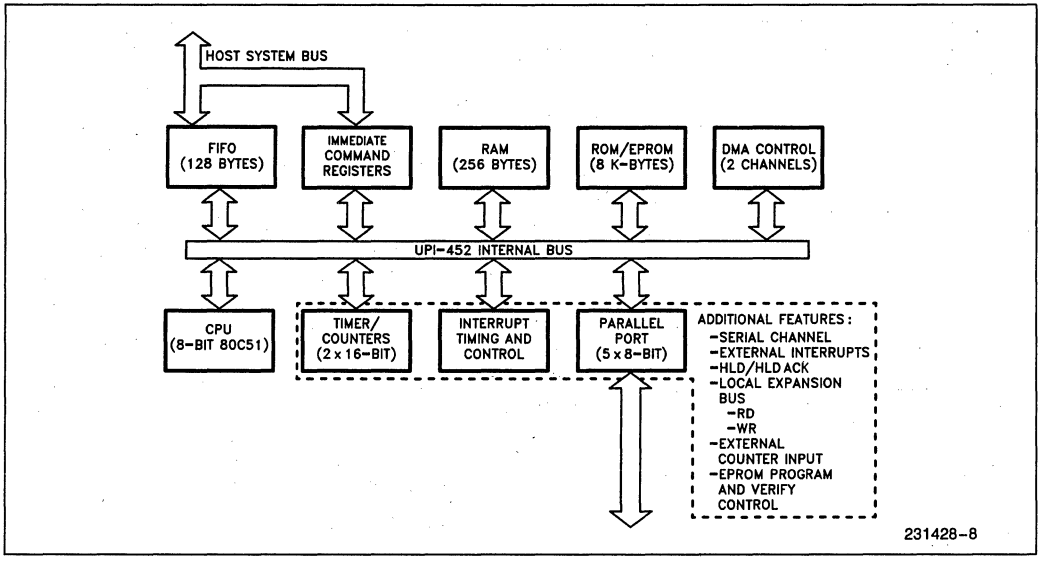


Figure 4. UPI-452 Functional Block Diagram

The division of the 128 bytes between Input and Output channels is user programmable allowing maximum flexibility. If the entire 128 byte FIFO is allocated to the Input channel, a high performance Host can transfer up to 128 bytes at one time, then dedicate its resources to other functions while the internal CPU processes the data in the FIFO. Various handshake signals allow the external Host to operate independently and without frequent monitoring of the UPI-452 internal CPU. The FIFO Buffer insures that the slave processor receives data in the same order that it was sent by the host without the need to keep track of addresses. Three slave bus interface handshake methods are supported by the UPI-452: DMA, Interrupt and Polled.

The FIFO is nine bits wide. The ninth bit acts as a command/data flag. Commands written to the FIFO by either the host or internal CPU are called Data Stream Commands or DSCs. DSCs are written to the input FIFO by the Host via a unique external address. DSCs are written to the output FIFO by the internal CPU via the COMMAND OUT Special Function Register (SFR). When encountered by the host or internal CPU a Data Stream Command can be used as an address vector to user defined service routines. DSCs provide synchronization of data and commands between the Host and internal CPU.

FIFO PROGRAMMABLE FEATURES

Size of Input/Output Channels

The 128 bytes of FIFO space can be allocated between the Input and Output channels via the Channel Boundary Pointer (CBP) SFR. This register contains the number of address locations assigned to the Input channel. The remaining address locations are automatically assigned to the Output FIFO. The CBP SFR can only be programmed by the internal CPU during FIFO DMA Freeze Mode (See FIFO-External Host Interface FIFO DMA Freeze Mode description). The CBP is initialized to 40H (64 bytes) upon reset.

The number in the Channel Boundary Pointer SFR is actually the first address location of the Output FIFO. Writing to the CBP SFR reassigns the Input and Output FIFO address space. Whenever the CBP is written, the Input FIFO pointers are reset to zero and the Output FIFO pointers are set to the value in the CBP SFR.

All of the FIFO space may be assigned to one channel. In such a situation the other channel's data path consists of a single SFR (FIFO IN/COMMAND IN or FIFO OUT/COMMAND OUT SFR) location.

CBP Register	Input FIFO Size	Output FIFO Size
0	1	128
1	1	128
2	2	126
3	3	125
4	4	124
•	•	•
7B	123	5
7C	124	4
7D	125	3
7E	128	1
7F	128	1

FIFO Read/Write Pointers

These normally operate in auto-increment (and auto-rollover) mode, but can be reassigned by the internal CPU during FIFO DMA Freeze Mode (See FIFO-External Host Interface FIFO DMA Freeze Mode description).

Threshold Register

The Input FIFO Threshold SFR contains the number of empty bytes that must be available in the Input FIFO to generate a Host interrupt. The Output FIFO Threshold SFR contains the number of bytes, data and/or DSC(s), that must be in the FIFO before an interrupt is generated. The Threshold feature prevents the Host from being interrupted each time the FIFO needs to load or unload one byte of data. The thresholds, therefore, allow the FIFO's operation to be adjusted to the speed of the Host, optimizing the overall interface performance.

Immediate Commands

The UPI-452 provides, in addition to data and DSCs, a third direct means of communication between the external Host and internal CPU called Immediate Commands. As the name implies, an Immediate Command is available to the receiving CPU immediately, via an interrupt, without being entered into the FIFO as are Data Stream Commands. Like Data Stream Commands, Immediate Commands are written either via a unique external address by the host CPU, or via dedicated SFR by the internal CPU.

The DSC and/or Immediate Command interface may be defined as either Interrupt or Polled under user program control via the Interrupt Enable (IE), Slave Control Register (SLCON), and Interrupt Enable Priority (IEP) Special Function Registers, for the internal CPU and via the Host Control SFR for the external Host CPU.

DMA

The UPI-452 contains a two channel internal DMA controller which allows transfer of data between any of the three writeable memory spaces: Internal Data Memory, External Load Expansion Bus Data Memory and the Special Function Register array. The Special Function Register array appears as a set of unique dedicated memory addresses which may be used as either the source or destination address of a DMA transfer. Each DMA channel is independently programmable via dedicated Special Function Registers for mode, source and destination addresses, and byte count to be transferred. Each DMA channel has four programmable modes:

— Alternate Cycle Mode

— Burst Mode

— FIFO or Serial Channel Demand Mode

— External Demand Mode

A complete description of each mode and DMA operation may be found in the section titled "General Purpose DMA Channels".

FIFO/SLAVE INTERFACE FUNCTIONAL DESCRIPTION

Overview

The FIFO is a 128 Byte RAM array with recirculating pointers to manage the read and write accesses. The FIFO consists of an Input and an Output channel. Access cycles to the FIFO by the internal CPU and external Host are interleaved and appear to be occurring concurrently to both the internal CPU and external Host. Interleaving access cycles ensures efficient use of this shared resource. The internal CPU accesses the FIFO in the same way it would access any of the Special Function Registers e.g., direct and register indirect addressing as well as arithmetic and logical instructions.

Input FIFO Channel

The Input FIFO Channel provides for data transfer from the external Host to the internal CPU (Figure 5). The registers associated with the Input Channel during normal operation are listed in Table 1*.

Table 1. Input FIFO Channel Registers*

	Register Name	Description
1)	Input Buffer Latch	Host CPU Write only
2)	FIFO IN SFR	Internal CPU Read only
3)	COMMAND IN SFR	Internal CPU Read only
4)	Input FIFO Read Pointer SFR	Internal CPU Read only
5)	Input FIFO Write Pointer SFR	Internal CPU Read only
6)	Input FIFO Threshold SFR	Internal CPU Read only

*See "FIFO-EXTERNAL HOST INTERFACE FIFO DMA FREEZE MODE" section for FIFO DMA Freeze Mode SFR characteristics description.

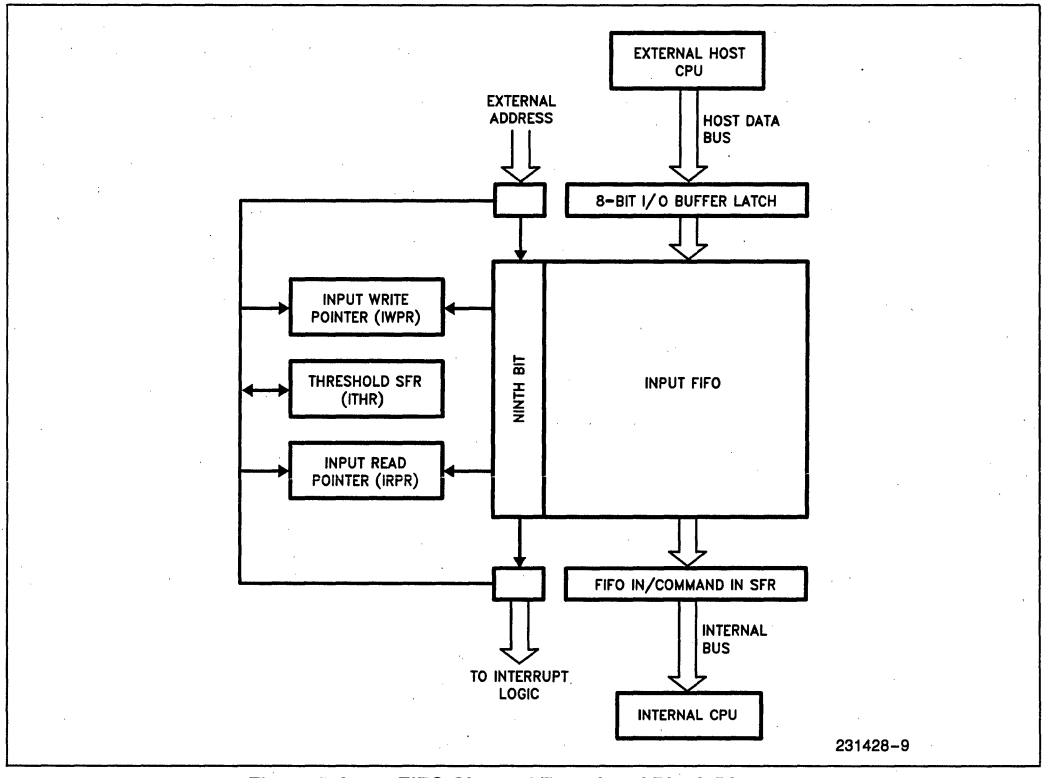


Figure 5. Input FIFO Channel Functional Block Diagram

The host CPU writes data and Data Stream Commands into the Input Buffer Latch on the rising edge of the external WR signal. External addressing determines whether the byte is a data byte or Data Stream Command and the FIFO logic sets the ninth bit of the FIFO accordingly as the byte is moved from the Input Buffer Latch into the FIFO. A "1" in the ninth bit indicates that the incoming byte is a Data Stream Command. The internal CPU reads data bytes via the FIFO IN SFR, and Data Stream Commands via the COMMAND IN SFR.

A Data Stream Command will generate an interrupt to the internal CPU prior to being read and after completion of the previous operation. The DSC can then be read via the COMMAND IN SFR. Data can only be read via the FIFO IN SFR and Data Stream Commands via the COMMAND IN SFR. Attempting to read Data Stream Commands as data by addressing the FIFO IN SFR will result in "0FFH" being read, and the Input FIFO Read Pointer will remain intact. (This prevents accidental misreading of Data Stream Commands.) Attempting to read data as Data Stream Commands will have the same consequence.

The Input FIFO Channel addressing is controlled by the Input FIFO Read and Write Pointer SFRs. These SFRs are read only registers during normal operation. However, during FIFO DMA Freeze Mode (See FIFO-External Host Interface FIFO DMA Freeze Mode description), the internal CPU has write access to them. Any write to these registers in normal mode will have no effect. The Input Write Pointer SFR contains the address location to which data/commands are written from the Input Buffer Latch. The write pointer is automatically incremented after each write and is reset to zero if equal to the CBP, as the Input FIFO operates as a circular buffer.

If a write is performed on an empty FIFO, the first byte is also written into the FIFO IN or COMMAND IN SFR. If the Host continues writing while the Input FIFO is full, an external interrupt, if enabled, is sent to the host to signal the overrun condition. The writes are ignored by the FIFO control logic. Similarly, an internal CPU read of an empty FIFO will cause an underrun error interrupt to be generated to the internal CPU and a value of "0FFH" will be read by the internal CPU.

The Read Pointer SFR holds the address of the next byte to be read from the Input FIFO. An Input FIFO read operation post-increments the Input Read Pointer SFR and loads a new data byte into the FIFO IN SFR or a Data Stream Command into the COMMAND IN SFR at the end of the read cycle.

An Input FIFO Request for Service (via DMA, Interrupt or a flag) is generated to the Host whenever more data can be written into the Input FIFO. For efficient utilization of the Host, a "threshold" value can be programmed into the Input FIFO Threshold SFR. The range of values of the Input FIFO Threshold SFR can be from 0 to (CBP-3). The Request for Service Interrupt is generated only after the Input FIFO has room to accommodate a threshold number of bytes or more. The threshold is equal to the total number of bytes assigned to the Input FIFO (CBP) minus the number of bytes programmed in the Input FIFO Threshold SFR. With this feature the Host is assured that it can write at least a threshold number of bytes to the Input FIFO channel without worrying about an overrun condition. Once the Request for Service is generated it remains active until the Input FIFO becomes full.

Output FIFO Channel

The Output FIFO Channel provides data transfer from the UPI-452 internal CPU to the external Host (Figure 6).

The registers associated with the Output Channel during normal operation are listed in Table 2*.

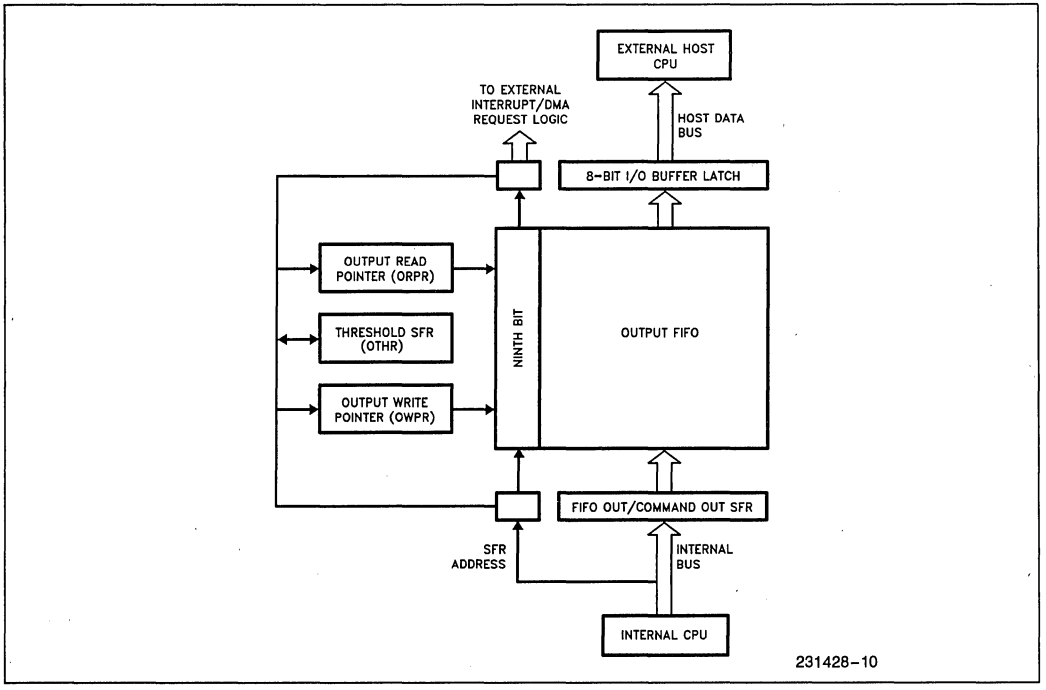


Figure 6. Output FIFO Channel Functional Block Diagram

Table 2. Output FIFO Channel Registers

	Register Name	Description
1)	Output Buffer Latch	Host CPU Read only
2)	FIFO OUT SFR	Internal CPU Read and Write
3)	COMMAND OUT SFR	Internal CPU Read and Write
4)	Output FIFO Read Pointer SFR	Internal CPU Read only
5)	Output FIFO Write Pointer SFR	Internal CPU Read only
6)	Output FIFO Threshold SFR	Internal CPU Read only

*See "FIFO-EXTERNAL HOST INTERFACE FIFO DMA FREEZE MODE" section for FIFO DMA Freeze Mode register characteristics description.

The UPI-452 internal CPU transfers data to the Output FIFO via the FIFO OUT SFR and commands via the COMMAND OUT SFR. If the byte is written to the COMMAND OUT SFR, the ninth bit is automatically set (=1) to indicate a Data Stream Command. If the byte is written to the FIFO OUT SFR the ninth bit is cleared (=0). Thus the FIFO OUT and COMMAND OUT SFRs are the same but the address determines whether the byte entered in the FIFO is a DSC or data byte.

The Output FIFO preloads a byte into the Output Buffer Latch. When the Host issues a RD/ signal, the data is immediately read from the Output Buffer Latch. The next data byte is then loaded into the Output Buffer Latch, a flag is set and an interrupt, if enabled, is generated if the byte is a DSC (ninth bit is set). The operation is carefully timed such that an interrupt can be generated in time for it to be recognized by the Host before its next read instruction. Internal CPU write and external Host read operations are interleaved at the FIFO so that they appear to be occurring concurrently.

The Output FIFO read and write pointer operation is the same as for the Input Channel. Writing to the FIFO OUT or COMMAND OUT SFRs will increment the Output Write Pointer SFR but reading from it will leave the write pointer unchanged. A rollover of the Output FIFO Write Pointer causes the pointer to be reset to the value in the Channel Boundary Pointer (CBP) SFR.

If the external host attempts to read a Data Stream Command as a data byte it will result in invalid data (0FFH) being read. The DSC is not lost because the invalid read does not increment the pointer. Similarly attempting to read a data byte as a Data Stream Command has the same result.

A Request for Service is generated to the external Host under the following two conditions:

1.) Whenever the internal CPU has written a threshold number of bytes or more into the Output FIFO (threshold = (OTHR) + 1). The threshold number should be chosen such that the bus latency time for the external Host does not result in a FIFO overrun error condition on the internal CPU side. The threshold limit should be large enough to make a bus request by the UPI-452 to the external host CPU worthwhile. Once a request for service is generated, the request remains active until the Output FIFO becomes empty. The range of values of the FIFO Output Threshold (OTHR) SFR is from 2 to {(80H-CBP)-1}. The threshold number can be programmed via the OTHR SFR.

2.) The second type of Request for Service is called "Flush Mode" and occurs when the internal CPU writes a Data Stream Command into the Output FIFO. Its purpose is to ensure that a data block entered into the Output FIFO, which is less than the programmed threshold, will generate a Request for Service interrupt, if enabled, and be read, or "Flushed" from the Output FIFO, by the external host CPU regardless of the status of the OTHR SFR.

Immediate Commands

Immediate Commands provide direct communication between the external Host and UPI-452. Unlike Data Stream Commands which are entered into the FIFO, the Immediate Command is available to the receiving CPU directly, bypassing the FIFO. The Immediate Command can serve as a program vector pointing into a jump table in the recipients software. Immediate Command Interrupts are generated, if enabled, and a bit in the appropriate Status Register is set when an Immediate Command is input or output. A similar bit is provided to acknowledge when an Immediate Command has been read and whether the register is available to receive another command. The bits are reset when the Immediate Commands are read. Two Special Function Registers are dedicated to the Immediate Command interface. External addressing determines whether the Host is accessing the Input FIFO or the Immediate Command IN (IMIN) SFR. The internal CPU writes Immediate Commands to the Immediate Command OUT (IMOUT) SFR.

Both processors have the ability to enable or disable Immediate Command Interrupts. By disabling the interrupt, the recipient of the Immediate Command can poll the status SFR and read the Immediate Command at its convenience. Immediate Commands should only be written when the appropriate Immediate Command SFR is empty (as indicated in the appropriate status SFR:HSTAT/SSTAT). Similarly, the Immediate Command SFR should only be read when there is data in the Register.

The flowcharts in Figure 7a and 7b illustrate the proper handshake mechanisms between the external Host and internal CPU when handling Immediate Commands.

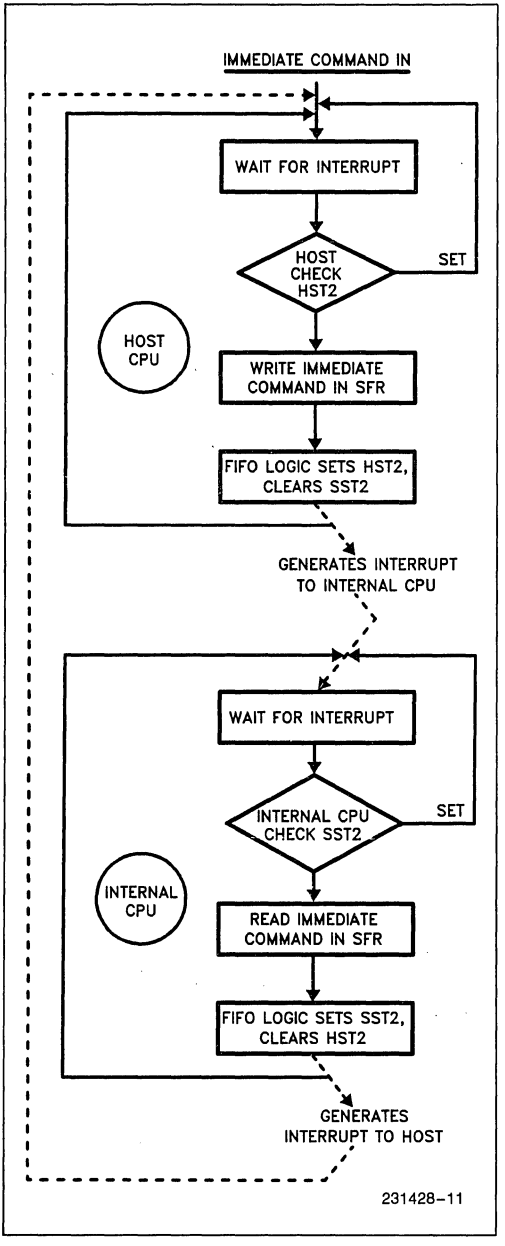


Figure 7a. Handshake Mechanisms for Handling Immediate Command IN Flowchart

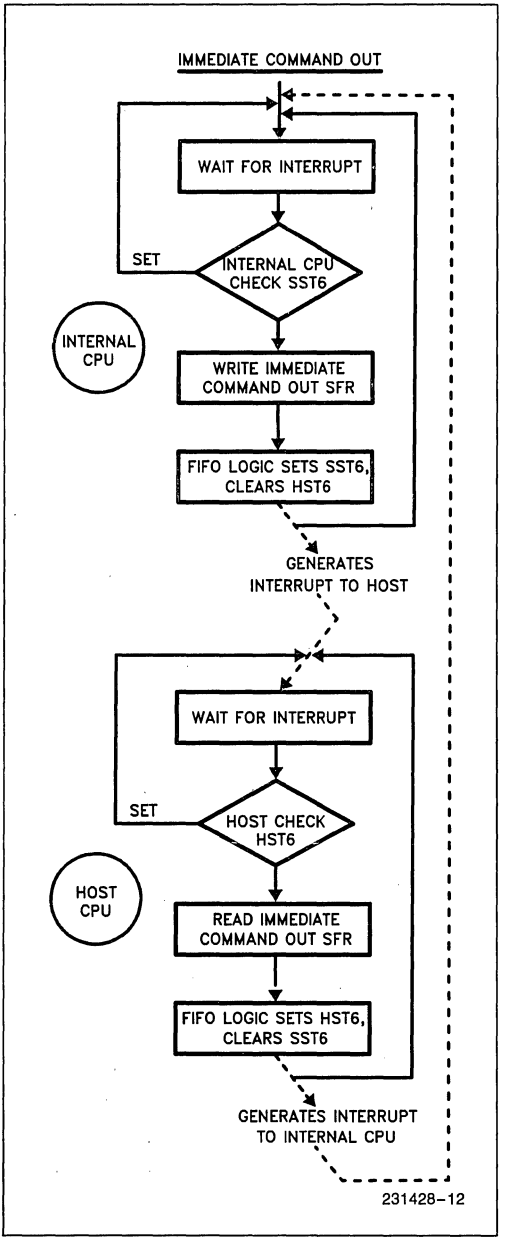


Figure 7b. Handshake Mechanisms for Handling Immediate Command OUT Flowchart

HOST & SLAVE INTERFACE SPECIAL FUNCTION REGISTERS

Slave Interface Special Function Registers

The Internal CPU interfaces with the FIFO slave module via the following registers:

1) Mode Special Function Register (MODE)

2) Slave Control Special Function Register (SLCON)

3) Slave Status Special Function Register (SSTAT)

Each register resides in the SFR Array and is accessible via all direct addressing modes except bit. Only the Slave Control Register (SLCON) is bit addressable.

1) MODE Special Function Register (MODE)

The MODE SFR provides the primary control of the external host-FIFO interface. It is included in the SFR Array so that the internal CPU can configure the external host-FIFO interface should the user decide that the UPI-452 slave initialize itself independent of the external host CPU.

The MODE SFR can be directly modified by the internal CPU through direct address instructions. It can also be indirectly modified by the external host CPU by setting up a MODE SFR service routine in the UPI-452 program memory and having the host issue a Command, either Immediate or DSC, to vector to that routine.

Symbolic Address								Physical Address
MODE	—	MD6	MD5	MD4	—	—	—	0F9H
	(MSB)							(LSB)
Status On Reset:								
	1*	0	0	0	1*	1*	1*	1*

MD7 (reserved)**

MD6 Request for Service to external CPU via;

1 = DMA (DRQIN/DRQOUT) request to external host when the Input or Output FIFO channel requests service

0 = Interrupt (INTRQIN/INTRQOUT or INTRQ) to external host when the Input or Output FIFO channel requests service or a DSC is encountered in the I/O Buffer Latch

MD5 Configure DRQIN/INTRQIN and DRQOUT/INTRQOUT to be either;

1 = Enable (Actively driven)

0 = Disable (Tri-state)

MD4 Configure INTRQ to be either;

1 = Enable (Actively driven)

0 = Disable (Tri-state)

MD3 (reserved) **

MD2 (reserved) **

MD1 (reserved) **

MD0 (reserved) **

2) Slave Control SFR (SLCON)

The Slave Control SFR is used to configure the FIFO-internal CPU interface. All interrupts are to the internal CPU.

Symbolic Address									Physical Address
SLCON	IFI	OFI	ICII	ICOI	FRZ	—	IFRS	OFRS	0E8H
	(MSB)							(LSB)	
	Status On Reset:								
	0	0	0	0	0	1*	0	0	

IFI Enable Input FIFO Interrupt (due to Underrun Error Condition, Data Stream Command or Request Service)

1 = Enable

0 = Disable

OFI Enable Output FIFO Interrupt (due to Overrun Error Condition or Request Service)

1 = Enable

0 = Disable

Note: If the DMA is configured to service a FIFO demand, then the Request for Service Interrupt is not generated.

ICII Generate Interrupt when a command is written to the Immediate Command in Register

1 = Enable

0 = Disable

ICOI Generate Interrupt when Immediate Command Out Register is Available

1 = Enable

0 = Disable

FRZ Enable FIFO DMA Freeze Mode

1 = Normal operation

0 = FIFO DMA Freeze Mode

SC2 (reserved) **

IFRS Input FIFO Channel Request for Service

1 = Request when Input FIFO not empty

0 = Request when Input FIFO full

OFRS Output FIFO Channel Request for Service

1 = Request when Output FIFO not full

0 = Channel Request when Output FIFO empty

NOTES:

*A '1' will be read from all SFR reserved locations except HCON SFR, HC0 and HC2.
**'reserved'—these locations are reserved for future use by Intel Corporation.

3) Slave Status SFR (SSTAT)

The bits in the Slave Status SFR reflect the status of the FIFO-internal CPU interface. It can be read during an internal interrupt service routine to determine the nature of the interrupt or read during a polling sequence to determine a course of action.

Symbolic Address									Physical Address
SSTAT	SST7	SST6	SST5	SST4	SST3	SST2	SST1	SST0	0E9H
	← Output FIFO Status →				← Input FIFO Status →				
	Status On Reset:								
	1	0	0	0	1	1	1	1	
	(MSB)							(LSB)	

SST7 Output FIFO Overrun Error Condition
- 1 = No Error
- 0 = Error (latched until Slave Status SFR is read)

SST6 Immediate Command Out Register Status
- 1 = Full (i.e. Host CPU has not read previous Immediate Command Out sent by internal CPU)
- 0 = Available

SST5 FIFO DMA Freeze Mode Status
- 1 = Normal Operation
- 0 = FIFO DMA Freeze Mode in Progress

SST4 Output FIFO Request for Service Flag
- 1 = Output FIFO does not request service
- 0 = Output FIFO requests service

SST3 Input FIFO Underrun Error Condition Flag
- 1 = No Underrun Error
- 0 = Underrun Error (latched until Slave Status SFR is read)

SST2 Immediate Command In SFR Status
- 1 = Empty
- 0 = Immediate Command received from host CPU

SST1 Data Stream Command/Data at Input FIFO Flag
- 1 = Data (not DSC)
- 0 = DSC (at COMMAND IN SFR)

SST0 Input FIFO Request For Service Flag
- 1 = Input FIFO Does Not Request Service
- 0 = Input FIFO Request for Service

EXTERNAL HOST INTERFACE SPECIAL FUNCTION REGISTERS

The external host CPU has direct access to the following SFRs:

1) Host Control Special Function Register
2) Host Status Special Function Register

It can also access other SFRs by commanding the internal CPU to change them accordingly via Data Stream Commands or Immediate Commands. The protocol for implementing this is entirely determined by the user.

1) Host Control SFR (HCON)

By writing to the Host Control SFR, the host can enable or disable FIFO interrupts and DMA requests and can reset the UPI-452.

Symbolic Address									Physical Address
HCON	HC7	HC6	HC5	HC4	HC3	—	HC1	—	0E7H
	(MSB)							(LSB)	
	Status On Reset:								
	0	0	0	0	0	0*	0	0*	

HC7 Enable Output FIFO Interrupt due to Underrun Error Condition, Data Stream Command or Service Request

1 = Enable

0 = Disable

HC6 Enable Input FIFO Interrupt due to Overrun Error Condition, or Service Request

1 = Enable

0 = Disable

HC5 Enable the generation of the Interrupt due to Immediate Command Out being present

1 = Enable

0 = Disable

HC4 Enable the Interrupt due to the Immediate Command In Register being Available for a new Immediate Command byte

1 = Enable

0 = Disable

HC3 Reset UPI-452

1 = Software RESET

0 = Normal Operation

HC2 (reserved) **

HC1 Select between INTRQ and INTRQIN/INTRQOUT as Request for Service interrupt signal when DMA is disabled

1 = INTRQ

0 = INTRQIN or INTRQOUT

HC0 (reserved) **

NOTES:

*A '1' will be read from all SFR reserved locations except HCON SFR, HC0 and HC2.
**'reserved'—these locations are reserved for future use by Intel Corporation.

2) Host Status SFR (HSTAT)

The Host Status SFR provides information on the FIFO-Host Interface and can be used to determine the source of an external interrupt during polling. Like the Slave Status SFR, the Host Status SFR reflects the current status of the FIFO-external host interface.

Symbolic Address									Physical Address
HSTAT	HST7	HST6	HST5	HST4	HST3	HST2	HST1	HST0	0E6H
	← Output FIFO Status →				← Input FIFO Status →				
Status On Reset:	1	1	1	1	1	1/0*	1	1	
	(MSB)							(LSB)	

HST7 Output FIFO Underrun Error Condition
1 = No Underrun Error
0 = Underrun Error (latched until Host Status Register is read)

HST6 Immediate Command Out SFR Status
1 = Empty
0 = Immediate Command Present

HST5 Data Stream Command/Data at Output FIFO Status
1 = Data (not DSC)
0 = DSC (present at Output FIFO COMMAND OUT SFR)
(Note: Only if HST4 = 0, if HST4 = 1 then undetermined)

HST4 Output FIFO Request for Service Status
1 = No Request for Service
0 = Output FIFO Request for Service due to:

a. Output FIFO containing the threshold number of bytes or more

b. Internal CPU sending a block of data terminated by a DSC (DSC Flush Mode)

HST3 Input FIFO Overrun Error Condition
1 = No Overrun Error
0 = Overrun Error (latched until Host Status Register is read)

HST2 Immediate Command In SFR Status
1 = Full (i.e. Internal CPU has not read previous Immediate Command sent by Host)
0 = Empty

* Reset value;

'1' — if read by the external Host

'0' — if read by internal CPU (reads shadow latch - see FIFO DMA Freeze Mode description)

HST1 FIFO DMA Freeze Mode Status
1 = Freeze Mode in progress.
(In Freeze Mode, the bits of the Host Status SFR are forced to a '1' initially to prevent the external Host from attempting to access the FIFO. The definition of the Host Status SFR bits during FIFO DMA Freeze Mode can be found in FIFO DMA Freeze Mode description)
0 = Normal Operation

HST0 Input FIFO Request Service Status
1 = Input FIFO does not request service
0 = Input FIFO request service due to the Input FIFO containing enough space for the host to write the threshold number of bytes or more

FIFO MODULE - EXTERNAL HOST INTERFACE

Overview

The FIFO-external Host interface supports high speed asynchronous bi-directional 8-bit data transfers. The host interface is fully compatible with Intel microprocessor local busses and with MULTIBUS. The FIFO has two specialized DMA request pins for Input and Output FIFO channel DMA requests. These are multiplexed to provide a dedicated Request for Service interrupt (DRQIN/INTRQIN, DRQOUT/INTRQOUT).

The external Host can program, under user defined protocol, thresholds into the FIFO Input and Output Threshold SFRs which determine when the FIFO Request for Service interrupt is generated to the Host CPU. The FIFO module external Host interface is configured by the internal CPU via the MODE SFR. "The external Host can enable and disable Host interface interrupts via the Host Control SFR." Data Stream Commands in the Input FIFO channel allow the Host to influence the processing of data blocks and are sent with the data flow to maintain synchronization. Data Stream Commands in the Output FIFO Channel allow the internal CPU to perform the same function, and also to set the Output FIFO Request Service status logic to the host CPU regardless of the programmed value in the Threshold SFR.

Slave Interface Address Decoding

The UPI-452 determines the desired Host function through address decoding. The lower three bits of the address as well as the READ, WRITE, Chip Select ($\overline{CS}$) and DMA Acknowledge ($\overline{DACK}$) are used for decoding. Table 3 shows the pin states and the Read or Write operations associated with each configuration.

Interrupts to the Host

The UPI-452 interrupts the external Host via the INTRQ pin. In addition, the DRQIN and DRQOUT pins can be multiplexed as interrupt request lines, INTRQIN and INTRQOUT respectively, when DMA is disabled. This provides two special FIFO "Request for Service" interrupts.

There are eight FIFO-related interrupt sources; two from The Input FIFO; three from The Output FIFO; one from the Immediate Command Out SFR; one from the Immediate Command IN SFR; and one due to FIFO DMA Freeze Mode.

INPUT FIFO: The Input FIFO interrupt is generated whenever:

a. The Input FIFO contains space for a threshold number of bytes.

Table 3. UPI-452 Address Decoding

$\overline{DACK}$	$\overline{CS}$	A2	A1	A0	Read	Write
1	1	X	X	X	No Operation	No Operation
1	0	0	0	0	Data or DMA from Output FIFO Channel	Data or DMA to Input FIFO Channel
1	0	0	0	1	Data Stream Command from Output FIFO Channel	Data Stream Command to Input FIFO Channel
1	0	0	1	0	Host Status SFR Read	Reserved
1	0	0	1	1	Host Control SFR Read	Host Control SFR Write
1	0	1	0	0	Immediate Command SFR Read	Immediate Command to SFR Write
1	0	1	1	X	Reserved	Reserved
0	X	X	X	X	DMA Data from Output FIFO Channel	DMA Data to Input FIFO Channel
1	0	1	0	1	Reserved	Reserved

NOTES:
1. Attempting to read a DSC as a data byte will result in invalid data being read. The read pointers are not incremented so that the DSC is not lost. Attempting to read a data byte as a DSC has the same result.
2. If $\overline{DACK}$ is active the UPI-452 will attempt a DMA operation when $\overline{RD}$ or $\overline{WR}$ becomes active regardless of the DMA enable bit (MD6) in the MODE SFR. Care should be taken when using $\overline{DACK}$. For proper operation, $\overline{DACK}$ must be driven high (+5V) when not using DMA.

b. When an Input FIFO overrun error condition exists. The appropriate bits in the Host Status SFR are set and the interrupt is generated only if enabled.

OUTPUT FIFO: The Output FIFO Request for Service Interrupt operates in a similar manner as the Input FIFO interrupt:

a. When the FIFO contains the threshold number of bytes or more.

b. Output FIFO error condition interrupts are generated when the Output FIFO is underrun.

c. Data Stream Command present in the Output Buffer Latch.

A Data Stream Command interrupt is used to halt normal processing, using the command as a vector to a service routine. When DMA is disabled, the user may program (through HC1) INTRQ to include FIFO Request for Service Interrupts or use INTRQIN and INTRQOUT as Request for Service Interrupts.

IMMEDIATE COMMAND INTERRUPTS:

a. An Immediate Command Out Interrupt is generated, if enabled, to the Host and the corresponding Host Status SFR bit (HSTAT HST6) is cleared, when the internal CPU writes to the Immediate Command OUT (IMOUT) SFR. When the Host reads the Immediate Command OUT (IMOUT) SFR the corresponding bit in the Host Status (HSTAT) SFR is set. This causes the Slave Status Immediate Command OUT Status bit (SSTAT SST6) to be cleared indicating that the Immediate Command OUT (IMOUT) SFR is empty. If enabled, a FIFO-Slave Interface will also be generated to the internal CPU. (See Figure 7b, Immediate Command OUT Flowchart.)

b. An Immediate Command IN interrupt is generated, if enabled, to the Host when the internal CPU has read a byte from the Immediate Command IN (IMIN) SFR. The read operation clears the Host Status SFR Immediate Command IN Status bit (HSTAT HST2) indicating that the Immediate Command IN SFR is empty. The corresponding Slave Status (SSTAT) SFR bit is also set to indicate an empty status. Setting the Slave Status SFR bit generates a FIFO-Slave Interface interrupt, if enabled, to the internal CPU. (See Figure 7a, Immediate Command IN Flowchart.)

NOTE:
Immediate Command IN and OUT interrupts are actually specific Request For Service interrupts to the Host.

FIFO DMA FREEZE MODE: When the internal CPU invokes FIFO DMA Freeze Mode, for example at reset or to reconfigure the FIFO interface, INTRQ is activated. The INTRQ can only be deactivated by the external Host reading the Host Status SFR (HST1 remains active until FIFO DMA Freeze Mode is disabled by the internal CPU).

Once an interrupt is generated, INTRQ will remain high until no interrupt generating condition exists. For a FIFO underrun/overrun error interrupt, the interrupt condition is deactivated by the external Host reading the Host Status SFR. An interrupt is serviced by reading the Host Status SFR to determine the source of the interrupt and vectoring the appropriate service routine.

DMA Requests to the Host

The UPI-452 generates two DMA requests, DRQIN and DRQOUT, to facilitate data transfer between the Host and the Input and Output FIFO channels. A DMA acknowledge, $\overline{DACK}$, is used as a chip select and initiates a data transfer. The external $\overline{READ}$ and $\overline{WRITE}$ signals select the Input and Output FIFO respectively. The $\overline{CS}$ and address lines can also be used as a DMA acknowledge for processors with onboard DMA controllers which do not generate a $\overline{DACK}$ signal.

The internal CPU can configure the UPI-452 to request service from the external host via DMA or interrupts by programming Mode SFR MD6 bit. In addition the external Host enables DMA requests through bits 6 and 7 of the Host Control SFR. When a DMA request is invoked the number of bytes transferred to the Input FIFO is the total number of bytes in the Input FIFO (as determined by the CBP SFR) minus the value programmed in the Input FIFO Threshold SFR. The DMA request line is activated only when the Input FIFO has a threshold number of bytes that can be transferred.

The Output FIFO DMA request is activated when a DSC is written by the internal CPU at the end of a less than threshold size block of data (Flush Mode) or when the Output FIFO threshold is reached. The request remains active until the Input FIFO becomes full or the Output FIFO becomes empty. If a DSC is encountered during an Output FIFO DMA transfer, the DMA request is dropped until the DSC is read. The DMA request will be reactivated after the DSC is read and remains active until the Output FIFO becomes empty or another DSC is encountered.

FIFO MODULE - INTERNAL CPU INTERFACE

Overview

The Input and Output FIFOs are accessed by the internal CPU through direct addressing of the FIFO IN/COMMAND IN and FIFO OUT/COMMAND OUT Special Function Registers. All of the 80C51 instructions involving direct addressing may be used to access the FIFO's SFRs. The FIFO IN, COMMAND IN and Immediate Command In SFRs are actually read only registers, and their Output counterparts are write only. Internal DMA transfers data between Internal memory, External Memory and the Special Function Registers. The Special Function Registers appear as another group of dedicated memory addresses and are programmed as the source or destination via the DMA0/DMA1 Source Address or Destination Address Special Function Registers. The FIFO module manages the transfer of data between the external host and FIFO SFRs.

Internal CPU Access to FIFO Via Software Instructions

The internal CPU has access to the Input and Output FIFOs via the FIFO IN/COMMAND IN and FIFO OUT/COMMAND OUT SFRs which reside in the Special Function Register Array. At the end of every instruction that involves a read of the FIFO IN/COMMAND IN SFR, the SFR is written over by a new byte from the Input FIFO channel when available. At the end of every instruction that involves a write to the FIFO OUT/COMMAND OUT SFR, the new byte is written into the Output FIFO channel and the write pointer is incremented after the write operation (post incremented).

The internal CPU reads the Input FIFO by using the FIFO IN/COMMAND IN SFR as the source register in an instruction. Those instructions which read the Input FIFO are listed below:

```
ADD  A,FIFO IN/COMMAND IN
ADDC A,FIFO IN/COMMAND IN
PUSH FIFO IN/COMMAND IN
ANL  A,FIFO IN/COMMAND IN
ORL  A,FIFO IN/COMMAND IN
XRL  A,FIFO IN/COMMAND IN
CJNE A,FIFO IN/COMMAND IN, rel
SUBB A,FIFO IN/COMMAND IN
MOV  direct,FIFO IN/COMMAND IN
MOV  @Ri,FIFO IN/COMMAND IN
MOV  Rn,FIFO IN/COMMAND IN
MOV  A,FIFO IN/COMMAND IN
```

After each access to these registers, they are overwritten by a new byte from the FIFO.

NOTE:
Instructions which use the FIFO IN or COMMAND IN SFR as both a source and destination register will have the data destroyed as the next data byte is rewritten into the FIFO IN register at the end of the instruction. These instructions are not supported by the UPI-452 FIFO. Data can only be read through the FIFO IN SFR and DSCs through the COMMAND IN SFR. Data read through the COMMAND IN SFR will be read as 0FFH, and DSCs read through the FIFO IN SFR will be read as OFFH. The Immediate Command in SFR is read with the same instructions as the FIFO IN and COMMAND IN SFRs.

The FIFO IN, COMMAND IN and Immediate Command In SFRs are read only registers. Any write operation performed on these registers will be ignored and the FIFO pointers will remain intact.

The internal CPU uses the FIFO OUT SFR to write to the Output FIFO and any instruction which uses the FIFO OUT or COMMAND OUT SFR as a destination will invoke a FIFO write. DSCs are differentiated from data by writing to the COMMAND OUT SFR. In the FIFO, Data Stream Commands have the ninth bit associated with the command byte set to "1". The instructions used to write to the Output FIFO are listed below:

```
MOV FIFO OUT/COMMOUT, A
MOV FIFO OUT/COMMOUT, direct
MOV FIFO OUT/COMMOUT, Rn
POP FIFO OUT/COMMOUT
MOV FIFO OUT/COMMOUT, #data
MOV FIFO OUT/COMMOUNT, @Ri
```

NOTE:
Instructions which use the FIFO OUT/COMMAND OUT SFRs as both a source and destination register cause invalid data to be written into the Output FIFO. These instructions are not supported by the UPI-452 FIFO.

GENERAL PURPOSE DMA CHANNELS

Overview

There are two identical General Purpose DMA Channels on the UPI-452 which allow high speed data transfer from one writeable memory space to another. As many as 64K bytes can be transferred in a single DMA operation. The following memory spaces can be used with DMA channels:

- Internal Data Memory
- External Data Memory
- Special Function Registers

The Special Function Register array appears as a limited group of dedicated memory addresses. The Special Function Registers may be used in DMA transfer operations by specifying the SFR as the source or destination address. The Special Function Registers which may be used in DMA transfers are listed in Table 4. Table 4 also shows whether the SFR may be used as Source or Destination only, or both.

The FIFO can be accessed during DMA by using the FIFO IN SFR as the DMA Source Address Register (SAR) or the FIFO OUT SFR as the Destination Address Register (DAR). (Note: Since the FIFO IN SFR is a read only register, the DMA transfer will be ignored if it is used as a DMA DAR. This is also true if the FIFO OUT SFR is used as a DMA SAR.)

Each DMA channel is software programmable to operate in either Block Mode or Demand Mode. In the Block Mode, DMA transfers can be further programmed to take place in Burst Mode or Alternate Cycle mode. In Burst Mode, the processor halts its execution and dedicates its resources to the DMA transfer. In Alternate Cycle Mode, DMA cycles and instruction cycles occur alternately.

In Demand Mode, a DMA transfer occurs only when it is demanded. Demands can be accepted from an external device (through External Interrupt pins, EXT0/EXT1) or from either the Serial Channel or FIFO flags. In this way, a DMA transfer can be synchronized to an external device, the FIFO or the Serial Port. If the External Interrupt is configured in Edge Mode, a single byte transfer occurs per transition. The external interrupt itself will occur if enabled. If the External Interrupt is configured in Level Mode, DMA transfers continue until the External Interrupt request goes inactive or the byte count becomes zero. The following flags activate Demand Mode transfers of one byte to/from the FIFO or Serial Channel:

RI - Serial Channel Receiver Buffer Full

TI - Serial Channel Transmitter Buffer Empty

Architecture

There are three 16 bit and one 8 bit Special Function Registers associated with each DMA channel.

- The 16 bit Source Address SFR (SAR) points to the source byte.
- The 16 bit Destination Address SFR (DAR) points to the destination.
- The 16 bit Byte Count SFR (BCR) contains the number of bytes to be transferred and is decremented when a byte transfer is accomplished.
- The DMA Control SFR (DCON) is eight bits wide and specifies the source memory space, destination memory space and the mode of operation.

In Auto Increment mode, the Source Address and/or Destination Address is incremented when a byte is transferred. When a DMA transfer is complete (BCR = 0), the DONE bit is set and a maskable interrupt is generated. The GO bit must be set to start any DMA transfer (also, the Slave Control SFR FRZ bit must be set to disable FIFO DMA Freeze Mode). The two DMA channels are designated as DMA0 and DMA1, and their corresponding registers are suffixed by 0 or 1; e.g. SAR0, DAR1, etc.

Table 4. DMA Accessible Special Function Registers

SFR	Symbol	Address	Source Only	Destination Only	Either
Accumulator	A/ACC	0E0H			Y
B Register	B	0F0H			Y
FIFO IN	FIN	0EEH	Y		
COMMAND IN	CIN	0EFH	Y		
FIFO OUT	FOUT	0FEH		Y	
COMMAND OUT	COUT	0FFH		Y	
Serial Data Buffer	SBUF	099H			Y
Port 0	P0	080H			Y
Port 1	P1	090H			Y
Port 2	P2	0A0H			Y
Port 3	P3	0B0H			Y
Port 4	P4	0C0H			Y

DMA Special Function Registers

DMA Control SFR: DCON0, DCON1

Symbolic Address									Physical Address
DCON0	DAS	IDA	SAS	ISA	DM	TM	DONE	GO	092H
DCON1	DAS	IDA	SAS	ISA	DM	TM	DONE	GO	093H
	(MSB)							(LSB)	

Reset Status: DCON0 and DCON1 = 00H

Bit Definition:

DAS	IDA	Destination Address Space
0	0	External Data Memory without Auto-Increment
0	1	External Data Memory with Auto-Increment
1	0	Special Function Register
1	1	Internal Data Memory

SAS	ISA	Source Address Space
0	0	External Data Memory without Auto-Increment
0	1	External Data Memory with Auto-Increment
1	0	Special Function Register
1	1	Internal Data Memory

DM	TM	DMA Transfer Mode
0	0	Alternate-Cycle Transfer Mode
0	1	Burst Transfer Mode
1	0	FIFO or Serial Channel Demand Mode
1	1	External Demand Mode

DONE DMA transfer Flag:

0 DMA transfer is not completed.

1 DMA transfer is complete.

NOTE:
This flag is set when contents of the Byte Count SFR decrements to zero. It is reset automatically when the DMA vectors to its interrupt routine.

GO Enable DMA Transfer:

0 Disable DMA transfer (in all modes).

1 Enable DMA transfer. If the DMA is in the Block mode, start DMA transfer if possible. If it is in the Demand mode, enable the channel and wait for a demand.

NOTE:
The GO bit is reset when the BCR decrements to zero.

DMA Transfer Modes

The following four modes of DMA operation are possible in the UPI-452.

1. ALTERNATE-CYCLE MODE

General

Alternate cycle mode is useful when CPU processing must occur during the DMA transfers. In this mode, a DMA cycle and an instruction cycle occur alternately. The interrupt request is generated (if enabled) at the end of the process, i.e. when BCR decrements to zero. The transfer is initiated by setting the GO bit in the DCON SFR.

Alternate-Cycle FIFO Demand Mode

Alternate cycle demand mode is useful for FIFO transfers of a less urgent nature. As mentioned before, CPU instruction cycles are interleaved with DMA transfer cycles, allowing true parallel processing.

This mode differs from FIFO Demand Mode in that CPU instruction cycles must be interleaved with DMA transfers, even if the FIFO is demanding DMA. In FIFO Demand Mode, CPU cycles would never occur if the FIFO demand was present.

Input Channel

The DMA is configured as in FIFO Demand Mode and transfers are initiated whenever an Input FIFO service request is generated. DMA transfer cycles are alternated with instruction execution cycles. DMA transfers are terminated as in FIFO Demand Mode.

Output Channel

The DMA is configured as in FIFO Demand Mode and transfers are initiated whenever an Output FIFO requests service. DMA transfer cycles are alternated with instruction execution cycles. DMA transfers are terminated as in FIFO Demand Mode.

The FIFO logic resets the interrupt flag after transferring the byte, so the interrupt is never generated.

Once the DMA is programmed to service the FIFO, the request for service interrupt for the FIFO is inhibited until the DMA is done (BCR = 0).

2. BURST MODE

In BURST mode the DMA is initiated by setting the GO bit in the DCON SFR. The DMA operation continues until BCR decrements to zero (zero byte count), then an interrupt is generated (if enabled). No interrupts are recognized during this DMA operation once it has started.

Input Channel

The FIFO Input Channel can be used in burst mode by specifying the FIFO IN SFR as the DMA Source Address. DMA transfers begin when the GO bit in the DMA Control SFR is set. The number of bytes to be transferred must be specified in the Byte Count SFR (BCR) and auto-incrementing of the SAR must be disabled. Once the GO bit is set nothing can interrupt the transfer of data until the BCR is zero. In this mode, a Data Stream Command encountered in the FIFO will be held in the COMMAND IN SFR with the pointers frozen, and invalid data (FFH) will be read through the FIFO IN SFR. If the input FIFO becomes empty during the block transfer, an 0FFH will be read until BCR decrements to zero.

Output Channel

The Output FIFO Channel can be used in burst mode by specifying the FIFO OUT or COMMAND OUT SFR as the DMA Destination Address. DMA transfers begin when the GO bit is set. This mode can be used to send a block of data or a block of Data Stream Commands. If the FIFO becomes full during the block transfer, the remaining data will be lost.

NOTE:
All interrupts including FIFO interrupts are not recognized in Burst Mode. Burst Mode transfers should be used to service the FIFO only when the user is certain that no Data Stream Commands are in the block to be transferred (Input FIFO) and that the FIFO contains enough space to store the block to be transferred. In all other cases Alternate Cycle or Demand Mode should be used.

3. FIFO AND SERIAL CHANNEL DEMAND MODES

NOTES:
1. If the output FIFO is configured as a one byte buffer and the user program consists of two-cycle instructions only, then Alternate-Cycle Mode should be used.
2. In non-auto increment mode for internal to external, or external to internal transfers, the lower 8 bits of the external address should not correspond to the FIFO or Serial Port address.

FIFO Demand Mode

Although any DMA mode is possible using the FIFO buffer, only FIFO Demand and Alternate Cycle FIFO Demand Modes are recommended. FIFO Demand Mode DMA transfers using the input FIFO Channel are set-up by setting the GO bit and specifying the FIFO IN register as the DMA Source Address Register. The BCR should be set to the maximum number of expected transfers. The user must also program bit 1 of the Slave Control Register (SC1) to determine whether the Slave Status (SSTAT) SFR FIFO Request For Service Flag will be activated when the FIFO becomes not empty or full. Once the Request For Service Flag is activated by the FIFO, the DMA transfer begins, and continues until the request flag is deactivated. While the request is active, nothing can interrupt the DMA (i.e. it behaves like burst mode). The DMA Request is held active until one of the following occurs:

1) The FIFO becomes empty.
2) A Data Stream Command is encountered (this generates a FIFO interrupt and DMA operation resumes after the Data Stream Command is read).
3) BCR = 0 (this generates a DMA interrupt and sets the DONE bit).

DMA transfers to the Output FIFO Channel are similar. The FIFO OUT or COMMAND OUT SFR is the DMA Destination Address SFR and a transfer is started by setting the GO bit. The user programs bit 0 of the Slave Control SFR (SC0) to determine whether a demand occurs when the Output FIFO is not full or empty. DMA transfers begin when the Request For Service Flag is activated by the FIFO logic and continue as long as the flag is active. The Flag remains active until one of the following occurs:

1) The FIFO becomes full
2) BCR = 0 (this generates a DMA interrupt and sets the DONE bit).

As in Alternate Cycle FIFO Demand Mode, the FIFO logic resets the interrupt flag after transferring the byte, so the interrupt is never generated.

After the GO bit is set, the DMA is activated if one of the following conditions takes place:
SAR(0/1) = FIFO IN and HIFRS flag is set
DAR(0/1) = FIFO OUT and HOFRS flag is set

The HIFRS and HOFRS signals are internal flags which are not accessible by software. These flags are similar to the SST0 and SST4 flags in the Slave Status Register except that they are of the opposite polarity and once set they are not cleared until the Input FIFO becomes empty (HIFRS) or the Output FIFO becomes full (HOFRS).

Serial Channel Demand Mode

Serial Channel Demand Mode is the logical choice when using the Serial Port. The DMAs can be activated by one of the Serial Channel Flags. Receiver interrupt (RI) or Transmitter Interrupt (TI).

SAR(0/1) = SBUF and RI flag is set
DAR(0/1) = SBUF and TI flag is set

NOTE:
TI flag must be set by software to initiate the first transfer.

When the DMA transfer begins, only one byte is transferred at a time. The serial port hardware automatically resets the flag after completion of the transfer, so an interrupt will not be generated unless DMA servicing is held off due to the DMA being done (BCR = 0) or when the Hold/Hold Acknowledge logic is used and the DMA does not own the bus. In this case a Serial Port interrupt may be generated if enabled because of the status of the RI or TI flags.

In FIFO demand mode, Alternate cycle FIFO demand mode or Serial Port demand mode only one of the following registers (SBUF, FIN or FOUT) should be used as either the SAR or DAR registers to prevent undesired transfers. For example if SAR0 = FIN and DAR0 = SBUF in demand mode, the DMA transfer will start if either the HIFRS or TI flags are set.

4. EXTERNAL DEMAND MODE

The DMA can be initiated by an external device via External interrupt 0 and 1 (INT0/INT1) pins. The INT0 pin demands DMA0 (Channel 0) and INT1 demands DMA1 (Channel 1). If the interrupts are configured in edge mode, a single byte transfer is accomplished for every request. Interrupts also result (INT0 and INT1) after every byte transfer (if enabled). If the interrupts are configured in level mode, the DMA transfer continues until the request goes inactive or BCR = 0. In either case, a DMA interrupt is generated (if enabled) when BCR = 0. The GO bit must be set for the transfer to begin.

EXTERNAL MEMORY DMA

When transferring data to or from external memory via DMA, the HOLD (HLD) and HOLD-ACKNOWLEDGE (HLDA) signals are used for handshaking. The HOLD and HOLD-ACKNOWLEDGE are active low signals which arbitrate control of the local bus. The UPI-452 can be used in a system where multimasters are connected to a single parallel Address/Data bus. The HLD/HLDA signals are used to share resources (memory, peripherals, etc.) among all the processors on the local bus. The UPI-452 can be configured in any of three different External Memory Modes controlled by bits 5 and 6 (REQ & ARB) in the PCON SFR (Table 5). Each mode is described below:

REQUESTER MODE: In this mode, the UPI-452 is not the bus master, but must request the bus from another device. The UPI-452 configures port pin P1.5 as a HLD output and pin P1.6 as a HLDA input. The UPI-452 issues a HLD signal when it needs external access for a DMA channel. It uses the local bus after receiving the HLDA signal from the bus master, and will not release the bus until its DMA operation is complete.

ARBITER MODE: In this mode, the UPI-452 is the bus master. It configures port pin P1.5 as HLD input and pin P1.6 as HLDA output. When a device asserts the HLD signal to use the local bus, the UPI-452 asserts the HLDA signal after current instruction execution is complete. If the UPI-452 needs an external access via a DMA channel, it waits until the requester releases the bus, HLD goes inactive.

DISABLE MODE: When external program memory is accessed by an instruction or by program counter overflow beyond the internal ROM address or external data memory is accessed by MOVX instructions, it is a local memory access and the HLD/HLDA logic is not initiated. When a DMA channel attempts data transfer to/from the external data memory, the HLD/HLDA logic is initiated as described below. DMA transfers from the internal memory space to the internal memory space does not initiate the HLD/HLDA logic.

The balance of the PCON SFR bits are described in the "80C51 Register Description: Power Control SFR" section below.

Latency

When the GO bit is set, the UPI-452 finishes the current instruction before starting the DMA operation. Thus the maximum latency is 3.5 microseconds (at 14 MHz).

DMA Interrupt Vectors

Each DMA channel has a unique vectored interrupt associated with it. There are two vectored interrupts associated with the two DMA channels. The DMA interrupts are enabled and priorities set via the Interrupt Enable and Priority SFR (see "Interrupts" section). The interrupt priority scheme is similar to the scheme in 80C51.

Table 5. DMA MODE CONTROL - PCON SFR

Symbolic Address								Physical Address
PCON	—*	ARB	REQ	—*	—*	—*	—*	—*	87H
	(MSB)							(LSB)

*Defined as per MLS-51 Data Sheet
Reset Status: 00H

Definition:

ARB	REQ	
0	0	HLD/HLDA logic is disabled.
0	1	The UPI-452 is in the Requester Mode.
1	0	The UPI-452 is in the Arbiter Mode.
1	1	Invalid

When a DMA operation is complete (BCR decrements to zero), the DONE flag in the respective DCON (DCON0 or DCON1) SFR is set. If the DMA interrupt is enabled, the DONE flag is reset automatically upon vectoring to the interrupt routine.

Interrupts When DMA is Active

If a Burst Mode DMA transfer is in progress, the interrupts are not serviced until the DMA transfer is complete. This is also true for level activated External Demand DMA transfers. During Alternate Cycle DMA transfers, however, the interrupts are serviced at the end of the DMA cycle. After that, DMA cycles and instruction execution cycles occur alternately. In the case of edge activated External Demand Mode DMA transfers, the interrupt is serviced at the end of DMA transfer of that single byte.

DMA Arbitration

Only one of the two DMA channels is active at a time, except when both are configured in the Alternate Cycle mode. In this case, the DMA cycles and Instruction Execution cycles occur in the following order:

1. DMA Cycle 0.
2. Instruction execution.
3. DMA Cycle 1.
4. Instruction execution.

DMA0 has priority over DMA1 during simultaneous activation of the two DMA channels. If one DMA channel is active, the other DMA channel, if activated, waits until the first one is complete.

If DMA0 is already in the Alternate Cycle mode and DMA1 is activated in Alternate Cycle Mode, it will take two instruction cycles before DMA1 is activated (due to the priority of DMA0). Once DMA1 becomes active, the execution will follow the normal sequence.

If DMA0 is already in the Alternate Cycle mode and DMA1 is activated in Burst Mode, the DMA1 Burst transfer will follow the DMA0 Alternate Cycle transfer (after the completion of the next instruction).

If the UPI-452 (as a Requester) asserts a HLD signal to request a DMA transfer (see "External Memory DMA")and its other DMA Channel requests a transfer before the HLDA signal is received, the channel having higher priority is activated first. A Burst Mode transfer on channel 0 can not be interrupted since DMA0 has the highest priority. A Demand Mode transfer on channel 0 is the only type of activity that can interrupt a block transfer on DMA1.

If, while executing a DMA transfer, the Arbiter receives a HLD signal, and then before it can acknowledge, its other DMA Channel requests a transfer, it then completes the second DMA transfer before sending the HLDA signal to release the bus to the HLD request.

DMA transfers may be held off under the following conditions:

1. A write to any of the DMA registers inhibits the DMA for one instruction cycle.

NOTE:
An instruction cycle may be executed in 1, 2 or 4 machine cycles dependent on the instruction being executed. DMA transfers are only executed after the completion of an instruction cycle never between machine cycles of a single instruction cycle. Similarly instruction cycles are only executed upon completion of a DMA transfer whether it be a one machine cycle transfer or two machine cycles (for ext. to ext. memory transfers).

2. A single machine cycle DMA register read operation (i.e. MOV A, DCON0) will inhibit the DMA for one instruction cycle. However a two cycle DMA register read operation will not inhibit the DMA (i.e. MOV P1, DCON0).

If the HOLD/HOLD Acknowledge logic is enabled in requestor mode the hold request will go active once the go bit has been set (for burst mode) and once the demand flag is set (for demand mode) regardless of whether the DMA is held off by one of the above conditions.

The DMA Transfer waveforms are in Figures 8-11.

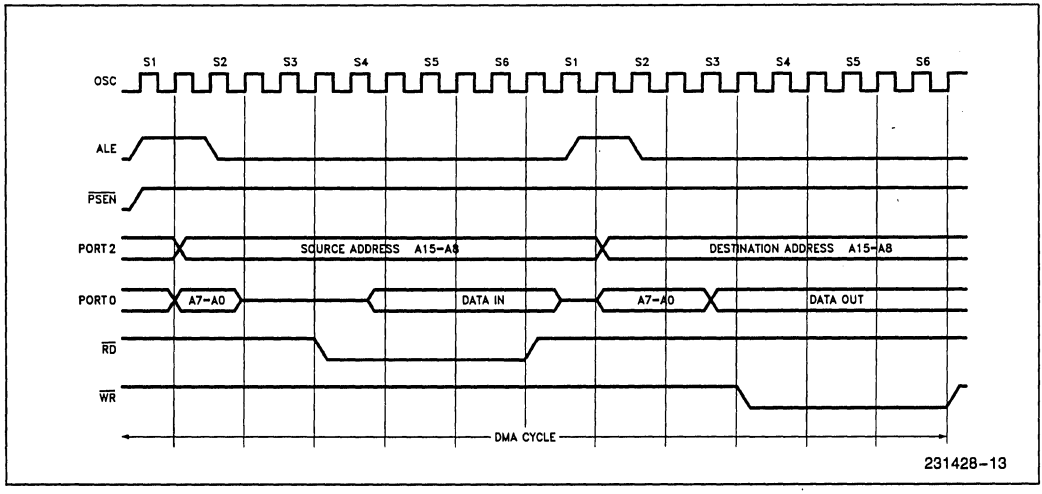


Figure 8. DMA Transfer from External Memory to External Memory

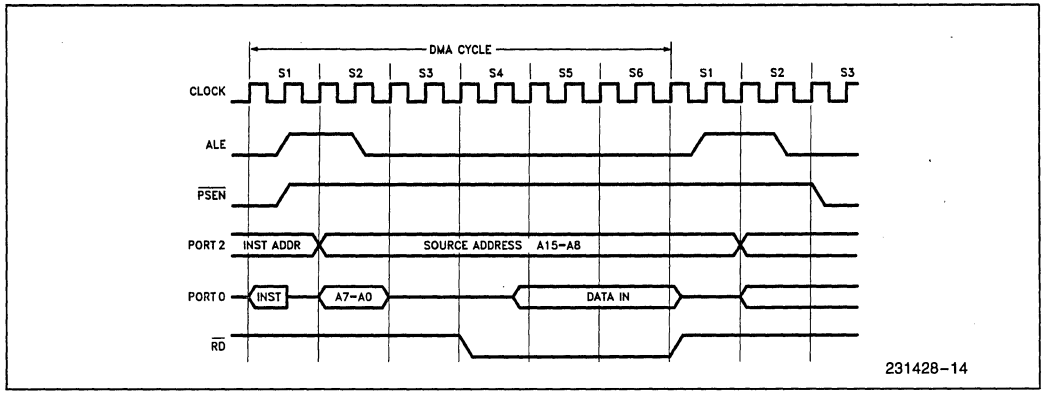


Figure 9. DMA Transfer from External Memory to Internal Memory

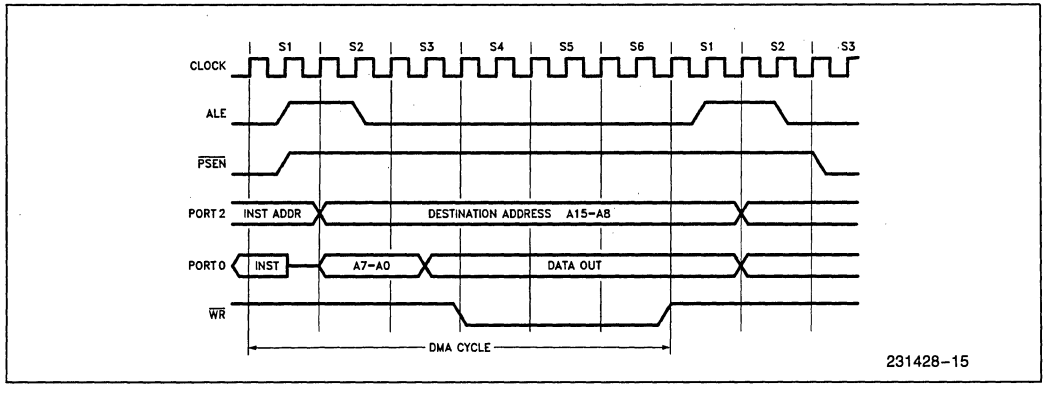


Figure 10. DMA Transfer from Internal Memory to External Memory

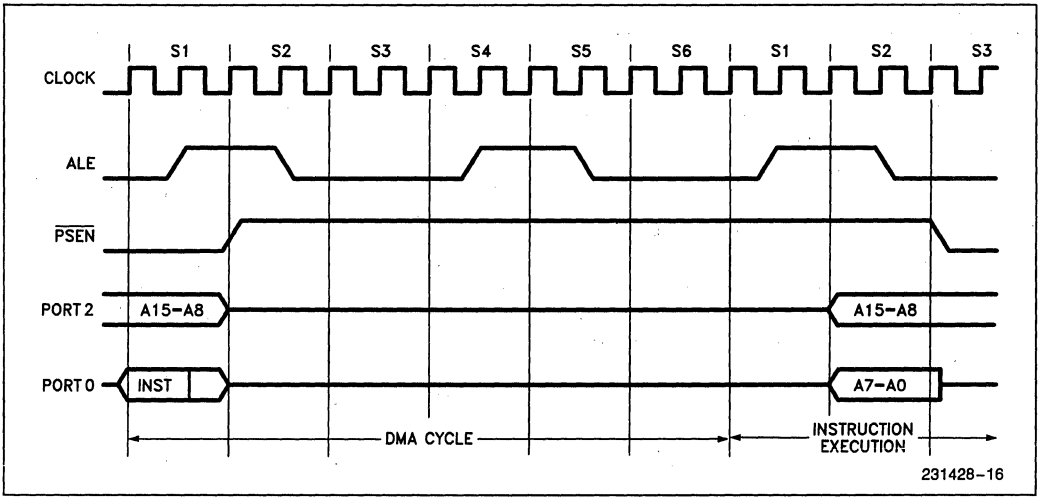


Figure 11. DMA Transfer from Internal Memory to Internal Memory

INTERNAL INTERRUPTS

Overview

The UPI-452 provides a total of eight interrupt sources (Table 6). Their operation is the same as in the 80C51, with the addition of three new interrupt sources for the UPI-452 FIFO and DMA features. These added interrupts have their enable and priority bits in the Interrupt Enable and Priority (IEP) SFR. The IEP SFR is in addition to the 80C51 Interrupt Enable (IE) and Interrupt Priority (IP) SFRs. The added interrupt sources are also globally enabled or disabled by the EA bit in the Interrupt Enable SFR. Table 6 lists the eight interrupt sources in order of priority. Table 7 lists the eight interrupt sources and their respective address vector location in program memory. (DMA interrupts are discussed in the "General Purpose DMA Channels" section. Additional interrupt information for Timer/Counter, Serial Channel, External Interrupt may be found in the Microcontroller Handbook for the 80C51.)

Table 6. Interrupt Priority

Interrupt Source	Priority Level
	(highest)
External Interrupt 0	0
Internal Timer/Counter 0	1
DMA Channel 0 Request	2
External Interrupt 1	3
DMA Channel 1 Request	4
Internal Timer/Counter 1	5
FIFO - Slave Bus Interface	6
Serial Channel	7
	(lowest)

Table 7. Interrupt Vector Addresses

Interrupt Source	Starting Address
External Interrupt 0	3 (003H)
Internal Timer/Counter 0	11 (00BH)
External Interrupt 1	19 (013H)
Internal Timer/Counter 1	27 (01BH)
Serial Channel	35 (023H)
FIFO - Slave Bus Interface	43 (02BH)
DMA Channel 0 Request	51 (033H)
DMA Channel 1 Request	59 (03BH)

FIFO Module Interrupts to Internal CPU

The FIFO module generates interrupts to the internal CPU whenever the FIFO requests service or when a Data Stream Command is in the COMMAND IN SFR. The Input FIFO will request service whenever it becomes full or not empty depending on bit 1 of the Slave Control SFR (IFRS). Similarly, the Output FIFO requests service when it becomes empty or not full as determined by bit 0 of the Slave Control SFR (OFRS). Request for Service interrupts are generated only if enabled by the internal CPU via the Interrupt Enable SFR, and the Slave Control Register.

A Data Stream Command Interrupt is generated whenever there is a Data Stream Command in the COMMAND IN SFR. The interrupt is generated to ensure that the internal interrupt is recognized before another instruction is executed.

Immediate Command Interrupts

a. An Immediate Command IN interrupt is generated, if enabled, to the internal CPU when the Host has written to the Immediate Command IN (IMIN) SFR. The write operation clears the Slave Status SFR bit (SSTAT SST2) and sets the Host Status SFR bit (HSTAT HST2) to indicate that a byte is present in the Immediate Command IN SFR. When the internal CPU reads the Immediate Command IN (IMIN) SFR the Slave Status SFR status bit is set, and the Host Status SFR status bit is cleared indicating the IMIN SFR is empty. Clearing the Host Status SFR bit will cause a Request For Service (INTRQ) interrupt, if enabled, to signal the Host that the IMIN SFR is empty. (See Figure 7a, Immediate Command IN Flowchart.)

b. An Immediate Command OUT interrupt is generated, if enabled, to the internal CPU when the Host has read the Immediate Command OUT SFR. The Host read causes the Slave Status Immediate Command OUT bit (SSTAT SST6) to be set and the corresponding Host Status bit (HSTAT HST6) to be cleared indicating the SFR is empty. When the internal CPU writes to the Immediate Command OUT SFR, the Host Status bit is set and Slave Status bit is cleared to indicate the SFR is full. (See Figure 7b, Immediate Command OUT Flowchart.)

NOTE:
Immediate Command IN and OUT interrupts are actually specific FIFO-Slave Interface interrupts to the internal CPU.

One instruction from the main program is executed between two consecutive interrupt service routines as in the 80C51. However, if the second interrupt service routine is due to a Data Stream Command Interrupt, the main program instruction is not executed (to prevent misreading of invalid data).

Interrupt Enabling and Priority

Each of the three interrupt special function registers (IE, IP and IEP) is listed below with its corresponding bit definitions.

Interrupt Enable SFR (IE)

Symbolic Address									Physical Address
IE	EA	—	—	ES	ET1	EX1	ET0	EX0	0A8H
	(MSB)							(LSB)	

Symbol	Position	Function
EA	IE.7	Enables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	(reserved)
—	IE.5	(reserved)
ES	IE.4	Serial Channel interrupt enable
ET1	IE.3	Internal Timer/Counter 1 Overflow Interrupt
EX1	IE.2	External Interrupt Request 1.
ET0	IE.1	Internal Timer/Counter 0 Overflow Interrupt
EX0	IE.0	External Interrupt Request 0.

Interrupt Priority SFR (IP)

A priority level of 0 or 1 may be assigned to each interrupt source, with 1 being higher priority level, through the IP and the IEP (Interrupt Enable and Priority) SFR. A priority level of 1 interrupt can interrupt a priority level 0 service routine to allow nesting of interrupts.

Symbolic Address									Physical Address
IP	—	—	—	PS	PT1	PX1	PT0	PX0	0B8H
	(MSB)							(LSB)	

Symbol	Position	Function	Priority Within A Level
			(lowest)
—	IP.7	(reserved)	—
—	IP.6	(reserved)	—
—	IP.5	(reserved)	—
PS	IP.4	Local Serial Channel	0.7
PT1	IP.3	Internal Timer/Counter 1	0.5
PX1	IP.2	External Interrupt Request 1	0.3
PT0	IP.1	Internal Timer/Counter 0	0.1
PX0	IP.0	External Interrupt Request 0	0.0
			(highest)

Interrupt Enable and Priority SFR (IEP)

The Interrupt Enable and Priority Register establishes the enabling and priority of those resources not covered in the Interrupt Enable and Interrupt Priority SFRs.

Symbolic Address									Physical Address
IEP	—	—	PFIFO	EDMA0	EDMA1	PDMA0	PDMA1	EFIFO	0F8H
	(MSB)							(LSB)	

Symbol	Position	Function	Priority Within a Level
—	IEP.7	(reserved)	
—	IEP.6	(reserved)	
PFIFO	IEP.5	FIFO Slave Bus Interface Interrupt Priority	0.6
EDMA0	IEP.4	DMA Channel 0 Interrupt Enable	
EDMA1	IEP.3	DMA Channel 1 Interrupt Enable	
PDMA0	IEP.2	DMA Channel 0 Priority	0.2
PDMA1	IEP.1	DMA Channel 1 Priority	0.4
EFIFO	IEP.0	FIFO Slave Bus Interface Interrupt Enable	

FIFO-EXTERNAL HOST INTERFACE FIFO DMA FREEZE MODE

Overview

During FIFO DMA Freeze Mode the internal CPU can reconfigure the FIFO interface. FIFO DMA Freeze Mode is provided to prevent the Host from accessing the FIFO during a reconfiguration sequence. The internal CPU invokes FIFO DMA Freeze Mode by clearing bit 3 of the Slave Control SFR (SC3). INTRQ becomes active whenever FIFO DMA Freeze Mode is invoked to indicate the freeze status. The interrupt can only be deactivated by the Host reading the Host Status SFR.

During FIFO DMA Freeze Mode only two operations are possible by the Host to the UPI-452 slave, the balance are disabled, as shown in Table 8. The internal DMA is disabled during FIFO DMA Freeze Mode, and the internal CPU has write access to all of the FIFO control SFRs (Table 9).

Initialization

At power on reset the FIFO Host interface is automatically frozen. The Slave Control Enable FIFO DMA Freeze Mode bit defaults to FIFO DMA Freeze Mode (SLCON FRZ=0). Below is a list of the FIFO Special Function Registers and their default power on reset values;

SFR Name	Label	Value
Channel Boundary Pointer	CBP	40H / 64D
Output Channel Read Pointers	ORPR	40H / 64D
Output Channel Write Pointers	OWPR	40H / 64D
Input Channel Read Pointers	IRPR	00H / 00D
Input Channel Write Pointers	IWPR	00H / 00D
Input Threshold	ITHR	80H / 128D
Output Threshold	OTHR	01H / 1D

The Input and Output FIFO channels can be reconfigured by programming any of these SFRs while the UPI-452 is in the Freeze Mode. The Host is notified when the Freeze Mode is active by a "1" in HST1 of the Host Status Register (HSTAT). The Host should interrogate HST1 to determine the status of the FIFO interface following reset before attempting to read from or write to the UPI-452 FIFO buffer.

NOTE:
During the initialization sequence of the UPI-452 FIFO SFRs, the OTHR should be changed from the default setting of 1 to a value between 2 and {(80H-CBP)-1}. Please refer to the section on Input and Output FIFO threshold SFRs for further information.

Table 8. Slave Bus Interface Status During FIFO DMA Freeze Mode

Interface Pins; $\overline{DACK}$	$\overline{CS}$	A2	A1	A0	$\overline{READ}$	$\overline{WRITE}$	Operation In Normal Mode	Status In FIFO DMA Freeze Mode
1	0	0	1	0	0	1	Read Host Status SFR	Operational
1	0	0	1	1	0	1	Read Host Control SFR	Operational
1	0	0	1	1	1	0	Write Host Control SFR	Disabled
1	0	0	0	0	0	1	Data or DMA Data from Output Channel	Disabled
1	0	0	0	0	1	0	Data or DMA Data to Input Channel	Disabled
1	0	0	0	1	0	1	Data Stream Command from Output Channel	Disabled
1	0	0	0	1	1	0	Data Stream Command to Input Channel	Disabled
1	0	1	0	0	0	1	Read Immediate Command Out from Output Channel	Disabled
1	0	1	0	0	1	0	Write Immediate Command In to Input Channel	Disabled
0	X	X	X	X	0	1	DMA Data from Output Channel	Disabled
0	X	X	X	X	1	0	DMA Data to Input Channel	Disabled

The UPI-452 can also be programmed to interrupt the Host following power on reset in order to indicate to the Host that FIFO DMA Freeze Mode is in progress. This is done by enabling the INTRQ interrupt output pin via the MODE SFR (MD4) before the Slave Control SFR Enable FIFO DMA Freeze Mode bit is set to Normal Mode. At power on reset the Mode SFR is forced to zero. This disables all interrupt and DMA output pins (INTRQ, DRQIN/INTRQIN and DRQOUT/INTRQOUT). Because the Host Status SFR FIFO DMA Freeze Mode In Progress bit is set, a Request For Service, INTRQ, interrupt is pending until the Host Status SFR is read. This is because the FIFO DMA Freeze Mode interrupt is always enabled. If the Slave Control FIFO DMA Freeze Mode bit (SLCON FRZ) is set to Normal Mode before the MODE SFR INTRQ bit is enabled, the INTRQ output will not go active when the MODE SFR INTRQ bit is enabled if the Host Status SFR has been read.

The default values for the FIFO and Slave Interface represents minimum UPI-452 internal initialization. No specific Special Function Register initialization is required to begin operation of the FIFO Slave Interface. The last initialization instruction must always set the UPI-452 to Normal Mode. This causes the UPI-452 to exit FIFO DMA Freeze Mode and enables Host read/write access of the FIFO.

Following reset, either hardware (via the RST pin) or software (via HCON SFR bit HC3) the UPI-452 requires 2 internal machine cycles (24 TCLCL) to update all internal registers.

Invoking FIFO DMA Freeze Mode During Normal Operation

When the UPI-452 is in normal operation, FIFO DMA Freeze Mode should not be arbitrarily invoked by clearing SC3 (SC3 = 0) because the external Host runs asynchronously to the internal CPU. Invoking FIFO DMA Freeze Mode without first stopping the external Host from accessing the UPI-452 will not guarantee a clean break with the external Host.

The proper way to invoke FIFO DMA Freeze Mode is by issuing an Immediate Command to the external host indicating that FIFO DMA Freeze Mode will be invoked. Upon receiving the Immediate Command, the external Host should complete servicing all pending interrupts and DMA requests, then send an Immediate Command back to the UPI-452 acknowledging the FIFO DMA Freeze Mode request. After issuing the first Immediate Command, the internal CPU should not perform any action on the FIFO until FIFO DMA Freeze Mode is invoked.

If FIFO DMA Freeze Mode is invoked without stopping the Host during Host transfers, only the last two bytes of data written into or read from the FIFO will be valid. The timing diagram for disabling the FIFO module to the external Host interface is illustrated in Figure 12. Due to this synchronization sequence, the UPI-452 might not go into FIFO DMA Freeze Mode immediately after SC3 is cleared. A special bit in the Slave Status Register (SST5) is provided to indicate the status of the FIFO DMA Freeze Mode. The FIFO DMA Freeze Mode operations described in this section are only valid after SST5 is cleared.

As FIFO DMA Freeze Mode is invoked, the DRQIN or DRQOUT will be deactivated (stopping the transferring of data), bit 1 of the Host Status SFR will be set (HST1 = 1), and SST5 will be cleared (SST5 = 0) to indicate to the external Host and internal CPU that the slave interface has been frozen. After the freeze becomes effective, any attempt by the external Host to access the FIFO will cause the overrun and underrun bits to be activated (bits HST7 (for reads) or HST3 (for writes)). These two bits, HST3 and HST7, will be set (deactivated) after the Host Status SFR has been read. If INTRQ is used to request service, the FIFO interface is frozen upon completion of any Host read or write operation in progress.

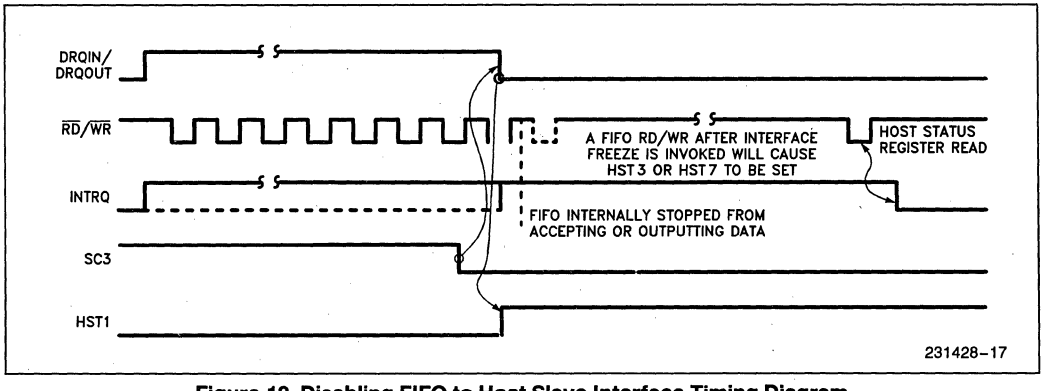


Figure 12. Disabling FIFO to Host Slave Interface Timing Diagram

External Host writing to the Immediate Command In SFR and the Host Control SFR is also inhibited when the slave bus interface is frozen. Writing to these two registers after FIFO DMA Freeze Mode is invoked will also cause HST3 (overrun) to be activated (HST3 = 0). Similarly, reading the Immediate Command Out Register by the external Host is disabled during FIFO DMA Freeze Mode, and any attempt to do so will cause the clearing (deactivating, "0") of HST7 bit (underrun).

After the slave bus interface is frozen, the internal CPU can perform the following operations on the FIFO Special Function Registers (these operations are allowed only during FIFO DMA Freeze Mode).

For FIFO Reconfiguration
1. Changing the Channel Boundary Pointer SFR.
2. Changing the Input and Output Threshold SFR.

To Enhance the Testability
3. Writing to the read and write pointers of the Input and Output FIFO's.
4. Writing to and reading the Host Control SFRs.
5. Controlling some bits of Host and Slave Status SFRS.
6. Reading the Immediate Command Out SFR and Writing to the Immediate Comand In SFR.

Description of each of these special functions are as follows:

FIFO Module SFRs During FIFO DMA Freeze Mode

Table 9 summarizes the characteristics of all the FIFO Special Function Registers during normal and FIFO DMA Freeze Modes. The registers that require special treatment in FIFO DMA Freeze Mode are: HCON, IWPR, IRPR, OWPR, ORPR, HSTAT, SSTAT, MIN & MOUT SFRs. They can be described in detail as follows:

Host Control SFR (HCON)

During normal operation, this register is written to or read by the external Host. However, in FIFO DMA Freeze Mode (i.e. SST5 = 0) the UPI-452 internal CPU has write access to the Host Control SFR and write operations to this SFR by the external Host will not be accepted. If the Host attempts to write to HCON, the Input Channel error condition flag (HST3) will be cleared.

Input FIFO Pointer Registers (IRPR & IWPR)

Once the FIFO module is in FIFO DMA Freeze Mode, error flags due to overrun and underrun of the Input FIFO pointers will be disabled. Any attempt to create an overrun or underrun condition by changing the Input FIFO pointers would result in an inconsistency in performance between the status flag and the threshold counter.

To enhance the speed of the UPI-452, read operations on the Input FIFO will look ahead by two bytes. Hence, every time the IRPR is changed during FIFO DMA Freeze Mode, two NOPs need to be executed so that the two byte pipeline can be updated with the new data bytes pointed to by the new IRPR. The Threshold Counter SFR also needs to change by the same number of bytes as the IRPR (increase Threshold Counter if IRPR goes forward or decrease if IRPR goes backward). This will ensure that future interrupts will still be generated only after a threshold number of bytes are available. (See "Input and Output FIFO Threshold SFR" section below.)

In FIFO DMA Freeze Mode, the internal CPU can also change the content of IWPR, and each change of IWPR also requires an update of the Threshold Counter SFR.

Normally, the internal CPU cannot write into the Input FIFO. It can, however, during FIFO DMA Freeze Mode by first reconfiguring the FIFO as an Output FIFO (Refer to "Input and Output FIFO Threshold SFR" section below). Changing the IRPR to be equal to IWPR generates an empty condition while changing IWPR to be equal to IRPR generates a full condition. The order in which the pointers are written determines whether a full or empty condition is generated.

Output FIFO Pointer SFR (ORPR and OWPR)

In FIFO DMA Freeze Mode the contents of OWPR can be changed by the internal CPU, but each change of OWPR or ORPR requires the Threshold Counter SFR to be updated as described in the next section. A NOP must be executed whenever a new value is written into ORPR, as just described for changes to IRPR. As before, changing ORPR to be equal to OWPR will generate an empty condition, Output FIFO overrun or underrun condition cannot be generated though. The FIFO pointers should not be set to a value outside of its range.

Table 9. FIFO SFR's Characteristics During FIFO DMA Freeze Mode

Label	Name	Normal Operation (SST5 = 1)	FIFO DMA Freeze Mode Operation (SST5 = 0)
HCON	Host Control	Not Accessible	Read & Write
HSTAT	Host Status	Read Only	Read & Write 4
SLCON	Slave Control	Read & Write	Read & Write
SSTAT	Slave Status	Read Only	Read & Write 4
IEP	Interrupt Enable & Priority	Read & Write	Read & Write
MODE	Mode Register	Read & Write	Read & Write
IWPR	Input FIFO Write Pointer	Read Only	Read & Write 5
IRPR	Input FIFO Read Pointer	Read Only	Read & Write 1, 5
OWPR	Output FIFO Write Pointer	Read Only	Read & Write 6
ORPR	Output FIFO Read Pointer	Read Only	Read & Write 2, 6
CBP	Channel Boundary Pointer	Read Only	Read & Write 3
IMIN	Immediate Command In	Read Only	Read & Write
IMOUT	Immediate Command Out	Read & Write	Read & Write
FIN	FIFO IN	Read Only	Read Only
CIN	COMMAND IN	Read Only	Read Only
FOUT	FIFO OUT	Read & Write	Read & Write
COUT	COMMAND OUT	Read & Write	Read & Write
ITHR	Input FIFO Threshold	Read Only	Read & Write
OTHR	Output FIFO Threshold	Read Only	Read & Write

NOTES:
1. Writing of IRPR will automatically cause the FIFO IN SFR to load the contents of the Input FIFO from that location.
2. Writing to ORPR will automatically cause the IOBL SFR to load the contents of the Output FIFO at that ORPR address.
3. Writing to the CBP SFR will cause automatic reset of the four pointers of the Input and Output FIFO channels.
4. The internal CPU cannot directly change the status of these registers. However, by changing the status of the FIFO channels, the internal CPU can indirectly change the contents of the status registers.
5. Changing the Input FIFO Read/Write Pointers also requires that a consistent update of the Input FIFO Threshold Counter SFR.
6. Changing the Output FIFO Read/Write Pointers also requires that a consistent update of the Output FIFO Threshold Counter SFR.

Input and Output FIFO Threshold SFR (ITHR & OTHR)

The Input and Output FIFO Threshold SFRs are also programmable by the internal CPU during FIFO DMA Freeze Mode. For proper operation of the Threshold feature, the Threshold SFR should be changed only when the Input and Output FIFO channels are empty, since they reflect the current number of bytes available to read/write before an interrupt is generated.

Table 10 illustrates the Threshold SFRs range of values and the number of bytes to be transferred when the Request For Service Flag is activated:

Table 10. Threshold SFRs Range of Values and Number of Bytes to be Transferred

ITHR (lower seven bits)	No. of Bytes Available to be Written	OTHR (lower seven bits)	No. of Bytes Available to be Read
0	CBP	2	3
1	CBP-1	3	4
2	CBP-2	•	•
•	•	•	•
•	•	•	•
•	•	(80H-CBP)-3	(80H-CBP)-2
CBP-3	3	(80H-CBP)-2	(80H-CBP)-1
		(80H-CBP)-1	(80H-CBP)

The eighth bit of the Input and Output FIFO Threshold SFR indicates the status of the service requests regardless of the freeze condition. If the eighth bit is a "1", the FIFO is requesting service from the external Host. In other words, when the Threshold SFR value goes below zero (2's complement), a service request is generated*. *The 8th bit of the ITHR SFR must be set during initialization if the Host interrupt request is desired immediately upon leaving Freeze Mode. Normally the ITHR SFR is decremented after each external Host write to the Input FIFO and incremented after each internal CPU read of the Input FIFO. The OTHR SFR is decremented by internal CPU writes and incremented by external Host reads. Thus if the pointers are moved when the FIFO's are not empty, these relationships can be used to calculate the offset for the Threshold SFRs. It is best to change the Threshold SFRs only when the FIFO's are empty to avoid this complication. The threshold registers should also be updated after the pointers have been manipulated.

NOTE:
The ITHR should only be programmed in the range from 0 to (CBP-3). An ITHR value of (CBP-2) could result in a failure to set the Input FIFO service request signal after the Input FIFO has been emptied. Correspondingly, the OTHR should be programmed in the range from 2 to {(80H-CBP)-1}. An OTHR value of 1 could result in a failure to set the Output FIFO service request after subsequent writes by the UPI-452 have filled the Output FIFO.

NOTE:
When programming the ITHR SFR, the eighth bit should be set to 1 (OR'd with 80H). This causes HSTAT SFR HST0 = 0, Input FIFO Request For Service. If ITHR bit 7 = 0 then HSTAT HST0 = 1, Input FIFO Does Not Request Service, and no interrupt will be generated.

Host Status SFR (HSTAT)

When in FIFO DMA Freeze Mode, some bits in the Host Status SFR are forced high and will not reflect the new status until the system returns to normal operation. The definition of the register in FIFO DMA Freeze Mode is as follows:

NOTE:
The internal CPU reads this shadow latch value when reading the Host Status SFR. The shadow latch will keep the information for these bits so normal operation can be resumed with the right status. The following bits are set (= 1) when FIFO DMA Freeze Mode is invoked;

HST7 Output FIFO Error Condition Flag

1 = No error.

0 = An invalid read has been done on the output FIFO or the Immediate Command Out Register by the host CPU.

NOTE:
The normal underrun error condition status is disabled. If an Immediate Command Out (IMOUT) SFR read is attempted during FIFO DMA Freeze Mode, the contents of the IMOUT SFR is output on the Data Buffer and the error status is cleared (= 0).

HST6 Immediate Command Out SFR Status

During normal operation, this bit is cleared (=0) when the IMOUT SFR is written by the UPI-452 internal CPU and set (= 1) when the IMOUT SFR is read by the external Host. Once the host-slave interface is frozen (i.e. SST5 = 0), this bit will be read as a 1 by the host CPU. A shadow latch will keep the information for this bit so normal operation can be resumed with the correct status.

Shadow latch:

1 = Internal CPU reads the IMOUT SFR

0 = Internal CPU writes to the IMOUT SFR

HST5 Data Stream Command at Output FIFO

This bit is forced to a "1" during FIFO DMA Freeze Mode to prevent the external host CPU from trying to read the DSC. Once normal operation is resumed, HST5 will reflect the Data/Command status of the current byte in the Output FIFO.

Shadow Latch (read by the internal CPU):

1 = No Data Stream Command (DSC)

0 = Data Stream Command at Output FIFO

HST4 Output FIFO Service Request Status

When FIFO DMA Freeze Mode is invoked, this bit no longer reflects the Output FIFO Request Service Status. This bit wll be forced to a "1".

HST3 Input FIFO Error Condition Flag

1 = No error.

0 = One of the following operations has been attempted by the external host and is invalid:

1) Write into the Input FIFO

2) Write into the Host Control SFR

3) Write into the Immediate Command In SFR

NOTE:

The normal Input FIFO overrun condition is disabled.

HST2 Immediate Command In SFR Status

This bit is normally cleared when the internal CPU reads the IMIN SFR and set when the external host CPU writes into the IMIN SFR. When the host-slave interface is frozen, reading and writing of the IMIN by the internal CPU will change the shadow latch of this bit. This bit will be read as a "1" by the external Host.

Shadow latch.

1 = Internal CPU writes into IMIN SFR

0 = Internal CPU reads the IMIN SFR

HST1 FIFO DMA Freeze Mode Status

1 = FIFO DMA Freeze Mode.

0 = Normal Operation (non-FIFO DMA Freeze Mode).

NOTE:

This bit is used to indicate to the external Host that the host-slave interface has been frozen and hence the external Host functions are now reduced as shown in Table 8.

HST0 Input FIFO Request Service Satus

When slave interface is frozen this bit no longer reflects the Input FIFO Request Service Status. This bit will be forced to a "1".

Slave Status SFR (SSTAT)

The Slave Status SFR is a read-only SFR. However, once the slave interface is frozen, most of the bits of this SFR can be changed by the internal CPU by reconfiguring the FIFO and accessing the FIFO Special Function Registers.

SST7 Output FIFO Overrun Error Flag

Inoperative in FIFO DMA Freeze Mode.

SST6 Immediate Command Out SFR Status

In FIFO DMA Freeze Mode, this bit will be cleared when the internal CPU reads the Immediate Command Out SFR and set when the internal CPU writes to the Immediate Command Out Register.

SST5 FIFO-External Interface FIFO DMA Freeze Mode Status

This bit indicates to the internal CPU that FIFO DMA Freeze Mode is in progress and that it has write access to the FIFO Control, Host control and Immediate Command SFRs.

SST4 Output FIFO Request Service Status

During normal operation, this bit indicates to the internal CPU that the Output FIFO is ready for more data. The status of this bit reflects the position of the Output FIFO read and write pointers. Hence, in FIFO DMA Freeze Mode, this flag can be changed by the internal CPU indirectly as the read and write pointers change.

SST3 Input FIFO Underrun Flag

Inoperative during FIFO DMA Freeze Mode.

During normal operation, a read operation clears (=0) this bit when there are no data bytes in the Input FIFO and deactivated (=1) when the Slave Status SFR is read. In FIFO DMA Freeze Mode, this bit will not be cleared by an Input FIFO read underrun error condition, nor will it be reset by the reading of the Slave Status SFR.

SST2 Immediate Command In SFR Status

This bit is normally activated (=0) when the external host CPU writes into the Immediate Command In SFR and deactivated (=1) when it is read by the internal CPU. In FIFO DMA Freeze Mode, this bit will not be activated (=0) by the external Host's writing of the Immediate Command IN SFR since this function is disabled. However, this bit will be cleared (=0) if the internal CPU writes to the Immediate Command In SFR and it will be set =1) if it reads from the register.

SST1 Data Stream Command at Input FIFO Flag

In FIFO DMA Freeze Mode, this bit operates normally. It indicates whether the next byte of data from the Input FIFO is a DSC or data byte. If it is a DSC byte, reading from the FIFO IN SFR will result in reading invalid data (FFH) and vice versa. In FIFO DMA Freeze Mode, this bit still reflects the type of data byte available from the Input FIFO.

SST0 Input FIFO Service Request Flag

During normal operation, this bit is activated (=0) when the Input FIFO contains bytes that can be read by the internal CPU and deactivated (=1) when the Input FIFO does not need any service from the internal CPU. In FIFO DMA Freeze Mode, the status of this bit should not change unless the pointers of the Input FIFO are changed. In this mode, the internal CPU can indirectly change this bit by changing the read and write pointers of the Input FIFO but cannot change it directly.

Immediate Command In/Out SFR (IMIN/IMOUT)

If FIFO DMA Freeze Mode is in progress, writing to the Immediate Command In SFR by the external host will be disabled, and any such attempt will cause HST3 to be cleared (=0). Similarly, the Immediate Command Out SFR read operation (by the host) will be disabled internally and read attempts will cause HST7 to be cleared (=0).

Internal CPU Read and Write of the FIFO During FIFO DMA Freeze Mode

In normal operation, the Input FIFO can only be read by the internal CPU and similarly, the Output FIFO can only be written by the internal CPU. During FIFO DMA Freeze Mode, the internal CPU can read the entire contents of the Input FIFO by programming the CBP SFR to 7FH, setting the IRPR SFR to zero, and then the IWPR SFR to zero. Programming the pointer registers in this order generates a FIFO full signal to the FIFO logic and enables internal CPU read operations. If the IWPR and IRPR are already zero, the write pointer should be changed to a non-zero value to clear the empty status then the pointers can be set to zero. Writing to the IRDR SFR automatically updates the look ahead registers.

In a similar manner, the internal CPU can write to all 128 bytes of the FIFO by setting the CBP SFR to zero, setting OWPR SFR to zero, and then setting ORPR SFR to zero. This generates a FIFO empty signal and allows internal CPU write operations to all 128 bytes of the FIFO. The Threshold registers also need to be adjusted when the pointers are changed. (See "Input and Output FIFO Threshold SFR" section below.)

MEMORY ORGANIZATION

The UPI-452 has separate address spaces for Program Memory and Data Memory like the 80C51. The Program Memory can be up to 64K bytes. The lower 8K of Program Memory may reside on-chip. The Data Memory consists of 256 bytes of on-chip RAM, up to 64K bytes of off-chip RAM and a number of "SFRs" (Special Function Registers) which appear as yet another set of unique memory addresses.

Table 11a. Internal Memory Addressing

Memory Space	Addressing Method
Lower 128 Bytes of Internal RAM	Direct or Indirect
Upper 128 Bytes of Internal RAM	Indirect Only
UPI-452 SFR's	Direct Only

The 80C51 Special Function Registers are listed in Table 11a, and the additional UPI-452 SFRs are listed in Table 11b. A brief description of the 80C51 core SFRs is also provided below.

Accessing External Memory

As in the 80C51, accesses to external memory are of two types: Accesses to external Program Memory and accesses to external Data Memory.

External Program Memory is accessed under two conditions:

1) Whenever signal $\overline{EA}$ = 0; or
2) Whenever the program counter (PC) contains a number that is larger than 1FFFH.

This requires that the ROMless versions have $\overline{EA}$ wired low to enable the lower 8K program bytes to be fetched from external memory.

External Data Memory is accessed using either the MOVX @DPTR (16 bit address) or the MOVX @Ri (8 bit address) instructions, or during external data memory transfers.

Table 11b. 80C51 Special Function Registers

Symbol	Name	Address	Contents
*ACC	Accumulator	0E0H	00H
*B	B Register	0F0H	00H
*PSW	Program Status Word	0D0H	00H
SP	Stack Pointer	81H	07H
DPTR	Data Pointer (consisting of DPH and DPL)	82H	0000H
*P0	Port 0	80H	0FFH
*P1	Port 1	90H	0FFH
*P2	Port 2	0A0H	0FFH
*P3	Port 3	0B0H	0FFH
*IP	Interrupt Priority Control	0B8H	0E0H
*IE	Interrupt Enable Control	0A8H	60H
TMOD	Timer/Counter Mode Control	89H	00H
*TCON	Timer/Counter Control	88H	00H
TH0	Timer/Counter 0 (high byte)	8CH	00H
TL0	Timer/Counter 0 (low byte)	8AH	00H
TH1	Timer/Counter 1 (high byte)	8DH	00H
TL1	Timer/Counter 1 (low byte)	8BH	00H
*SCON	Serial Control	98H	00H
SBUF	Serial Data Buff	99H	I
PCON	Power Control	87H	I0H

I = Indeterminate

The SFRs marked with an asterisk (*) are both bit- and byte- addressable. The functions of the SFRs are as follows:

Table 11c. UPI-452 Additional Special Function Registers

Symbol	Name	Address	Contents
BCRL0	DMA Byte Count Low Byte/	0E2H	I
BCRH0	High Byte/ Channel 0	0E3H	I
BCRL1	Low Byte/	0F2H	I
BCRH1	Hi Byte/ Channel 1	0F3H	I
CBP	Channel Boundary Pointer	0ECH	40H
CIN	COMMAND IN	0EFH	I
COUT	COMMAND OUT DMA Destination Address	0FFH	I

Table 11c. UPI-452 Additional Special Function Registers (Continued)

Symbol	Name	Address	Contents
DARL0	Low Byte/	0C2H	I
DARH0	Hi Byte/ Channel 0	0C3H	I
DARL1	Low Byte/	0D2H	I
DARH1	Hi Byte/ Channel 1	0D3H	I
DCON0	DMA0 Control	92H	00H
DCON1	DMA1 Control	93H	00H
FIN	FIFO IN	0EEH	I
FOUT	FIFO OUT	0FEH	I
HCON	Host Control	0E7H	00H
HSTAT	Host Status	0E6H	0FBH
*IEP	Interrupt Enable and Priority	0F8H	0C0H
IMIN	Immediate Command In	0FCH	I
IMOUT	Immediate Command Out	0FDH	I
IRPR	Input Read Pointer	0EBH	00H
ITHR	Input FIFO Threshold	0F6H	80H
IWPR	Input Write Pointer	0EAH	00H
MODE	Mode Register	0F9H	8FH
ORPR	Output Read Pointer	0FAH	40H
OTHR	Output FIFO Threshold	0F7H	01H
OWPR	Output Write Threshold	0FBH	40H
*P4	Port 4 DMA Source Address	0C0H	0FFH
SARL0	Low Byte/	0A2H	I
SARH0	Hi Byte/ Channel 0	0A3H	I
SARL1	Low Byte/	0B2H	I
SARH1	Hi Byte/ Channel 1	0B3H	I
*SLCON	Slave Control	0E8H	04H
SSTAT	Slave Status	0E9H	08FH

I = Indeterminate

The SFRs marked with an asterisk (*) are both bit- and byte- addressable. The functions of the SFRs are as follows:

Miscellaneous Special Function Register Description

80C51 SFRs

ACCUMULATOR

ACC is the Accumuator SFR. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.

B REGISTER

The B SFR is used during multiply and divide operations. For other instructions it can be treated as another scratch pad regster.

PROGRAM STATUS WORD

The PSW SFR contains program status information as detailed in Table 12.

STACK POINTER

The Stack Pointer register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

DATA POINTER

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

PORTS 0 TO 4

P0, P1, P2, P3 and P4 are the SFR latches of Ports 0, 1, 2, 3 and 4, respectively.

SERIAL DATA BUFFER

The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

TIMER/COUNTER SFR

Register pairs (TH0, TL0), and (TH1, TL1) are the 16-bit counting registers for Timer/Counters 0 and 2.

POWER CONTROL SFR (PCON)

The PCON Register (Table 13) controls the power down and idle modes in the UPI-452, as well as providing the abilty to double the Serial Channel baud rate. There are also two general purpose flag bits available to the user. Bits 5 and 6 are used to set the HOLD/HOLD Acknowledge mode (see "General Purpose DMA Channels" section), and bit 4 is not used.

Table 12. Program Status Word

Symbolic Address									Physical Address
PSW	CY	AC	FO	RS1	RS0	OV	—	P	0D0H
	(MSB)							(LSB)	

Symbol	Position	Name
CY	PSW.7	Carry Flag
AC	PSW.6	Auxiliary Carry (For BCD operations)
F0	PSW.5	Flag 0 (user assignable)
RS1	PSW.4	Register Bank Select bit 1*
RS0	PSW.3	Register Bank Select bit 0*
OV	PSW.2	Overflow Flag
—	PSW.1	(reserved)
P	PSW.0	Parity Flag

*(RS1, RS0) enable internal RAM register banks as follows:

RS1	RS0	Internal RAM Register Bank
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

Table 13. PCON Special Function Register

Symbolic Address									Physical Address
PCON	SMOD	ARB	REQ	—	GF1	GF0	PD	IDL	087H
	(MSB)							(LSB)	

Symbol	Position	Function
SMOD	PCON7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either Mode 1, 2 or 3.
ARB	PCON6	HLD/HLDA Arbiter control bit *
REQ	PCON5	HLD/HLDA Requestor control bit *
—	PCON4	(reserved)
GF1	PCON3	General-purpose flag bit
GF0	PCON2	General-purpose flag bit
PD	PCON1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON0	Idle Mode bit. Setting this bit activates idle mode operation.

*See "Ext. Memory DMA" description.

NOTE:
If 1's are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (000X0000).

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C†
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin to V_{SS} .....−0.5V to V_{CC} + 0.5V
Voltage on V_{CC} to V_{SS} .....−0.5V to +6.5V
Power Dissipation.....1.0W**

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage (except XTAL1, RST)	2.0	V_{CC} + 0.5	V	
V_{IH1}	Input High Voltage (XTAL1, RST)	3.9	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage (Ports 1, 2, 3, 4)		0.45	V	I_{OL} = 1.6 mA (Note 1)
V_{OL1}	Output Low Voltage (except Ports 1, 2, 3, 4)		0.45	V	I_{OL} = 3.2 mA (Note 1)
V_{OH}	Output High Voltage (Ports 1, 2, 3, 4)	2.4		V	I_{OH} = −60 μA, V_{CC} = 5V ±10%
		0.9 V_{CC}		V	I_{OH} = −10 μA
V_{OH1}	Output High Voltage (except Ports 1, 2, 3, 4 and Host Interface (Slave) Port)	2.4		V	I_{OH} = −400 μA, V_{CC} = 5V ±10%
		0.9 V_{CC}		V	I_{OH} = −40 μA (Note 2)
V_{OH2}	Output High Voltage (Host Interface (Slave) Port)	2.4		V	I_{OH} = −400 μA, V_{CC} = 5V ±10%
		V_{CC} − 0.4		V	I_{OH} = −10 μA
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3, 4)		−50	μA	V_{IN} = 0.45V
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3, 4)		−650	μA	V_{IN} = 2V

D.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V (Continued)

Symbol	Parameter	Min	Max	Units	Test Conditions
I_{LI}	Input Leakage Current (except Ports 1, 2, 3, 4)		±10	μA	0.45V < V_{IN} < V_{CC}
I_{OZ}	Output Leakage Current (except Ports 1, 2, 3, 4)		±10	μA	0.45V < V_{OUT} < V_{CC}
I_{CC}	Operating Current		50	mA	V_{CC} = 5.5V, 14 MHz (Note 4)
I_{CCI}	Idle Mode Current		25	mA	V_{CC} = 5.5V, 14 MHz (Note 5)
I_{PD}	Power Down Current		100	μA	V_{CC} = 2V (Note 3)
RRST	Reset Pulldown Resistor	50	150	KΩ	
CIO	Pin Capacitance		20	pF	1 MHz, T_A = 25°C (sampled, not tested on all parts)

NOTES:
1. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OLS} of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
2. Capacitive loading on Ports 0 and 2 may cause the V_{OH} on ALE and PSEN to momentarily fall before the 0.9 V_{CC} specification when the address bits are stabilizing.
3. Power DOWN I_{CC} is measured with all output pins disconnected; EA = Port 0 = V_{CC}; XTAL2 N.C.; RST = V_{SS}; DB = V_{CC}; $\overline{WR}$ = $\overline{RD}$ = $\overline{DACK}$ = $\overline{CS}$ = A0 = A1 = A2 = V_{CC}. Power Down Mode is not supported on the 87C452P.
4. I_{CC} is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} − 0.5V; XTAL2 N.C.; EA = RST = Port 0 = V_{CC}; $\overline{WR}$ = $\overline{RD}$ = $\overline{DACK}$ = $\overline{CS}$ = A0 = A1 = A2 = V_{CC}. I_{CC} would be slightly higher if a crystal oscillator is used.
5. Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} − 0.5V; XTAL2 N.C.; Port 0 = V_{CC}; EA = RST = V_{SS}; $\overline{WR}$ = $\overline{RD}$ = $\overline{DACK}$ = $\overline{CS}$ = A0 = A1 = A2 = V_{CC}.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for:

A: Address.

C: Clock.

D: Input data.

H: Logic level HIGH.

I: Instruction (program memory contents).

L: Logic level LOW, or ALE.

P: PSEN.

Q: Output data.

R: READ signal.

T: Time.

V: Valid.

W: WRITE signal.

X: No longer a valid logic level.

Z: Float.

EXAMPLE

TAVLL = Time for Address Valid to ALE Low.
TLLPL = Time for ALE Low to $\overline{PSEN}$ Low.

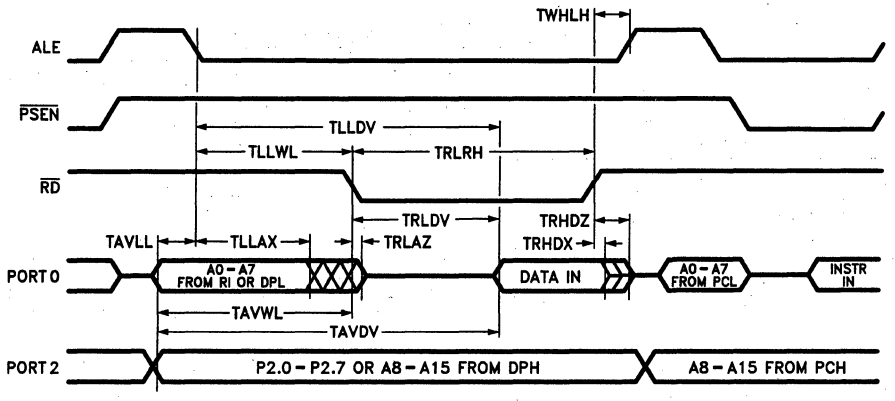
A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%, V_{SS} = 0V, Load Capacitance for Port 0, ALE, and PSEN = 100 pF, Load Capacitance for All Other Outputs = 80 pF

EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

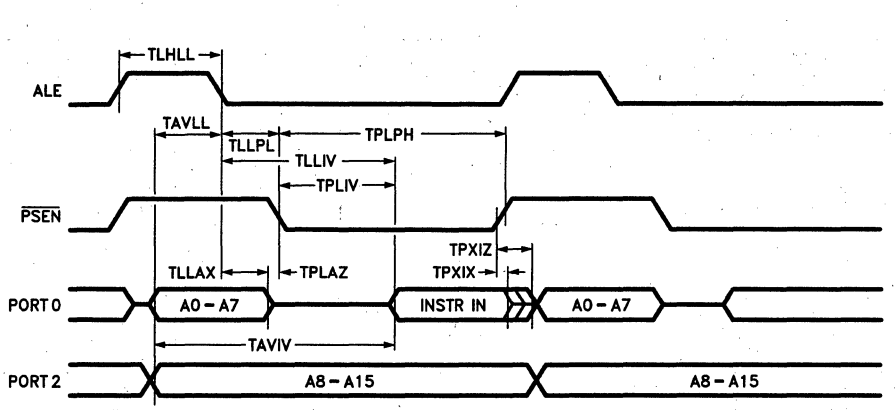
Symbol	Parameter	14 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
1/TCLCL	Oscillator Frequency	3.5	14			MHz
TLHLL	ALE Pulse Width	103		2TCLCL−40		ns
TAVLL	Address Valid to ALE Low (Note 1)	25		TCLCL−55		ns
TLLAX	Address Hold after ALE Low	36		TCLCL−35		ns
TLLIV	ALE Low to Valid Instr In		185		4TCLCL−100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	31		TCLCL−40		ns
TPLPH	$\overline{PSEN}$ Pulse Width	169		3TCLCL−45		ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr In		110		3TCLCL−105	ns
TPXIX	Input Instr Hold after $\overline{PSEN}$	0		0		ns
TPXIZ	Input Instr Float after $\overline{PSEN}$ (Note 1)		57		TCLCL−25	ns
TAVIV	Address to Valid Instr In		252		5TCLCL−105	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		10		10	ns
TRLRH	$\overline{RD}$ Pulse Width	329		6TCLCL−100		ns
TWLWH	$\overline{WR}$ Pulse Width	329		6TCLCL−100		ns
TRLDV	$\overline{RD}$ Low to Valid Data In		192		5TCLCL−165	ns
TRHDX	Data Hold after $\overline{RD}$	0		0		ns
TRHDZ	Data Float after $\overline{RD}$		73		2TCLCL−70	ns
TLLDV	ALE Low to Valid Data In		422		8TCLCL−150	ns
TAVDV	Address to Valid Data In		478		9TCLCL−165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	164	264	3TCLCL−50	3TCLCL+50	ns
TAVWL	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	156		4TCLCL−130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	11		TCLCL−60		ns
TWHQX	Data Hold after $\overline{WR}$	21		TCLCL−50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	31	111	TCLCL−40	TCLCL+40	ns
TQVWH	Data Valid to $\overline{WR}$ (Setup Time)	350		7TCLCL−150		ns

NOTE:
1. Use the value of 14 MHz specification or variable oscillator specification, whichever is greater.

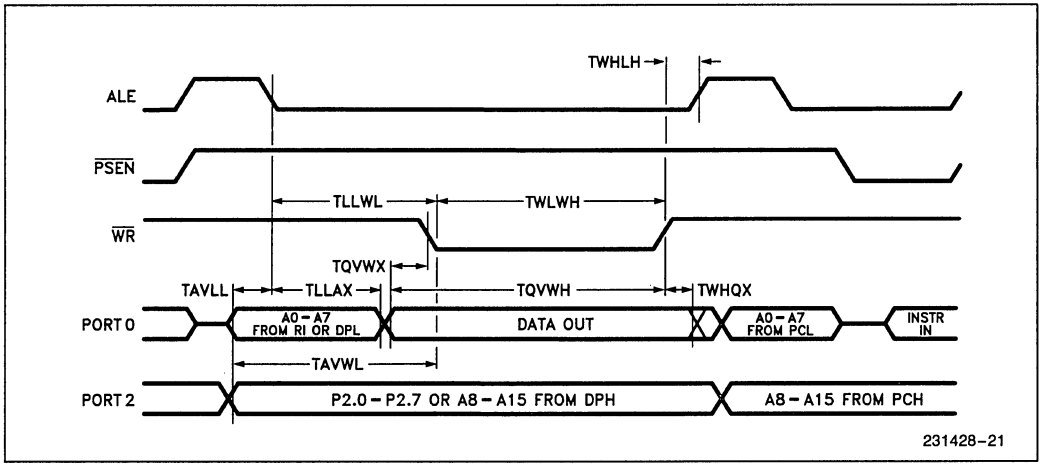
EXTERNAL DATA MEMORY READ CYCLE



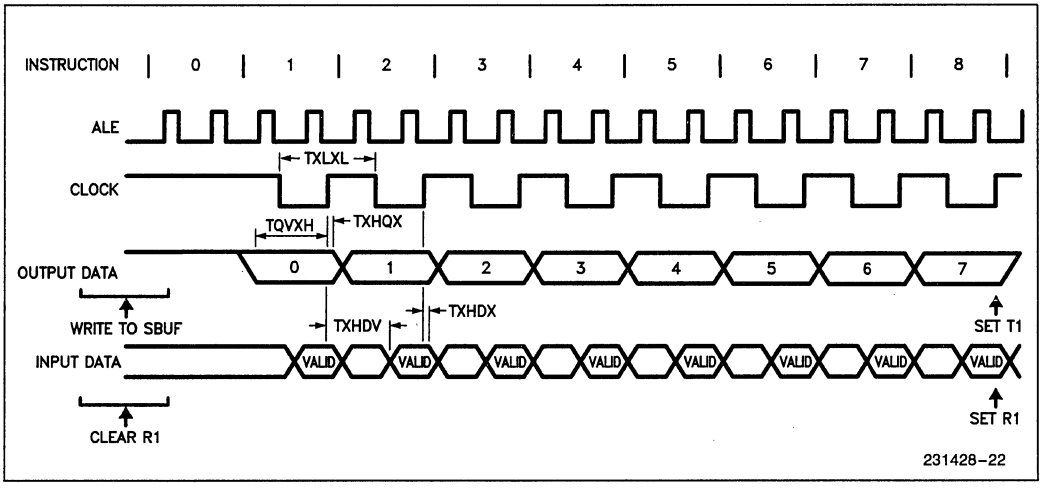
EXTERNAL PROGRAM MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



SHIFT REGISTER MODE TIMING WAVEFORMS



EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
1/TCLCL	Oscillator Frequency	3.5	14	MHz
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

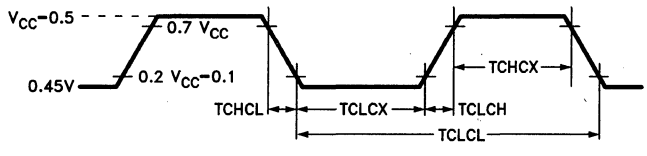
NOTE:
External clock timings are sampled, not tested on all parts.

SERIAL PORT TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

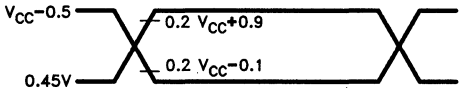
Symbol	Parameter	14 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TXLXL	Serial Port Clock Cycle Time	857		12TCLCL		ns
TQVXH	Output Data Setup to Clock Rising Edge	581		10TCLCL−133		ns
TXHQX	Output Data Hold after Clock Rising Edge	26		2TCLCL−117		ns
TXHDX	Input Data Hold after Clock Rising Edge	0		0		ns
TXHDV	Clock Rising Edge to Input Data Valid		581		10TCLCL−133	ns

EXTERNAL CLOCK DRIVE WAVEFORM



231428-23

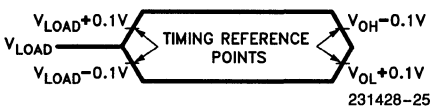
AC TESTING INPUT, OUTPUT WAVEFORMS



231428-24

AC inputs during testing are driven at V_{CC} −0.5V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at V_{IH} min. for a logic "1" and V_{IL} max. for a logic "0".

FLOAT WAVEFORMS

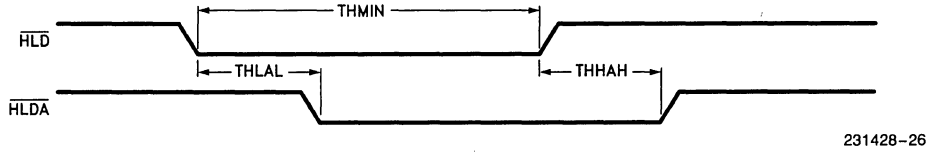


231428-25

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

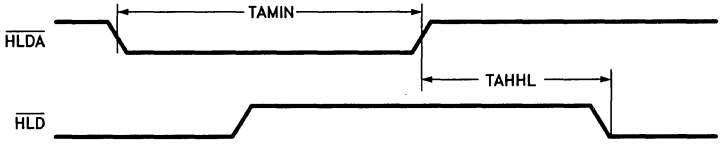
HLD/HLDA WAVEFORMS

Arbiter Mode



231428-26

Requestor Mode



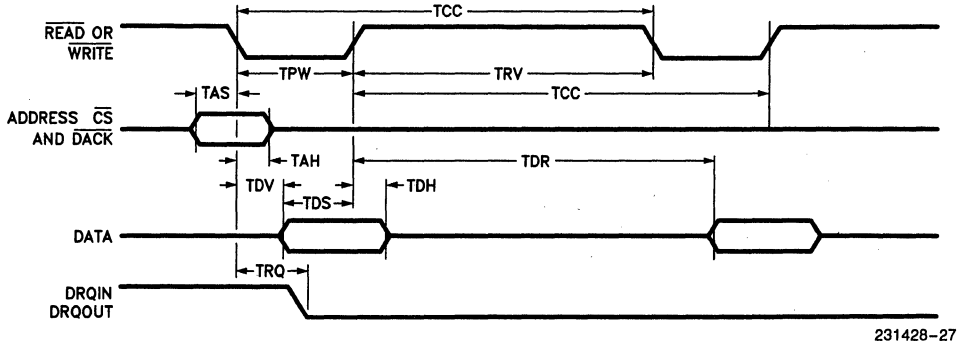
231428-31

HLD/HLDA TIMINGS

Test Conditions: T_A = 0°C to +70°C; V_{CC} = 5V ±10%, V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	14 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
THMIN	$\overline{HLD}$ Pulse Width	386		4TCLCL+100		ns
THLAL	$\overline{HLD}$ to $\overline{HLDA}$ Delay if $\overline{HLDA}$ is Granted	186	672	4TCLCL−100	8TCLCL+100	ns
THHAH	HLD to HLDA Delay	186	672	4TCLCL−100	8TCLCL+100	ns
TAMIN	$\overline{HLDA}$ Pulse Width	386		4TCLCL+100		ns
TAHHL	$\overline{HLDA}$ Inactive to $\overline{HLD}$ Active	186		4TCLCL−100		ns

HOST PORT WAVEFORMS



HOST PORT TIMINGS

Test Conditions: T_A = 0°C to 70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	14 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
TCC	Cycle Time	429		6TCLCL		ns
TPW	Command Pulse Width	100		100		ns
TRV	Recovery Time	60		60		ns
TAS	Address Setup Time	5		5		ns
TAH	Address Hold Time	30		30		ns
TDS	$\overline{\text{WRITE}}$ Data Setup Time	30		30		ns
TDHW	$\overline{\text{WRITE}}$ Data Hold Time	5		5		ns
TDHR	$\overline{\text{READ}}$ Data Hold Time	5	40	5	40	ns
TDV	$\overline{\text{READ}}$ Active to Read Data Valid Delay		92		92	ns
TDR	$\overline{\text{WRITE}}$ Inactive to Read Data Valid Delay (Applies only to Host Control SFR)		343		4.8TCLCL	ns
TRQ	$\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ Active to DRQIN or DRQOUT Inactive Delay		150		150	ns

REVISION HISTORY

DOCUMENT: UPI-452 Data Sheet

OLD REVISION NUMBER: 231428-004

NEW REVISION NUMBER: 231428-005

1. Maximum Clock Rate was changed from 16 MHz to 14 MHz. This change is reflected in all Maximum Timing specifications.
2. The proper range of values for ITHR has been changed from [0 to (CBP-2)] to [0 to (CBP-3)] to ensure proper setting of the Input FIFO request for service bit. See the following sections: INPUT FIFO CHANNEL, and INPUT AND OUTPUT FIFO THRESHOLD SFR (ITHR & OTHR).
3. The proper range of values for OTHR has been changed from [1 to {(80H-CBP)-1}] to [2 to {(80-CBP)-1}] to ensure proper setting of the Output FIFO request for service bit. See the following sections: OUTPUT FIFO CHANNEL, FIFO-EXTERNAL HOST INTERFACE FIFO DMA FREEZE MODE, and INPUT AND OUTPUT FIFO THRESHOLD SFR (ITHR & OTHR).
4. The following D.C. Characteristics were deleted from the data sheet:

   V_{OH} = 0.75* V_{CC} @ I_{OH} = −25 μA,

   V_{OH1} = 0.75* V_{CC} @ I_{OH} = 150 μA,

   V_{OH2} = 3.0V @ I_{OH} = 1 mA, and

   I_{CC1} = 15 mA @ V_{CC} = 5.5V (87C452P).

   See D.C. CHARACTERISTICS TABLE.
5. The parameter descriptions for THHAH and THLAL has been reversed and their maximum specification for clock rates less than 14 MHz has been changed from [4TCLC + 100 ns] to [8TCLC + 100 ns]. See HLD/HLDA TIMINGS.
6. TAMIN specification has been removed from the Arbiter Mode waveform diagram and added to the Requestor Mode waveform diagram. See HLD/HLDA WAVEFORMS.

CHMOS EPROMS

14

27C64/87C64
64K (8K x 8) CHMOS PRODUCTION AND UV ERASABLE PROMS

- **CHMOS Microcontroller and Microprocessor Compatible**
  - **87C64-Integrated Address Latch**
  - **Universal 28 Pin Memory Site, 2-line Control**
- **Low Power Consumption**
  - **100 μA Maximum Standby Current**
- **Noise Immunity Features**
  - **$\pm$10% V_{CC} Tolerance**
  - **Maximum Latch-up Immunity Through EPI Processing**
- **High Performance Speeds**
  - **150 ns Maximum Access Time**
- **New Quick-Pulse Programming™ Algorithm (1 second programming)**
- **Available in 28-Pin Cerdip and Plastic DIP Package and 32-Lead PLCC Package.**

(See Packaging Spec, Order #231369)

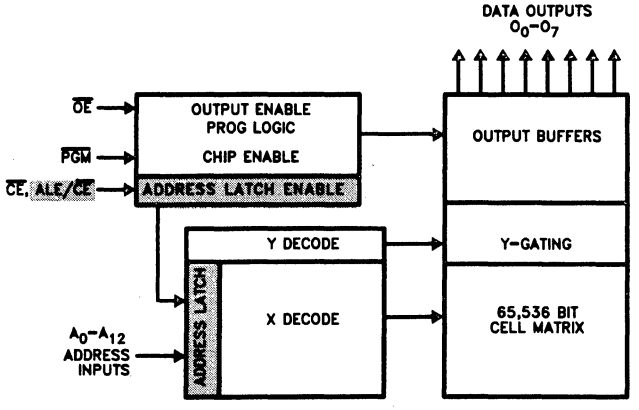
Intel's 27C64 and 87C64 CHMOS EPROMs are 64K bit 5V only memories organized as 8192 words of 8 bits. They employ advanced CHMOS*II-E circuitry for systems requiring low power, high performance speeds, and immunity to noise. The 87C64 has been optimized for multiplexed bus microcontroller and microprocessor compatibility while the 27C64 has a non-multiplexed addressing interface and is plug compatible with the standard Intel 2764A (HMOS II-E).

The 27C64 and 87C64 are offered in both a ceramic DIP, Plastic DIP, and Plastic Leaded Chip Carrier (PLCC) Packages. Cerdip packages provide flexibility in prototyping and R&D environments, whereas Plastic DIP and PLCC EPROMs provide optimum cost effectiveness in production environments. A new Quick-Pulse Programming™ Algorithm is employed which can speed up programming by as much as one hundred times.

The 87C64 incorporates an address latch on the address pins to minimize chip count in multiplexed bus systems. Designers can eliminate an external address latch by tieing address and data pins of the 87C64 directy to the processor's multiplexed address/data pins. On the falling edge of the ALE input (ALE/$\overline{CE}$), address information at the address inputs (A_0–A_{12}) of the 87C64 is latched internally. The address inputs are then ignored as data information is passed on the same bus.

The highest degree of protection against latch-up is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins from -1V to V_{CC} + 1V.

*HMOS and CHMOS are patented processes of Intel Corporation.



290000-1

Shaded Areas represent the 87C64 version

Figure 1. Block Diagram

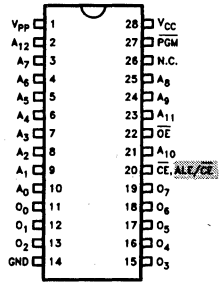
February 1988
Order Number: 290000-007

Pin Names

A_0-A_{12}	ADDRESSES
O_0-O_7	OUTPUTS
$\overline{OE}$	OUTPUT ENABLE
$\overline{CE}$	CHIP ENABLE
ALE/$\overline{CE}$	ADDRESS LATCH ENABLE /CHIP ENABLE
$\overline{PGM}$	PROGRAM STROBE
N.C.	NO CONNECT
D.U.	DON'T USE

27256	27128	2732A	2716
V_{PP}	V_{PP}		
A_{12}	A_{12}		
A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2
Gnd	Gnd	Gnd	Gnd

27C64/87C64
P27C64/P87C64

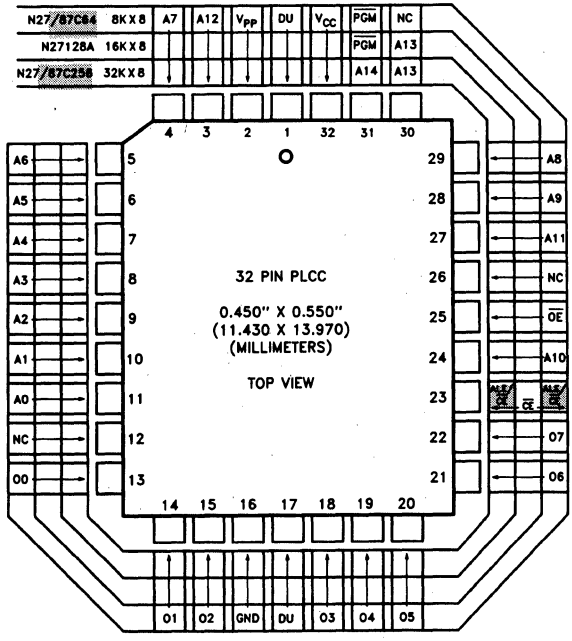


290000-2

2716	2732A	27128	27256
		V_{CC}/$\overline{PGM}$	V_{CC}
			A_{14}
V_{CC}	V_{CC}	A_{13}	A_{13}
A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$
A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3

NOTE:
Intel "Universal Site" Compatible EPROM Pin Configurations are shown in the adjacent blocks to 27C64 Pins.
Shaded Areas represent the 87C64 version

Figure 2. Pin Configuration



290000-11

Figure 3. PLCC(N) Lead Configuration

Extended Temperature (Express) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications.

EXPRESS EPROM products are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are available along with automotive temperature range (−40°C to +125°C) products. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

EXPRESS EPROM Product Family

PRODUCT DEFINITIONS

Type	Operating Temperature (°C)	Burn-in 125°C (hr)
Q	0 to +70	168 ±8
T	−40 to +85	NONE
L	−40 to +85	168 ±8
A	−40 to +125	NONE
B	−40 to +125	168 ±8

EXPRESS Options

27C64/87C64 Versions

Packaging Options			
Speed Versions	**Cerdip**	**PLCC**	**Plastic DIP**
−1	T, L, Q	T	T
−15	T, L, Q	T	T
−2	T, L, Q, A, B	T, A	T, A
−20	T, L, Q, A	T	T
−STD	T, L, Q, A, B	T, A	T, A
−25	T, L, Q, A	T	T
−3	T, L, Q, A, B	T, A	T, A
−30	T, L, Q, A	T	T

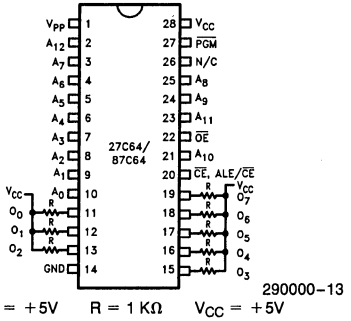
READ OPERATION

D.C. CHARACTERISTICS

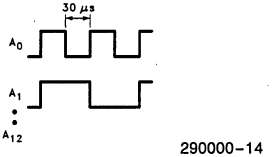
Electrical Parameters of EXPRESS EPROM products are identical to standard EPROM parameters except for:

Symbol	Parameter		27C64 87C64 Min	27C64 87C64 Max	Test Conditions
I_{SB}	V_{CC} Standby Current (mA)	CMOS		0.1	$\overline{CE} = V_{CC}$, $\overline{OE} = V_{IL}$
		TTL		1.0	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC_1}(1)	V_{CC} Active Current (mA)	TTL		20, 30	$\overline{OE} = \overline{CE} = V_{IL}$
	V_{CC} Active Current at High Temperature	TTL		20, 30	$\overline{OE} = \overline{CE} = V_{IL}$ $V_{PP} = V_{CC}$, $T_{ambient} = 85°C$

NOTE:
1. See notes 4 and 6 of Read Operation D.C. Characteristics.



$\overline{OE}$ = +5V R = 1 KΩ V_{CC} = +5V
V_{PP} = +5V GND = 0V $\overline{CE}$ = 33.3 KHz
$\overline{PGM}$ = +5V



Binary Sequence from A_0 to A_{12}

Burn-In Bias and Timing Diagrams

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature
During Read ..... 0°C to +70°C(2)

Temperature Under Bias ..... −10°C to +80°C

Storage Temperature ..... −65°C to +150°C

Voltage on Any Pin with
Respect to Ground ..... −2.0V to 7V(1)

Voltage on Pin A_9 with
Respect to Ground ..... −2.0V to +13.5V(1)

V_{PP} Supply Voltage with Respect to Ground
During Programming ..... −2.0V to +14V(1)

V_{CC} Supply Voltage with
Respect to Ground ..... −2.0V to +7.0V(1)

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

READ OPERATION D.C. CHARACTERISTICS $0°C \le T_A \le +70°C$

Symbol	Parameter		Notes	Min	Typ(3)	Max	Unit	Test Condition
I_{LI}	Input Leakage Current				0.01	1.0	µA	V_{IN} = 0V to 5.5V
I_{LO}	Output Leakage Current					±10	µA	V_{OUT} = 0V to 5.5V
I_{PP_1}	V_{PP} Current Read		6			100	µA	$V_{PP} = V_{CC}$
I_{SB}	V_{CC} Current Standby with Inputs—	CMOS	5			100	µA	$\overline{CE} = V_{CC}$
		TTL	4			1.0	mA	$\overline{CE} = V_{IH}$
I_{CC_1}	V_{CC} Current Active		4, 6			20, 30	mA	$\overline{CE} = V_{IL}$, f = 5 MHz, I_{OUT} = 0 mA
V_{IL}	Input Low Voltage (±10% Supply) (TTL)			−0.5		0.8	V	$V_{PP} = V_{CC}$
	Input Low Voltage (CMOS)			−0.2		0.2		
V_{IH}	Input High Voltage(±10% Supply) (TTL)			2.0		V_{CC}+0.5	V	$V_{PP} = V_{CC}$
	Input High Voltage (CMOS)			V_{CC}−0.2		V_{CC}+0.2		
V_{OL}	Output Low Voltage					0.45	V	I_{OL}=2.1 mA
V_{OH}	Output High Voltage			3.5			V	I_{OH}= −2.5 mA
I_{OS}	Output Short Circuit Current		7			100	mA	
V_{PP}	V_{PP} Read Voltage		8	V_{CC}−0.7		V_{CC}	V	

NOTES:
1. Minimum D.C. input voltage is −0.5V. During transitions, the inputs may undershoot to −2.0V for periods less than 20 ns. Maximum D.C. Voltage on output pins is V_{CC} + 0.5V which may overshoot to V_{CC} + 2V for periods less than 20 ns.
2. Operating temperature is for commercial product defined by this specification. Extended temperature options are available in EXPRESS and Military version.
3. Typical limits are at V_{CC} = 5V, T_A = +25°C.
4. 20 mA for STD and −3 versions; 30 mA for −2 and 150 ns versions.
V_{IL}, V_{IH} levels at TTL inputs.
5. ALE/$\overline{CE}$ or $\overline{CE}$ is V_{CC} ± 0.2V. All other inputs can have any value within spec.
6. Maximum Active power usage is the sum I_{PP} + I_{CC}. The maximum current value is with Outputs O_0 to O_7 unloaded.
7. Output shorted for no more than one second. No more than one output shorted at a time. I_{OS} is sampled but not 100% tested.
8. V_{PP} may be one diode voltage drop below V_{CC}. It may be connected directly to V_{CC}.

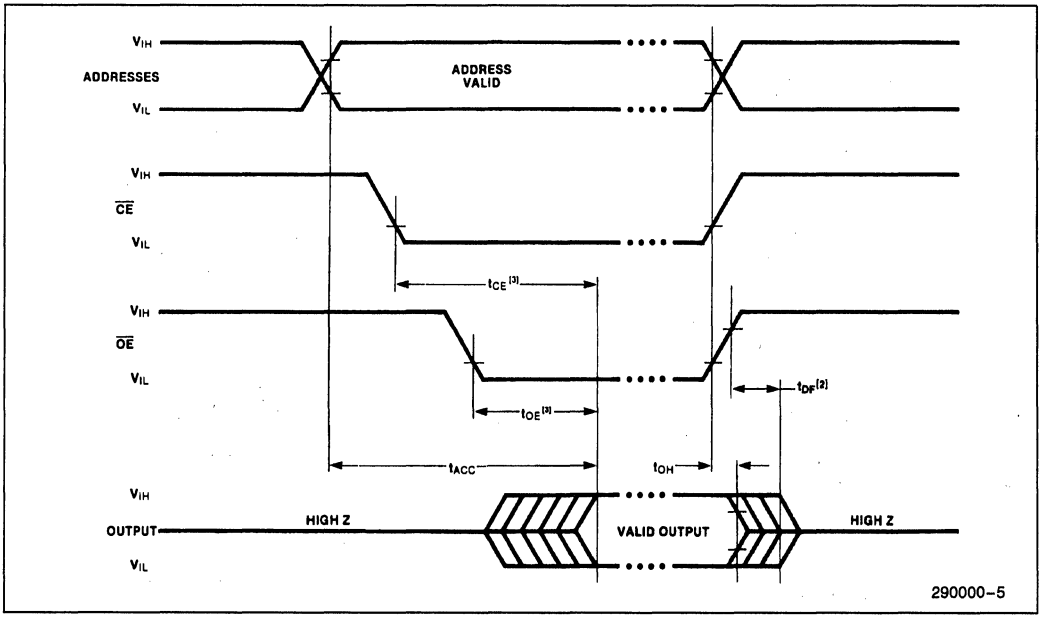
READ OPERATION

A.C. CHARACTERISTICS 27C64(1) $0°C \leq T_A \leq +70°C$

Versions (3)		$V_{CC} \pm 5\%$		27C64-1 N27C64-1 P27C64-1		27C64-2 N27C64-2 P27C64-2		27C64 N27C64		27C64-3 N27C64-3		Unit
		$V_{CC} \pm 10\%$		27C64-15 N27C64-15 P27C64-15		27C64-20 N27C64-20 P27C64-20		27C64-25 N27C64-25		27C64-30 N27C64-30		
Symbol	**Characteristic**			**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{ACC}	Address to Output Delay				150		200		250		300	ns
t_{CE}	$\overline{CE}$ to Output Delay				150		200		250		300	ns
t_{OE}	$\overline{OE}$ to Output Delay				75		75		100		120	ns
t_{DF}(2)	$\overline{OE}$ High to Output High Z				35		55		60		105	ns
t_{OH}(2)	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change-Whichever is First			0		0		0		0		ns

NOTES:
1. A.C. characteristics tested at V_{IH} = 2.4V and V_{IL} = 0.45V.
   Timing measurements made at V_{OL} = 0.8V and V_{OH} = 2.0V.
2. Guaranteed and sampled.
3. Model Number Prefixes: No prefix = Cerdip; P = Plastic DIP; N = PLCC.

A.C. WAVEFORMS 27C64



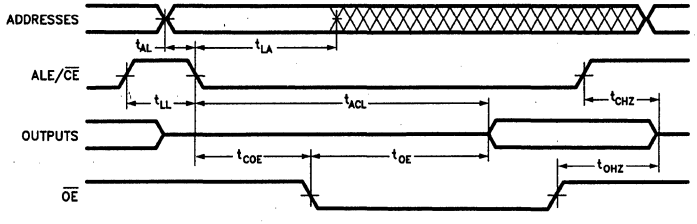
NOTES:
1. Typical values are for T_A = 25°C and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested.
3. $\overline{OE}$ may be delayed up to $t_{CE} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.

A.C. CHARACTERISTICS 87C64(1) $0°C \le T_A \le +70°C$

Versions (3)		$V_{CC} \pm 5\%$		87C64-1 N87C64-1 P87C64-1		87C64-2 N87C64-2 P87C64-2		87C64 N87C64		87C64-3 N87C64-3		Unit
		$V_{CC} \pm 10\%$		87C64-15 N87C64-15 P87C64-15		87C64-20 N87C64-20 P87C64-20		87C64-25 N87C64-25		87C64-30 N87C64-30		
Symbol	**Parameter**			**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	
t_{LL}	Chip Deselect Width			50		50		60		75		ns
t_{AL}	Address to $\overline{CE}$-Latch Set-up			7		20		25		30		ns
t_{LA}	Address Hold from $\overline{CE}$-LATCH			30		45		50		60		ns
t_{ACL}	CE-Latch Access Time				150		200		250		300	ns
t_{OE}	Output Enable to Output Valid				75		75		100		120	ns
t_{COE}	ALE/$\overline{CE}$ to Output Enable			30		45		50		60		ns
t_{CHZ}(2)	Chip Deselect to Output in High Z				45		50		60		75	ns
t_{OHZ}(2)	Output Disable to Output in High Z				35		50		60		75	ns

NOTES:
1. A.C. characteristics tested at V_{IH} = 2.4V and V_{IL} = 0.45V. Timing measurements made at V_{OL} = 0.8V and V_{OH} = 2.0V.
2. Guaranteed and sampled.
3. Model Number Prefixes: No prefix = Cerdip; P = Plastic DIP; N = PLCC.

A.C. WAVEFORMS 87C64



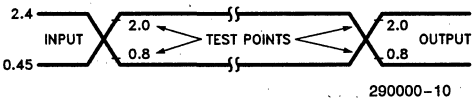
290000-6

CAPACITANCE(1) T_A = 25°C, f = 1.0 MHz

Symbol	Parameter	Max	Unit	Conditions
C_{IN}	Address/Control Capacitance	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	12	pF	V_{OUT} = 0V

NOTE:
1. Sampled. Not 100% tested.

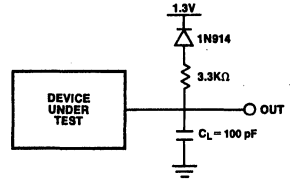
A.C. TESTING INPUT/OUTPUT WAVEFORM



290000-10

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT



290000-3

C_L = 100 pF
C_L Includes Jig Capacitance

DEVICE OPERATION

The modes of operation of the 27C64/87C64 are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for V_{PP} and 12V on A_9 for int$_e$ligent Identifier mode.

Table 1. Mode Selection for 27C64 and 87C64

Mode \ Pins	ALE/$\overline{CE}$ $\overline{CE}$	$\overline{OE}$	$\overline{PGM}$ (7)	A_9	A_0	V_{PP} (7)	V_{CC}	Outputs
Read	V_{IL}	V_{IL}	V_{IH}	X(1)	X	V_{CC}	5.0V	D_{OUT}
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	X	V_{CC}	5.0V	High Z
Standby	V_{IH}	X	X	X	X	V_{CC}	5.0V	High Z
Programming	V_{IL}	V_{IH}	V_{IL}	X	X	(4)	(4)	D_{IN}
Program Verify	V_{IL}	V_{IL}	V_{IH}	X	X	(4)	(4)	D_{OUT}
Program Inhibit	V_{IH}	X	X	X	X	(4)	(4)	HIGH Z
int$_e$ligent Identifier(3) -Manufacturer	V_{IL}	V_{IL}	V_{IH}	V_H(2)	V_{IL}	V_{CC}	V_{CC}	89 H (6) 88 H (6)
int$_e$ligent Identifier(3) -27C64	V_{IL}	V_{IL}	V_{IH}	V_H(2)	V_{IH}	V_{CC}	V_{CC}	07 H
int$_e$ligent Identifier(3, 5) -87C64	V_{IL}	V_{IL}	V_{IH}	V_H(2)	V_{IH}	V_{CC}	V_{CC}	37 H

NOTES:
1. X can be V_{IL} or V_{IH}.
2. V_H = 12.0V ± 0.5V.
3. A_1–A_8, A_{10-12} = V_{IL}.
4. See Table 2 for V_{CC} and V_{PP} voltages.
5. ALE/$\overline{CE}$ has to be toggled in order to latch in the addresses and read the signature codes.
6. The Manufacturer's identifier reads 89H for Cerdip devices; 88H for Plastic DIP and PLCC devices.
7. In Read Mode tie $\overline{PGM}$ and V_{PP} to V_{CC}.

Read Mode: 27C64

The 27C64 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after a delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least t_{ACC}–t_{OE}.

Read Mode: 87C64

The 87C64 was designed to reduce the hardware interface requirements when incorporated in processor systems with multiplexed address-data busses. Chip count (and therefore power and board space) can be minimized when the 87C64 is designed as shown in Figure 4. The processor's multiplexed bus (AD_{0-7}) is tied to both address and data pins of the 87C64. All address inputs of the 87C64 are latched when ALE/$\overline{CE}$ is brought low, thus eliminating the need for a separate address latch.

The 87C64 internal address latch is directly enabled through the use of the ALE/$\overline{CE}$ line. As the transition occurs on the ALE/$\overline{CE}$ from the TTL high to the low state, the last address presented at the address pins is retained. Data is then enabled onto the bus from the EPROM by the $\overline{OE}$ pin.

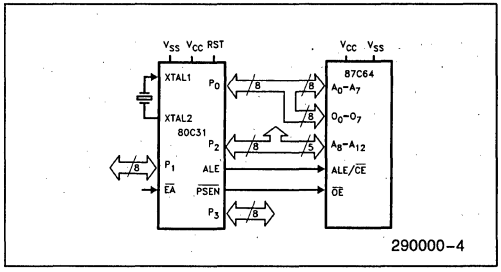


Figure 4. 80C31 with 87C64 System Configuration

Standby Mode

The 27C64 and 87C64 have Standby modes which reduce the maximum V_{CC} current to 100 μA. Both are placed in the Standby mode when $\overline{CE}$ or ALE/$\overline{CE}$ are in the CMOS-high state. When in the Standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Two Line Output Control

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (or ALE/$\overline{CE}$) should be decoded and used as the primary device selecting function, while $\overline{OE}$ should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

SYSTEM CONSIDERATIONS

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient and inductive current peaks is dependent on the output capacitive and inductive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor for low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effect of PC board-traces.

PROGRAMMING MODES

Caution: Exceeding 14V on V_{PP} will permanently damage the device.

Initially, and after each erasure, all bits of the EPROM are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The device is in the programming mode when V_{PP} is raised to its programming voltage (See Table 2) and $\overline{CE}$ (or ALE/$\overline{CE}$) and $\overline{PGM}$ are both at TTL low and $\overline{OE}$ = V_{IH}. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

Program Inhibit

Programming of multiple EPROMS in parallel with different data is easily accomplished by using the Program Inhibit mode. A high-level $\overline{CE}$ (or ALE/$\overline{CE}$) or $\overline{PGM}$ input inhibits the other devices from being programmed.

Except for $\overline{CE}$ (or ALE/$\overline{CE}$), all like inputs (including $\overline{OE}$) of the parallel EPROMs may be common. A TTL low-level pulse applied to the $\overline{PGM}$ input with V_{PP} at its programming voltage and $\overline{CE}$ (or ALE/$\overline{CE}$) = V_{IL} will program the selected device.

Program Verify

A verify (read) should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with $\overline{OE}$ and $\overline{CE}$ (or ALE/$\overline{CE}$) at V_{IL}, $\overline{PGM}$ at V_{IH}, and V_{CC} and V_{PP} at their programming voltages. Data should be verified a minimum of t_{OE} after the falling edge of $\overline{OE}$.

int$_e$ligent Identifier™ Mode

The int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ±5°C ambient temperature range that is required when programming the device.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 of the EPROM. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during the int$_e$ligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. These two identifier bytes are given in Table 1. ALE/$\overline{CE}$ of the 87C64 has to be toggled in order to latch in the addresses and read the Signature Codes.

ERASURE CHARACTERISTICS (FOR CERDIP EPROMS)

The erasure characteristics are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the EPROM in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the device is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 Wsec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The EPROM should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose an EPROM can be exposed to without damage is 7258 Wsec/cm^2 (1 week @ 12000 μW/cm^2). Exposure of the device to high intensity UV light for longer periods may cause permanent damage.

CHMOS NOISE CHARACTERISTICS

Special EPI processing techniques have enabled Intel to build CHMOS with features adding to system reliability. These include input/output protection to latch-up. Each of the data and address pins will not latch-up with currents up to 100 mA and voltages from −1V to V_{CC} + 1V.

Additionally, the V_{PP} (programming) pin is designed to resist latch-up to the 14V maximum device limit.

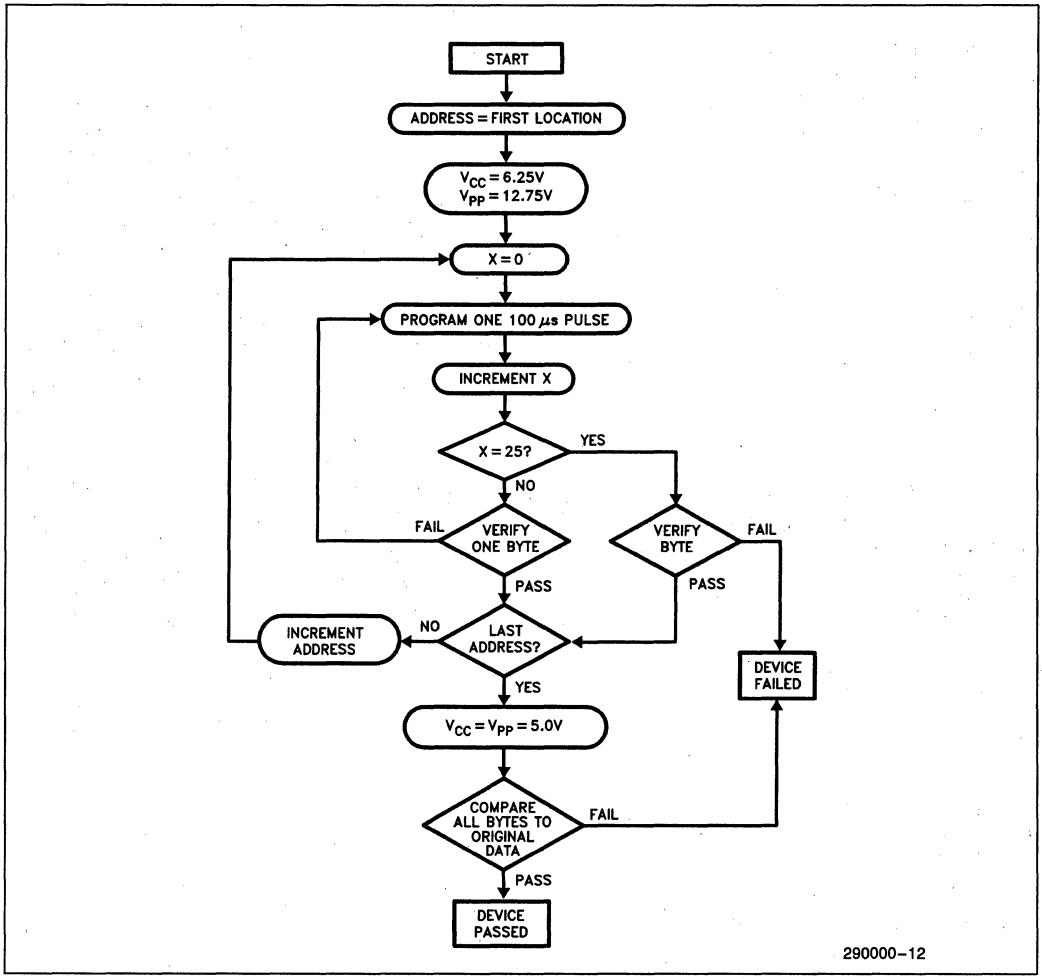


Figure 5. Quick-Pulse Programming™ Algorithm

Quick-Pulse Programming™ Algorithm

Intel's 27C64 and 87C64 EPROMs can now be programmed using the Quick-Pulse Programming Algorithm, developed by Intel to substantially reduce the throughput time in the production environment. This algorithm allows these devices to be programmed in under one second, almost a hundred fold improvement over previous algorithms. Actual programming time is a function of the PROM programmer being used.

The Quick-Pulse Programming Algorithm uses initial pulses of 100 microseconds followed by a byte verification to determine when the address byte has been successfully programmed. Up to 25 100 µs pulses per byte are provided before a failure is recognized. A flowchart of the Quick-Pulse Programming Algorithm is shown in Figure 5.

For the Quick Pulse Programming Algorithm, the entire sequence of programming pulses and byte verifications is performed at V_{CC} = 6.25V and V_{PP} at 12.75V. When programming of the EPROM has been completed, all bytes should be compared to the original data with V_{CC} = V_{PP} = 5.0V.

D.C. PROGRAMMING CHARACTERISTICS (27C64/87C64) T_A = 25°C ±5°C

Table 2

Symbol	Parameter	Limits			Test Conditions (Note 1)
		Min	Max	Unit	
I_{LI}	Input Current (All Inputs)		1.0	µA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.1	0.8	V	
V_{IH}	Input High Level	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage During Verify		0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage During Verify	3.5		V	I_{OH} = −2.5 mA
I_{CC2}(3)	V_{CC} Supply Current		30	mA	
I_{PP2}(3)	V_{PP} Supply Current (Program)		30	mA	$\overline{CE} = V_{IL}$
V_{ID}	A_9 intelligent Identifier Voltage	11.5	12.5	V	
V_{PP}	Programming Voltage	12.5	13.0	V	
V_{CC}	Supply Voltage During Programming	6.0	6.5	V	

A.C. PROGRAMMING CHARACTERISTICS 27C64

T_A = 25°C ±5°C, See Table 2 for V_{CC} and V_{PP} Voltages

Symbol	Parameter	Limits				Conditions (Note 1)
		Min	Typ	Max	Unit	
t_{AS}	Address Setup Time	2			µs	
t_{OES}	$\overline{OE}$ Setup Time	2			µs	
t_{DS}	Data Setup Time	2			µs	
t_{AH}	Address Hold Time	0			µs	
t_{DH}	Data Hold Time	2			µs	
t_{DFP}	$\overline{OE}$ High to Output Float Delay	0		130	ns	(Note 2)
t_{VPS}	V_{PP} Setup Time	2			µs	
t_{VCS}	V_{CC} Setup Time	2			µs	
t_{CES}	$\overline{CE}$ Setup Time	2			µs	
t_{PW}	$\overline{PGM}$ Program Pulse Width	95	100	105	µs	Quick-Pulse
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%) ..... 20 ns

Input Pulse Levels ..... 0.45V to 2.4V

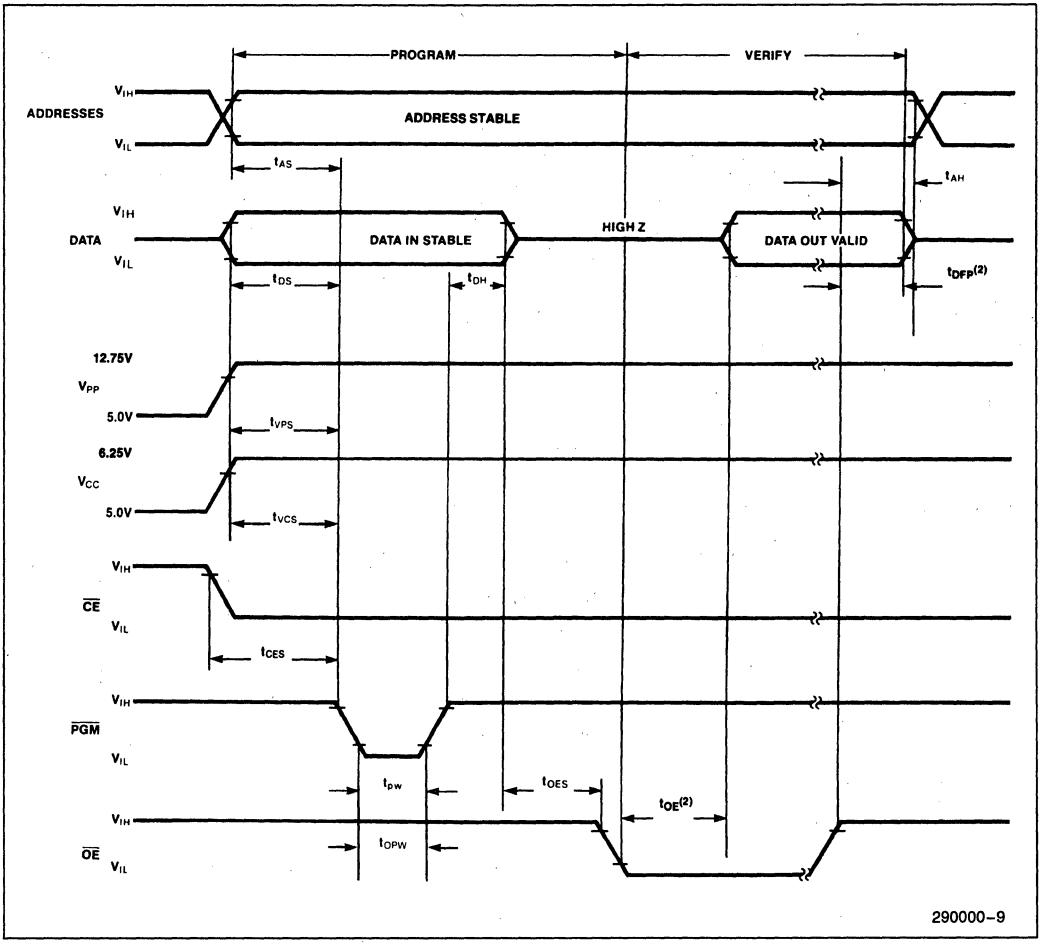
Input Timing Reference Level ..... 0.8V and 2.0V

Output Timing Reference Level ..... 0.8V and 3.5V

NOTES:

1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram.
3. The maximum current value is with outputs O_0 to O_7 Unloaded.

PROGRAMMING WAVEFORMS 27C64



NOTES:
1. The Input Timing Reference Level is 0.8V for V_{IL} and 2V for a V_{IH}.
2. t_{OE} and t_{DFP} are characteristics of the device but must be accommodated by the programmer.
3. When programming the 27C64, a 0.1 μF capacitor is required across V_{PP} and ground to suppress spurious voltage transients which can damage the device.

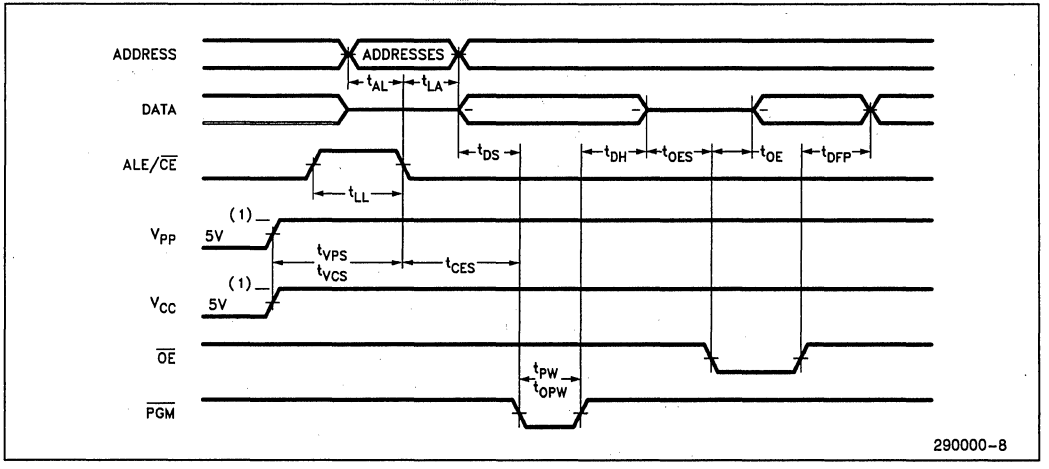
A.C. PROGRAMMING CHARACTERISTICS 87C64

T_A = 25°C ±5°C, See Table 2 for V_{CC} and V_{PP} Voltages.

Symbol	Parameter	Limits			Unit	Conditions
		Min	Typ	Max		
t_{VPS}	V_{PP} Setup Time	2			µs	
t_{VCS}	V_{CC} Setup Time	2			µs	
t_{LL}	Chip Deselect Width	2			µs	
t_{AL}	Address to Chip Select Setup	1			µs	
t_{LA}	Address Hold from Chip Select	1			µs	
t_{PW}	$\overline{PGM}$ Pulse Width	95	100	105	µs	Quick-Pulse
t_{DS}	Data Setup Time	2			µs	
t_{DFP}	$\overline{OE}$ High to Data Float	0		130	ns	
t_{OES}	Output Enable Setup Time	2			µs	
t_{OE}	Data Valid from Output Enable			150	ns	
t_{DH}	Data Hold Time	2			µs	
t_{CES}	$\overline{CE}$ Setup Time	2			µs	

NOTE:
1. Programming tolerances and test conditions are the same as 27C64.

PROGRAMMING WAVEFORMS 87C64



NOTE:
1. 12.75V V_{PP} & 6.25V V_{CC} for Quick-Pulse Programming Algorithm.

87C257
256K (32K x 8) CHMOS UV ERASABLE PROM

- **CHMOS/NMOS Microcontroller and Microprocessor Compatible**
  - **87C257-Integrated Address Latch**
  - **Universal 28 Pin Memory Site, 2-line Control**
- **Low Power Consumption**
- **High Performance Speeds**
  - **170 ns Maximum Access Time**
- **Noise Immunity Features**
  - **± 10% V_{CC} Tolerance**
  - **Maximum Latch-up Immunity Through EPI Processing**
- **New Quick-Pulse Programming™ Algorithm**
  - **4 Second Programming**
- **28-Pin Cerdip and 32-Lead PLCC Packages**

(See Packaging Spec., Order #231369)

Intel's 87C257 CHMOS EPROM is a 256K-bit 5V-only memory organized as 32,768 8-bit words. It employs advanced CHMOS*II-E circuitry for systems requiring low power, high speed performance, and noise immunity. The 87C257 is optimized for compatibility with multiplexed address/data bus microcontrollers such as Intel's 16 MHz 8051- and 8096- families.

The 87C257 incorporates latches on all address inputs to minimize chip count, reduce cost, and simplify design of multiplexed bus systems. The 87C257's internal address latch allows address and data pins to be tied directly to the processor's multiplexed address/data pins. Address information (inputs A_0–A_{14}) is latched early in the memory-fetch cycle by the falling edge of the ALE input. Subsequent address information is ignored while ALE remains low. The EPROM can then pass data (from pins O_0–O_7) on the same bus during the last part of the memory-fetch cycle.

The 87C257 is offered in ceramic DIP and Plastic Leaded Chip Carrier (PLCC) packages. The Cerdip package provides flexibility in prototyping and R&D environments while the PLCC version is used in surface mount and automated manufacturing. The 87C257 employs the Quick-Pulse Programming™ Algorithm for fast and reliable programming.

Intel's EPI processing achieves the highest degree of latch-up protection. Address and data pin latch-up prevention is provided for stresses up to 100 mA from $-1V$ to $V_{CC} + 1V$.

*HMOS and CHMOS are patented processes of Intel Corporation.

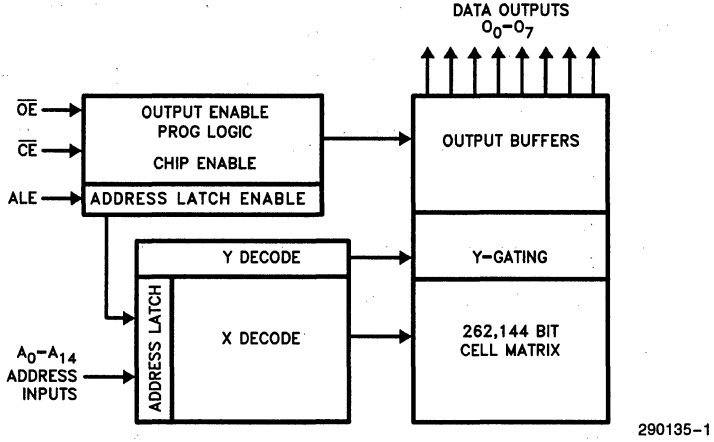


Figure 1. Block Diagram

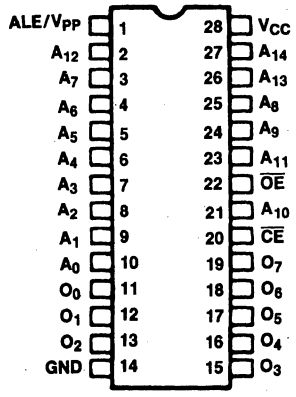
October 1988
Order Number: 290135-005

Pin Names

A_0–A_{14}	ADDRESSES
O_0–O_7	OUTPUTS
$\overline{OE}$	OUTPUT ENABLE
$\overline{CE}$	CHIP ENABLE
ALE/V_{PP}	Address Latch Enable/V_{PP}
N.C.	NO CONNECT
D.U.	Don't Use

87C64
V_{PP}
A_{12}
A_7
A_6
A_5
A_4
A_3
A_2
A_1
A_0
O_0
O_1
O_2
Gnd

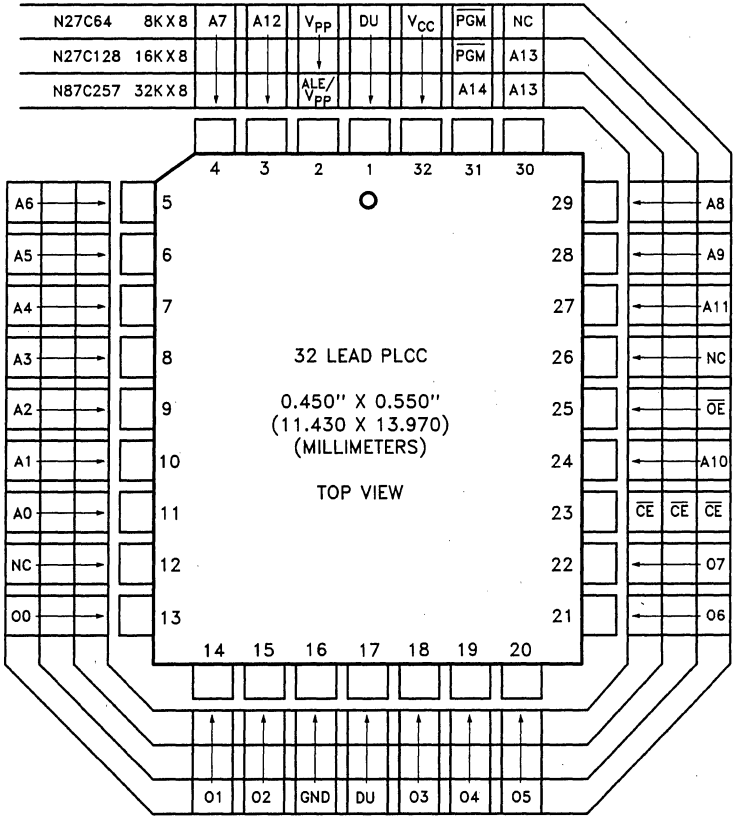
87C257



87C64
V_{CC}
$\overline{PGM}$
N.C
A_8
A_9
A_{11}
$\overline{OE}$
A_{10}
ALE/$\overline{CE}$
O_7
O_6
O_5
O_4
O_3

290135-2

FIGURE 2. DIP Pin Configuration



290135-13

Figure 3. PLCC Lead Configuration

NOTE:
Intel "Universal Site"-Compatible EPROM Pin Configurations are Shown in the Blocks Adjacent.

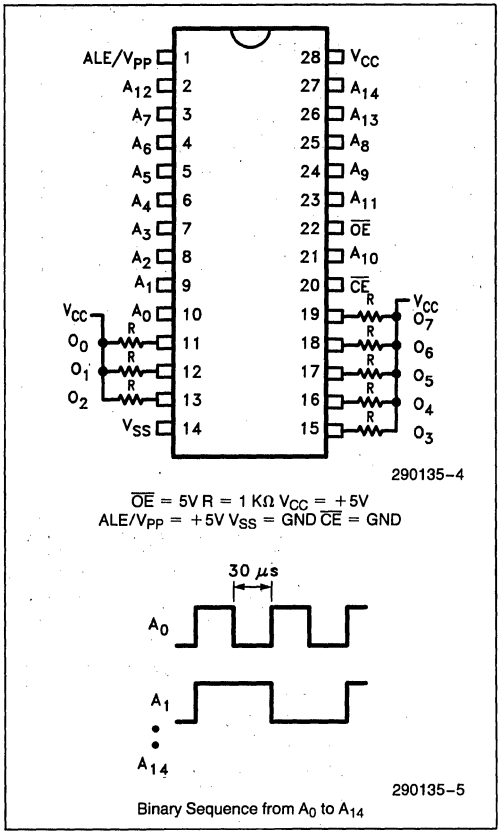
EXTENDED TEMPERATURE (EXPRESS) EPROMs

The Intel EXPRESS EPROM family receives additional processing to enhance product characteristics. EXPRESS processing is available for several EPROM densities allowing the appropriate memory size to match system applications. EXPRESS EPROMs are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process meets or exceeds most industry burn-in specifications. The standard EXPRESS EPROM operating temperature range is 0°C to +70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS and automotive temperature range (−40°C to +125°C) products are also available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This allows reduction or elimination of incoming testing.

AUTOMOTIVE AND EXPRESS OPTIONS

Versions

Speed Versions	Packaging Options	
	Cerdip	PLCC
−200V05	A	A
−200V10	A, L	A, T
−250V10	A, L	A, T
−250V05	A	A



Burn-In Bias and Timing Diagrams

AUTOMOTIVE AND EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITIONS

Type	Operating Temperature (°C)	Burn-in 125°C (hr)
T	−40°C to +85°C	NONE
L	−40°C to +85°C	168 ±8
A	−40°C to +125°C	NONE

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature During Read ..... 0°C to + 70°C(2)
Temperature Under Bias ..... −10°C to +80°C(2)
Storage Temperature ..... −65°C to +150°C
Voltage on any Pin with Respect to Ground ..... −2V to +7V(1)
Voltage on A_9 with Respect to Ground ..... −2V to +13.5V(1)
V_{PP} Supply Voltage with Respect to Ground During Programming ..... −2V to +14.0V(1)
V_{CC} Supply Voltage with Respect to Ground ..... −2V to +7.0V(1)

Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

NOTICE: Specifications contained within the following tables are subject to change.

READ OPERATION

D.C. CHARACTERISTICS TTL and NMOS Inputs

Symbol	Parameter		Notes	Min	Typ(3)	Max	Units	Test Condition
I_{LI}	Input Load Current				0.01	1.0	µA	V_{IN} = 0V to 5.5V
I_{LO}	Output Leakage Current					±10	µA	V_{OUT} = 0V to 5.5V
I_{SB}	V_{CC} Current Standby with Inputs—	Switching				10	mA	$\overline{CE}$ = ALE = V_{IH}
		Stable				1.0	mA	$\overline{CE}$ = V_{IH}, ALE = V_{IL}
I_{CC_1}	V_{CC} Current Active		5			30	mA	$\overline{CE}$ = V_{IL}, ALE = V_{IH} f = 5 MHz, I_{OUT} = 0 mA
V_{IL}	Input Low Voltage (±10% Supply)		1	−0.5		0.8	V	
V_{IH}	Input High Voltage (±10% Supply)			2.0		V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage					0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage			2.4			V	I_{OH} = −400 µA
I_{OS}	Output Short Circuit Current		6			100	mA	

D.C. CHARACTERISTICS CMOS Inputs

Symbol	Parameter		Notes	Min	Typ(3)	Max	Units	Test Condition
I_{LI}	Input Load Current				0.01	1.0	µA	V_{IN} = 0V to 5.5V
I_{LO}	Output Leakage Current					±10	µA	V_{OUT} = 0V to 5.5V
I_{SB}	V_{CC} Current Standby with Inputs—	Switching	4			6	mA	$\overline{CE}$ = ALE = V_{CC}
		Stable				100	µA	$\overline{CE}$ = V_{CC}, ALE = GND
I_{CC_1}	V_{CC} Current Active		5			15	mA	$\overline{CE}$ = V_{IL}, ALE = V_{IH} f = 5 MHz, I_{OUT} = 0 mA
V_{IL}	Input Low Voltage (±10% Supply)			−0.2		0.8	V	
V_{IH}	Input High Voltage (±10% Supply)			0.7 V_{CC}		V_{CC} + 0.2	V	
V_{OL}	Output Low Voltage					0.4	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage			V_{CC} − 0.8			V	I_{OH} = −2.5 mA
I_{OS}	Output Short Circuit Current		6			100	mA	

NOTES:
1. Minimum D.C. input voltage is −0.5V. During transitions, the inputs may undershoot to −2.0V for periods less than 20 ns. Maximum D.C. voltage on output pins is V_{CC} + 0.5V which may overshoot to V_{CC} + 2V for periods less than 20 ns.
2. Operating temperature is for commercial product defined by this specification. Extended temperature options are available in EXPRESS and Automotive versions.
3. Typical limits are at V_{CC} = 5V, T_A = +25°C.
4. $\overline{CE}$ is V_{CC} ±0.2V. All other inputs can have any value within spec.
5. Maximum current value is with outputs O_0 to O_7 unloaded.
6. Output shorted for no more than one second. No more than one output shorted at a time. I_{OS} is sampled but not 100% tested.

READ OPERATION

A.C. CHARACTERISTICS(1) $0°C \le T_A \le +70°C$

Versions(3)	$V_{CC} \pm 5\%$		N87C257-200V05				Units
	$V_{CC} \pm 10\%$	87C257-170V10	87C257-200V10 N87C257-200V10		87C257-250V10 N87C257-250V10		

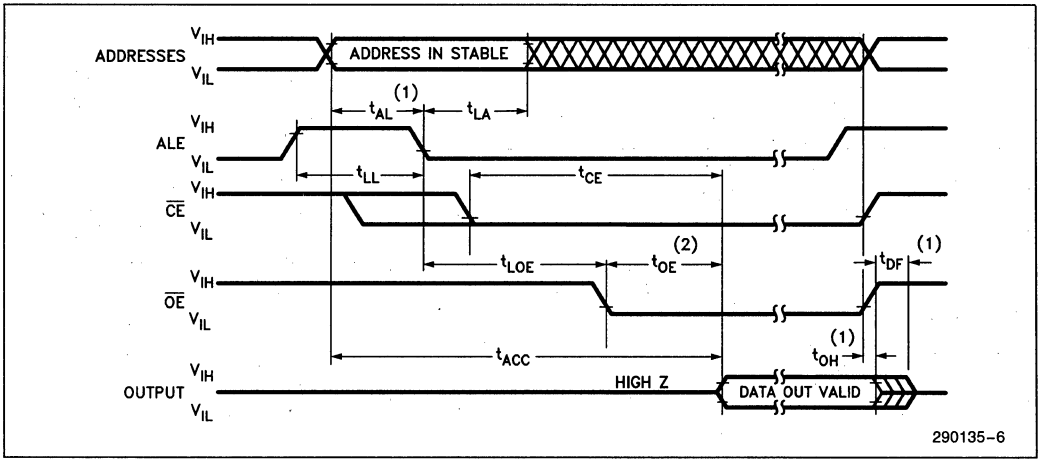
Symbol	Characteristic	Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay		170		200		250	ns
t_{CE}	$\overline{CE}$ to Output Delay		170		200		250	ns
t_{OE}	$\overline{OE}$ to Output Delay		58		75		100	ns
t_{DF}(2)	$\overline{OE}$ High to Output High Z		35		40		55	ns
t_{OH}(2)	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Change-Whichever is First	0		0		0		ns
t_{LL}	Latch Deselect Width	35		50		60		ns
t_{AL}(2)	Address to Latch Set-Up	7		15		25		ns
t_{LA}	Address Hold from LATCH	23		30		40		ns
t_{LOE}	ALE to Output Enable	23		30		40		ns

NOTES:
1. See A.C. Testing Input/Output Waveforms for timing measurements.
2. Guaranteed and sampled.
3. Model Number Prefixes: No Prefix = CERDIP.

A.C. CONDITIONS OF TEST

Input Rise and Fall Times (10% to 90%) ..... 10 ns
Input Pulse Levels ..... V_{OL} to V_{OH}
Input Timing Reference Level 1.5V
Output Timing Reference Level ..... V_{IL} and V_{IH}

A.C. WAVEFORMS



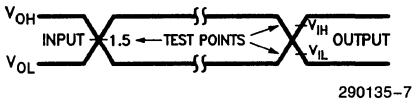
NOTES:
1. This parameter is only sampled and is not 100% tested.
2. $\overline{OE}$ may be delayed up to $t_{CE} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE}.

CAPACITANCE(1) T_A = 25°C, f = 1.0 MHz

Symbol	Parameter	Max	Units	Conditions
C_{IN}	Address/Control Capacitance	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	12	pF	V_{OUT} = 0V

NOTE:
1. Sampled. Not 100% tested.

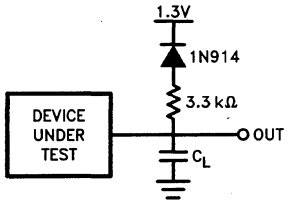
A.C. TESTING INPUT/OUTPUT WAVEFORM



290135-7

A.C. testing inputs are driven at V_{OH} for a Logic "1" and V_{OL} for a Logic "0". Timing measurements are made at V_{IH} for a Logic "1" and V_{IL} for a Logic "0".

A.C. TESTING LOAD CIRCUIT



290135-8

C_L = 100 pF
C_L Includes Jig Capacitance

DEVICE OPERATION

Table 1 lists 87C257 operating modes. Read mode requires a single 5V power supply. All input levels are TTL or CMOS except A9 in int$_e$ligent Identifier mode and V_{PP}.

Table 1. Mode Selection

Mode \ Pins	$\overline{CE}$	$\overline{OE}$	A_9	A_0	ALE/V_{PP}	V_{CC}	Outputs
Read	V_{IL}	V_{IL}	X(1)	X	X	5.0V	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	X	X	5.0V	High Z
Standby	V_{IH}	X	X	X	X	5.0V	High Z
Programming	V_{IL}	V_{IH}	X	X	(Note 4)	(Note 4)	D_{IN}
Program Verify	V_{IH}	V_{IL}	X	X	(Note 4)	(Note 4)	D_{OUT}
Optional Program Verify	V_{IL}	V_{IL}	X	X	V_{CC} (Note 4)	(Note 4)	D_{OUT}
Program Inhibit	V_{IH}	V_{IH}	X	X	(Note 4)	(Note 4)	High Z
int$_e$ligent Identifier(3) -Manufacturer	V_{IL}	V_{IL}	V_H(2)	V_{IL}	X	V_{CC}	89 H
int$_e$ligent Identifier(3) -87C257	V_{IL}	V_{IL}	V_H(2)	V_{IH}	X	V_{CC}	24 H

NOTES:
1. X can be V_{IL} or V_{IH}.
2. V_H = 12.0V ±0.5V.
3. A_1–A_8, A_{10-12} = V_{IL}, A_{13-14} = X.
4. See Table 2 for V_{CC} and V_{PP} programming voltages.

Read Mode

The 87C257 has two control functions; both must be logically active to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and the device-select. Output enable ($\overline{OE}$) gates data to the output pins by controlling the output buffer. When the address is stable (ALE = V_{IH}) or latched (ALE = V_{IL}), the address access time (t_{ACC}) equals the delay from $\overline{CE}$ to output (t_{CE}). Outputs display valid data t_{OE} after the falling edge of $\overline{OE}$, assuming t_{ACC} and t_{CE} times are met.

The 87C257 reduces the hardware interface in multiplexed address-data bus systems. Figure 4 shows a low power, small board space, minimal chip 87C257/microcontroller design. The processor's multiplexed bus (AD_{0-7}) is tied to the 87C257's address and data pins. No separate address latch is needed because the 87C257 latches all address inputs when ALE is low.

The ALE input controls the 87C257's internal address latch. As ALE transitions from V_{IH} to V_{IL}, the last address present at the address pins is retained. The $\overline{OE}$ control can then enable EPROM data onto the bus.

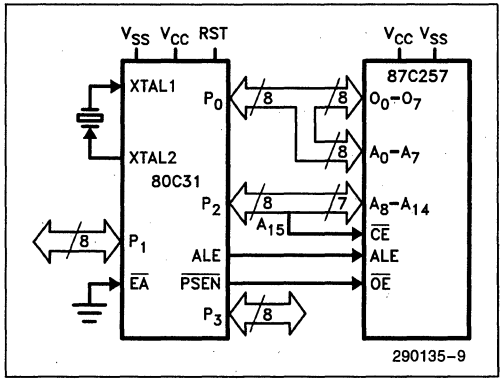


Figure 4. 80C31 with 87C257 System Configuration

Standby Mode

The standby mode substantially reduces V_{CC} current. When $\overline{CE}$ = V_{IH}, the standby mode places the outputs in a high impedance state, independent of the $\overline{OE}$ input.

Two Line Output Control

EPROMs are often used in larger memory arrays. Intel provides two contol inputs to accommodate multiple memory connections. Two-line control provides for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To efficiently use these two control inputs, an address decoder should enable $\overline{CE}$ while $\overline{OE}$ should be connected to all memory-array devices and the system's $\overline{READ}$ control line. This assures that only selected memory devices have active outputs while deselected memory devices are in low-power standby mode.

SYSTEM CONSIDERATIONS

EPROM power switching characteristics require careful device decoupling. System designers are interested in three supply current (ICC) issues—standby current levels, active current levels, and transient current peaks produced by falling and rising edges of Chip Enable. Transient current magnitudes depend on the device outputs' capacitive and inductive loading. Two-Line Control and proper decoupling capacitor selection will suppress transient voltage peaks. Each device should have a 0.1 μF ceramic capacitor connected between its V_{CC} and GND. This high frequency, low inherent-inductance capacitor should be placed as close as possible to the device. Additionally, for every eight devices, a 4.7 μF electrolytic capacitor should be placed between V_{CC} and GND at the array's power supply connection. The bulk capacitor will overcome voltage slumps caused by PC board trace inductances.

PROGRAMMING MODES

Caution: Exceeding 14V on V_{PP} will permanently damage the device.

Initially, and after each erasure, all EPROM bits are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" are programmed, the data word

can contain both "1s" and "0s". Ultraviolet light erasure is the only way to change "0s" to "1s".

The programming mode is entered when V_{PP} is raised to its programming voltage (see Table 2). Data is programmed by applying an 8-bit word to the output pins (O_{0-7}). Pulsing $\overline{CE}$ to TTL-low while $\overline{OE} = V_{IH}$ will program data. TTL levels are required for address and data inputs.

Program Inhibit

The Program Inhibit mode allows parallel programming of multiple EPROMs with different data. With V_{PP} at its programming voltage, a $\overline{CE}$-low pulse programs the desired EPROM. $\overline{CE}$-high inputs inhibit programming of non-targeted devices. Except for $\overline{CE}$ and $\overline{OE}$, parallel EPROMs may have common inputs.

Program Verify

With V_{PP} and V_{CC} at their programming voltages, a verify (read) determines that bits are correctly programmed. The verify is performed with $\overline{CE} = V_{IH}$ and $\overline{OE} = V_{IL}$. Valid data is available t_{OE} after $\overline{OE}$ falls low.

Optional Program Verify

The optional verify allows parallel programming and verification when several devices share a common bus. It is performed with $\overline{CE} = \overline{OE} = V_{IL}$ and $V_{PP} = V_{CC} = 6.25V$. The normal read mode is then used for program verify. Outputs will tri-state depending on $\overline{OE}$ and $\overline{CE}$.

int$_e$ligent Identifier™ Mode

The int$_e$ligent Identifier Mode will determine an EPROM's manufacturer and device type. Programming equipment can automatically match a device with its proper programming algorithm.

This mode is activated when programming equipment forces 12V ±0.5V on the EPROM's A_9 address line. With A_1-A_8, $A_{10}-A_{12} = V_{IL}$ (A_{13-14} are don't care), address line $A_0 = V_{IL}$ will present the manufacturer's code and $A_0 = V_{IH}$ the device code (see Table 1). When $A_9 = V_H$, ALE need not be toggled to latch each identifier address. This mode functions in the 25°C ±5°C ambient temperature range required during programming.

ERASURE CHARACTERISTICS (FOR CERDIP EPROMS)

Exposure to light of wavelength shorter than 4000 Angstroms (Å) begins EPROM erasure. Sunlight and some fluorescent lamps have wavelengths in the 3000–4000Å range. Constant exposure to room-level fluorescent light can erase an EPROM in about 3 years (about 1 week for direct sunlight). Opaque labels over the window will prevent unintentional erasure under these lighting conditions.

The recommended erasure procedure is exposure to 2537Å ultraviolet light. The minimum integrated dose (intensity x exposure time) is 15 Wsec/cm^2. Erasure time using a 12000 μW/cm^2 ultraviolet lamp is approximately 15 to 20 minutes. The EPROM should be placed about 1 inch from the lamp. The maximum integrated dose is 7258 Wsec/cm^2 (1 week @ 12000 μW/cm^2). High intensity UV light exposure for longer periods can cause permanent damage.

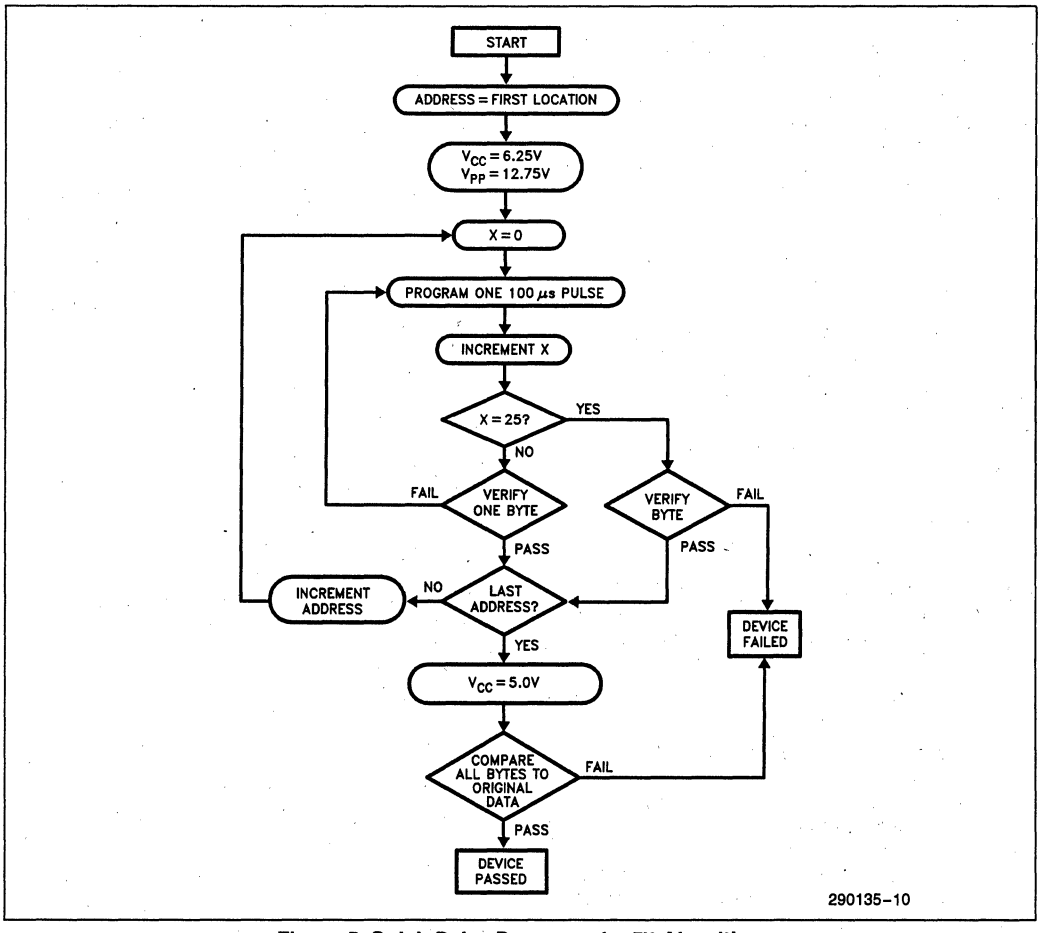


Figure 5. Quick-Pulse Programming™ Algorithm

CHMOS NOISE CHARACTERISTICS

System reliability is enhanced by Intel's CHMOS EPI-process techniques. Protection on each data and address pin prevents latch-up; even with 100 mA currents and voltages from $-1V$ to $V_{CC} + 1V$. Additionally, the V_{PP} pin is designed to resist latch-up to the 14V maximum device limit.

Quick-Pulse Programming™ Algorithm

The Quick-Pulse Programming algorithm programs Intel's 87C257 EPROM. Developed to substantially reduce production programming throughput time, this algorithm can program a 87C257 in under four seconds. Actual programming time depends on the PROM programmer used.

The Quick-Pulse Programming algorithm uses a 100 microsecond initial-pulse followed by a byte verification to determine when the addressed byte is correctly programmed. The algorithm terminates if 25 100µs pulses fail to program a byte. Figure 5 shows the Quick-Pulse Programming algorithm flowchart.

The entire program-pulse/byte-verify sequence is performed with $V_{CC} = 6.25V$ and $V_{PP} = 12.75V$. When programming is complete, all bytes should be compared to the original data with $V_{CC} = 5.0V$.

Alternate Programming

Intel's 27C256 and 27256 Quick-Pulse Programming algorithms will also program the 87C257. By overriding a check for the int$_e$ligent Identifier, older or non-upgraded PROM programmers can program the 87C257. See Intel's 27C256 and 27256 data sheets for programming waveforms of these alternate algorithms.

D.C. PROGRAMMING CHARACTERISTICS $T_A = 25°C \pm 5°C$

Table 2

Symbol	Parameter	Limits			Test Conditions
		Min	Max	Unit	
I_{LI}	Input Current (All Inputs)		1.0	μA	$V_{IN} = V_{IL}$ or V_{IH}
V_{IL}	Input Low Level (All Inputs)	−0.2	0.8	V	
V_{IH}	Input High Level	2.0	$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage During Verify		0.4	V	$I_{OL} = 2.1$ mA
V_{OH}	Output High Voltage During Verify	$V_{CC} - 0.8$		V	$I_{OH} = -400$ μA
I_{CC2}(3)	V_{CC} Supply Current		30	mA	
I_{PP2}(3)	V_{PP} Supply Current (Program)		50	mA	$\overline{CE} = V_{IL}$
V_{ID}	A_9 intelligent Identifier Voltage	11.5	12.5	V	
V_{PP}(1)	Programming Voltage	12.5	13.0	V	
V_{CC}(1)	Supply Voltage During Programming	6.0	6.5	V	

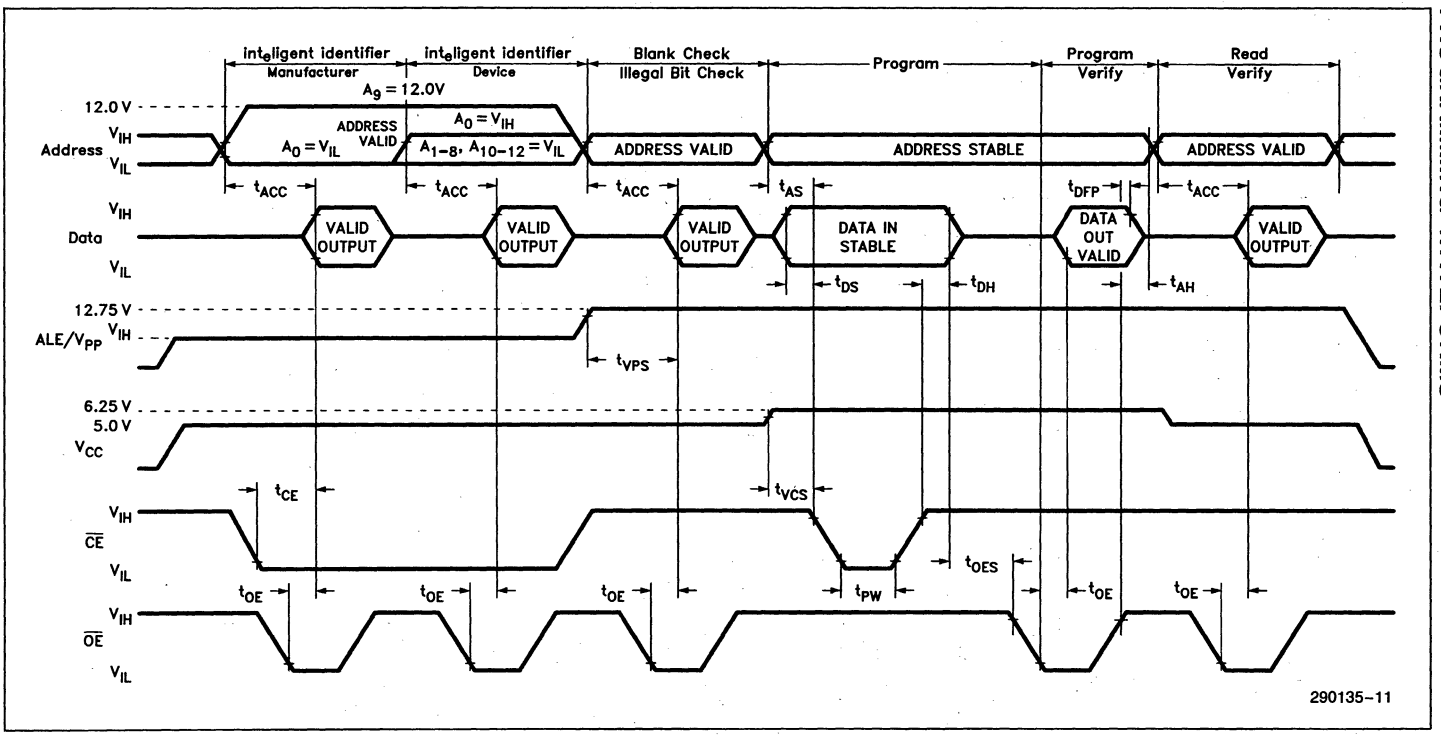
A.C. PROGRAMMING CHARACTERISTICS

$T_A = 25°C \pm 5°C$; see Table 2 for V_{CC} and V_{PP} voltages.

Symbol	Parameter	Limits				Conditions
		Min	Typ	Max	Unit	
t_{AS}	Address Setup Time	2			μs	
t_{OES}	$\overline{OE}$ Setup Time	2			μs	
t_{DS}	Data Setup Time	2			μs	
t_{AH}	Address Hold Time	0			μs	
t_{DH}	Data Hold Time	2			μs	
t_{DFP}(2)	$\overline{OE}$ High to Output Float Delay	0		130	ns	
t_{VPS}(1)	V_{PP} Setup Time	2			μs	
t_{VCS}(1)	V_{CC} Setup Time	2			μs	
t_{PW}	$\overline{CE}$ Program Pulse Width	95	100	105	μs	
t_{OE}	Data Valid from $\overline{OE}$			150	ns	

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram.
3. The maximum current value is with outputs O_0 to O_7 unloaded.

PROGRAMMING WAVEFORMS



NOTES:
1. The input timing reference level is V_{IL} = 0.8V and V_{IH} = 2V.
2. t_{OE} and t_{DFP} are device characteristics but must be accommodated by the programmer.
3. To prevent device damage during programming, a 0.1 μF capacitor is required between V_{PP} and ground to suppress spurious voltage transients.
4. During programming, the address latch function is bypassed whenever V_{PP} = 12.75V or $A_9 = V_H$. When V_{PP} and A_9 are at TTL levels, the address latch function is enabled, and the device functions in read mode.
5. V_{PP} can be 12.75V during Blank Check and Final Verify; if so, $\overline{CE}$ must be V_{IH}.

87C75PF
MICROCONTROLLER PERIPHERAL I/O PORT EXPANDER WITH 32Kx8 EPROM

- 2 Configurable 8-bit I/O Ports
  - Open Drain
  - Quasi-bi-directional
  - CMOS
- 32K x 8 EPROM
  - 200nS Access Time
- Quick-Pulse Programming™ Algorithm
  - 4 Second Programming
- Configuration Registers
  - Relocate the EPROM in Memory
  - Relocate the SFRs in Memory
  - Programmable RESET Level
  - Double or Single Plane Operation
- No-Glue Microcontroller Interface
  - Programmable Memory Map
  - Programmable Control Signals
  - Built-in Address Latches
  - Integrated Address Decoder
- Special Function Registers (SFRs)
  - Port Latch Read/Write
  - Port Pin Read
- Low Power CHMOS-II-E
  - TTL Compatible
- 40-Pin DIP, 44-Lead PLCC
  (See Packaging Spec., Order #231369)

The microcontroller peripheral Port Expander contains two 8-bit bi-directional I/O ports, a 32K x 8 EPROM, fully multiplexed address/data pins, and a user-configurable architecture. A microcontroller that accesses external memory must use two of its 8-bit I/O ports for multiplexed address/data lines. The Port Expander recovers these two ports while supplying needed EPROM memory. Considerable board space and design time can be saved by replacing discrete memory, port, address-decoder, address-latch, and glue chips with a single Port Expander chip.

User-programmable options allow "no-glue" interfacing with 8051, 8096, and 80188 microcontroller families. EPROM and port addresses can be relocated within dual-64K-byte memory planes. Non-standard-architecture microcontrollers (68xx, 63xx, Z8xx, etc.) require only minimal "glue" chips to interface with the Port Expander. The programmable RESET input will conform to various microcontrollers. Its flexible architecture allows applications to use multiple Port Expander chips.

The device's flexibility accommodates several microcontroller architectures. Its default mode is ideal for dual-memory-plane 8051 applications. A single plane option conforms to 80188, 68xx, and 8-bit-mode 8096 architectures. The memory-plane overlap option allows address-constrained systems and 8051 systems that have code compiled from high-level languages to use multiple Port Expanders.

*CHMOS is a patented process of Intel Corporation.

October 1988
Order Number: 290165-002

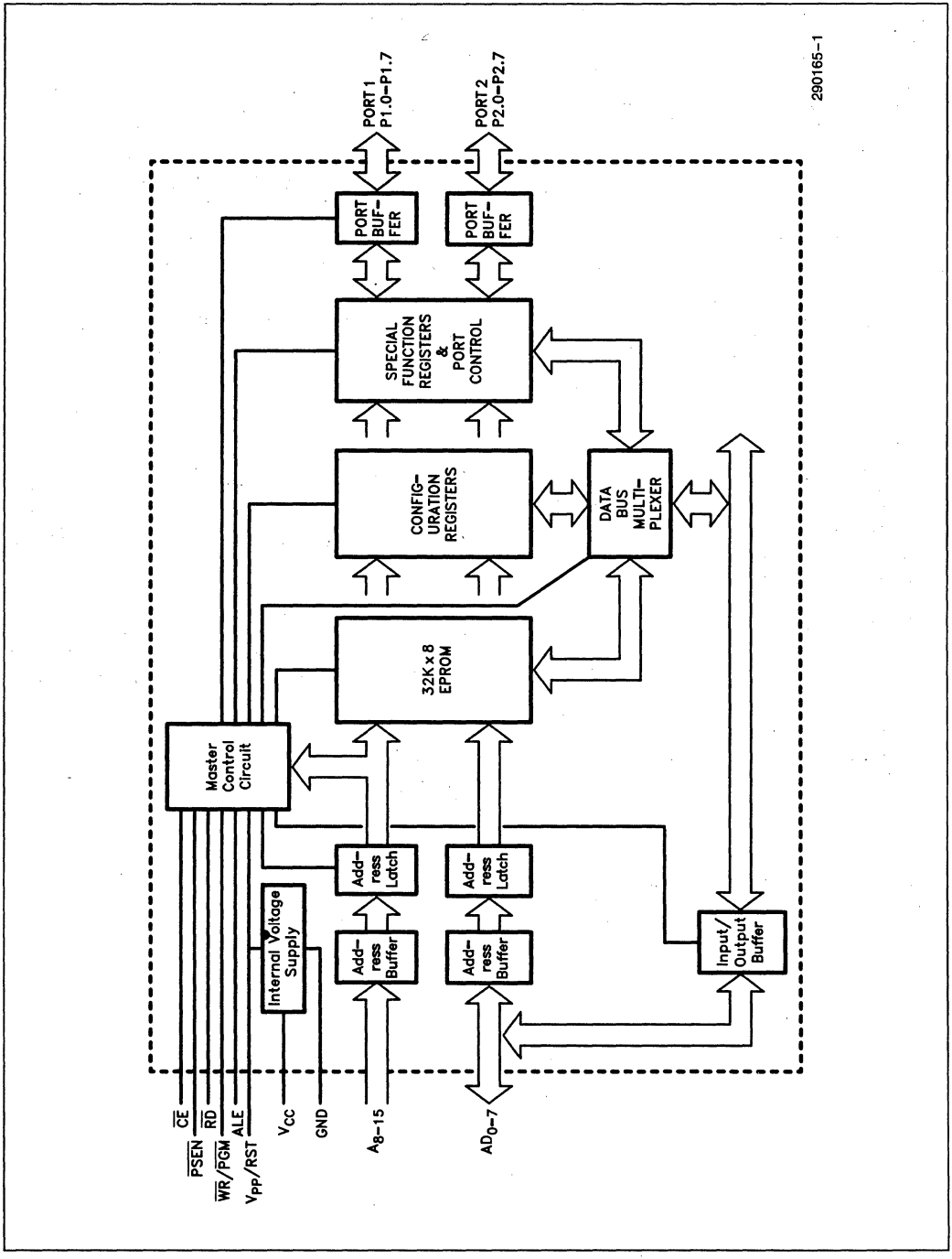


Figure 1. Block Diagram

ARCHITECTURE

Intel's 8051-family and the 87C75PF form the most versatile, integrated microcontroller combination in the industry. No other solution provides a microcontroller, 32K-bytes of EPROM and port expansion in only two chips. Also, the 87C75PF takes full advantage of the 8051's separate program- and data-memory planes. The 87C75PF uses all sixteen address/data lines and all of the 8051's control signals to access two 64K-byte memory planes. In fact, this architecture accommodates two 87C75PFs — 64K-bytes of EPROM and 4 ports — still leaving room for 60K of RAM and other features.

The 87C75PF's versatility makes possible minimum-chip solutions for other microcontroller architectures, too. Single memory-plane modes are user programmable for no-glue interfaces to 8096BH, 80C196, 8098, and 80188 controllers.

Flexible Memory Map

Programmable memory map options will customize the 87C75PF for any application. Intel's 8051 and 8096 microcontrollers have boot-up locations in the lower half of their memory maps. The 87C75PF's EPROM defaults to low memory for these controllers. 80188, 68xx, and 63xx microcontrollers use high-memory boot-up (code and vector) addresses. A user programmable option will move the 87C75PF's EPROM to the device's high-memory addresses. Special Function Registers and port addresses can also be moved to any 2K-byte address boundary.

Programmable Control

Reset level varies depending on the microcontroller family. The 87C75PF's reset (RST) is active-high to match the 8051. Other microcontrollers have active-low reset. A programmable active-low reset option will configure the 87C75PF for these controllers.

Versatile I/O Ports

The 87C75PF has two 8-bit I/O ports. Port 1 is open-drain and port 2 is quasi-bi-directional. The open-drain port can be used for high impedance inputs or "wire-ORed" input/outputs. The quasi-bi-directional port can be used as inputs with built-in pull-up resistors or as low-current-drive outputs. Alternate modes allow either port to have active pull-up (CMOS) outputs. This output mode provides higher current, faster switching, and low power port drive.

Minimum Chip Microcontroller Solution

Primary applications are: 1) single-chip microcontroller systems that have outgrown the controller's internal code-memory and 2) multiple-chip systems that need features-integration, such as redesigned applications that recover ports with discrete components. Typical memory expansion requires EPROM, port expander chips, address latches, address decoder and glue-logic chips — all are incorporated in the 87C75PF (Figure 1).

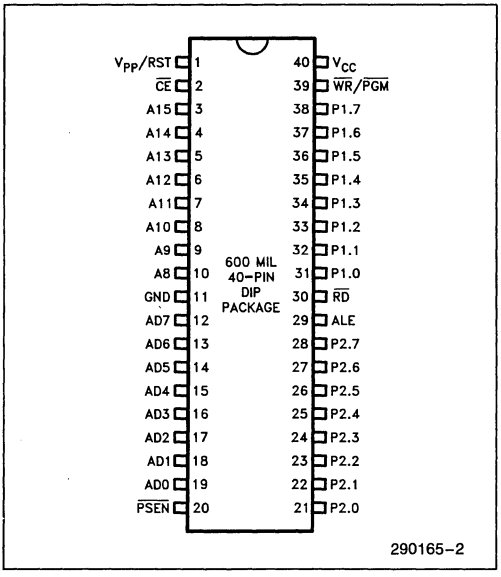


Figure 2. 40-Pin Dip Package

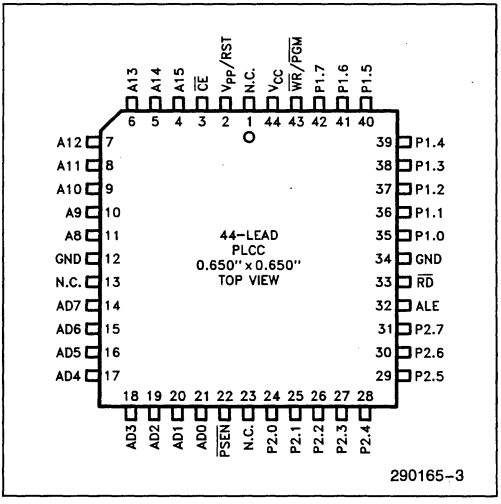


Figure 3. 44-Lead PLCC Package

PIN DESCRIPTIONS

Symbol	Pin Number		Function
	DIP	PLCC	
VPP/RST	1	2	In operating mode, VPP/RST is V_{IL} or V_{IH} and serves as the reset input. RST is user programmable as active-high or active-low via the Control Level Register (CLR.7). When RST is asserted, ports are set to inputs in non-CMOS mode or 1's in CMOS mode. With RST asserted, port-writes have no affect; port-latch-reads return "1s". VPP is the programming supply-voltage input.
$\overline{CE}$	2	3	$\overline{CE}$, the master device enable input, is active-low. When asserted, data can be written and read to/from the device. When $\overline{CE}$ is not asserted, the memory is in standby and cannot be accessed; ports cannot be accessed but maintain their current active states.
A_{15}–A_8	3–10	4–11	High-order addresses flow into the device when ALE = V_{IH} and are latched when ALE = V_{IL}.
GND	11	12,34	VSS (Ground) pins.
AD_7–AD_0	12–19	14–21	Multiplexed low-order address/data. After ALE latches addresses, these pins input or output data depending on $\overline{RD}$, $\overline{WR}/\overline{PGM}$, and $\overline{PSEN}$.
$\overline{PSEN}$	20	22	This active-low pin is the Program Store ENable. EPROM or non-volatile registers are read if this pin is asserted. If bit ELR.6 is programmed ("0"), $\overline{PSEN}$ and $\overline{RD}$ are internally combined. If either or both of these signals is V_{IL}, EPROM or SFR data is accessed depending on the address. When VPP is at its programming voltage, $\overline{PSEN}$ and $\overline{RD}$ are internally combined, as described above. This allows a resident microcontroller to use its $\overline{READ}$ signal to verify programmed data during in-system programming.
P2.0–P2.7	21–28	24–31	8-bit I/O port pins with Quasi-bi-directional (internal pull-up) outputs. All Port 2 pins can be configured as CMOS outputs by programming Control Level Register bit CLR.5.
ALE	29	32	Addresses flow through the latches to address decoders when ALE = V_{IH}. ALE's falling edge latches all addresses independent of $\overline{CE}$. $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}/\overline{PGM}$ are non-functional when ALE is V_{IH}. Read and write modes are possible only when ALE is V_{IL}.
$\overline{RD}$	30	33	During normal operation, $\overline{RD}$ is used to read information from the SFRs. If bit ELR.6 = "0", $\overline{RD}$ and $\overline{PSEN}$ are internally combined (see $\overline{PSEN}$ pin description). During programming, $\overline{RD}$ and $\overline{PSEN}$ are internally combined when VPP is at its programming voltage. This pin's location is the same as a megabit EPROM's GND pin. For compatibility with PROM programmers that force this pin to ground, $\overline{RD}$ becomes non-functional when P1.0 is at V_H.
P1.0–P1.7	31–38	35–42	General purpose 8-bit open-drain I/O port pins. When P1.0 is at V_H (12V) the Configuration Plane can be accessed (see the Mode table) and $\overline{RD}$ is internally disabled. To prevent device damage, Port 1 must be reset, by RST, or have a "1" written to P1.0 before V_H is applied to P1.0. All Port 1 pins can be configured as CMOS outputs by programming Control Level Register CLR.6.
$\overline{WR}/\overline{PGM}$	39	43	The active-low $\overline{WR}/\overline{PGM}$ is used to write data to the SFRs. During programming (VPP = 12.75V), the SFRs cannot be written, and this signal becomes the program-pulse control input.
VCC	40	44	This pin is the supply voltage input.

EXTENDED TEMPERATURE (EXPRESS) uC PERIPHERAL

Intel's EXPRESS microcontroller and application-specific peripheral families receive additional processing to enhance product characteristics. EXPRESS processing is available for several microcontrollers, EPROMs, and peripheral products allowing the appropriate device to match custom system applications. EXPRESS devices are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process meets or exceeds most industry burn-in specifications. The standard EXPRESS operating temperature range is 0°C to +70°C. EXPRESS extended operating temperature range (−40°C to +85°C) and automotive temperature range (−40°C to +125°C) products are also available. Like all Intel products, the EXPRESS family is inspected to 0.1% electrical AQL. This allows reduction or elimination of incoming testing.

AUTOMOTIVE AND EXPRESS PRODUCT FAMILY

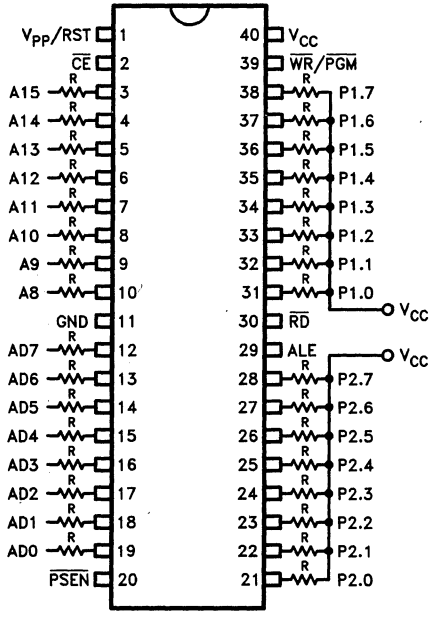
PRODUCT DEFINITIONS

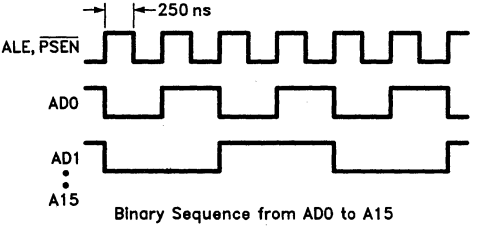
Type	Operating Temperature (°C)	Burn-in 125°C (hr)
Q	0°C to +70°C	168 ±8
T	−40°C to +85°C	NONE
L	−40°C to +85°C	168 ±8

AUTOMOTIVE AND EXPRESS OPTIONS

Speed Versions	Packaging Options	
	CERDIP	PLCC
	Contact your local Intel Sales Office for EXPRESS product availability	

Burn-In Bias and Timing Diagrams





Binary Sequence from AD0 to A15

This cycle also includes port read/write accesses.

V_{CC} = +5V, V_{PP}/RST = +5V, $\overline{CE}$ = GND, $\overline{RD}$ = +5V, $\overline{WR}$/$\overline{PGM}$ = +5V, R = 10K

290165-20

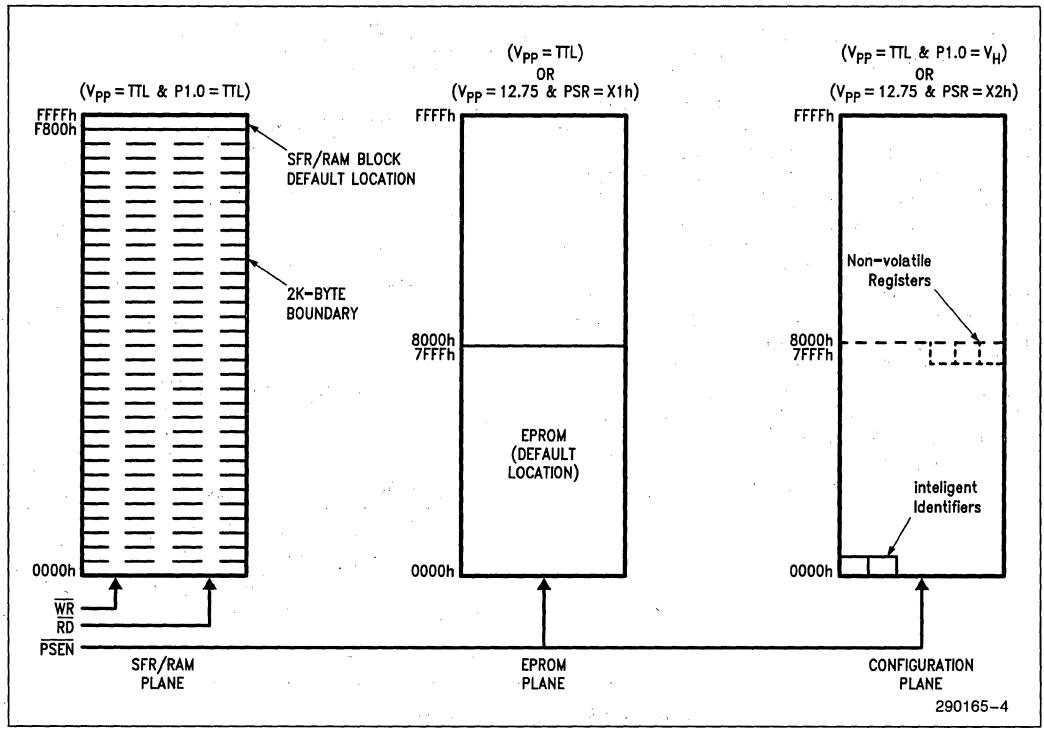


Figure 4. The Port Expander's 3-Plane Memory Map

MEMORY MAP

The Port Expander contains three memory planes — EPROM plane, special function register/RAM (or SFR/RAM) plane, and the configuration plane. Figure 4 shows the three memory planes.

The EPROM's default location (in an erased device) is at the bottom 32K bytes (0000h to 7FFFh) of the 64K-byte EPROM plane. The default location of the special function register block is F800h to FFFFh in the SFR/RAM plane. Non-volatile registers (NVRs) are mapped at addresses 7FFDh through 7FFFh in the configuration plane.

Non-volatile registers are used to program the locations of the EPROM, SFR block, and other features (see Figure 7). In normal operating mode, the configuration plane cannot be accessed; only the EPROM and SFR/RAM planes are available. During programming/verification, the plane select register, PSR, (SFR default location F810h) determines which plane — EPROM or configuration — is accessed. The EPROM array is programmed/verified if PSR contains xxxxxx01b (X1h). The configuration plane is programmed/verified if PSR contains xxxxxx10b (X2h) before V_{PP} is raised to 12.75V.

NVRs in the configuration plane are also read if pin P1.0 = V_H (12V) while V_{PP} = TTL. This allows PROM programmers to identify the device, download its configuration, and program duplicates accordingly.

ARCHITECTURE FLEXIBILITY

The Port Expander can operate in several configurations. The configuration plane's non-volatile registers configure the device for microcontroller-architecture compatibility.

8051 architecture accommodates two 64K-byte memory planes — program-memory and data-memory planes. In its default mode (erased) the device is configured with these two independently addressable planes — a perfect companion for the 8051 family.

Many other 8-bit microcontrollers (8096BH, 8098, Z8xx, 68xx, etc.), and 8051s with code compiled from high-level languages, can handle only one 64K-byte memory plane. Another mode configures the device for single plane operation — again, a perfect 8-bit microcontroller companion device.

Often, more than two external ports and greater than 32K-bytes of external EPROM are required in single-memory-plane applications. Another mode allows two Port Expanders to supply 60K-bytes of EPROM and four 8-bit I/O ports — still leaving 4K-bytes for other read/write devices.

Double-plane Applications

The default configuration has two memory planes; program (EPROM) and data (SFR/RAM). This configuration is consistent with the 8051 architecture. The EPROM plane is read-only and is accessed by $\overline{PSEN}$. The SFR/RAM plane is a read/write plane that is accessed by the $\overline{RD}$ and $\overline{WR}/\overline{PGM}$ inputs. These signals and the sixteen address inputs provide two 64K-byte memory-planes.

Single-plane Applications

Many microcontroller architectures have only one 64K-byte memory plane. One way to configure the device for a single-plane is to simply tie $\overline{PSEN}$ and $\overline{RD}$ together and connect the combined read signal to the system's $\overline{READ}$ line.

8051 machine code compiled from high-level languages often can't deal with separate program- and data-planes. Systems using high-level languages usually form one 64K-byte memory plane by combining $\overline{PSEN}$ and $\overline{RD}$ into a common $\overline{READ}$ signal (by using an AND gate).

The Port Expander provides a better solution. If the EPROM Location Register bit ELR.6 is programmed, $\overline{PSEN}$ and $\overline{RD}$ are combined internally to form a common $\overline{READ}$ signal. Either of these signals can be used to gate data from the EPROM plane and/or SFR/RAM plane to the outputs. In effect, this mode forms a single 64K-byte memory plane. For 8051 high-level-language systems, no external glue is required to "AND" $\overline{PSEN}$ with $\overline{RD}$. The 8051's $\overline{PSEN}$ and $\overline{RD}$ signals can be connected directly to the Port Expander's corresponding inputs. Single-plane, non-8051 microcontroller systems need to route their $\overline{READ}$ line to either, or both, $\overline{PSEN}$ or $\overline{RD}$. If only one input is used, the other must be tied high.

Overlapped Single Plane

Two Port Expanders can fit in a single-memory-plane system by programming the configuration plane's non-volatile registers. To accomplish this, each device must have its SFR block mapped over a portion of its EPROM array. The SFR block can be placed on any 2K-byte boundary by programming the SFRLR. EPROM Location Register bit ELR.7 allows the EPROM to be moved to high memory or to remain in its default low-memory location. Programming ELR.6, the overlap bit, allows the EPROM plane to be mapped over the SFR/RAM plane; this also internally combines $\overline{PSEN}$ and $\overline{RD}$. 2K EPROM bytes located at the SFR block's base-address are disabled and replaced by the 2K-byte SFR block.

Figure 5 shows various memory configurations.

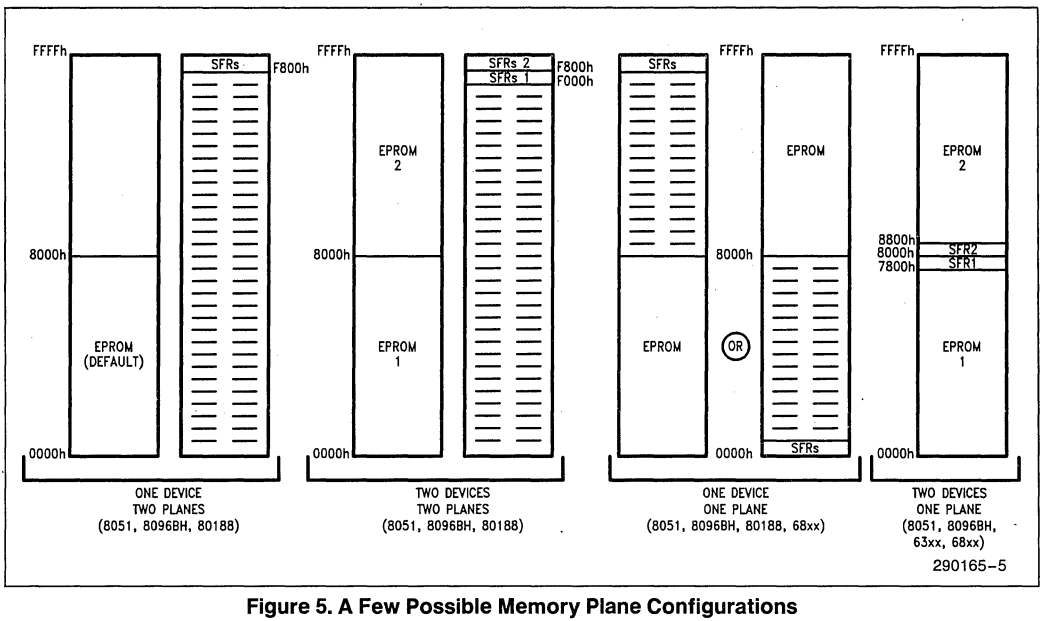


Figure 5. A Few Possible Memory Plane Configurations

EPROM

The Port Expander contains a 32,768 x 8-bit EPROM. When erased, the EPROM is located between EPROM plane addresses 0000h and 7FFFh. This is a common boot-up address range for most microcontrollers, including 8051 and 8096 families. For microcontrollers that reset in high addresses, the EPROM can be relocated to device addresses 8000h through FFFFh via the EPROM Location Register. This also allows systems to use two Port Expanders — one with EPROM at low-memory and the other with EPROM relocated at high-memory.

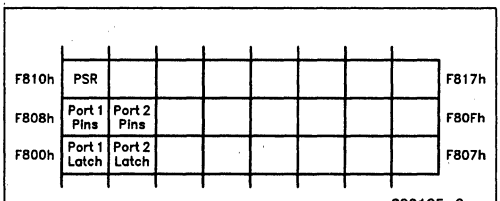
When a valid address is present, $\overline{PSEN}$ controls EPROM access. Asserting $\overline{PSEN}$ during non-valid addresses places device outputs at high impedance.

SPECIAL FUNCTION REGISTERS (SFRs)

SFR addresses described below and in Figure 6 are default in an erased device. A 2K-byte block (default locations F800h through FFFFh) is reserved for ports, plane select register (PSR), and future features. The SFR/RAM-block base address can change depending on the SFRLR's five most-significant bits (Figure 10). Only five SFR/RAM-block locations are defined. Accessing any other addresses in this block places the external bus in a high impedance state allowing external devices to occupy these locations.

Ports are accessed by reading or writing the SFRs. Port 1 and Port 2 latch data is read/written by accessing locations F800h and F801h. F802h through F807h are reserved for future port latches. Port 1 and Port 2 pins are read at F808h and F809h. Writing to these locations has no effect. F80Ah through F80Fh are reserved for future port-pin locations.

F810h is a two-bit read/write plane select register (PSR). During program/verify, PSR's value before V_{PP} = 12.75V determines whether the EPROM- or configuration-plane is accessed. If PSR contains xxxxxx01b, the EPROM plane is programmed and verified. If PSR contains xxxxxx10b, configuration plane registers will be programmed and verified. In operating mode, the configuration plane cannot be accessed. However, in the configuration read mode (P1.0 = V_H and V_{PP} = TTL) configuration registers can be read (only) and the SFRs can be written (only).



NOTE:
SFRs are shown in their default locations.

Figure 6. SFR Memory Map

CONFIGURATION PLANE

Non-Volatile Registers (NVRs)

The configuration plane contains the inteligent Identifier™ and non-volatile registers. This plane is read:

1) if P1.0 = V_H (V_H = 12V ±1V) while V_{PP} = TTL
   or
2) if PSR contains xxxxxx10b while V_{PP} = 12.75V.

Inteligent Identifier codes are at 0000h (manufacturer) and 0001h (device). NVRs are at 7FFDh (CLR), 7FFEh (ELR), and 7FFFh (SFRLR).

NVRs are programmed/verified by writing xxxxxx10b to PSR before V_{PP} = 12.75V; inteligent Identifier bytes are read-only. Figure 7 shows the configuration plane's NVR locations. Condition 1) above allows PROM programmers to check the device's configuration and locate the SFRs and EPROM. NVRs are EPROM cells which, when erased, contain "1s".

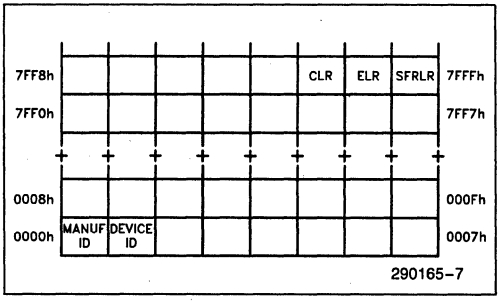


Figure 7. Configuration Plane Memory Map

Control Level Register (CLR)

The Control Level Register, CLR (7FFDh), is used to change the RST pin's active level and port output drive. RST is active-high and CMOS port-drive is disabled in an erased device. If the reset level bit, RSTL (CLR.7), is programmed ("0"), RST is active-low. Port 1 and/or Port 2 outputs will be CMOS if P1C (CLR.6) and/or P2C (CLR.5) are programmed.

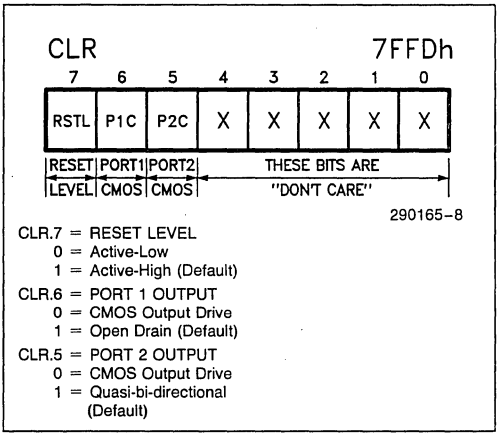


Figure 8. Control Level Register (CLR)

EPROM Location Register (ELR)

The EPROM Location Register (ELR) is at 7FFEh. The EPROM location bit, EL (ELR.7), places the EPROM at either top or bottom EPROM-plane addresses. When erased, the EPROM array is at 0000h–7FFFh in the 64K-byte address space. Programming ELR.7="0" places the EPROM at 8000h–FFFFh.

When erased, the overlap option is disabled. EPROM and SFR/RAM blocks are in default locations. $\overline{PSEN}$ accesses EPROM- and $\overline{RD}$ accesses the SFR-data.

If the OVERLAP bit, OVLP (ELR.6), is programmed, EPROM and SFR/RAM planes overlap. $\overline{PSEN}$ and $\overline{RD}$ are internally combined. If either is V_{IL}, EPROM or SFR data is accessed depending on the address.

If the SFR/RAM block's 2K-byte boundary overlaps the EPROM array and ELR.6=0, the SFR/RAM block replaces 2K EPROM bytes. Accessing non-defined bytes in the 2K-byte space places the external bus in a high-Z state. By programming ELR.6, one-memory-plane microcontrollers (8096, 80188, 68xx) and 8051s with high-level-language-compiled code can use two 87C75PFs.

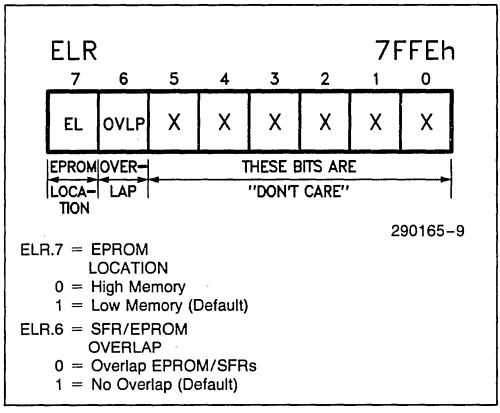


Figure 9. EPROM Location Register (ELR)

SFR Location Register (SFRLR)

The SFRLR (7FFFh) determines the SFR/RAM block's five most-significant base-address bits; SFRLR.7 = A15, SFRLR.6 = A14, SFRLR.5 = A13, SFRLR.4 = A12, and SFRLR.3 = A11. Programming this register places the SFR/RAM block on any 2K-byte boundary. For example, the SFRs are placed at 2800h by programming 00101xxxb.

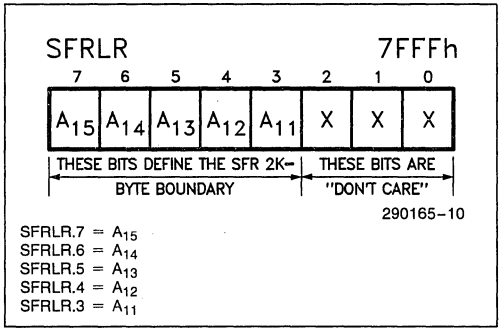


Figure 10. SFR Location Register (SFRLR)

Port 1

Port 1 has 8 open-drain, bi-directional pins. Pins float if "1s" are written to latches or RST is asserted. They can then serve as high impedance inputs.

P1.0 receives high voltage (V_H = 12V) during the inteligent Identifier/NVR Mode. P1.0 MUST BE RESET (BY RST OR BY WRITING "1" TO P1.0) BEFORE APPLYING V_H. For megabit PROM programmer compatibility, P1.0 = V_H disables $\overline{RD}$.

All Port 1 pins are CMOS outputs (TTL level in 200ns, CMOS level in 1us) if P1C is programmed (CLR6="0"). Asserting RST or writing "1s" will present CMOS V_{OH} levels. CMOS-configured Port 1 pins should not be used as inputs. Port latch writes occur on $\overline{WR}$'s rising edge to prevent glitches when changing individual bits; other bits are not affected.

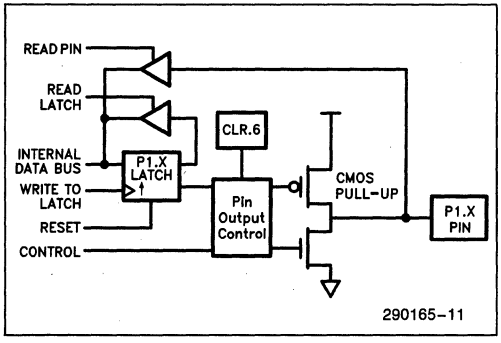


Figure 11. Port 1 Pin Diagram

Port 2

Port 2 is an 8-bit quasi-bi-directional port. Writing "1s" asserts short-duration active pull-ups to guarantee CMOS V_{OH} levels within 200ns. Port 2 pins are held high by internal pull-ups allowing them to serve as inputs. Pins pulled low externally source current (I_{IL}). Port 2 latches are set to "1s" upon reset.

Programming P2C (CLR.5="0") configures Port 2 as CMOS outputs. Asserting RST or writing "1s" outputs CMOS V_{OH} levels. CMOS-configured Port 2 pins should not be used as inputs. Port latch writes occur on $\overline{WR}$'s rising edge to prevent glitches when changing individual port bits; other bits are not affected.

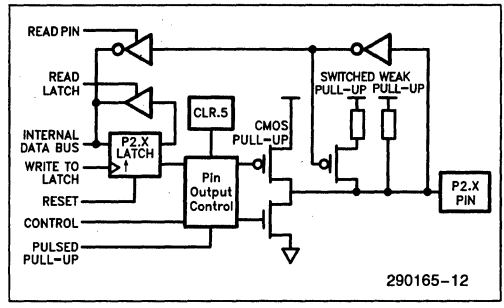


Figure 12. Port 2 Pin Diagram

SYSTEM APPLICATIONS

The 87C75PF significantly reduces chip count and interfacing hardware in multiplexed address/data bus systems. Figure 13 shows a low power, small board space, minimum chip design. The controller's multiplexed bus (AD_{0-7}) is tied to the 87C75PF's address/data pins. Separate address latches and address decoders are not needed because the 87C75PF latches all sixteen addresses and decodes internal features within its two 64K-byte memory planes.

ALE controls the 87C75PF's internal address latches. A V_{IH} to V_{IL} transition latches the present address. $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$ control data-flow between the controller and 87C75PF. 8051, 8096, and 80188 families benefit from the 87C75PF's "no-glue" interface.

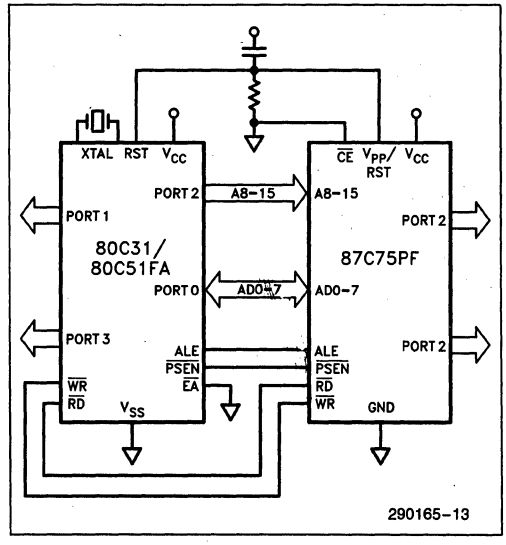


Figure 13. "No-glue" 80C51 with 87C75PF

Table 1. MODE SELECTION 87C75PF (default configuration shown)

MODE	$\overline{CE}$	$\overline{PSEN}$	$\overline{RD}$	$\overline{WR}/\overline{PGM}$	ALE(6)	V_{PP}/RST(4)	V_{CC}	P1.0(2)	AD_{0-7}
Reset	X(1)	X	X	X	X	V_{IH}	5V	X(10)	X
Read EPROM(12)	V_{IL}	V_{IL}	V_{IH}	V_{IH}	V_{IL}	X	5V	X	D Out
Read SFR(12)	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	5V	X	D Out
Single Plane Read(8)	V_{IL}	V_{IL} or V_{IL}		V_{IH}	V_{IL}	X	5V	X	D Out
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{IL}	X	5V	X	High Z
Write SFR	V_{IL}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IL}(14)	5V	X or V_H	D In
Write Disable	V_{IL}	X	X	V_{IH}	V_{IL}	X	5V	X	High Z
Read/Write Disable	V_{IL}	X	X	X	V_{IH}	X	5V	X	High Z
Standby	V_{IH}	X	X	X	X	X	5V	X	High Z
Program EPROM/NVR(5)	V_{IL}	V_{IH}	X(11)	V_{IL}	V_{IL}	V_{PP}(3)	V_{CP}(3)	V_H(7)	D In
EPROM/NVR Verify(5)	V_{IL}	V_{IL}	X(11)	V_{IH}	V_{IL}	V_{PP}	V_{CP}	V_H	D Out
Program Inhibit	V_{IH}	X	X	X	X	V_{PP}	V_{CP}	X	High Z
Alternate Program	V_{IL}	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{PP}	V_{CP}	X	D In
Alternate Verify(9)	V_{IL}	V_{IL} or V_{IL}		V_{IH}	V_{IL}	V_{PP}	V_{CP}	X	D Out
NVR Config Read(13)	V_{IL}	V_{IL}	X	V_{IH}	V_{IL}	X(7)	5V	V_H	D Out
inteligent(13) Identifier - Manuf	V_{IL}	V_{IL}	X	V_{IH}	V_{IL}	X(7)	5V	V_H	89h
inteligent(13) Identifier - Device	V_{IL}	V_{IL}	X	V_{IH}	V_{IL}	X(7)	5V	V_H	D0h

NOTES:
1. X can be V_{IL} or V_{IH}.
2. $V_H = 12.0V \pm 1V$.
3. $V_{PP} = 12.75V$ and $V_{CP} = 6.25v$ during programming.
4. RST is active-high (erase default shown) or programmable via CLR.7 as active-low.
5. The EPROM array is programmed/verified if PSR = X1h. The NVR array is programmed/verified if PSR = X2h. NVRs and inteligent Identifier can be read when P1.0 = V_H and V_{PP} = TTL.
6. Data cannot be read/written when ALE = V_{IH}. ALE must toggle — V_{IH} to V_{IL} — to latch addresses.
7. Reset must occur via V_{PP}/RST or "1" written to P1.0 before P1.0 = V_H.
8. If ELR.6 = 0, $\overline{PSEN}$ and $\overline{RD}$ are internally combined.
9. If $V_{PP} = 12.75V$, $\overline{PSEN}$ and $\overline{RD}$ are internally combined. If either is V_{IL}, EPROM (PSR = X1h) or NVR (PSR = X2h) data is verified. If P1.0 = V_H, $\overline{RD}$ is non-functional. If V_{PP} = TTL and P1.0 = V_H, only NVRs and inteligent identifier can be read.
10. RST sets port latches to "1s". After reset, P1.0 (= "1") is protected when V_H is applied.
11. For programmer compatibility, the 87C75PF's $\overline{RD}$ is disabled when P1.0 = V_H.
12. $\overline{PSEN}$ and $\overline{RD}$ can be asserted simultaneously unless the EPROM and SFRs overlap & ELR.6 = 1.
13. Addresses must be latched during Identifier/NVR reads.
14. RST not asserted.

DEVICE OPERATION

Table 1 lists 87C75PF operating and programming modes. Operating modes require a 5V power supply. Programming modes require 12.75V V_{PP}, 6.25V V_{CC}, and 12.0V Identifier/NVR-read voltages. All input levels are TTL or CMOS except V_{PP}, V_{CP}, and V_H.

OPERATING MODES

Reset

RST is an active-high input in an erased device. Programming CLR.7 ("0") makes RST active-low. Asserting RST for 500ns sets port latches to "1s". RST affects no other writable locations. Before a PROM programmer enters the inteligent Identifier/NVR read mode, RST should be asserted (or "1" written to P1.0) to set P1.0's pin. This protects P1.0 from damage by the 12.0V identifier voltage.

EPROM Read Mode

$\overline{PSEN}$ enables EPROM data onto AD_{0-7} and controls the device's output buffer. This active-low pin functions only when $\overline{CE}$ and ALE are asserted. When an address is latched (ALE = V_{IL}), access time (t_{AVDV}) equals the $\overline{CE}$ to output delay (t_{CLDV}). Outputs display valid data t_{ELDV} after $\overline{PSEN}$'s falling edge, assuming t_{AVDV} and t_{CLDV} times are met.

SFR Read Mode

$\overline{RD}$ enables SFR data onto AD_{0-7} and controls the device's output buffer. This active-low pin functions only when $\overline{CE}$ and ALE are V_{IL}. EPROM read mode timing requirements apply to this mode.

Single Plane Read Mode

This mode allows single-plane microcontrollers and 8051-family controllers with high-level-language-compiled code to use an 87C75PF without "glue" devices. It is possible to assert $\overline{PSEN}$ and $\overline{RD}$ simultaneously. Data bus conflict will not occur if the SFRs are not memory mapped over EPROM array addresses. If SFR and EPROM addresses overlap, bus conflict can be avoided if the EPROM location register's "Overlap" bit (ELR.6) is programmed. Programming this bit also internally combines $\overline{PSEN}$ and $\overline{RD}$. Asserting either (or both) enables EPROM or SFR data, depending on the address, onto AD_{0-7}. See the "Overlapped Single Plane" section for details.

Output Disable Mode

If $\overline{PSEN}$ and $\overline{RD}$ are not asserted, the device's output buffers (AD_{0-7}) are disabled. Data can be written to the 87C75PF or transferred to/from other devices.

SFR Write Mode

$\overline{WR}/\overline{PGM}$ enables data on AD_{0-7} to be written into the SFRs. This active-low pin functions only when $\overline{CE}$ and ALE are V_{IL}. When an address is latched (ALE = V_{IL}) and data has been present for t_{DVWH}, $\overline{WR}$'s rising edge latches data into an SFR. Other A.C. timing parameters must be observed.

Write Disable

SFR data cannot be written when $\overline{WR}/\overline{PGM}$ is high. Low-address and data share common pins, but the device allows new addresses only when ALE is high; data can be written only when ALE is low.

Read/Write Disable

Since the Port Expander uses a multiplexed address/data bus, data can be read or written only if a valid address is latched. To prevent erroneous reads or spurious writes of invalid data, $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}/\overline{PGM}$ are non-functional when ALE is high; however, new address information can enter the address latches. ALE's falling edge latches the address and enables $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}/\overline{PGM}$.

Standby Mode

Standby mode substantially reduces V_{CC} current. $\overline{CE}$ = V_{IH} places output buffers in low-power, high impedance mode independent of $\overline{PSEN}$, $\overline{RD}$, or $\overline{WR}$. Two-line output control ($\overline{CE}+\overline{PSEN}$ or $\overline{CE}+\overline{RD}$) provides:

a) minimum memory power dissipation, and

b) assurance that data bus contention will not occur.

To efficiently use two-line control, address decoding circuitry should enable $\overline{CE}$. $\overline{PSEN}$ should be connected to the microcontroller's program-store enable ($\overline{PSEN}$), $\overline{RD}$ to the controller's data-read enable ($\overline{RD}$), and $\overline{WR}/\overline{PGM}$ to its write control ($\overline{WR}$). This assures that only selected memory and peripheral devices have active inputs and outputs while non-selected devices are in low-power standby mode.

PROGRAMMING MODES

EPROM/Configuration (NVR) Programming Mode

Initially and after each erasure, all EPROM and NVR bits are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" are programmed, the data word can contain both "1s" and "0s". Ultraviolet light erasure is the only way to change "0s" to "1s".

The programming mode is entered when V_{PP} is raised to its programming voltage. After latching an address, data is programmed by applying an 8-bit word to data pins AD_{0-7}. Pulsing $\overline{WR}/\overline{PGM}$ to TTL-low while $\overline{CE}$ and ALE are V_{IL} will program data. TTL levels are required for address and data inputs.

To accommodate PROM programmers that force the $\overline{RD}$ pin to ground (DIP pin 30), applying 12V to port pin P1.0 will internally disable the 87C75PF's $\overline{RD}$ input. When V_{PP} is not at its programming voltage the device is in the inteligent Identifier mode. When V_{PP} is raised for programming, the inteligent Identifier mode is disabled.

EPROM and NVR Verify

With V_{PP} and V_{CC} at their programming levels and $\overline{CE}$ asserted, EPROM or configuration data (depending on PSR's contents) can be verified. To simplify on-board and in-system programming, $\overline{PSEN}$ and $\overline{RD}$ are internally combined when V_{PP} is at its programming level. Either signal can be used to verify programmed data (if P1.0 is not V_H).

For compatibility with PROM programmers equipped for word-wide megabit EPROMs, DIP-pin 30 — the 87C75PF's $\overline{RD}$ pin — is internally disabled when P1.0 is V_H.

Program Inhibit

The Program Inhibit mode allows parallel programming and verification of multiple devices with different data. With V_{PP} at its programming voltage, a $\overline{WE}/\overline{PGM}$ pulse programs any device that has $\overline{CE}$ asserted. Programming is inhibited on any device with $\overline{CE}$ not asserted.

Alternate Programming and Verification Modes

For programmers that can apply V_{IH} or V_{CC} to $\overline{RD}$, the EPROM and NVRs can be programmed using a more conventional slow-motion write-mode-type algorithm. 12V need not be applied to P1.0 to disable the $\overline{RD}$ pin during the alternate programming mode. $\overline{PSEN}$ and $\overline{RD}$ are internally combined when V_{PP} is applied, and either signal can be used to enable EPROM or NVR data during program verification. See the Quick-Pulse Programming algorithm flowchart and waveforms at the end of this data sheet.

inteligent Identifier™/NVR Mode

Programming equipment determines the device's manufacturer, type, and configuration (NVR contents) by using the inteligent Identifier/NVR Mode. A programmer can read a master device's identifier and NVRs, select the proper algorithm, and program duplicates accordingly.

The configuration plane is accessed by raising port pin P1.0 to V_H = 12.0V. Before P1.0 is brought to V_H, Port 1 must be reset by asserting the V_{PP}/RST pin or by writing a "1" to P1.0's latch. When ALE latches a valid address and $\overline{PSEN}$ is V_{IL}, identifier/NVR data appears on Address/Data pins AD_{0-7}. For compatibility with programmers that support megabit EPROMs, $\overline{RD}$, which is usually forced to ground, is "don't-care" when P1.0 = V_H. When $\overline{CE}$, ALE, and $\overline{PSEN}$ are V_{IL} and P1.0 = V_H, identifier/NVR data can be read. While in this mode, the SFR/RAM plane cannot be read but can be written. The PSR register can be configured so that either the EPROM or configuration plane is programmed when V_{PP} is raised. This mode's temperature range is 25°C +5°C.

ABSOLUTE MAXIMUM RATINGS*

Read Operating Temperature . . . . . 0°C to +70°C(2)
Case Temperature Under Bias . . −10°C to +80°C(2)
Storage Temperature . . . . . −65°C to +150°C
All Input or Output Voltages . . . . . −2.0V to +7.0V(1)
with Respect to Ground
Voltage on Pin P1.0 . . . . . −2.0V to +13.5V(1)
with Respect to Ground
V_{PP} Supply Voltage . . . . . −2.0V to +14.0V(1)
with Respect to Ground
V_{CC} Supply Voltage . . . . . +2.0V to +7.0V(1)
with Respect to Ground

Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

NOTICE: Specifications contained within the following tables are subject to change.

READ/WRITE BUS OPERATION

D.C. CHARACTERISTICS TTL and NMOS Inputs; see A.C. Characteristics for V_{CC} versions offered

Symbol	Parameter	Notes	Min	Max	Units	Test Conditions
I_{LI}	Input Load Current (A_{8-15})			1.0	μA	V_{IN} = 0V to V_{CC}
I_{LO}	Output Leakage Current (AD_{0-7})			10	μA	V_{OUT} = 0V to V_{CC}
I_{SB}	V_{CC} Current Standby	6		5	mA	$\overline{CE}$- inactive, ALE = V_{IL}
I_{CC}	V_{CC} Current Active	4		60	mA	$\overline{CE}$-active, ALE = V_{IH} f(Hz) = $1/t_{AVDV}$, I_{OUT} = 0 mA
V_{IL}	Input Low Voltage	1	−0.5	0.8	V	
V_{IH}	Input High Voltage		2.0	V_{CC} + 0.5V	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	1	2.4		V	I_{OH} = −400 μA
I_{OS}	Output Short Circuit Current	5		100	mA	

D.C. CHARACTERISTICS CMOS Inputs; V_{CC} = see A.C. Characteristics for versions offered.

Symbol	Parameter	Notes	Min	Max	Units	Test Conditions
I_{LI}	Input Load Current (A_{8-15})			1.0	μA	V_{IN} = 0V to V_{CC}
I_{LO}	Output Leakage Current (AD_{0-7})			10	μA	V_{OUT} = 0V to V_{CC}
I_{SB}	V_{CC} Current Standby	6		5	mA	$\overline{CE}$-inactive, ALE = V_{IL}
I_{CC}	V_{CC} Current Active	3 4		45	mA	$\overline{CE}$-active, ALE = V_{IH} f(Hz) = $1/t_{AVDV}$, I_{OUT} = 0 mA
V_{IL}	Input Low Voltage	1	−0.2	0.8	V	
V_{IH}	Input High Voltage		0.7 V_{CC}	V_{CC} + 0.2V	V	
V_{OL}	Output Low Voltage			0.40	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	1	V_{CC} − 0.8		V	I_{OH} = −400 μA
I_{OS}	Output Short Circuit Current	5		100	mA	

NOTES:
1. Minimum DC input voltage is −0.5V during transitions. Inputs may undershoot to −2.0V for periods less than 20 ns. Maximum output-pin DC voltage is V_{CC} + 0.5V; overshoot may be V_{CC} + 2.0V for periods less than 20 ns.
2. This specification defines commercial-product operating temperatures. EXPRESS and Automotive versions are available as noted.
3. $\overline{CE}$ is V_{CC} ± 0.2V (87C75PF inactive) or ± 0.2V (87C75PF active). Other inputs can have any value within specification.
4. Maximum current value with outputs unloaded.
5. One output shorted for no more than one second. I_{OS} is sampled but not 100% tested.
6. Port latches set to "1s"; outputs unloaded.

PORTS/RESET

D.C. CHARACTERISTICS V_{CC} = see A.C. Characteristics for versions offered.

Symbol	Parameter		Notes	Min	Max	Units	Test Conditions
I_{LI}	Input Leakage Current Port 1 Open Drain		1		10	μA	0.4V ≤ V_{IN} ≤ V_{CC}
I_{IL}	Logic 0 current Port 2 (Quasi-bi-directional)		2		−50	μA	P2.x latch = "1", V_{IN} = 0V
V_{IL}	Input Low Voltage	Ports 1&2		−0.5	0.2 V_{CC} − 0.1	V	
		RST Input	3	−0.5	0.2 V_{CC} − 0.1	V	
V_{IH}	Input High Voltage	Ports 1&2		0.2 V_{CC} + .9	V_{CC} + 0.5	V	
		RST Input	3	0.7 V_{CC}	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage				0.40	V	I_{OL} = 3.2 mA
V_{OH}	Output High Voltage	CMOS mode		2.4		V	I_{OH} = −400 μA
		Ports 1&2		0.9 V_{CC}		V	I_{OH} = −40 μA
		Quasi-bi-		2.4		V	I_{OH} = −60 μA
		dir Port 2		0.9 V_{CC}		V	I_{OH} = −10 μA

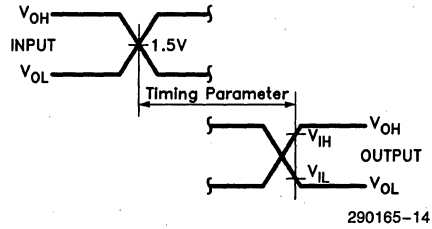
NOTES:
1. Input Leakage current does not apply to Port 2.
2. This specification assumes that Port 2 pins are internally driven to "1" but are externally pulled low
3. RST has hysteresis. V_{IL} is valid at or below 0.2 V_{CC} − 0.1V. V_{IH} is valid at or above 0.7 V_{CC}.

CAPACITANCE(1) T_A = 25°C, f = 1.0MHz

Symbol	Parameter	Max	Units	Conditions
C_{IN}	Address/Control Capacitance	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	12	pF	V_{OUT} = 0V
C_{VPP}	RST/VPP Capacitance	25	pF	V_{IN} = 0V
$C_{I/O}$	Port Pin Capacitance	10	pF	V_{OUT} = 0V

NOTE:
1. Sampled. Not 100% tested.

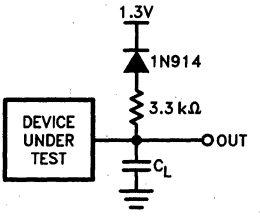
A.C. INPUT/OUTPUT REFERENCE WAVEFORMS



290165-14

A.C. test inputs are driven at V_{OH} (2.4 V_{TTL}) for a logic "1" and V_{OL} (0.45 V_{TTL}) for a logic "0".
Input timing begins at 1.5V.
Output timing ends at V_{IH} (2.0 V_{TTL}) and V_{IL} (0.8 V_{TTL}).
Input Rise and Fall Times (10% to 90%) ≤ 10 ns.

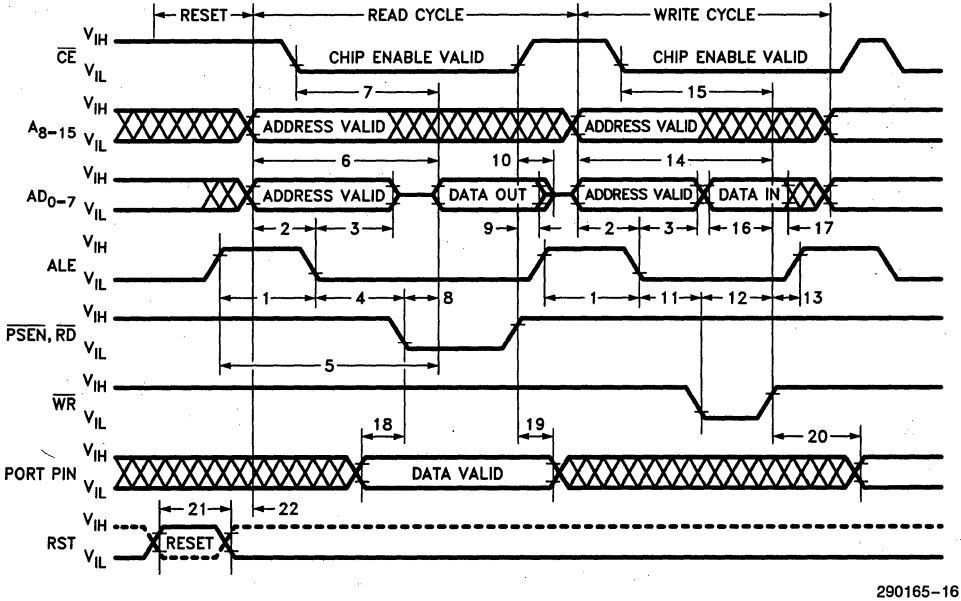
A.C. TESTING LOAD CIRCUIT



290165-15

C_L = 100 pF
C_L Includes Jig Capacitance

READ AND WRITE WAVEFORMS



290165-16

EXPLANATION OF A.C. SYMBOLS

Each timing symbol has five characters. The first is always a "t" (for time). Second and fourth characters represent signal names. Third and fifth represent the signal's logical state. The following list shows character representations.

A: Address
C: Chip Enable or VCC Supply Voltage
D: Data (or instruction)
E: $\overline{PSEN}$ or $\overline{RD}$ Enable
G: $\overline{PGM}$ (Program Strobe)
H: Logic High level
L: ALE or Latch Enable
P: VPP Programming Voltage
Q: Port Output
S: RST (Reset Pin)
T: Time
V: Valid
W: Write Enable
X: No longer a valid "driven" logic level
Z: Float or High-Z level

For example,
t_{AVLL} = Time from Address Valid to ALE Low
t_{LLEL} = Time form ALE Low to Enable ($\overline{PSEN}$ or $\overline{RD}$) Low

A.C. CHARACTERISTICS: READ & WRITE $0°C \leq T_A \leq +70°C$

Parameter		Versions	$V_{CC} \pm 5\%$	87C75PF-200V05		87C75PF-250V05		Unit
No	Symbol	Characteristic	Notes	Min	Max	Min	Max	
1	t_{LHLL}	ALE Pulse Width		50		50		ns
2	t_{AVLL}	Address Valid to ALE Low		7		15		ns
3	t_{LLAX}	Address Hold after ALE Low		20		30		ns
4	t_{LLEL}	ALE Low to $\overline{PSEN}$ or $\overline{RD}$ Low		20		30		ns
5	t_{LHDV}	ALE High to Valid Data			235		305	ns
6	t_{AVDV}	Address Valid to Data Valid	3		200		250	ns
7	t_{CLDV}	$\overline{CE}$ Active to Data Valid	1,3		200		250	ns
8	t_{ELDV}	$\overline{PSEN}$ or $\overline{RD}$ Low to Data Valid	2,3		75		100	ns
9	t_{EHDX}	$\overline{PSEN}$, $\overline{RD}$, $\overline{CE}$, or Address Invalid — Whichever is first — to Data Invalid		0		0		ns
10	t_{EHDZ}	$\overline{PSEN}$ or $\overline{RD}$ High to Data High-Z	4		35		45	ns
11	t_{LLWL}	ALE Low to $\overline{WR}$ Low		20		30		ns
12	t_{WLWH}	$\overline{WR}$ Pulse Width		60		80		ns
13	t_{WHLH}	$\overline{WR}$ High to ALE High		20		30		ns
14	t_{AVWH}	Address Valid to $\overline{WR}$ High		200		250		ns
15	t_{CVWH}	$\overline{CE}$ Active to $\overline{WR}$ High		200		250		ns
16	t_{DVWH}	Data Valid to $\overline{WR}$ High		60		80		ns
17	t_{WHDX}	$\overline{WR}$ High to Data Invalid		10		20		ns
18	t_{QVEL}	Port Input Valid to $\overline{RD}$ Low		15		25		ns
19	t_{EHQX}	Data Hold after $\overline{RD}$ High		0		0		ns
20	t_{WHQV}	$\overline{WR}$ High to Port Output Valid			225		250	ns
21	t_{SVSX}	RST Pulse Width		500		500		ns
22	t_{SXAV}	RST Inactive to Address Valid		0		0		ns

NOTES:
1. t_{CLDV} is 1 μs during inteligent Identifier/NVR Mode.
2. t_{ELDV} is 750 ns during inteligent Identifier/NVR Mode.
3. Output load is 100 pF for t_{AVDV}, t_{CLDV}, and t_{ELDV}.
4. Output Load is 5 pF for t_{EHDZ}, which is measured at high-Z $\pm$500 mV.

PROGRAMMING

Caution: Exceeding 14V on V_{PP} will permanently damage the device.

Program and Data Planes

During programming (V_{PP} = 12.75V), the SFR/RAM plane is not available, only the EPROM and configuration planes (depending on PSR's value) can be accessed. The SFR/RAM plane is accessed only when V_{PP} is V_{IL} or V_{IH}.

Programming the EPROM Plane

The EPROM array is programmed if the plane select register (PSR) contains xxxxxx01b (X1h) when V_{PP} is raised to 12.75V. In an erased device, the EPROM array occupies addresses 0000h through 7FFFh and is programmed at these locations. After programming, the array can be relocated to addresses 8000h–FFFFh by programming the EPROM location register bit ELR.7 (EL) in the configuration plane. Alternately, the EPROM array can be relocated first via ELR.7 and programmed at addresses 8000h–FFFFh.

Programming the Configuration Plane

The configuration plane contains five information bytes. Addresses 0000h and 0001h contain read-only inteligent identifiers. The control level register, EPROM location register, and SFR location register are at 7FFDh, 7FFEh, and 7FFFh. These latter three non-volatile registers (NVRs) are made of EPROM cells. EPROM registers allow the device to be configured, or erased and reconfigured, for various microcontroller architectures.

These registers are programmed if the plane select register (PSR) contains xxxxxx10b (X2h) when V_{PP} is raised to 12.75V. Once this plane is entered, it is programmed and verified just like the EPROM plane.

ERASURE CHARACTERISTICS (FOR CERAMIC, WINDOWED EPROMS)

Exposure to light of wavelength shorter than 4000 Angstroms (Å) begins erasure. Sunlight and some fluorescent lamps have wavelengths in the 3000-4000Å range. Constant exposure to room-level fluorescent light can erase an EPROM in about 3 years (about 1 week for direct sunlight). Opaque labels over the window will prevent unintentional erasure under these lighting conditions.

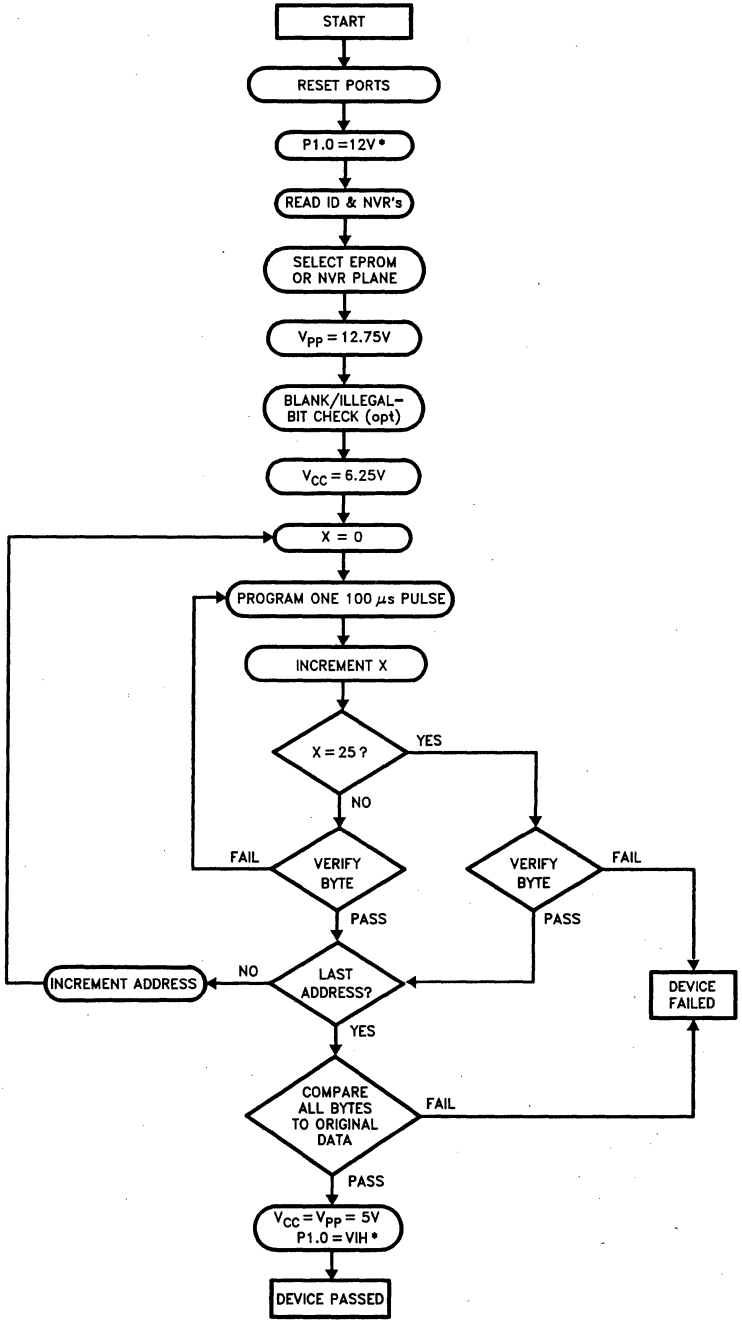
The recommended erasure procedure is exposure to 2537Å ultraviolet light. The minimum integrated Erasure time using a 12000 uW/cm^2 ultraviolet lamp is approximately 15 to 20 minutes. The EPROM should be placed about 1 inch from the lamp. The maximum integrated dose is 7258 Wsec/cm^2 (1 week - 12000 uW/cm^2). High intensity UV light exposure for longer periods can cause permanent damage.

QUICK-PULSE PROGRAMMING™ ALGORITHM

The Quick-Pulse Programming algorithm programs Intel's 87C75PF Port Expander. Developed to substantially reduce production programming throughput time, this algorithm allows optimized programming equipment to program an 87C75PF in under four seconds. Actual programming time depends on the PROM programmer used.

The Quick-Pulse Programming algorithm uses a 100 microsecond initial-pulse followed by a byte verification to determine when the addressed byte is correctly programmed. The algorithm terminates if 25 100us pulses fail to program a byte. Figure 14 shows the 87C75PF Quick-Pulse Programming algorithm flowchart.

The entire program-pulse/byte-verify and final verify sequence is performed with V_{CC} = 6.25V and V_{PP} = 12.75V. When programming is complete, all bytes should be compared to the original data with V_{CC} = 5.0V.



290165-17

*P1.0 = 12V is used to intrnally disable the devices $\overline{RD}$ input for PROM programmers that ground this pin.

Figure 14. 87C75PF Quick-Pulse Programming™ Algorithm

D.C. PROGRAMMING CHARACTERISTICS T_A = 25°C ±5°C

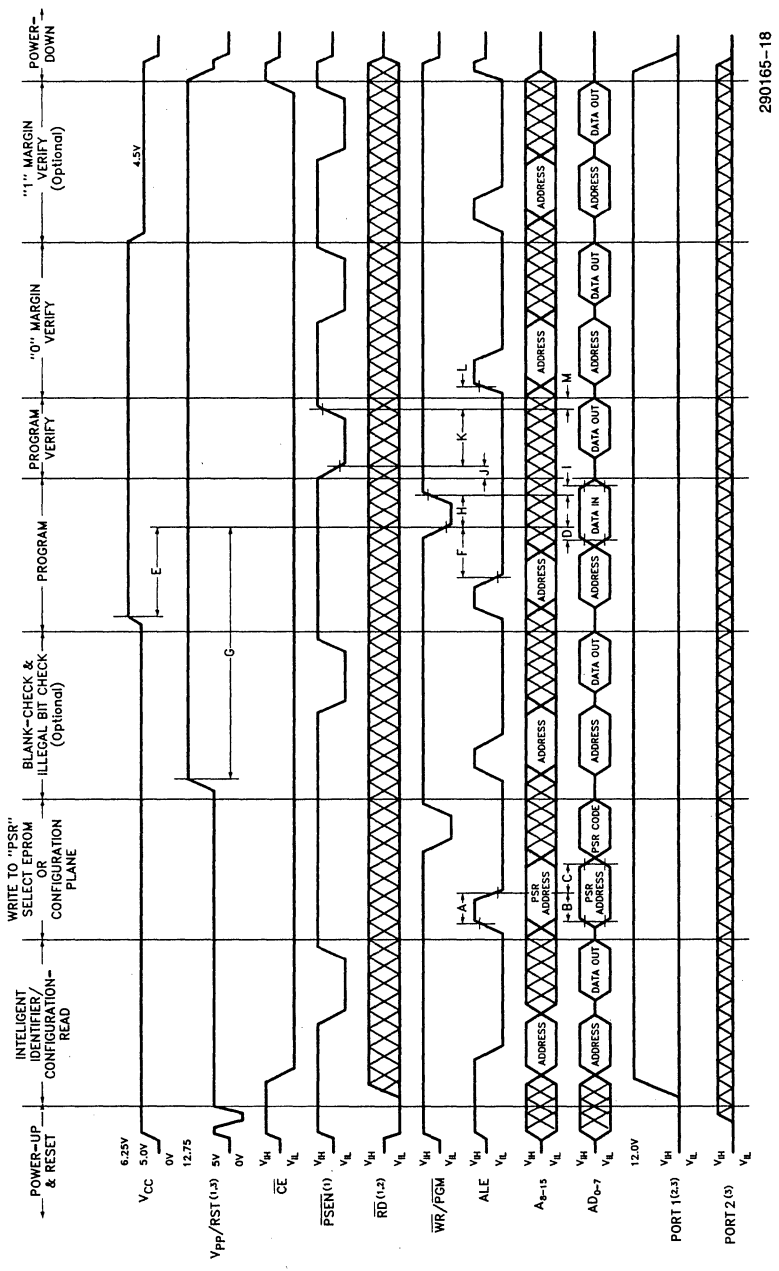
Symbol	Parameter	Notes	Min	Nominal	Max	Units	Test Conditions
I_{LI}	Input Load Current				1.0	μA	$V_{IN} = V_{IL}$ or V_{IH}
I_{CC2}	V_{CC} Supply Current	2			60	mA	$\overline{CE}$ = ALE = V_{IL}
I_{PP}	V_{PP} Supply Current	2			50	mA	$\overline{CE}$ = $\overline{WR}$ = ALE = V_{IL}
V_{IL}	Input Low Voltage		−0.2		0.8	V	
V_{IH}	Input High Voltage		2.0		V_{CC}+0.5V	V	
V_{OL}	Verify Output Low Voltage				0.40	V	I_{OL} = 2.1 mA
V_{OH}	Verify Output High Voltage		V_{CC}−0.8			V	I_{OH} = −400 μA
V_{ID}	P1.0 Identifier/Configuration-Read Voltage		11.0	12.0	13.0	V	
V_{PP}	Programming Voltage	1	12.5	12.75	13.0	V	
V_{CC}	Supply Voltage During Programming	1	6.0	6.25	6.5	V	

NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. Maximum current value is with Address/Data pins AD_{0-7} in write mode; port pins are unloaded.

A.C. PROGRAMMING CHARACTERISTICS T_A = 25°C ±5°C

Parameter		Characteristic	Limits			Units
No	Symbol		Min	Nominal	Max	
A	t_{LHLL}	ALE Pulse Width	100			ns
B	t_{AVLL}	Address Valid to ALE Low	20			ns
C	t_{LLAX}	Address Hold after ALE Low	50			ns
D	t_{DVGL}	Data valid to $\overline{WR}/\overline{PGM}$ Low	2			μs
E	t_{CHGL}	V_{CC} Setup Time to $\overline{WR}/\overline{PGM}$ Low	2			μs
F	t_{LLGL}	ALE to $\overline{WR}/\overline{PGM}$	2			μs
G	t_{PHGL}	V_{PP} Setup Time to $\overline{WR}/\overline{PGM}$ Low	2			μs
H	t_{GLGH}	$\overline{WR}/\overline{PGM}$ Program Pulse Width	95	100	105	μs
I	t_{GHDX}	Data Hold after $\overline{WR}/\overline{PGM}$ High	2			μs
J	t_{DXEL}	Data In Float to $\overline{PSEN}$ or $\overline{RD}$ Low	2			μs
K	t_{ELEH}	$\overline{PSEN}$ or $\overline{RD}$ Verify Pulse Width	150			ns
L	t_{EHLH}	$\overline{PSEN}$ or $\overline{RD}$ High to ALE High	0			ns
M	t_{EHDZ}	$\overline{PSEN}$ or $\overline{RD}$ High to Instruction/Data High-Z	0		35	ns

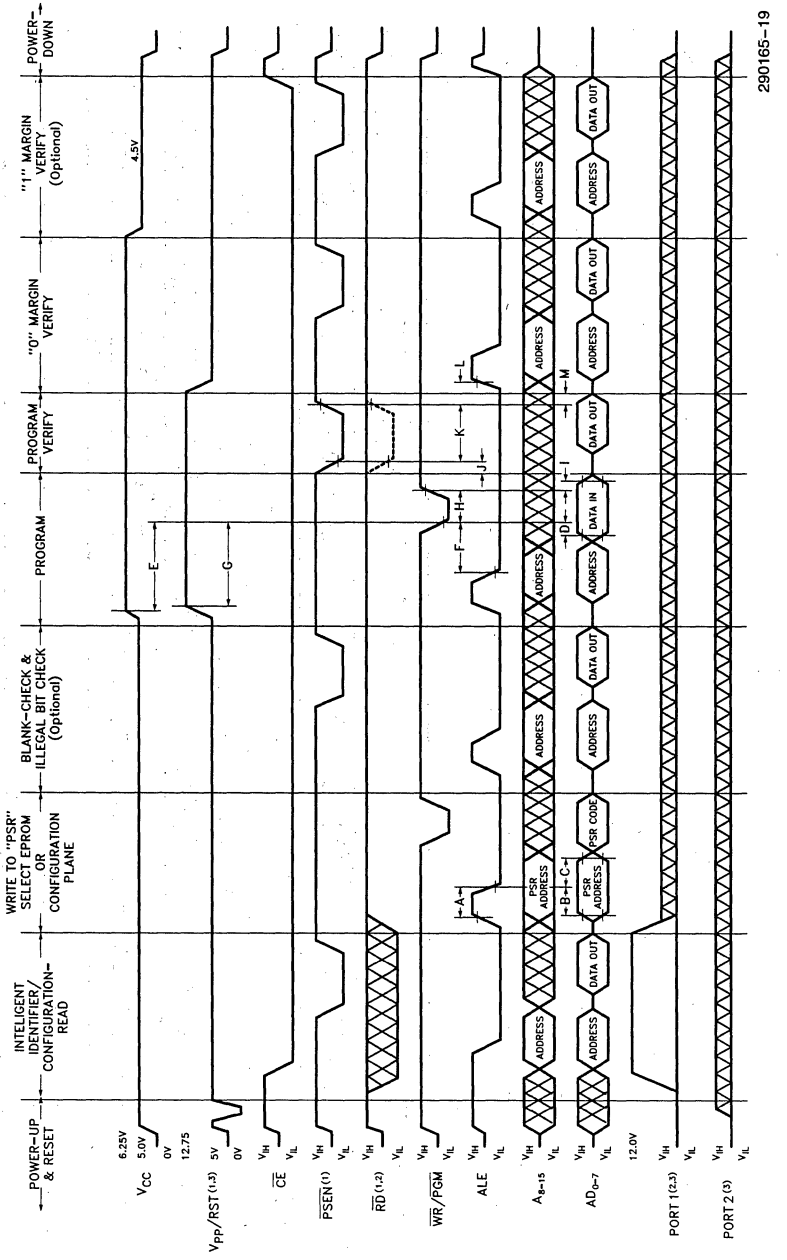
PROGRAMMING WAVEFORMS (For PROM Programmers with $\overline{RD}$ = GND)



NOTES:

1. When V_{PP} = Programming Voltage, either $\overline{PSEN}$ or $\overline{RD}$ will access EPROM (if PSR = x1h) or Configuration (if PSR = x2h) Planes and inteligent Identifier mode is disabled.
2. $\overline{RD}$ is Don't-care when P1.0 = V_H.
3. Port 1 must be RESET or "1" written to P1.0 before V_H is applied to P1.0. All other Port 1 and Port 2 pins should be driven at High-Z or V_{IH} during programming.

PROGRAMMING WAVEFORMS (For PROM Programmers that can have $\overline{RD}$ = V_{IH} or V_{CC})

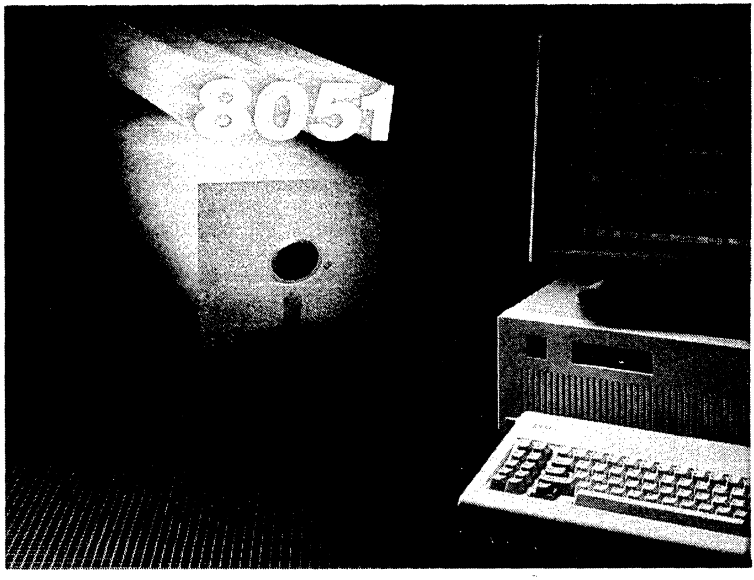


NOTES:
1. When V_{PP} = Programming Voltage, either $\overline{PSEN}$ or $\overline{RD}$ will access EPROM (if PSR = x1h) or Configuraton (if PSR = x2h) Planes and inteligent Identifier mode is disabled.
2. $\overline{RD}$ is Don't-care when P1.0 = V_H.
3. Port 1 must be RESET or "1" written to P1.0 before V_H is applied to P1.0. All other Port 1 and Port 2 pins should be driven at High-Z or V_{IH} during programming.

MCS®-51 Development Support Tools

15

8051 SOFTWARE DEVELOPMENT PACKAGES



COMPLETE SOFTWARE DEVELOPMENT SUPPORT FOR THE MCS®-51 FAMILY OF MICROCONTROLLERS

Intel supports application development for its MCS®-51 family of microcontrollers with a complete set of development languages and utilities. These tools include a macroassembler, a PL/M compiler, linker/relocator program, a librarian utility, and an object-to-hex utility. Develop code in the language(s) you desire, then combine object modules from different languages into a single, fast program. These tools were designed to work with each other, with the MCS-51 architecture, and with the Intel ICE5100 in-circuit emulator.

FEATURES

- Support for all members of the Intel MCS-51 family of embedded microcontrollers
- ASM-51 Macroassembler
- PL/M-51 high-level language
- Linker/Relocator program
- Library utility
- Object to hexadecimal converter
- Hosted on IBM PC XT/AT, V. 3.0 or later
- Worldwide service and support

September, 1988
Order Number: 280819-001

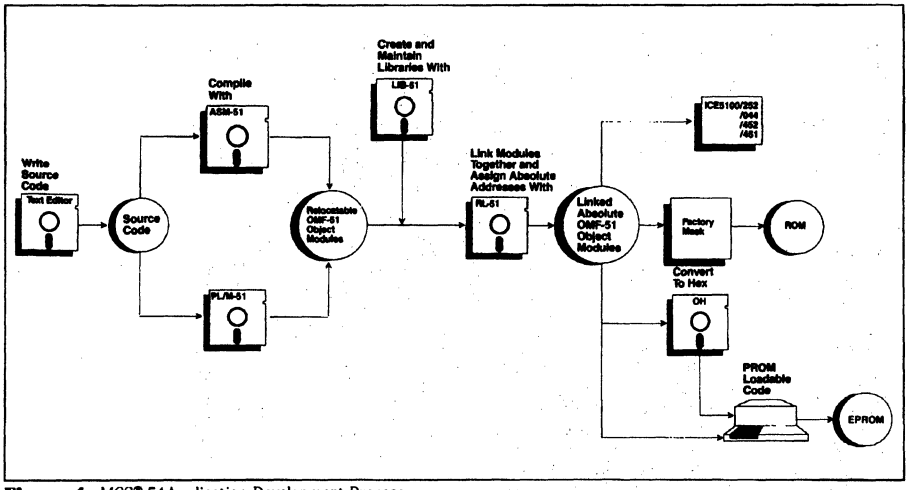


Figure 1. MCS®-51Application Development Process

ASM-51 MACROASSEMBLER

ASM-51 is the macroassembler for the MCS-51 family of microcontrollers. ASM-51 provides full and accurate support for all of the specific component's instructions. It also provides symbolic access to the many features of the MCS-51 family of microcontrollers. Also provided is an "include" file with all the appropriate component registers and memory spaces defined.

The macro facility in ASM-51 saves development and maintenance time, since common code sequences need only be developed once.

PL/M-51 COMPILER

PL/M-51 is a high-level language designed to support the software requirements of the MCS-51 family of microcontrollers. The PL/M-51 compiler translates PL/M high-level language statements into MCS-51 relocatable object code. Major features of the PL/M-51 compiler include:

- **Structured programming for ease of maintenance and enhancement.** The PL/M-51 language supports modular and structured programming, making programs easier to understand, maintain, and debug.
- **Data types facilitate various common functions.** PL/M-51 supports three data types to facilitate various arithmetic, logic and address functions. The language also uses BASED variables that map more than one variable to the same memory location to save memory space.
- **Interrupt attribute speeds coding effort.** The INTERRUPT attribute allows you to easily define interrupt handling procedures. The compiler will generate code to save and restore the program status word for INTERRUPT procedures.
- **Code optimization reduces memory requirements.** The PL/M-51 compiler has four different levels of optimization for significantly reducing the size of the program.
- **Language compatibility saves development time.** PL/M-51 object modules are compatible with object modules generated by all other MCS-51 language translators. This compatibility allows for easy linking of all modules and the ability to do symbolic debugging with the Intel ICE5100 in-circuit emulator.

RL-51 Linker/Relocator

Intel's RL-51 utility is used to link multiple MCS-51 object modules into a single program, resolve all references between modules and assign absolute addresses to all relocatable segments. Modules can be written in either ASM-51 or PL/M-51.

LIB-51

The Intel LIB-51 utility creates and maintains libraries of software object modules. Standard modules can be placed in a library and linked into your applications programs using RL-51. When using libraries, the linker will link only those modules that are required to satisfy external references.

LIB-51 and RL-51 make it easy to reuse software modules that are fully debugged and used by various applications, thus shortening the software development cycle.

OH OBJECT TO HEXADECIMAL CONVERTER

The OH utility converts Intel OMF-51 object modules into standard hexadecimal format. This allows the code to be loaded directly into PROM via non-Intel PROM programmers.

SERVICE, SUPPORT, AND TRAINING

Intel augments its MCS-51 architecture family of development tools with a full array of seminars, classes, and workshops; on-site consulting services; field application engineering expertise; telephone hot-line support; and software and hardware maintenance contracts. This full line of services will ensure your design success.

ORDERING INFORMATION

D86ASM51*	MCS-51 Assembler for PC XT or AT system (or compatible), running DOS 3.0 or higher
D86PLM51*	PL/M-51 Software Package for PC XT or AT system (or compatible), running DOS 3.0 or higher

*Also Includes: Relocator/Linker, Object-to-hex converter, and Librarian.

AEDIT SOURCE CODE AND TEXT EDITOR

PROGRAMMER SUPPORT

AEDIT is a full-screen text editing system designed specifically for software engineers and technical writers. With the facilities for automatic program block indentation, HEX display and input, and full macro support, AEDIT is an essential tool for any programming environment. And with AEDIT, the output file is the pure ASCII text (or HEX code) you input—no special characters or proprietary formats.

Dual file editing means you can create source code and its supporting documents at the same time. Keep your program listing with its errors in the background for easy reference while correcting the source in the foreground. Using the split-screen windowing capability, it is easy to compare two files, or copy text from one to the other. The DOS system-escape command eliminates the need to leave the editor to compile a program, get a directory listing, or execute any other program executable at the DOS system level.

There are no limits placed on the size of the file or the length of the lines processed with AEDIT. It even has a batch mode for those times when you need to make automatic string substitutions or insertions in a number of separate text files.

AEDIT FEATURES

- Complete range of editing support—from document processing to HEX code entry and modification
- Supports system escape for quick execution of PC-DOS System level commands
- Full macro support for complex or repetitive editing tasks
- Hosted on PC-DOS and RMX operating systems
- Dual file support with optional split-screen windowing
- No limit to file size or line length
- Quick response with an easy to use menu driven interface
- Configurable and extensible for complete control of the editing process

September, 1988
Order Number: 280804-002

FEATURES

POWERFUL TEXT EDITOR

As a text editor, AEDIT is versatile and complete. In addition to simple character insertion and cursor positioning commands, AEDIT supports a number of text block processing commands. Using these commands you can easily move, copy, or delete both small and large blocks of text. AEDIT also provides facilities for forward or reverse string searches, string replacement and query replace.

AEDIT removes the restriction of only inserting characters when adding or modifying text. When adding text with AEDIT you may choose to either insert characters at the current cursor location, or over-write the existing text as you type. This flexibility simplifies the creation and editing of tables and charts.

USER INTERFACE

The menu-driven interface AEDIT provides makes it unnecessary to memorize long lists of commands and their syntax. Instead, a complete list of the commands or options available at any point is always displayed at the bottom of the screen. This makes AEDIT both easy to learn and easy to use.

FULL FLEXIBILITY

In addition to the standard PC terminal support provided with AEDIT, you are able to configure AEDIT to work with almost any terminal. This along with user-definable macros and full adjustable tabs, margins, and case sensitivity combine to make AEDIT one of the most flexible editors available today.

MACRO SUPPORT

AEDIT will create macros by simply keeping track of the command and text that you type, "learning" the function the macro is to perform. The editor remembers your actions for later execution, or you may store them in a file to use in a later editing session.

Alternatively, you can design a macro using AEDIT's powerful macro language. Included with the editor is an extensive library of useful macros which you may use or modify to meet your individual editing needs.

TEXT PROCESSING

For your documentation needs, paragraph filling or justification simplifies the chore of document formatting. Automatic carriage return insertion means you can focus on the content of what you are typing instead of how close you are to the edge of the screen.

SERVICE, SUPPORT, AND TRAINING

Intel augments its development tools with a full array of seminars, classes, and workshops; on-site consulting services; field application engineering expertise; telephone hot-line support; and software and hardware maintenance contracts. This full line of services will ensure your design success.

SPECIFICATIONS

HOST SYSTEM

AEDIT for PC-DOS has been designed to run on the IBM* PC XT, IBM PC AT, and compatibles. It has been tested and evaluated for the PC-DOS 3.0 or greater operating system.

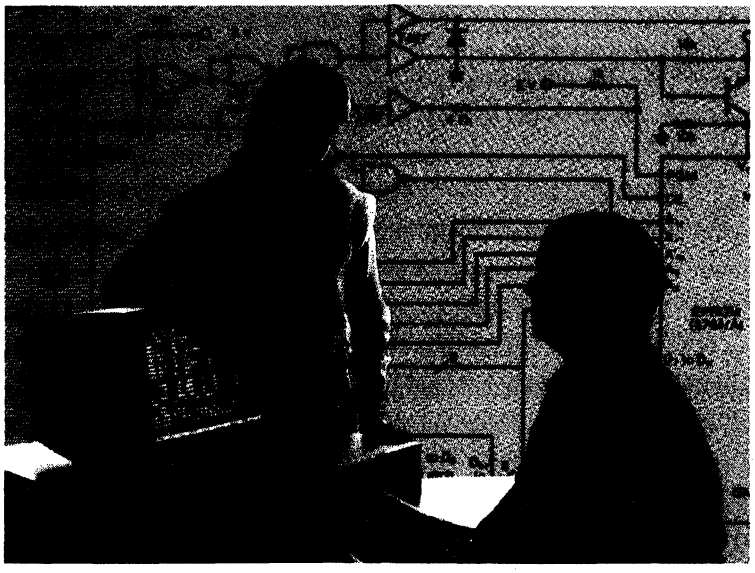
Versions of AEDIT are available for the iRMX™-86 and RMX II Operating System.

ORDERING INFORMATION

D86EDINL	AEDIT Source Code Editor Release 2.2 for PC-DOS with supporting documentation
122716	AEDIT-DOS Users Guide
122721	AEDIT-DOS Pocket Reference
RMX864WSU	AEDIT for iRMX-86 Operating System
R286EDI286EU	AEDIT for iRMX II Operating System

For direct information on Intel's Development Tools, or for the number of your nearest sales office or distributor, call 800-874-6835 (U.S.). For information or literature on additional Intel products, call 800-548-4725 (U.S. and Canada).

ICE™-5100/252 In-Circuit Emulator



IN-CIRCUIT EMULATOR FOR THE MCS® 51 FAMILY OF MICROCONTROLLERS

The ICE-5100/252 In-Circuit Emulator is a complete hardware/software debug environment for developing embedded control applications based on the Intel MCS-51 family of microcontrollers. With high-performance 16 MHz emulation, symbolic debugging, and flexible memory mapping, the ICE-5100/252 emulator expedites all stages of development: hardware development, software development, system integration, and system test; shortening your project's time to market.

FEATURES

- Full speed to 16 MHz.
- 64KB of emulation mapped memory.
- 254 frames of execution trace.
- Symbolic debug.
- Serial link to an IBM PC XT, AT, 100% compatible.
- Four address breakpoints with in-range, out-of-range, and page breaks.
- On-line disassembler and single line assembler.
- Source code display.
- ASM-51 and PL/M-51 language support.
- Pop-up help.
- DOS shell escape.
- On-line tutorial.
- Built-in CRT based editor.
- System self-test diagnostics.
- Worldwide service and support.

September, 1988

Order Number: 280798-002

ONE TOOL FOR ENTIRE DEVELOPMENT CYCLE

The ICE-5100/252 emulator speeds target system development by allowing hardware and software design to proceed simultaneously. You can develop software even before prototype hardware is finished. And because the ICE-5100/252 emulator precisely matches the component's electrical and timing characteristics, it's a valuable tool for hardware development and debug. Thus, the ICE-5100/252 emulator can debug a prototype or production system at any stage in its development, without introducing extraneous hardware or software test tools.

HIGH-SPEED, REAL-TIME EMULATION

The ICE-5100/252 emulator provides full-speed, real-time emulation up to 16 MHz. Because the emulator is fully transparent to the target system, you have complete control over hardware and software debug and system integration.

64KB of zero wait-state emulation memory is available to replace target system code memory, allowing software debug to begin even before prototype hardware is finished.

FLEXIBLE BREAKPOINTING FOR QUICK PROBLEM ISOLATION

The ICE-5100/252 emulator supports three different types of break specifications: specific address breaks on up to 64,000 possible addresses; range breaks, both within and outside a user-defined range; and page breaks, up to 256 pages on 256-byte boundaries. 254 frames of execution trace memory provide ample debug information, with each frame divided into 16 bits of program execution address and 8 bits of external event information. A maximum of four tracepoints allows qualified trace for a variety of debug conditions.

SYMBOLIC DEBUGGING FOR FAST DEVELOPMENT

Design team productivity is enhanced by the use of symbolic debug references to program line, high-level statements, and module and variable names. The terms used to develop programs are the same used for system debugging.

PATCH CODE WITHOUT RECOMPILING

Code-patching is easy with the ICE-5100/252 emulator's single-line assembler. Machine code can be disassembled to mnemonics for significantly easier debugging and project development.

EASY TO LEARN AND USE

The ICE-5100/252 is accompanied by a full tutorial that explains all system functions and provides many examples. Additional features such as on-line help, a built-in CRT-based editor, and DOS shell escape make the emulator fast and easy to use for both novice and experienced users. You can develop your own test suites or save frequently-used debug routines as debug procedures (PROCs) that can be invoked with a single command.

WORLDWIDE SERVICE AND SUPPORT

The ICE-5100/252 emulator is supported by Intel's worldwide service and support organization. In addition to an extended warranty, you can choose from hotline support, on-site system engineering assistance, and a variety of hands-on training workshops.

ICE™-5100/252 Emulator Supported Components

Part	On-Chip Program Memory	On-Chip Data Memory
8031	None	128 bytes
8051	4K ROM	128 bytes
8751	4K EPROM	128 bytes
80C31	None	128 bytes
80C51	4K ROM	128 bytes
87C51	4K EPROM	128 bytes
8032	None	256 bytes
8052	8K ROM	256 bytes
8752	8K EPROM	256 bytes
80C51FA	None	256 bytes
83C51FA	8K ROM	256 bytes
87C51FA	8K EPROM	256 bytes
80C51FB	None	256 bytes
83C51FB	16K ROM	256 bytes
87C51FB	16K EPROM	256 bytes

ELECTRICAL CONSIDERATIONS

The emulation processor's user-pin timings and loadings are identical to the 80C51FA component except as follows.

Maximum Operating ICC and Idle ICC (ma)*

V_{cc}	Maximum Operating ICC (ma)*			Maximum Idle ICC (ma)**		
	4V	5V	6V	4V	5V	6V
	Frequency					
0.5 MHz	0.87	1.62	3.0	0.58	1.21	2.5
3.5 MHz	4.8	6.82	9.76	2.2	4.97	6.33
8.0 MHz	10.5	15.0	20.5	6.0	8.98	11.76
12.0 MHz	15.2	22.2	30.2	9.2	13.34	17.46
16.0 MHz	19.4	28.6	38.7	11.8	17.4	23.4

*ICC is measured with all output pins disconnected
XTAL1 driven with TCLCH,TCHCL = 10ns, $V_{il} = V_{ss} + .5V$, $V_{ih} = V_{cc} - .5V$. XTAL2 not connected.
For maximum operating ICC
$\overline{EA}$ = RST = Port0 = V_{cc}.
**For maximum idle ICC
$\overline{EA}$ = Port0 = V_{cc}, RST = V_{cc}, internal clock to PCA gated off.

- Up to 25 pf of additional pin capacitance is contributed by the processor module and target adaptor assemblies.
- Pin 31, $\overline{EA}$, has approximately 32 pf of additional capacitance loading due to sensing circuitry.
- Pins 18 and 19, XTAL1 and XTAL2, respectively, have approximately 15 to 16 pf of additional capacitance when configured for crystal operation.

PROCESSOR MODULE DIMENSIONS

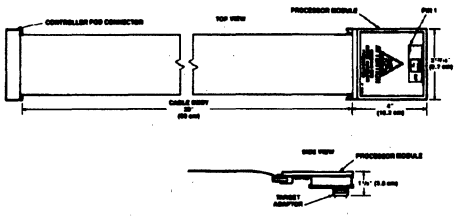


Figure 1. Processor Module Dimensions

CHMOS AND HMOS DESIGN DIFFERENCES

Chip Function	HMOS Component 8031	CHMOS Component 80C31
RST trigger threshold	2.5V	70% V_{cc}(3.5V @ V_{cc} = 5V)
RST input impedance	4K-10K ohms	50K-150K ohms
Port 1_{il}	−800μA	−50μA
Clock threshold	2.5V	70% V_{cc}(3.5V @ V_{cc} = 5V)

SPECIFICATIONS

Host Requirements:

IBM PC-XT, AT or compatible
PC-DOS 3.0 or later
512K RAM
One floppy drive and hard disk

Physical Characteristics:

The ICE-5100/252 emulator consists of the following components:

Unit	Width		Height		Length	
	Inch	Cm	Inch	Cm	Inch	Cm
Controller Pod	8.25	21.0	1.5	3.8	13.5	34.3
User Cable					39.0	99.0
Processor Module*	3.8	9.7	1.5	3.8	4.0	10.2
Power Supply	7.6	18.1	4.0	10.2	11.0	28.0
Serial Cable					144.0	360.0

*with supplied target adaptor.

Electrical Characteristics:

Power supply
100-120V or 220-240V selectable
50-60 Hz
2 amps (AC max) @ 120V
1 amp (AC max) @ 240V

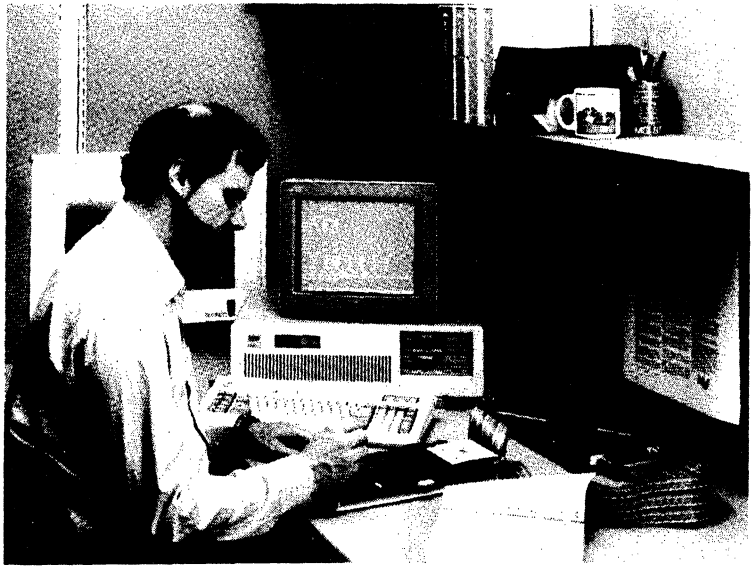
Environmental Characteristics:

Operating temperature: +10°C to +40°C (50°F to 104°F)
Operating humidity: Maximum of 85% relative humidity, non-condensing

ORDERING INFORMATION

Order Code	Description
pI252KITAD	Kit contains ICE-5100/252 user probe assembly, power supply and cables, serial cables, target adapter, crystal power accessory, emulator controller pod, emulator software, DOS host communication, ASM-51 and AEDIT text editor (requires software license).
pI252KITD	Kit contains the same components as pI252KITAD, excluding ASM-51 and the AEDIT text editor (requires software license).
pC252KITD	Conversion kit for ICE-5100/452, ICE-5100/451, or ICE-5100/044 running PC-DOS 3.0 or later, to provide emulation support for MCS-51 components (requires software license).
TA252D	Target adapter converting 48-pin DIP to 44-pin PLCC package.
D86ASM51	ASM/RL 51 package for PC-DOS (requires software license).
D86PLM51	PL/M/RL 51 package for PC-DOS (requires software license).
D86EDINL	AEDIT text editor for PC-DOS.

ICE™-5100/451 In-Circuit Emulator



IN-CIRCUIT EMULATOR FOR THE MCS™-51 FAMILY OF MICROCONTROLLERS

The ICE-5100/451 In-Circuit Emulator is a complete hardware/software debug environment for developing embedded control applications based on the Intel MCS 51 family of microcontrollers. With high-performance 12 MHz emulation, symbolic debugging, and flexible memory mapping, the ICE-5100/451 emulator expedites all stages of development: hardware development, software development, system integration, and system test; shortening your project's time to market.

FEATURES

- Full speed to 12 MHz.
- 64KB of emulation mapped memory.
- 254 frames of execution trace.
- Symbolic debug.
- Serial link to an IBM PC XT, AT, 100% compatible.
- Four address breakpoints with in-range, out-of-range, and page breaks.
- On-line disassembler and single line assembler.
- Source code display.
- ASM-51 and PL/M-51 language support.
- Pop-up help.
- DOS shell escape.
- On-line tutorial.
- Built-in CRT based editor.
- System self-test diagnostics.
- Worldwide service and support.

September, 1988
Order Number: 280802-002

ONE TOOL FOR ENTIRE DEVELOPMENT CYCLE

The ICE-5100/451 emulator speeds target system development by allowing hardware and software design to proceed simultaneously. You can develop software even before prototype hardware is finished. And because the ICE-5100/451 emulator precisely matches the component's electrical and timing characteristics, it's a valuable tool for hardware development and debug. Thus, the ICE-5100/451 emulator can debug a prototype or production system at any stage in its development, without introducing extraneous hardware or software test tools.

HIGH-SPEED, REAL-TIME EMULATION

The ICE-5100/451 emulator provides full-speed, real-time emulation up to 12 MHz. Because the emulator is fully transparent to the target system, you have complete control over hardware and software debug and system integration.

64KB of zero wait-state emulation memory is available to replace target system code memory, allowing software debug to begin even before prototype hardware is finished.

FLEXIBLE BREAKPOINTING FOR QUICK PROBLEM ISOLATION

The ICE-5100/451 emulator supports three different types of break specifications: specific address breaks on up to 64,000 possible addresses; range breaks, both within and outside a user-defined range; and page breaks, up to 256 pages on 256-byte boundaries. 254 frames of execution trace memory provide ample debug information, with each frame divided into 16 bits of program execution address and 8 bits of external event information. A maximum of four tracepoints allows qualified trace for a variety of debug conditions.

SYMBOLIC DEBUGGING FOR FAST DEVELOPMENT

Design team productivity is enhanced by the use of symbolic debug references to program line, high-level statements, and module and variable names. The terms used to develop programs are the same used for system debugging.

PATCH CODE WITHOUT RECOMPILING

Code-patching is easy with the ICE-5100/451 emulator's single-line assembler. Machine code can be disassembled to mnemonics for significantly easier debugging and project development.

EASY TO LEARN AND USE

The ICE-5100/451 is accompanied by a full tutorial that explains all system functions and provides many examples. Additional features such as on-line help, a built-in CRT-based editor, and DOS shell escape make the emulator fast and easy to use for both novice and experienced users. You can develop your own test suites or save frequently-used debug routines as debug procedures (PROCs) that can be invoked with a single command.

WORLDWIDE SERVICE AND SUPPORT

The ICE-5100/451 emulator is supported by Intel's worldwide service and support organization. In addition to an extended warranty, you can choose from hotline support, on-site system engineering assistance, and a variety of hands-on training workshops.

ELECTRICAL CONSIDERATIONS

The emulation processor's user-pin timings and loadings are identical to the 80C451 component except as follows.

Maximum Operating ICC and Idle ICC (ma)*

V_{cc}	Maximum Operating ICC (ma)*	Maximum Idle ICC (ma)**
	5V	5V
0.5 MHz	4.97	4.93
3.5 MHz	9.53	10.39
8.0 MHz	13.0	12.75
12.0 MHz	22.2	18.19

*ICC is measured with all output pins disconnected
XTAL1 driven with TCLCH,TCHCL = 5ns, $V_{il} = V_{ss} + 0.5V$,
$V_{ih} = V_{cc} - 0.5V$. XTAL2 not connected.
For maximum operating ICC
$\overline{EA}$ = RST = Port0 = V_{cc}.

**Maximum idle ICC is measured with all output pins disconnected.
XTAL1 driven with TCLCH, TCHCL = 5ns
$V_{il} = V_{ss} + 0.5V$,
$V_{ih} = V_{cc} - 0.5V$,
XTAL2 not connected,
Port 0 = V_{cc},
$\overline{EA}$ = RST = V_{cc}.
Up to 25 pf of additional pin capacitance is contributed by the processor module and target adapter assemblies. Pin 1 (EA) has approximately 32 pf of additional capacitance loading due to sensing circuitry. Pins 53 and 52, XTAL1 and XTAL2, respectively, have approximately 15 to 16 pf of additional capacitance when configured for crystal operation.

PROCESSOR MODULE DIMENSIONS

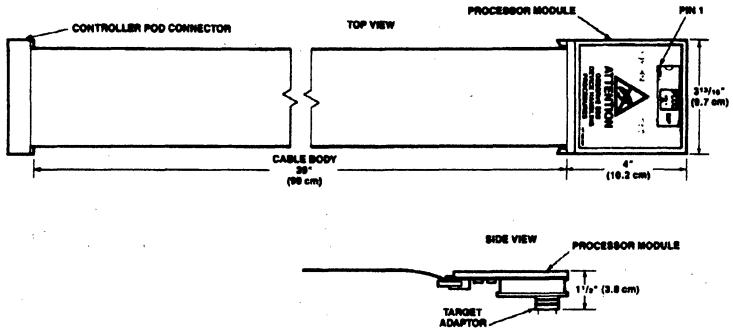


Figure 1: Processor Module Dimensions

CHMOS DESIGN DIFFERENCES

Chip Function	CHMOS Component 80C31
RST trigger threshold	70% V_{cc}(3.5V @ V_{cc} = 5V)
RST input impedance	50K-150K ohms
Port 1_{il}	$-50\mu A$
Clock threshold	70% V_{cc}(3.5V @ V_{cc} = 5V)

SPECIFICATIONS

Host Requirements:

IBM PC-XT, AT or compatible
PC-DOS 3.0 or later
512K RAM
One floppy drive and hard disk

Physical Characteristics:

The ICE-5100/451 emulator consists of the following components:

Unit	Width		Height		Length	
	Inch	Cm	Inch	Cm	Inch	Cm
Controller Pod	8.25	21.0	1.5	3.8	13.5	34.3
User Cable					39.0	99.0
Processor Module*	3.8	9.7	1.5	3.8	4.0	10.2
Power Supply	7.6	18.1	4.0	10.2	11.0	28.0
Serial Cable					144.0	360.0

*with supplied target adapter.

Electrical Characteristics:

Power supply
100-120V or 220-240V selectable
50-60 Hz
2 amps (AC max) @ 120V
1 amp (AC max) @ 240V

Environmental Characteristics:

Operating temperature: + 10°C to + 40°C (50°F to 104°F)
Operating humidity: Maximum of 85% relative humidity, non-condensing

ORDERING INFORMATION

Order Code	Description
pl451KITAD	Kit contains ICE-5100/451 user probe assembly, power supply and cables, serial cables, target adapter, crystal power accessory, emulator controller pod, emulator software, DOS host communication cables, ASM-51 and AEDIT text editor (requires software license).
pl451KITD	Kit contains the same components as pl451KITAD, excluding ASM-51 and the AEDIT text editor (requires software license).
pC451KITD	Conversion kit for ICE-5100/452, ICE-5100/252, or ICE-5100/044 running PC-DOS 3.0 or later, to provide emulation support for 80C451 components (requires software license).
TA451E	Target adapter for 68-pin PLCC package support.
D86ASM51	ASM/RL 51 package for PC-DOS (requires software license).
D86PLM51	PL/M/RL 51 package for PC-DOS (requires software license).
D86EDINL	AEDIT text editor for PC-DOS.

ICE™-5100/044 In-Circuit Emulator

IN-CIRCUIT EMULATOR FOR THE RUPI™-44 FAMILY OF PERIPHERALS

The ICE-5100/044 In-Circuit Emulator is a complete hardware/software debug environment for developing embedded control applications based on the Intel RUPI™-44 family of peripherals, including the 8044-based BITBUS™ board products. With high-performance 12 MHz emulation, symbolic debugging, and flexible memory mapping, the ICE-5100/044 emulator expedites all stages of development: hardware development, software development, system integration, and system test; shortening your project's time to market.

FEATURES

- Full speed to 12 MHz.
- 64KB of emulation mapped memory.
- 254 frames of execution trace.
- Symbolic debug.
- Serial link to an IBM PC XT, AT, 100% compatible.
- Four address breakpoints with in-range, out-of-range, and page breaks.
- On-line disassembler and single line assembler.
- Source code display.
- ASM-51 and PL/M-51 language support.
- Pop-up help.
- DOS shell escape.
- On-line tutorial.
- Built-in CRT based editor.
- System self-test diagnostics.
- Worldwide service and support.

intel

September, 1988

Order Number: 280818-001

ONE TOOL FOR ENTIRE DEVELOPMENT CYCLE

The ICE-5100/044 emulator speeds target system development by allowing hardware and software design to proceed simultaneously. You can develop software even before prototype hardware is finished. And because the ICE-5100/044 emulator precisely matches the component's electrical and timing characteristics, it's a valuable tool for hardware development and debug. Thus, the ICE-5100/044 emulator can debug a prototype or production system at any stage in its development, without introducing extraneous hardware or software test tools.

HIGH-SPEED, REAL-TIME EMULATION

The ICE-5100/044 emulator provides full-speed, real-time emulation up to 12 MHz. Because the emulator is fully transparent to the target system, you have complete control over hardware and software debug and system integration.

64KB of zero wait-state emulation memory is available to replace target system code memory, allowing software debug to begin even before prototype hardware is finished.

FLEXIBLE BREAKPOINTING FOR QUICK PROBLEM ISOLATION

The ICE-5100/044 emulator supports three different types of break specifications: specific address breaks on up to 64,000 possible addresses; range breaks, both within and outside a user-defined range; and page breaks, up to 256 pages on 256-byte boundaries. 254 frames of execution trace memory provide ample debug information, with each frame divided into 16 bits of program execution address and 8 bits of external event information. A maximum of four tracepoints allows qualified trace for a variety of debug conditions.

SYMBOLIC DEBUGGING FOR FAST DEVELOPMENT

Design team productivity is enhanced by the use of symbolic debug references to program line, high-level statements, and module and variable names. The terms used to develop programs are the same used for system debugging.

PATCH CODE WITHOUT RECOMPILING

Code-patching is easy with the ICE-5100/044 emulator's single-line assembler. Machine code can be disassembled to mnemonics for significantly easier debugging and project development.

EASY TO LEARN AND USE

The ICE-5100/044 is accompanied by a full tutorial that explains all system functions and provides many examples. Additional features such as on-line help, a built-in CRT-based editor, and DOS shell escape make the emulator fast and easy to use for both novice and experienced users. You can develop your own test suites or save frequently-used debug routines as debug procedures (PROCs) that can be invoked with a single command.

WORLDWIDE SERVICE AND SUPPORT

The ICE-5100/044 emulator is supported by Intel's worldwide service and support organization. In addition to an extended warranty, you can choose from hotline support, on-site system engineering assistance, and a variety of hands-on training workshops.

ELECTRICAL CONSIDERATIONS

The emulation processor's user-pin timings and loadings are identical to the 8044 component except as follows.

- Up to 25 pf of additional pin capacitance is contributed by the processor module and target adaptor assemblies.
- Pin 31, $\overline{EA}$, has approximately 32 pf of additional capacitance loading due to sensing circuitry.
- Pins 18 and 19, XTAL1 and XTAL2, respectively, have approximately 15 to 16 pf of additional capacitance when configured for crystal operation.

DESIGN CONSIDERATIONS

Execution of user programs that contain interrupt routines causes incorrect data to be stored in the trace buffer. When an interrupt occurs, the next instruction to be executed is placed into the trace buffer before it is actually executed. Following completion of the interrupt routine, the instruction is executed and again placed into the trace buffer.

PROCESSOR MODULE DIMENSIONS

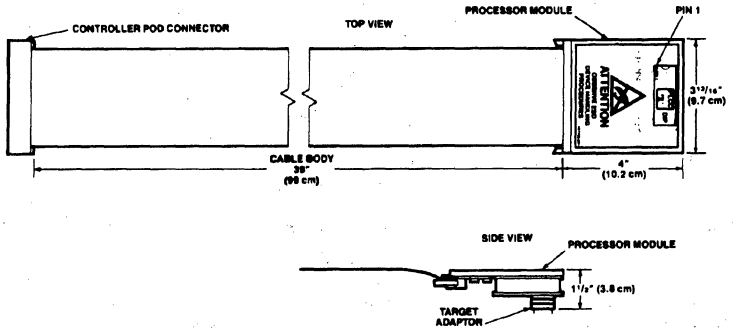


Figure 1. Processor Module Dimensions

SPECIFICATIONS

Host Requirements:

IBM PC-XT, AT or compatible
PC-DOS 3.0 or later
512K RAM
One floppy drive and hard disk

Physical Characteristics:

The ICE-5100/044 emulator consists of the following components:

Unit	Width		Height		Length	
	Inch	Cm	Inch	Cm	Inch	Cm
Controller Pod	8.25	21.0	1.5	3.8	13.5	34.3
User Cable					39.0	99.0
Processor Module*	3.8	9.7	1.5	3.8	4.0	10.2
Power Supply	7.6	18.1	4.0	10.2	11.0	28.0
Serial Cable					144.0	360.0

*with supplied target adaptor.

Electrical Characteristics:

Power supply
100-120V or 220-240V selectable
50-60 Hz
2 amps (AC max) @ 120V
1 amp (AC max) @ 240V

Environmental Characteristics:

Operating temperature: + 10°C to + 40°C (50°F to 104°F)
Operating humidity: Maximum of 85% relative humidity, non-condensing

ORDERING INFORMATION

Order Code	Description
pI044KITAD	Kit contains ICE-5100/044 user probe assembly, power supply and cables, serial cables, target adapter, crystal power accessory, emulator controller pod, emulator software, DOS host communication, ASM-51 and AEDIT text editor (requires software license).
pI044KITD	Kit contains the same components as pI044KITAD, excluding ASM-51 and the AEDIT text editor (requires software license).
pC044KITD	Conversion kit for ICE-5100/452, ICE-5100/451, or ICE-5100/252 running PC-DOS 3.0 or later, to provide emulation support for RVPI-44 family of peripherals (requires software license).
D86ASM51	ASM/RL 51 package for PC-DOS (requires software license).
D86PLM51	PL/M/RL 51 package for PC-DOS (requires software license).
D86EDINL	AEDIT text editor for PC-DOS.

ASIC: Intel Cell-Based Design

16

INTRODUCTION TO THE INTEL 1.5 MICRON CHMOS III CELL LIBRARY

PREFACE

A growing number of system designers are turning to Application Specific Integrated Circuits (ASICs) for the solution to their system design needs. ASIC design methodologies bridge the gap between standard ICs and fully-customized devices. Using sophisticated software design tools and a collection of predefined circuit elements, system designers realize the benefits of custom ICs without incurring the high cost and long development times associated with full custom designs.

The Intel Design Environment provides a flexible system for designing and manufacturing Intel ASIC products. The design environment includes a comprehensive set of CAE/CAD tools, design libraries running on the Daisy and Mentor engineering workstations, and design and manufacturing services. Intel cell-based designs are backed by a proven manufacturing capability and the same strict adherence to quality and reliability applied to Intel standard products.

This chapter introduces the Intel 1.5 Micron CHMOS III Cell Library. It provides cell data specifications and describes Intel design services as they relate to cell-based designs executed at Intel design centers or at the customer site.

The Intel 1.5 Micron CHMOS III Cell Library provides customers with the building blocks necessary to design complex semicustom integrated circuits. The Intel 1.5 Micron CHMOS III Cell Library is composed of pre-designed, fully characterized equivalents of common circuit elements and Intel standard products. The library includes SSI, MSI, and LSI circuit elements commonly used in system design. In addition, the library contains VLSiCEL™ elements, functional equivalents of Intel standard microcontroller, microprocessor, and microprocessor support peripheral products.

FEATURES/BENEFITS

- Over 150 SSI/MSI/LSI logic functions.
- A comprehensive cell library provides a wide range of cell-based ASIC solutions. The library is highlighted by VLSiCEL elements, cell versions of popular Intel standard microprocessors, microcontrollers, and microprocessor support peripherals. VLSiCEL elements offer the highest level of microcomputer-based system integration. Intel's cell-based design methodology offers customers the ability to use predefined circuit elements as building blocks to design complex circuits. The current library includes:

80C51BH	8-Bit Microcontroller
82C37A	Programmable DMA Controller
82C54	Programmable Interval Timer
82C59A	Programmable Interrupt Controller
82C84A	8086/8088 Clock Generator
82C284	80286 Clock Generator
82C88	8086/8088 Bus Controller
82288	80286 Bus Controller

  A complete set of test vectors is provided for each VLSiCEL building block, providing a guaranteed 0.1% AQL (Acceptable Quality-Level) or better.
- The emulator kit for 80C51-based ASICs includes a complete set of tools for hardware and software debug of ASIC core-based designs.
- UCS family is based on the 80C51BH standard product and includes both the microcontroller core and peripheral functions. This family makes the internal communications bus (SFR Bus) available to the designer.
- User-configurable n-bit counters, registers, multipliers, and magnitude comparators built from "telescoping" cells achieve high performance for repetitive functions.

- CHMOS III—An advanced 1.5 micron, double-layer metal CMOS process technology providing high performance, high density, and low power semicustom integrated circuits with proven manufacturability.
- CMOS, TTL, and Schmitt Trigger compatible I/O cells available with a variety of drive levels and ESD protection to 2000V.
- A complete set of packaging options with lead counts up to 208 pins. Special packaging configurations and higher pin count packages are available upon request.
- The Intel Design Environment provides a comprehensive set of CAE/CAD tools, logic libraries, and customer support and design services that enable users without IC design expertise to design their own cell-based ASICs.
- Intel's 1.5 Micron CHMOS III Cell Library is fully supported on Mentor and Daisy compatible engineering workstations. Mainframe simulation capability is supported through Intel technology centers and direct dial-up to Intel factory mainframes.

CELL-BASED DESIGN

Why Cell-Based ASICs?

Semicustom integrated circuits are designed from a variety of functional building blocks ranging in complexity from individual logic gates to LSI and VLSI functions. Cell-based ASICs offer the highest level of semicustom integration and are capable of implementing complex VLSI functions (microprocessors and microcontrollers). They also offer high performance, increased functionality and better silicon utilization because the individual cells have been hand-packed to the highest possible densities. Better silicon utilization means lower-cost production in high volume. Full custom ICs can provide the same benefits as cell-based ICs. However, while full custom designs often require years to develop, semicustom chips can be developed in weeks or months. Well-characterized, easy-to-use automated design methodologies also typify semicustom chip development, making ASIC design accessible to system engineers without specialized IC design experience. System manufacturers realize a faster time to market, thus giving more time to concentrate on system rather than IC issues.

Why Intel?

Intel believes that to successfully serve its ASIC customers, it must provide customers with a comprehensive product offering, advanced manufacturing capabilities, a complete CAE tool set, and design services to support the entire ASIC design process.

- Product offering. Intel offers a complete cell-based library including ASIC versions of Intel standard products.
- Manufacturing expertise. Intel ASIC manufacturing draws on the recognized strengths of Intel CMOS technology, advanced packaging, and a demonstrated expertise in assembly and test.
- Design tools. Customers may access the Intel libraries on a variety of platforms, including the Daisy Systems and Mentor Graphics compatible workstations. Simulations for complex designs are supported on mainframes through Intel Technology Centers and direct dial-up to the Intel factory. The UC51-EDK supports code development and "bread board" simulation of 80C51-based ASICs.
- Design services. Intel provides complete documentation for ASIC product lines. Technology centers offer comprehensive hands-on training courses.

What is a Standard Cell?

A standard cell can be thought of as a well characterized module containing an individual, independent, logic circuit. It is a complete functional block with predesigned and precharacterized logic. Intel has designed these cells for optimum electrical performance and silicon utilization. The standard SSI/MSI cells in the library have been designed with a fixed-height and variable-width configuration. These cells are arranged horizontally in rows with routing channels on either side. The height of these routing channels is variable, and is determined by required metalization interconnect between the cells.

Customers who design with standard cells need only look at a "black box" view of the functions. Knowledge of or training on gate/transistor level functionally is not necessary, making the cell-based approach similar to designing with commodity logic or standard products.

Figure 1 is an example of a standard cell.

Large scale functions including the VLSiCEL elements and Intel's special function cells are designed as "both" variable-width and variable-height cell structures.

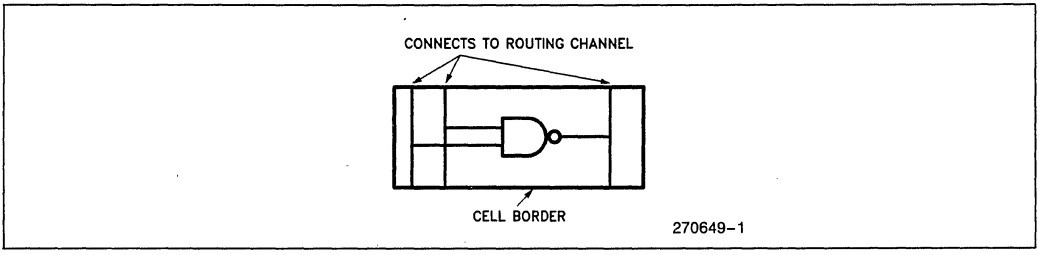


Figure 1. Example Standard Cell

An Example of a Cell-Based Design

Figure 2 depicts a 25,000 "gate" cell-based design where a gate is defined as equal to 4 transistors (the typical equivalent complexity of a 2-input NAND function). When VLSiCEL and special function cells are used in a cell-based ASIC, higher densities are achieved via the large, custom-designed cells. Intel cell-based ASICs often have 100,000 or more transistors.

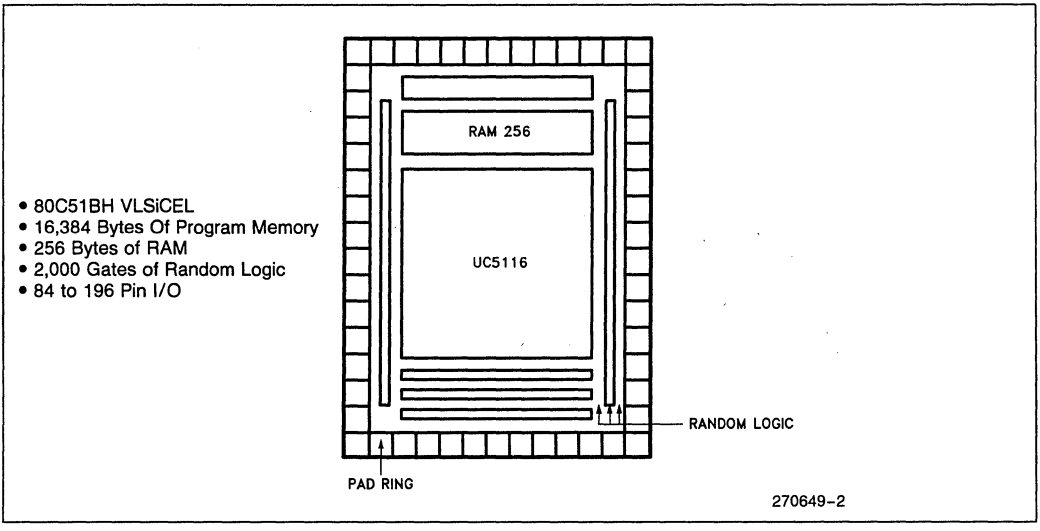


Figure 2. Example Cell-Based Design

CELL TYPES

Standard Cells

The 1.5 Micro CHMOS III Cell Library is composed of over 150 hard-wired basic logic building blocks. These standard cells represent SSI and MSI cells, and are as easy to use as their commodity logic counterparts.

The cell library offers an extensive variety of random logic cells. Included in this category are AND, OR, NAND, NOR, AND-OR, AND-OR-INVERT, EXCLUSIVE OR and EXCLUSIVE NOR gates. Inverters and buffers are available with normal drive, high drive, or 3-state outputs.

The Intel cell library contains a broad range of flip-flops and latches. Flip-flops include D, JK, and Toggle; latches include transparent, SR, and $\overline{SR}$. All bistable devices in the Intel 1.5 Micron CHMOS III Cell Library contain an asynchronous master reset input to aid in system design. Flip-flop and latch configurations provide enable, 3-state, scan input, and set functionality. Other standard cell functions in the Intel 1.5 Micron CHMOS III Cell Library include multiplexers, decoders/demultiplexers and a parity generator/checker.

Telescoping Cells

Registers, shift registers, counters, adders, and magnitude comparators are all available in the Intel 1.5 Micron CHMOS III Cell Library as "telescoping" cells. Telescoping cells provide silicon-efficient, user-configurable implementation of repetitive logic functions.

Designing a telescoping multistage device may require three types of telescoping cells: body cells, control cells, and cap cells. Body cells provide the basic function required by a telescoping component. Control cells provide an interface between a body cell and an external network. Cap cells are used as a driver for any final output (such as carry-out) related to the overall operation of the device. The use of cap cells is optional.

Telescoping cells allow the designer to implement the exact function width required by the design. Silicon utilization is optimized. Front-end workstation tools make the creation of telescoping blocks transparent to the user.

Cluster Macros

Intel provides customers with the capability to group cells which when placed together perform a higher level function. During the layout phase of the design, these cells are physically placed together on-chip—minimizing delays due to interconnect. These cell groups, or cluster macros, are particularly useful for controlling critical path timing.

I/O Cells

The Intel cell library contains over 30 types of I/O buffer cells. Input, output, and I/O cells are available in TTL or CMOS compatible configurations. Input cells are available in both inverting or non-inverting configurations, and with Schmitt Trigger inputs. Each type of input cell contains bonding pads as well as input static protection networks. Output cells are also available in both inverting or non-inverting configurations and with open drain and 3-state outputs. Output cells include bonding pads and output static protection networks. ESD protection to 2000V is provided.

Table 1 shows the available I/O cells and the associated options.

Memory Cells

The most popular Static RAM sizes are available as standard blocks in the cell library. All RAM cells feature byte-wide organization, with densities ranging from 512 to 8K bits.

VLSiCEL™ Elements

VLSiCEL building blocks provide the customer with the ability to use Intel standard products in their design. The VLSiCEL elements are captured and simulated as complete functions with fully supported simulation models. Intel has addressed the long-standing test issues that have prevented the incorporation of embedded microprocessors and complex functions into semicustom ICs by the building in elements of design for testability. All VLSiCEL building blocks come with a pre-defined set of test vectors, assuring circuit validation and eliminating test "bottlenecks." Complex chips can be designed in a fraction of the time required for gate-level implementations.

CELL SIZE

The standard cells (SSI, MSI) in the 1.5 Micron CHMOS III Cell Library are fixed-height, variable-width cells. Intel expresses the size of these cells in terms of grids. Grid counts provide a relative measure of the physical size of these cells. The grid count for each cell, indicated on the cell data sheet, can be used to determine with circuit configuration that yields the optimum silicon area for a given design.

Table 1. Available I/O Cells and Associated Options

I/O Cell Types	Options
Input	TTL and CMOS compatible Inverting/non-inverting buffers Schmitt Trigger inputs Pull-up resistors Normal and high drive
Output	TTL and CMOS compatible Inverting/non-inverting buffers 3-state, open drain, push-pull Multiple output drive levels
Bidirectional	TTL and CMOS compatible 3-state outputs, latched inputs Multiple output drive levels

THE INTEL DESIGN ENVIRONMENT

Intel's design environment provides the customer with a comprehensive set of CAE/CAD tools, logic libraries, customer support and design services. The design environment enables users without IC design expertise to design their own Application Specific Integrated Circuits (ASICs). This section will focus on the design interface for Intel cell-based ASICs.

AN OVERVIEW OF THE INTEL DESIGN ENVIRONMENT

Intel's design environment includes all the design tools and services required to design cell-based ASICs.

The cell-based ASIC design sequence begins with the entering of the design schematic followed by the generation of a netlist (a netlist defines the interconnections between the cells used in the design). Once a netlist has been specified, simulation tools allow the engineer to evaluate the functionality of the design, including timing verification. The design engineer defines the stimulus to exercise the design and verify performance during the simulation phase. After a successful simulation, automatic place and route tools are used to lay out the design. The design is then re-simulated using the delay times that are computed from the layout database. After a successful resimulation, prototypes are manufactured, tested, and delivered.

During all phases of the design process, Intel provides a wide range of support services. Dedicated regional ASIC specialists provide local design analysis and consultation. Technology centers offer comprehensive customer training and technical support, along with access to the software tools running on a variety of hardware platforms. Extensive documentation is available for all software tools and libraries.

AN OVERVIEW OF THE CELL-BASED DESIGN SEQUENCE

Cell-based ASIC designs can be separated into three major phases: design, layout and verification, and manufacturing.

DESIGN PHASE

The design phase includes device specification, logic design, schematic capture, and simulation of the ASIC device. Figure 3 shows the typical flow of events that occur during the design phase of a cell-based ASIC design.

After becoming familiar with the CAE tools and the Intel 1.5 Micron CHMOS III Cell Library, the engineer begins the design process by creating a chip specification. This includes functionality, logic partitioning, and physical requirements (i.e., I/O limitations, packaging, power requirements, die size limitations, operating conditions, and performance requirements). Intel recommends that the engineer adhere to several cell-based design guidelines and design for testability considerations which are covered in the *Introduction to Cell-Based Design Course.* During this phase, customers may also develop functional and timing delay simulation vectors and identify critical paths.

Once the preliminary design is complete, an optional pre-design review can be held. The engineer will then begin selecting cells from the Intel 1.5 Micron CHMOS III Cell Library. Cells are organized into three general categories within the library: VLSiCEL elements, special function cells, and standard cells. Data specification sheets can be used to select functions the same way a component catalog is used to evaluate standard components for board level design.

Upon completion of the chip specification and cell selection, the engineer enters the schematic on a CAE workstation. Schematic capture can be done at an Intel technology center or on an Intel-supported workstation at the customer site. Customers may wish to hold an optional pre-simulation design review with Intel at this time.

A functional simulation and a full timing analysis are required to verify that the design will meet performance requirements. Simulation may be done using workstation simulation or Intel-supported mainframe simulators, depending on the complexity of the design. Designs greater than the approximately 10K gates in complexity often require the computational power of a mainframe for efficient simulation. This mainframe capability gives designers maximum flexibility for simulating large designs such as those including VLSiCEL elements. This phase may require several iterations.

During the design and simulation phase, it is important to consider the testability requirements for the circuit. Intel requires customers to produce designs with 100% observability when performing an industry standard node toggle test. Fault grading is an available option.

After the design is successfully simulated, a pre-layout design review must occur before layout can begin. Once approved by both Intel and the customer, the design progresses to the layout and verification phase.

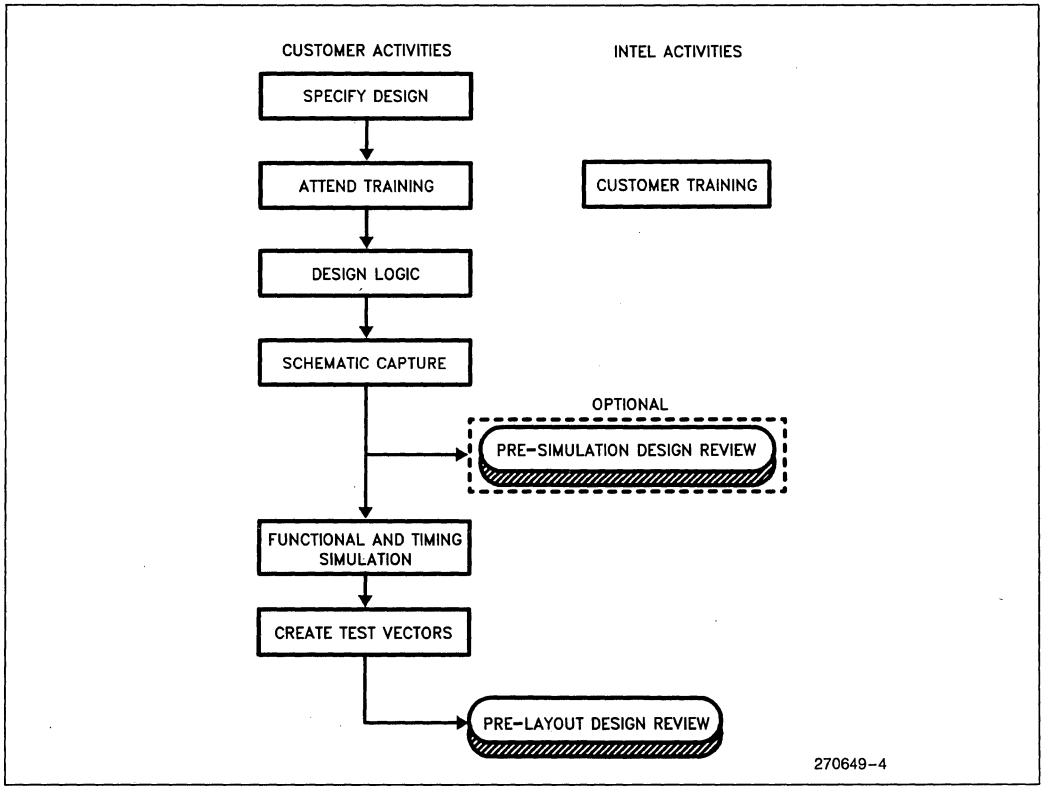


Figure 3. Design Phase Flowchart (Technology Center)

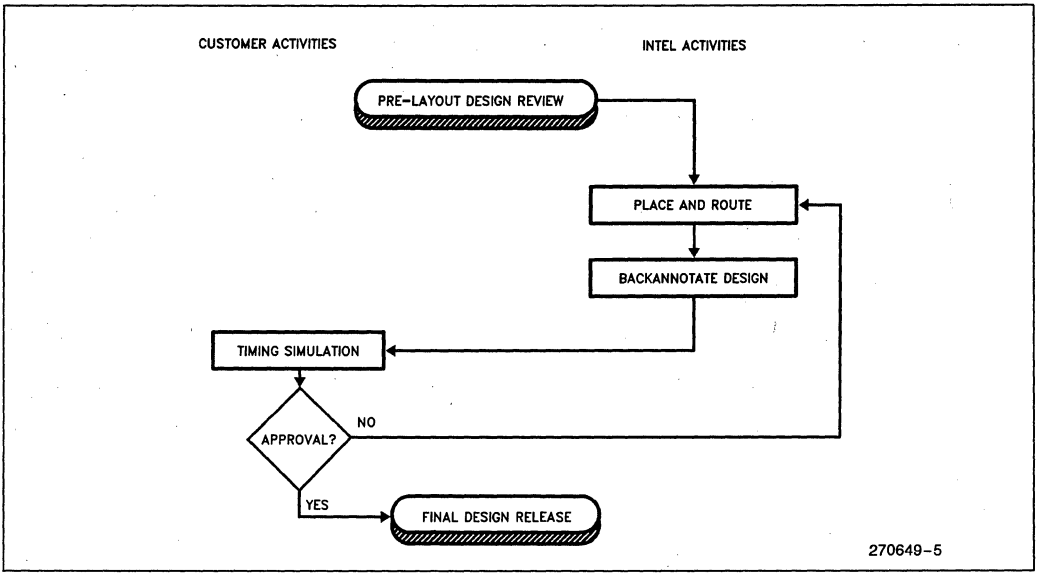


Figure 4. Layout and Verification Phase Flowchart

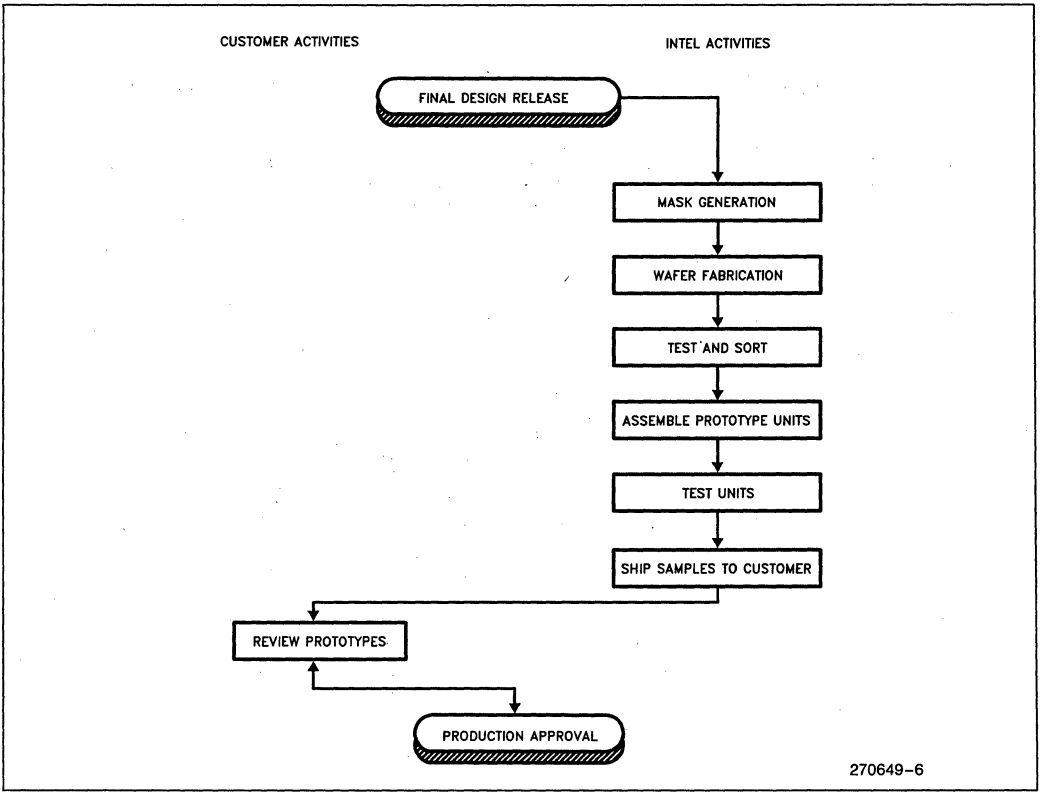


Figure 5. Manufacturing Phase Flowchart

LAYOUT AND VERIFICATION

Figure 4 shows the sequence of events in the layout and verification phase. Placement and routing of the ASIC design is performed using automatic place and route software. As a final check Intel factors in the actual delay times as determined from the layout database (this is called back annotation). The design is then resimulated and post- pre-layout simulation results are compared for consistency. When requirements are met, and both Intel and the customer are satisfied with the results, a final design specification is approved. The design specification becomes the governing document against which prototype and production components are evaluated. The design then enters the manufacturing phase.

MANUFACTURING PHASE

Figure 5 shows the sequence of events in the manufacturing phase. After the layout and verification phase has been completed, Intel will produce prototypes. These prototypes will be submitted to the customer for a final review and production approval.

Intel offers rapid turnaround times for its ASIC products. The ASIC circuits are fabricated, tested, sorted, assembled into packages, and tested again as finished devices. Customer-defined test patterns are used to verify the device, and standard parametric tests are used to confirm performance over temperature and supply voltage extremes.

PACKAGING

Intel provides a variety of standard IC packages for use with cell-based ASIC designs. Among the available packaging options are ceramic and plastic Dual-In-Line Packages (DIP) with up to 48 pins, Plastic Leaded Chip Carriers (PLCC) with up to 84 pins, ceramic and plastic Pin Grid Arrays (PGA) with up to 180 pins, and ceramic and plastic quad flatpacks for up to 208 pin configurations.

QUALITY AND RELIABILITY

Intel is committed to the highest possible standards of quality, reliability and customer satisfaction in its products. All ASIC products must meet the same quality and reliability standards as Intel standard products. Intel insists on building-in quality and reliability for every product from the very beginning of a technology and product development cycle. Strict controls and monitors are applied in the manufacturing process to ensure high quality and reliability. All processes are audited regularly to ensure that they meet specifications. For additional details on Intel's quality and reliability programs, refer to the *Components Quality/Reliability Handbook,* Order Number 210997.

Intel's cell-based products are shipped to a 0.1% AQL level and less than 200 FITs (Failures In Time).

DESIGN SUPPORT

Intel customers have the option to specify as much or as little design support as needed to complete their ASIC. Table 2 describes two preferred ASIC design interfaces: design tasks performed at the customer's site or at an Intel technology center. Customers may also opt for full service design support.

Intel's design environment provides the necessary design tools and services for all these design interface alternatives. Customers who choose to develop their ASIC on-site port Intel libraries onto their own CAE systems and design the device within their own development environment. Customers who decide to use Intel technology centers for ASIC development take advantage of Intel's on-site CAE systems, libraries, and applications support to assist them during their design effort.

Table 2 lists the major steps required to execute a design and specifies the responsibilities for each party.

Table 2. Design Sequence Responsibilities

Activity	Technology Center	Customer Site
Pre-design Start Design Review	Intel/Customer	Intel/Customer
Cell Selection	Intel/Customer	Customer
Schematic Capture	Intel/Customer	Customer
Simulation (Functional)	Intel/Customer	Customer
Simulation (Timing)	Intel/Customer	Customer
Pre-Layout Design Review	Intel/Customer	Intel/Customer
Test Vector Generation	Intel/Customer	Customer
Autolayout	Intel	Intel
Post-Layout Simulation	Intel/Customer	Customer
Post-Layout Approval	Intel/Customer	Intel/Customer
Mask Generation	Intel	Intel
Wafer Fab	Intel	Intel
Assembly/Test	Intel	Intel
Prototype Approval	Customer	Customer

COMPUTER AIDED ENGINEERING (CAE) TOOLS

The 1.5 Micron CHMOS III Cell Library runs on all engineering workstations from Daisy Systems and Mentor Graphics. This wide range of compatibility gives designers the flexibility to execute cell-based designs using a variety of CAE hardware.

MAINFRAME-BASED SIMULATION

Simulation requirements for complex designs are often best served by mainframe computational power. Using an Intel mainframe simulator, customers can run simulations that are too time consuming or not possible to do using a desktop workstation environment. An integrated design database allows for portability between the mainframe environment and the workstation environment.

Intel's mainframe computers may be accessed through dial-up from the customer site or via an Intel technology center.

INTEL TECHNOLOGY CENTERS

Intel technology centers offer training classes, design consultation and technical workstation support for semicustom chip design, cell-based design libraries, and CAE workstations as well as access to workstations for schematic capture and simulation. Customers may use the technology centers to take advantage of Intel's on-site systems, libraries and services. Each center is equipped with Daisy Systems and Mentor Graphics workstations. Direct access to the Intel mainframe is also available for efficient simulation of complex designs.

Training classes and technical support are available from Intel's technology center ASIC specialists. The *Introduction to Intel Cell-Based Design Course* consists of both lecture and labs emphasizing "hands-on" experience. Lectures address Intel-specific design practices; labs offer hands-on training in the Intel design environment. Contact your local Intel field sales office for scheduling.

INTEL TECHNOLOGY CENTERS

CALIFORNIA, USA

Intel Technology Center
3065 Bowers Avenue
Santa Clara, CA USA 95051
Tel: (408) 765-2252

MASSACHUSETTS, USA

Intel Technology Center
3 Carlisle Rd.
Westford, MA USA 01886
Tel: (617) 692-3222

FRANCE

Intel Technology Center
1 Rue Edison, BP303
78054 St. Quentin EN
Yvelines Cedex, France
Tel: 33 1-30-57-7000

UNITED KINGDOM

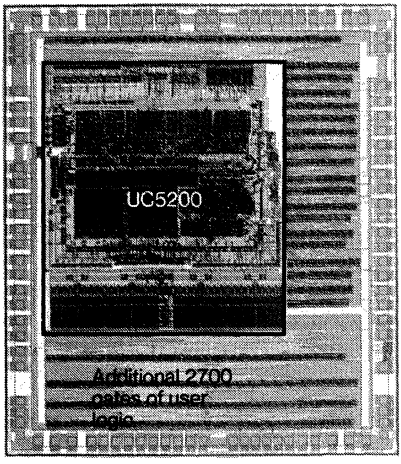
Intel Technology Center
Piper's Way
Swindon SN3IRJ
Wiltshire, U.K.
Tel: 0793 696000

RELATED PUBLICATIONS

- Cell-Based Design—Daisy Environment Order #83002
- Cell-Based Design—Mentor Environment Order #830000
- Microprocessor and Peripheral Handbook Order #230843
- Components Quality/Reliability Handbook Order #210997

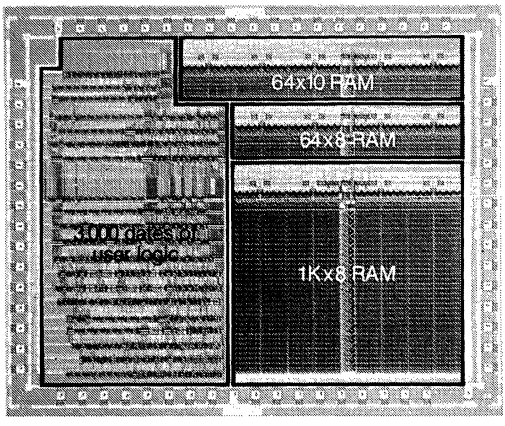
1.5 MICRON CHMOS III CELL LIBRARY

- 1.5 Micron CMOS
- User Configurable N-Bit Counters, Registers, Adders, and Magnitude Comparators Built from "Telescoping" Cells Achieve High Performance and Silicon Efficiency for Repetitive Functions
- RAM Configurations up to 8K bits
- All VLSI Cells are Tested and Verified Using the Equivalent Standard Product Test Program, Guaranteeing 0.1% AQL or Better.
- VLSiCEL™ Elements, Cell Versions of Popular Intel Standard Microprocessors, Microcontrollers, and Microprocessor Support Peripherals, Offer the Highest Level of Micro-Computer Based System Integration. The Current Library Includes:
  - — 80C51BH 8-Bit Microcontroller
  - — 82C37A Programmable DMA Controller
  - — 82C54 Programmable Interval Timer
  - — 82C59A Programmable Interrupt Controller
  - — 82C84 8086/8088 Clock Generator
  - — 82C284 80286 Clock Generator
  - — 82C88 8086/8088 Bus Controller
  - — 82288 80286 Bus Controller



240051-1

The Cell-Based IC Shown here Contains the 80C51BH ASIC Core. This Design also Integrates Approximately 2700 Gates of User Control Logic



240051-2

The Cell-Based IC Shown above Depicts a Standard Cell Design with 3000 Gates of User Logic and 3 RAM Configurations

*Daisy is a registered trademark of Daisy Systems Corporation.
*Mentor Graphics is a registered trademark of Mentor Graphics Corporation.

December 1988
Order Number: 240051-004

DESCRIPTION

Intel's advanced cell-based family of integrated circuits is based on a comprehensive set of predesigned, fully-characterized functions for the integration of system logic into high-performance, cost effective semicustom devices. The basic cell library contains over 150 logic, I/O and special function cells. Also available are VLSiCEL™ elements such as the 80C51BH 8-bit microcontroller, and cell equivalents of Intel microprocessor support peripheral functions. Intel libraries run on several industry standard CAE platforms, including Daisy Systems* and Mentor Graphics* workstations. Customers may receive expert technical support via Intel's fully equipped technology centers. Once a design has been completed to the customer's satisfaction, Intel produces its cell-based IC using a 1.5 micron double layer metal CMOS process. This process is also used on standard components such as the 80C51BH, thus ensuring manufacturability and high quality and reliability.

FEATURES/BENEFITS

Logic Cells

- Over 150 SSI/MSI/LSI logic functions to obtain high performance and high density.
- CMOS, TTL, and Schmitt Trigger compatible I/O cells available with a variety of drive levels and ESD protection to 2000V.

Special Function Cells

- Fixed configuration RAM to 8K bits.
- User configurable n-bit counters, shift registers, adders and magnitude comparators built from "Telescoping" cells achieve high performance and silicon efficiency for repetitive functions.

ASIC Emulators

- ASIC emulators support code development and "breadboard" simulation of 80C51BH-based ASICs.

Packaging

- A complete set of packaging options with lead counts to 208.

VLSiCEL™ ELEMENTS

- Cell versions of popular Intel standard microprocessors, microcontrollers and microprocessor support peripherals are available in the library. These "VLSiCEL" elements offer the highest level of microcomputer based system integration.

80C51BH	8-bit Microcontroller
82C37A	Programmable DMA Controller
82C54	Programmable Interval Timer
82C59A	Programmable Interrupt Controller
82C84A	8086 Clock Generator
82C284	80286 Clock Generator
82C88	8086 Bus Controller
82288	80286 Bus Controller

- UCS Family is based on the 80C51BH standard product and includes both the microcontroller core and peripheral functions. This family makes the internal communications bus (SFR Bus) available to the designer.
- The 12.5 MHz microprocessor support peripheral family includes 82CXX peripheral cells compatible with 86/186/286 environments.
- All VLSI cells are tested and verified using the equivalent standard product test program, guaranteeing 0.1% AQL or better.
- Functional relationships identical to the standard product, including standard product code compatibility.

Design Environment

- Libraries compatible with workstations from Daisy Systems and Mentor Graphics.
- Mainframe interface accessible to customers via Intel technology centers or dial-up from customer site. Supports high complexity designs.
- All design environments include Intel developed utilities for back annotation of actual delays from layout database for post layout simulations.
- Software utilities provided by Intel automatically converts your simulation stimulus to Intel tester compatible vectors.
- For those designs incorporating the 80C51-based VLSI cell and which are designed in the Daisy environment, customers may also simulate the core using Daisy's Physical Modeling Extension (PMX) board. The PMX is a hardware add-on chassis and control software that allows the user to perform software modeling using an actual physical component in the Daisy environment.

CHMOS III: 1.5 Micron CMOS

- Intel's cell based products are produced using CHMOS III—an advanced 1.5 micron, double-layer metal CMOS process technology providing high performance, high density, and low power consumption semi-custom integrated circuits with proven manufacturability.
- Intel's 80C51BH standard product also manufactured in CHMOS III.

Manufacturing Reliability

- 0.1% Average Quality Level (AQL) against submitted test vectors.
- < 200 Failures In Time (FITs).

LSI and VLSI Functions

UCS51 FAMILY: UCS51XX and UCS52XX cells listed below are fully compatible with the 80C51BH standard product. In addition, these cells make the internal bus (Special Function Register Bus) of the 80C51BH accessible to the designer. A series of SFR peripheral cells have been designed to interface to this bus. These peripherals are also listed below. Peripheral communications are optimized (less code required) and the designer can also define his own peripherals and connect them directly to the bus.

Name	Cell Description
UCS5100	80C51BH Microcontroller Core with No ROM, 128 Bytes RAM
UCS5104	80C51BH Microcontroller Core with 4K Bytes ROM, 128 Bytes RAM
UCS5108	80C51BH Microcontroller Core with 8K Bytes ROM, 128 Bytes RAM
UCS5116	80C51BH Microcontroller Core with 16K Bytes ROM, 128 Bytes RAM
UCS5200	80C51BH Microcontroller Core with No ROM, 256 Bytes RAM
UCS5204	80C51BH Microcontroller Core with 4K Bytes ROM, 256 Bytes RAM
UCS5208	80C51BH Microcontroller Core with 8K Bytes ROM, 256 Bytes RAM
UCS5216	80C51BH Microcontroller Core with 16K Bytes ROM, 256 Bytes RAM
UCS51BRG	Baud Rate Generator
UCS51SIO	Serial I/O
UCS51T2	Timer 2
UCS51BIU	SFR Bus Interface Unit
UCS51AD	8-Bit Analog to Digital Converter with Sample and Hold
UCS51IEU	Interrupt Expansion Unit

NOTE:
All specifications within tables are subject to change.

UC51 FAMILY: UC51XX cells listed below are fully compatible with the 80C51BH standard product and are available in a variety of ROM configurations. In addition, the signals from the standard product "CORE" have been demultiplexed, making all 116 signals available for use.

Name	Cell Description
UC5100	80C51BH Microcontroller Core with No ROM, 128 Bytes RAM
UC5104	80C51BH Microcontroller Core with 4K Bytes ROM, 128 Bytes RAM
UC5108	80C51BH Microcontroller Core with 8K Bytes ROM, 128 Bytes RAM
UC5116	80C51BH Microcontroller Core with 16K Bytes ROM, 128 Bytes RAM
UC5200	80C51BH Microcontroller Core with No ROM, 256 Bytes RAM
UC5204	80C51BH Microcontroller Core with 4K Bytes ROM, 256 Bytes RAM
UC5208	80C51BH Microcontroller Core with 8K Bytes ROM, 256 Bytes RAM
UC5216	80C51BH Microcontroller Core with 16K Bytes ROM, 256 Bytes RAM

80C51 Based Core Companion Cells: The following cells are used in conjunction with the UC51XX cells noted above.

Name	Cell Description
POSC	Oscillator
PWOSC	Oscillator with Power Down
PADB	Address/Data Bus I/O Buffer
PRESET	Reset Input Buffer
PTNQB	Quasi-Bidirectional I/O Buffer
PRGPIN	UC51 Programmable I/O Buffer
PRGUCS	UCS Programmable I/O Buffer

MICROPROCESSOR SUPPORT PERIPHERAL COMPANION CELLS: The following cells are used in conjunction with the microprocessor support peripheral cells noted above:

For use with SP8284/SP82284	
POSC2	Oscillator, frequency range to 37.5 MHz
PCNO4	Non-inverting CMOS Output Buffer, High Drive
For use with SP8288/SP82288	
PCO2	Inverting CMOS Output Buffer, High Drive
PCOT6	3-State Inverting CMOS Output Buffer with Enable, High Drive

MICROPROCESSOR SUPPORT PERIPHERAL FAMILY: These cells are equivalent in functionality to the corresponding Intel standard products.

Name	Cell Description
SP8237	Programmable DMA Controller
SP8254	Programmable Interval Timer
SP8259	Programmable Interrupt Controller
SP8284	8086/8088 Clock Generator and Driver
SP82284	80286 Clock Generator and Ready Interface
SP8288	8086/8088 Bus Controller
SP82288	80286 Bus Controller

Fixed Configuration Memory

Name	Cell Description
RAM64	64 x 8 Static Random Access Memory
RAM128	128 x 8 Static Random Access Memory
RAM256	256 x 8 Static Random Access Memory
RAM512	512 x 8 Static Random Access Memory
RAM1K	1024 x 8 Static Random Access Memory

NOTE:
All specifications within tables are subject to change.

Telescoping Cells

Telescoping cells are building blocks from which certain multi-stage logic functions can be implemented. They are the preferred method of construction for multi-stage logic functions because they allow the designer to implement the exact function width required by the design. Silicon utilization is optimized. Front end work station tools make the creation of telescoping blocks transparent to the user.

Name	Cell Description
REGC	Telescoping Register Control
REGB	Telescoping Register Body
REGCT	Telescoping 3-State Register Control
REGBT	Telescoping 3-State Register Body
SHRC	Telescoping Shift Register Control
SHRB	Telescoping Shift Register Body
SHLC	Telescoping Shift Register with Load, Control
SHLB	Telescoping Shift Register with Load, Body
CULC	Telescoping Up Counter Control
CULB	Telescoping Up Counter Body
CULP	Telescoping Up Counter Carry Out Driver
CUPC	Telescoping Up/Down Counter Control
CUPB	Telescoping Up/Down Counter Body
CUPP	Telescoping Up/Down Counter End Count Driver
CUPP2	Telescoping Up/Down Counter End Count/Carry/Borrow Driver
ADDC	Telescoping Adder Control
ADDB	Telescoping Adder Body
ADDP	Telescoping Adder Carry Out Driver
CMPC	Telescoping Magnitude Comparator Control
CMPB	Telescoping Magnitude Comparator Body
CMPP	Telescoping Magnitude Comparator Equal/Greater Than/Less Than Driver

NOTICE: All specifications within tables are subject to change.

Inverters, Buffers and Gates

Name	Cell Description
INVN	Inverter, Normal Drive
INVNH	Inverter, High Drive
INVTE	3-State Inverter with Active Low Output Enable, Normal Drive
INVTD	3-State Inverter with Active High Output Enable, Normal Drive
BUF	Buffer, Normal Drive
BUFH	Buffer, High Drive
BUF2	Buffer with Dual Output, Normal Drive
BUFTD	3-State Buffer with Active High Output Enable, Normal Drive
BUFTE	3-State Buffer with Active Low Output Enable, Normal Drive
NAN2	2 Input NAND, Normal Drive
NAN3	3 Input NAND, Normal Drive
NAN4	4 Input NAND, Normal Drive
NAN5	5 Input NAND, Normal Drive
NAN6	6 Input NAND, Normal Drive
NAN7	7 Input NAND, Normal Drive
NAN8	8 Input NAND, Normal Drive
NOR2	2 Input NOR, Normal Drive
NOR3	3 Input NOR, Normal Drive
NOR4	4 Input NOR, Normal Drive
NOR5	5 Input NOR, Normal Drive
NOR6	6 Input NOR, Normal Drive
NOR7	7 Input NOR, Normal Drive
NOR8	8 Input NOR, Normal Drive
AND2	2 Input AND, Normal Drive
AND3	3 Input AND, Normal Drive
AND4	4 Input AND, Normal Drive
AND5	5 Input AND, Normal Drive
AND6	6 Input AND, Normal Drive
AND7	7 Input AND, Normal Drive
AND8	8 Input AND, Normal Drive
OR2	2 Input OR, Normal Drive
OR3	3 Input OR, Normal Drive
OR4	4 Input OR, Normal Drive
OR5	5 Input OR, Normal Drive
OR6	6 Input OR, Normal Drive
OR7	7 Input OR, Normal Drive
OR8	8 Input OR, Normal Drive
AOR22	2 AND2 into OR2, Normal Drive
AOI22	2 AND2 into NOR2, Normal Drive
EXR2	2 Input EXCLUSIVE OR, Normal Drive
EXN2	2 Input EXCLUSIVE NOR, Normal Drive

NOTICE: All specifications within tables are subject to change.

Flip-Flops and Latches

Name	Cell Description
FFT	Toggle Flip-Flop with Master Reset
FFTE	Toggle Flip-Flop with Enable and Master Reset
FFJK	JK Flip-Flop with Master Reset
FLJK	JK Flip-Flop with Master Set and Master Reset
FLJKT	3-State JK Flip-Flop with Master Set and Master Reset
FFD	D Flip-Flop with Master Reset
FFDE	D Flip-Flop with Enable and Master Reset
FLDE	D Flip-Flop with Enable, Master Set and Master Reset
FLDET	3-State D Flip-Flop with Enable, Master Set and Master Reset
FFDM2	D Flip-Flop with 2 to 1 Data Multiplexer and Master Reset
FLDM2	D Flip-Flop with 2 to 1 Data Multiplexer, Master Set and Master Reset
FFDHI	Positive Edge Event Trigger with Master Reset
LAD	Transparent D Latch with Master Reset
LSR	S-R Latch with Master Reset
LASR	S-R Latch with Enable and Master Reset
LNSR	$\overline{S}$-$\overline{R}$ Latch with Master Reset
LANSR	$\overline{S}$-$\overline{R}$ Latch with Enable and Master Reset

Multiplexers, Decoders and Arithmetic Functions

Name	Cell Description
MUX21	2-Line to 1-Line Multiplexer
MUX41	4-Line to 1-Line Multiplexer
DMX2	2-Line to 4-Line Demultiplexer/Decoder with 2 Enables
DMX3	3-Line to 8-Line Demultiplexer/Decoder
CPR	8/9-Bit Parity Checker/Generator

NOTICE: All specifications within tables are subject to change.

Input/Output Functions

Name	Cell Description
PCI	Non-Inverting CMOS Input Buffer, Normal Drive
PCIH	Non-Inverting CMOS Input Buffer, High Drive
PTI	Non-Inverting TTL Input Buffer, Normal Drive
PTIH	Non-Inverting TTL Input Buffer, High Drive
PTIRH	Non-Inverting TTL Input Buffer with Pull-Up Resistor, High Drive
PISH	Non-Inverting TTL Schmitt Trigger Input Buffer, High Drive
PISRH	Non-Inverting TTL Schmitt Trigger Input Buffer with Pull-Up Resistor, High Drive
PCO	Inverting CMOS Output Buffer, Low Drive
PCNO	Non-Inverting CMOS Output Buffer, Low Drive
PCOT	3-State Inverting CMOS Output Buffer, Low Drive
PTO	Inverting TTL Output Buffer, Low Drive
PTNO	Non-Inverting TTL Output Buffer, Low Drive
PTNO3	Non-Inverting TTL Output Buffer, Medium Drive
PTNO5	Non-Inverting TTL Output Buffer, High Drive
PTOT	3-State Inverting TTL Output Buffer, Low Drive
PTOT3	3-State Inverting TTL Output Buffer, Medium Drive
PTOT5	3-State Inverting TTL Output Buffer, High Drive
PTND	Non-Inverting TTL Open-Drain Output Buffer, Low Drive
PTND3	Non-Inverting TTL Open-Drain Output Buffer, Medium Drive
PTND5	Non-Inverting TTL Open-Drain Output Buffer, High Drive
PCIO	CMOS I/O Buffer; Latched Non-Inverting Input, 3-State Inverting Output, Low Drive
PTIO	TTL I/O Buffer; Latched Non-Inverting Input, 3-State Inverting Output, Low Drive
PTIO3	TTL I/O Buffer; Latched Non-Inverting Input, 3-State Inverting Output, Medium Drive
PTIO5	TTL I/O Buffer; Latched Non-Inverting Input, 3-State Inverting Output, High Drive

NOTICE: All specifications within tables are subject to change.

INTEL DESIGN ENVIRONMENT

Intel's design environment includes the software tools necessary to assist customers with their Intel cell-based design. Intel provides workstation library models and utilities to support schematic capture, netlist generation, and functional and timing simulation in the Daisy Systems or Mentor Graphics environments. For complex designs requiring simulation capability exceeding the capabilities of the workstation, Intel provides mainframe simulation. Automatic test vector generation software is provided to assist the customer in the development of input test stimuli. The table below shows the typical flow of Intel cell-based design activities.

Task	Intel Feature	Responsibility
System Design		Customer
Schematic Capture, User Logic and Test Logic Design	Customer Maintains Control of the Design Specification.	Customer
Netlist Generation	Electrical Rules Check. Flags Illegal use of Cells.	Customer
Simulation	Built in Timing Verifier.	Customer
Test Vector Generation	Automatic Conversion of User's Simulation Vectors to Intel-tester Compatible Format. 100% Explainable Toggle Node Count Required.	Customer
Design Review		Customer
Auto Place and Route	Proprietary Tools for Automatic Place and Route, Mask design.	Intel
Back Annotation/ Post-Layout Simulation	Resistance and Capacitance Values Extracted from Layout Database.	Intel/Customer
Design Review		Intel/Customer
Fab, Assembly, Test		Intel
Prototype Delivery	On-Time Delivery; Conformance to Specifications.	Intel

ABSOLUTE MAXIMUM RATINGS*

Case Temperature under Bias
- Plastic ..... −40°C to +85°C
- Ceramic ..... −55°C to +125°C

Storage Temperature ..... −65°C to +150°C

DC Supply Voltage (V_{DD}) ..... 0V to 7.0V

Voltage to Any Pin with Respect to Ground ..... −0.5V to V_{DD} + 0.5V

Power Dissipation 1.0W

RECOMMENDED OPERATING CONDITIONS

DC Supply Voltage, V_{DD} ..... +4.5V to 5.5V

Case Temperature ..... 0°C to +70°C

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

PACKAGING

Intel's cell-based ICs are available in a wide variety of both through-board and surface-mount packages. DIP, chip carrier, pin grid array and flatpack configurations are offered. Special packages are available by request.

Intel ASIC Components Packaging Alternatives

Package Type	Leadcount	Ceramic	Plastic
Dual-In-Line(1)	16		X
	18		X
	20	X	X
	24	X	X
	28	X	X
	40	X	X
	48	X	X
Leaded Chip Carrier	20		X
	28		X
	32		X
	44	X(2)	X
	52		X
	68	X(2)	X
	84	X(2)	X
Leadless Chip Carrier	68	X	
	84	X(2)	
Quad Flat Pack	84	X(2)	X
	100	X(2)	X
	132	X(2)	X
	164	X(2)	X
Pin Grid Array	68	X	
	72	X	
	84	X	
	88	X	
	100	X	
	132	X	
	144	X	
	180	X	
	208	X	

NOTES:
1. Hermetic DIPs may be side-brazed or CERDIP.
2. Prototype only.

D.C. CHARACTERISTICS/I-O CELLS 0°C–70°C, 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Low Level Input Voltage				
	TTL Inputs		0.8	V	
	CMOS Inputs		1.2	V	
V_{IH}	High Level Input Voltage				
	TTL Inputs	2.0		V	
	CMOS Inputs	3.5		V	
I_{IL}	Low Level Input Current				
	TTL Inputs		−10	μA	$V_{IL} = V_{SS}$
	CMOS Inputs		−10	μA	$V_{IL} = V_{SS}$
I_{IH}	High Level Input Current				
	TTL Inputs		10	μA	$V_{IH} = V_{DD}$
	CMOS Inputs		10	μA	$V_{IH} = V_{DD}$
V_{OL}	Low Level Output Voltage				
	TTL Inputs		0.45	V	
	CMOS Inputs		0.4	V	
V_{OH}	High Level Output Voltage				
	TTL Inputs	2.4		V	
	CMOS Inputs	3.6		V	
I_{OL}	Low Level Output Current				
	TTL Outputs		3.2	mA	Low Drive
	TTL Outputs		7	mA	Medium Drive
	TTL Outputs		12	mA	High Drive
	CMOS Outputs		3.2	mA	
I_{OH}	High Level Output Current				
	TTL Outputs		−0.08	mA	All Drives
	CMOS Outputs		−2.4	mA	

NOTICE: All specifications within tables are subject to change.

DESIGN SUPPORT AND INTERFACE

Intel's design environment provides customers with expert technical support, either locally or in dedicated ASIC Technology Centers. Intel training and field application engineers offer training classes, design consultation, and technical support for semicustom chip design, cell-based design libraries, and engineering workstation (CAE) tools. ASIC Technology Centers are equipped with Daisy and Mentor workstations. These design centers also provide customers with access to factory mainframes for efficient simulation of complex designs.

ORDERING INFORMATION

Contact your local Intel sales office.

RELATED LITERATURE

Item	Order Number
Introduction to Intel Cell-Based Design	231816

UC51 TIMING CALCULATION PACKAGE

Daisy Environment

- **Full-functional simulation for the UC51 core cell**
- **Physical model enables fast, accurate timing verification**
- **Test generator for UC51-to-user logic (ExtASM51)**
- **Operates with Physical Modeling Extension (PMX) from Daisy**
- **Timing models correlated with Intel's referent simulator to reduce design iterations**
- **Post-layout timing values incorporated into simulation (back-annotation)**

PRODUCT DESCRIPTION

The Daisy-based UC51 Timing Calculation Package contains software and hardware for performing full-functional simulation of UC51-based ASIC devices. The package includes a software timing shell, a "Daisy hardware modeling kit" (DHMK), and the "Extended ASM51" for creating test patterns from 8051 instruction mnemonics. The DHMK is used with Daisy's Physical Modeling Extension (PMX) to perform accurate timing simulations.

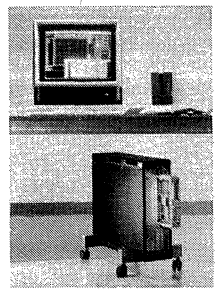
FEATURES

Timing Calculation

The timing calculation shell is installed on the Daisy along with the other full-functional models in the 1.5 micron CHMOS cell library. The timing shell is the software interface between the DLS-II simulator, and the UC51 physical model. The timing package supports functional simulation, full-timing simulation, and post-layout simulation with back-annotated delay values. The timings are correlated with the UC51 timing model on the referent simulator.

Daisy Hardware Modeling Kit

The UC51-DHMK consists of a bonded-out version of the UC51 core cell, and the installation/wiring instructions for use with the Daisy PMX. When simulating a UC51 design, the DHMK enables the user to attach a segment of application code (in hex) that will be executed during the simulation. This flexibility enables the validation of device confidence tests in addition to the logic functionality and timing of the ASIC device. The DHMK kit also explains how a user can connect the completed UC51-based ASIC to the PMX for doing system-level simulation.



intel

AUGUST 1988
ORDER NUMBER: 240272-001

UC51 Testability

The UC51 core cell has been developed to be tested in two modes. The user is only responsible for developing test vectors for the logic surrounding the UC51 core. Through the use of 13 pins on the device, Intel guarantees it can test the final ASIC — including the core. Most of the 13 pins (such as the data bus) can also be used in the application.

In "core isolation" mode, Intel tests the UC51 core to the same conditions it tests the standard 8051. In "user-test" mode, the user-developed test-vectors are used to verify the application-specific logic connected to the UC51. The user develops only one set of test-vectors for both logic simulation and production-test, using Intel-supplied conversion programs between the different environments. To aid development of these user-specific stimuli, Intel provides a software tool call ExtASM51.

ExtASM51

The Extended Assembler-51 is a modified assembler that generates test-patterns in sequence with 8051 instruction mnemonics. Specific keywords in the comment fields indicate to ExtASM51 what test patterns should be generated, and how the generated vectors should be synchronized with the executed instructions. For example, to read in specific values on UC51 ports, during specific tester cycles, ExtASM51 will generate the appropriate stimulus, given a few simple mnemonics.

```
MOV DPTR,#0AAAAH                    #0018 E0    99    MOVX A,@DPTR  ;)DATABUS=AAH
MOVX A,@DPTR     ;)DATABUS=00H       3   1 1 1 0 0 0 0 0   1
MOV P1,A         ;)IOPORT1=0FF      12   1 1 1 1 1 1 1 1   0
MOV A,P1         ;)IOPORT1=FFH       4   1 0 1 0 1 0 1 0   1
MOV A,P1         ;)IOPORT1=00H       5   1 1 1 1 1 1 1 1   0
MOVX A,@DPTR     ;)DATABUS=AAH      #0019 F590  100   MOV P1,A      ;)IOPORT1=0FF
MOV P1,A         ;)IOPORT1=0FF       3   1 1 1 1 0 1 0 1   1
MOV A,P1         ;)IOPORT1=55H       3   1 1 1 1 1 1 1 1   0
MOV A,P1         ;)IOPORT1=AAH       3   1 0 0 1 0 0 0 0   1
MOV DPTR,#1234H                     3   1 1 1 1 1 1 1 1   0
ORL C,ACC.7
END;

Sample ExtASM51 source              Excerpted TPDL output from ExtASM51
```

ExtASM51 generates a test file (in TPDL format) that can be input to the simulator or IC tester, as well as a .ROM object file and a .LST file.

SPECIFICATIONS

Hardware Required:
For Schematic Entry and Simulation: Daisy Systems' Logician, Personal Logician (286 or 386), or MegaLogician
Physical Modeling Extension (PMX), with Two (2) Dynamic PEM's
Two (2) PMX "Pin Grid Daughter Boards"

Documentation Supplied:
UC-51 Daisy Hardware Modeling Kit User's Guide

Software Required:
Daisy DNIX operating system, release 5.02 or later
Intel Daisy Design Entry Package
Intel Daisy Cell-Based Basic Timing Package

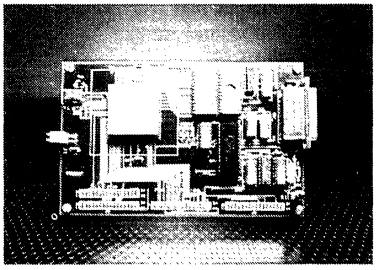
Media:
5.25" and 8" Diskettes Supplied

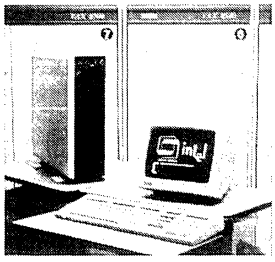
ORDERING INFORMATION

DC-51TP *Daisy UC51 Timing Package*

Printed in USA/AC003/888/3K/RR SM

ASIC TOOLS AND SERVICES





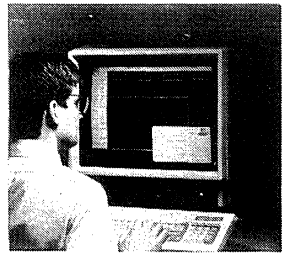
- **Complete development cycle support for Intel cell-based ASIC designs**
- **Complete CAE support on Daisy and Mentor engineering workstations**
- **Mainframe simulation support for complex designs**
- **Emulator kits for software development and prototyping core-based ASICs**
- **Productivity-approaches to ASIC testability**
- **Training courses on Intel cell-based ASIC design**
- **Engineering services in Technology Centers**

Intel's ASIC Tools and Services provide a low-risk approach to high-performance ASIC design. The offering includes a comprehensive set of tools for design entry, logic simulation, test-program development, and system design, as well as Technology Center services that minimizes the number of design changes required to achieve working silicon.

Workstation Libraries

Intel ASICs are supported by several software libraries that operate on Daisy and Mentor Engineering Workstations. The "Design Entry Package" consists of schematic-entry symbols and functional timing models for Intel cell-based designs. The package also contains several utilities for test-program development, netlisting, rules checking, and timing comparison. The design-entry package contains support for Intel-unique elements such as telescoping cells, microprocessor cores, and peripherals.

The "Timing Calculation Packages" contain full-functional simulation models for performing design verification on the engineering workstation. The models are correlated to our referent simulator, to reduce the risk of design changes when we accept the netlist and do a referent simulation. The Timing Calculation Packages will incorporate post-layout timing delays (back-annotation) into a final simulation prior to the manufacture of prototype silicon. Separate Timing Calculation Packages are available for the Basic Standard-Cell library, and the VLSiCEL™ families (for Daisy and Mentor).



Mainframe Simulation

Customers who use the Design Entry Packages may also access Intel's mainframe-based referent simulator for doing design verification. This "Mainframe Design Verification System" is accessible through an Intel Technology Center or dial-up via modem. The benefits of this approach is the increased simulation throughput time (for larger designs), and the ability to design in the same environment in which the cells were originally developed.

intel

AUGUST 1988
ORDER NUMBER: 240269-001

Test Methodology

Intel's approach to testing ASICs enables users to write a single set of test vectors, that will be used for Workstation timing simulation, Mainframe timing simmulation, and Production test of wafers and assembled devices. Upon design-acceptance, we can ensure that prototype and production devices are tested with the same test vectors used during design verification. The format conversions between Daisy, Mentor, mainframe, and tester environments are performed by utility programs provided with the Design Entry Packages. Also included is a utility that eases the process of writing test vectors; it provides a shell-file with predefined and design-specific values already filled in.

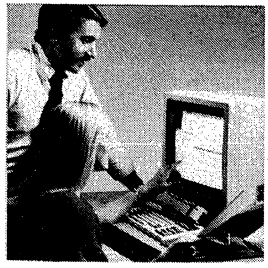
VLSI cells, such as microprocessor cores and peripherals, are designed with a "test ring" of circuitry that enables two modes of testing to take place. In "core isolation mode," Intel tests the ASIC core to the same stringent requirements as our standard parts. In "user test mode," we run the same user-supplied vectors that were used during simulation. Our Mainframe Design Verification System, as well as workstation-based timing packages for VLSI cells, provide special tools to test the interoperation of the VLSI cells and user logic.

System Design

Breadboarding and software development for programmable cores are addressed with in-circuit emulation products. These Emulator Design Kits make use of "bond-out" versions of the ASIC cores. Such a kit may then be used to build a breadboard version of the ASIC, which includes the core and user logic — this is useful for building prototype systems prior to the completion of the ASIC device. The Emulator kits also contain PC-based software to download and debug system software.

ASIC Training

Training courses for Intel ASIC design, using Daisy or Mentor workstations, are offered periodically in Intel Technology Centers. More detailed courses on designing with microprocessor cores and peripherals are also available, and may be combined with the basic courses. All courses contain a lecture portion on the design rules and methodology, and extensive hands-on lab exercises. Contact the local Technology Center for a complete listing of training courses and to schedule course times.



Engineering Services

Intel Technology Centers are capable of assisting in all or part of an ASIC design. Engineers are available for consultation, or for complete design specification, verification, and test.

Intel Technology Centers

California
Intel Technology Center
3065 Bowers Avenue
Santa Clara, CA 95051
Tel: (408) 987-5400

Massachusetts
Intel Technology Center
3 Carlisle Rd
Westford, MA 01886
Tel: (617) 692-3222

United Kingdom
Intel Technology Center
Piper's Way
Swindon SN31RJ
Wiltshire, U.K.
Tel: 0793 696000

France
Intel Technology Center
1 Rue Edison, BP 303
78054 Saint-Quentin-
en-Yvelines Cedex
Paris, France
Tel: 33-1-30-57-7000

Printed in USA/AC003/888/3K/RR SM

MENTOR-BASED ASIC LIBRARIES

- **Support for Intel's entire ASIC library, including complex cells**
- **Provides design-entry and simulation capability on Mentor Idea Stations**
- **Added-value utilities for rules-checking, test-pattern development, and design-flow audit**
- **Timing models correlated with Intel's referent simulator to reduce design iterations**
- **Post-layout timing values incorporated into simulation (back-annotation)**

PRODUCT DESCRIPTION

The Mentor Design Entry Package and Timing Calculation Packages provide complete ASIC design support to users of Mentor Idea Stations. These software libraries contain symbols, timing models, and added-value utilities to support Intel cell-based ASIC designs. The utilities and libraries are well-integrated with Mentor's NETED and QuickSim to provide a productive development environment with minimal design risk.

FEATURES

Schematic Entry Symbols

The Mentor Design Entry Package contains symbols for 150 logic cells in the standard-cell library, plus microprocessor core and peripheral cells, including the UC51, 8284, 8288, 82284, 82288, 8237, 8254, 8259. Also included is support for 11 different telescoping cells (adders, counters, comparitors, registers) with widths of 1-12 bits. The symbols are used in conjunction with Mentor's NETED software.

Functional Simulation

The Mentor Design Entry Package contains the intrinsic-delay timing models for the logic elements in Intel's ASIC library. This will enable the user to debug the logic of a design, but not the timing accuracy. The models are used with QuickSim.

Full-Functional Timing Simulation

The Mentor-based Basic Timing Package contains simulation models for the SSI/MSI elements of Intel's ASIC library. These contain the load-dependent delays for each pin of each cell, under best-case, worst-case, and normal conditions. Both pre-layout delays (algorithmically determined) and post-layout delays (back-annotated from the layout database) are supported, using QuickSim.

AUGUST 1988
ORDER NUMBER: 240271-001

IDE Utilities

The Intel Development Environment is the term used to describe the tools and services that run on engineering workstations and the internal mainframe environment. A set of utilities common to the development platforms has been created to ease the development of an ASIC design, as well as reduce the number of design iterations required for acceptance. The utilities are outlined in the table below.

IDE_AUDIT	Reports the execution time, version number, and errors for the tools executed in the design flow.
IDE_AUTOTPDL	Generates a skeleton TPDL file, using design elements extracted from the database.
IDE_BA	Backannotates post-layout interconnect delays into the design database.
IDE_ERC	Performs electrical rules checks, such as pad connections, floating inputs, character names, etc.
IDE_EXPAND	Invokes EXPAND with IDE-specific property, parameter, and primitive information.
IDE_ITC	Incorporates estimated load-dependent delays into the design database.
IDE_NETLIST	Generates a hierarchical netlist in SDL format.
IDE_SIM	Establishes the simulation environment and invokes QuickSim.
IDE_STATS	Creates a report file that profiles the number and types of cells used in the design.
IDE_TIF	Converts simulation outputs to test pattern (.TPN) format.
IDE_TIFCOMP	Compares .TPN files for slow, fast, and typical simulations, and reports cycle slips.
IDE_TOGGLE	Generates a list of all untoggled nodes.
IDE_TPDL	Converts user-developed production test program (in TPDL format) to MISL format for simulation stimuli.
IDE_TRANSFER	Copies all necessary design files to floppy disk for design acceptance and transfer.

SPECIFICATIONS

Hardware Required:
Apollo DOMAIN System with at least 4 Mb RAM; eg, DN3000 or DN4000

Software Required:
Mentor Idea Station SW Rel 6.0 or later
For Design Entry: Mentor Capture Station
For Simulation: Mentor Idea Station

Media:
1/4" Cartridge Tape shipped with order
9-track MagTape available by request
5-1/4" Floppy disks available by request

Documentation Supplied:
Intel Cell-Based Design, Mentor Environment

ORDERING INFORMATION

M-DEP	*Mentor Design Entry Package*
MC-BTP	*Mentor Cell-Based Basic Timing Package*

Printed in USA/AC003/888/3K/RR SM

DAISY-BASED ASIC LIBRARIES

- **Support for Intel's entire ASIC library, including complex cells**
- **Provides design-entry and simulation capability on Daisy workstations**
- **Added-value utilities for rules-checking, test-pattern development, and design-flow audit**
- **Timing models correlated with Intel's referent simulator to reduce design iterations**
- **Post-layout timing values incorporated into simulation (back-annotation)**

PRODUCT DESCRIPTION

The Daisy Design Entry Package and Timing Calculation Packages provide complete ASIC design support to users of Daisy workstations (Logician, Personal Logician, MegaLogician, and Entry! system). These software libraries contain symbols, timing models, and added-value utilities to support Intel cell-based ASIC designs. The utilities and libraries are well-integrated with Daisy's ACE, DED, and DLS2, to provide a productive development environment with minimal design risk.

FEATURES

Schematic Entry Symbols

The Daisy Design Entry Package contains symbols for 150 logic cells in the standard-cell library, plus microprocessor core and peripheral cells, including the UC51, 8284, 8288, 82284, 82288, 8237, 8254, 8259. Also included is support for 11 different telescoping cells (adders, counters, comparitors, registers) with widths of 1-12 bits. The symbols are used in conjunction with Daisy's ACE and DED2 software.

Functional Simulation

The Daisy Design Entry Package contains the intrinsic-delay timing models for the logic elements in Intel's ASIC library. This will enable the user to debug the logic of a design, but not the timing accuracy. The models are used with DED2.

Full-Functional Timing Simulation

The Daisy-based Basic Timing Package contains simulation models for the SSI/MSI elements of Intel's ASIC library. These contain the load-dependent delays for each pin of each cell, under best-case, worst-case, and normal conditions. Both pre-layout delays (algorithmically determined) and post-layout delays (back-annotated from the layout database) are supported, using DLS2.

AUGUST 1988
ORDER NUMBER: 240270-001

IDE Utilities

The Intel Development Environment is the term used to describe the tools and services that run on engineering workstations and the internal mainframe environment. A set of utilities common to the development platforms has been created to ease the development of an ASIC design, as well as to reduce the number of design iterations required for acceptance. The utilities are outlined in the table below.

IDE_AUDIT	Reports the execution time, version number, and errors for the tools executed in the design flow.
IDE_AUTOTPDL	Generates a skeleton TPDL file, using design elements extracted from the database with SING.
IDE_NETLIST	Generates a hierarchical netlist in SDL format.
IDE_SOM2TPDL	Converts a stimulus file from SOM format to the TTABLE section appended to a TPDL file.
IDE_STATS	Creates a report file that profiles the number and types of cells used in the design.
IDE_TCAL	Executes TCAL (for load-dependent delay calculation) with specific user options.
IDE_TIF	Converts simulation outputs to test pattern (.TPN) format.
IDE_TIFCOMP	Compares .TPN files for slow, fast, and typical simulations, and reports cycle slips.
IDE_TOGGLE	Activates the toggle-test environment, invoking SIFT, SOM, DLS, and the toggle utility.
IDE_TPDL	Converts user-developed production test program (in TPDL format) to SOM format for simulation stimuli.
IDE_TRANSFER	Copies all necessary design files to floppy disk for design acceptance and transfer.

SPECIFICATIONS

Hardware Required:
For Schematic Entry and Simulation: Daisy Systems' Logician, Personal Logician (286 or 386), MegaLogician, or DOS-based "Entry!" system.
Required: 4 Mb RAM, 40 Mb Disk
Suggested: 8 Mb RAM, 80-140 Mb Disk

Software Required:
Daisy DNIX operating system, release 5.02 or later

Media:
5.25" or 8" Diskettes

Documentation Supplied:
Intel Cell-based Design, Daisy Environment

ORDERING INFORMATION

D-DEP	*Daisy Design Entry Package*
DC-BTP	*Daisy Cell-Based Basic Timing Package*

Printed in USA/ AC/003/ 888/ 3K/ RR SM

MCS®-96 8098

17

intel

MCS®-96
8098 ARCHITECTURAL OVERVIEW

The 8098/8398 microcontrollers are pin compatible eight-bit data bus interface members of Intel Corporation's MCS®-96 architectural family of 16-bit microcontrollers; all of which feature a unique 256-byte Register Arithmetic Logic Unit (RALU) as part of the Central Processing Unit (CPU). There are actually three devices associated with what is generally referred to as the 8098. The product designation of 8398 is the nomenclature for the core RALU architecture intended for standalone designs with 8K × 8-bit on-chip read only memory (ROM) addresses. This on-chip memory is mask programmable for executable code and 8-, 16-, or with some arithmetic instructions 32-bit memory data types. The 8795BH is the 8K byte on-chip EPROM component for prototyping and low volume production; and the 8098 is the ROMless RALU only device intended for applications with external memory addressing to 64K × 8. The EPROM version is the highest functionality device, in that it is user programmable, but is provided in the same package and pinout as the 8098/8398 for easy prototyping. All software is upward compatible to 8X9XBH and the 80C196KX architectures. Notations will be made when features and functionality differ between the various devices.

The on-chip peripheral subsystems (see Figure 1) under microcontrol of the RALU can be separated into several sections for the purpose of describing its operation. There is a 17-bit arithmetic unit associated with the 256-byte on-chip register file, a programmable High Speed Input/Output (HSIO) engine, a four input analog aquisition system (with 10-bit Analog to Digital Converter), an internal interrupt controller and wait state ready logic, a synchronous/asynchronous serial port, and a Pulse Width Modulated (PWM) output for use with digital to analog conversion circuits. There are also clock generators as well as software and hardware timers that help support the overall operation of the device. The 17-bit RALU, 10-bit A/D system, 8-bit PWM, and programmable High Speed I/O make the 8098 a unique high performance product in the 8-bit microcontroller industry. Let's examine each feature.

1.0 CPU OPERATION

The major components of the 8098 CPU are the fast on-chip Register File, Special Function Registers (SFRs), Memory Controller and RALU (Register/Arithmetic Logic Unit). Communication with the outside world is done through either the SFRs or the Memory Controller. The RALU does not use an accumulator, it operates directly on the 256-byte address space made up of the Register File and the SFRs. Efficient I/O, A/D, PWM and serial port operations are possible by directly controlling the I/O through the SFRs. The main benefits of this structure are the ability to quickly change context, the absence of accumulator bottleneck, and fast throughput and I/O times.

1.1 CPU Buses

A Control Unit and two busses connect the Register File and RALU. Figure 1 shows the CPU with its major bus connections. The two buses are the "A-Bus" which is 8 bits wide, and the "D-Bus" which is 16 bits wide. The D-Bus transfers data only between the RALU and the Register File or Special Function Registers (SFRs). The A-Bus is used as the address bus for the above transfers or as a multiplexed address/data bus connecting to the "Memory Controller". Any accesses of either the internal ROM or external memory are done through the Memory Controller.

1.2 CPU Register File

The Register File contains 232 bytes of RAM which can be accessed as bytes, words, or double-words. Since each of these locations can be used by the RALU, there are essentially 232 "accumulators". The first word in the Register File is reserved for use as the stack pointer so it can not be used for data when stack manipulations are taking place. Addresses for accessing the Register File and SFRs are temporarily stored in two 8-bit address registers by the CPU hardware.

1.3 RALU Control

1.4 RALU

Most calculations performed by the 8098 take place in the RALU. The RALU, shown in Figure 2, contains a 17-bit ALU, the Program Status Word (PSW), the Program Counter (PC), a loop counter, and three temporary registers. All of the registers are 16 bits or 17 bits (16 plus sign extension) wide. Some of the registers have the ability to perform simple operations to offload the ALU.

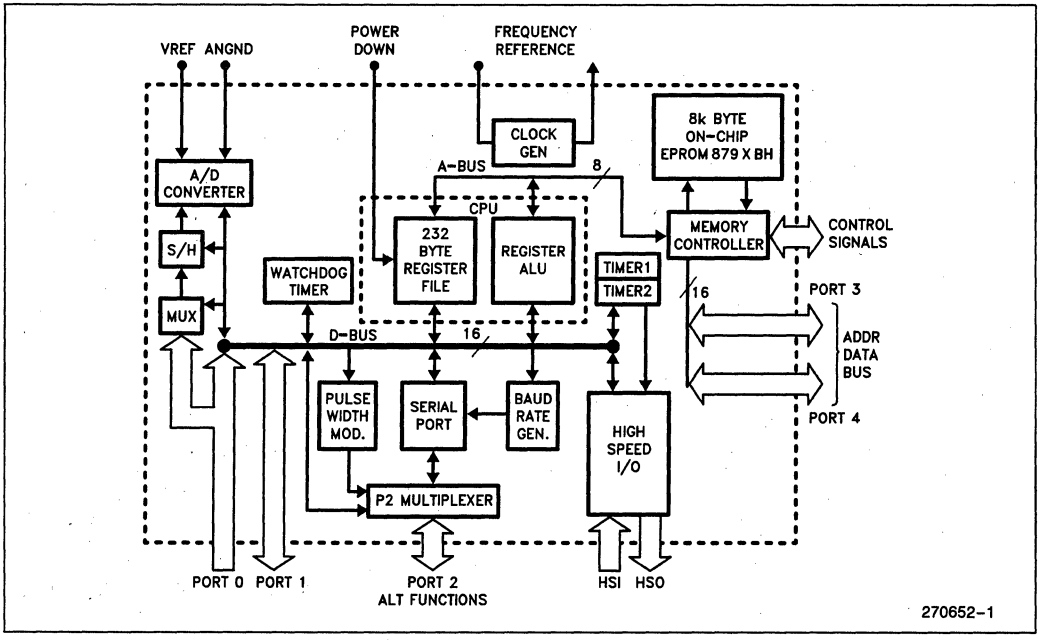


Figure 1. Block Diagram

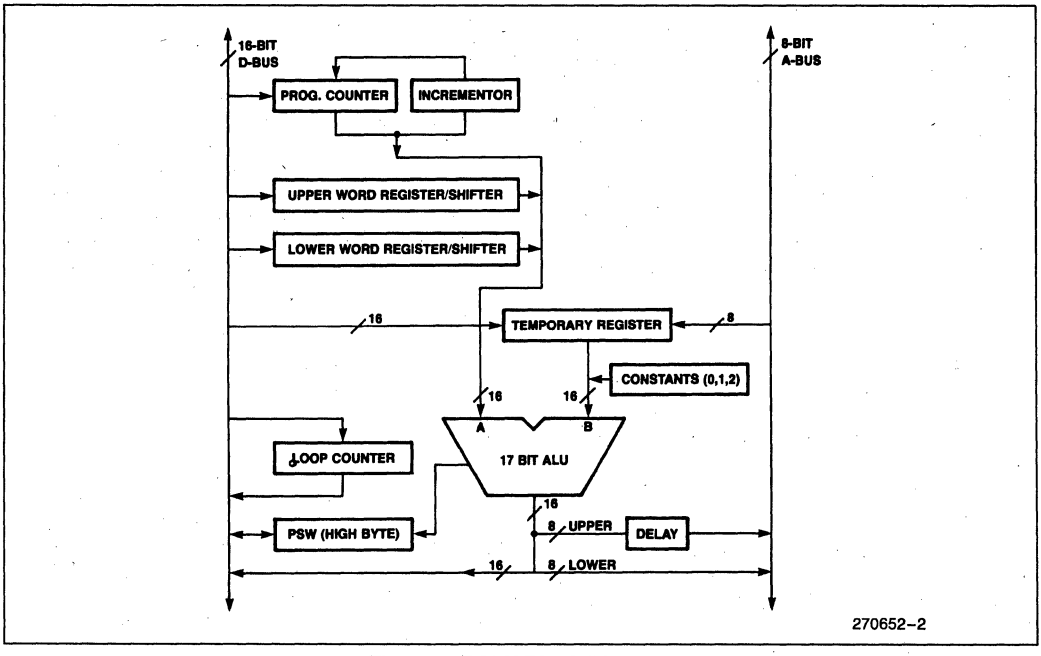


Figure 2. RALU Block Diagram

A separate incrementor is used for the PC; however, jumps must be handled through the ALU. Two of the temporary registers have their own shift logic. These registers are used for the operations which require logical shifts, including Normalize, Multiply and Divide. The "Lower Word" register is used only when double-word quantities are being shifted, the "Upper Word" register is used whenever a shift is performed or as a temporary register for many instructions. Repetitive shifts are counted by the 5-bit "Loop Counter".

The DELAY shown in Figure 2 is used to convert the 16-bit bus into an 8-bit bus. This is required as all addresses and instructions are carried on the 8-bit A-Bus. Several constants, such as 0, 1 and 2 are stored in the RALU for use in speeding up certain calculations. These come in handy when the RALU needs to make a 2's complement number or perform an increment or decrement instruction.

1.5 Internal Timing

The 8098 requires an input clock frequency of between 6.0 MHz and 12 MHz to function. This frequency can be applied directly to XTAL1. Alternatively, since XTAL1 and XTAL2 are inputs and outputs of an inverter, it is also possible to use a crystal to generate the clock. A block diagram of the oscillator section is shown in Figure 3.

The crystal or external oscillator frequency is divided by 3 to generate the three internal timing phases as shown in Figure 4. Each of the internal phases repeat every 3 oscillator periods: 3 oscillator periods are referred to as one "state time", the basic time measurement for 8098 operations. Most internal operations are synchronized to either Phase A, B or C, each of which have a 33% duty cycle. Phase A is represented externally by CLKOUT, a signal available on the 68-pin part. Phases B and C are not available externally. The relationships of XTAL1, CLKOUT, and Phases A, B and C are shown in Figure 4. It should be noted that propagation delays have not been taken into account in this diagram.

The $\overline{\text{RESET}}$ line can be used to start the 8098 at an exact time to provide for synchronization of test equipment and multiple chip systems. Use of this feature is fully explained under $\overline{\text{RESET}}$, Section 13.

2.0 MEMORY SPACE

The addressable memory space on the 8098 consists of 64 Kbytes, most of which is available to the user for program or data memory. Locations which have special purposes are 0000H through 00FFH and 1FFEH through 2080H. All other locations can be used for either program or data storage or for memory mapped peripherals. A memory map is shown in Figure 5.

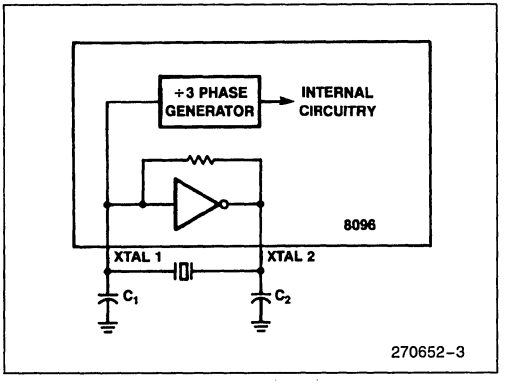


Figure 3. Block Diagram of Oscillator

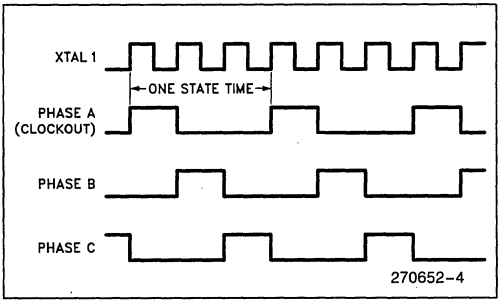


Figure 4. Internal Timings Relative to XTAL 1

2.1 Register File

Locations 00H through 0FFH contain the Register File and Special Function Registers (SFRs). No code can be executed from this internal RAM section. If an attempt to execute instructions from locations 000H through 0FFH is made, the instructions will be fetched from external memory. This section of external memory is reserved for use by Intel development tools.

The RALU can operate on any of the 256 internal register locations. Locations 00H through 17H are used to access the SFRs. Locations 18H and 19H contain the stack pointer. These are not SFRs, and may be used as standard RAM if stack operations are not being performed. The stack pointer must be initialized by the user program and can point anywhere in the 64K memory space. The stack builds down. There are no restrictions on the use of the remaining 230 locations except that code cannot be executed from them.

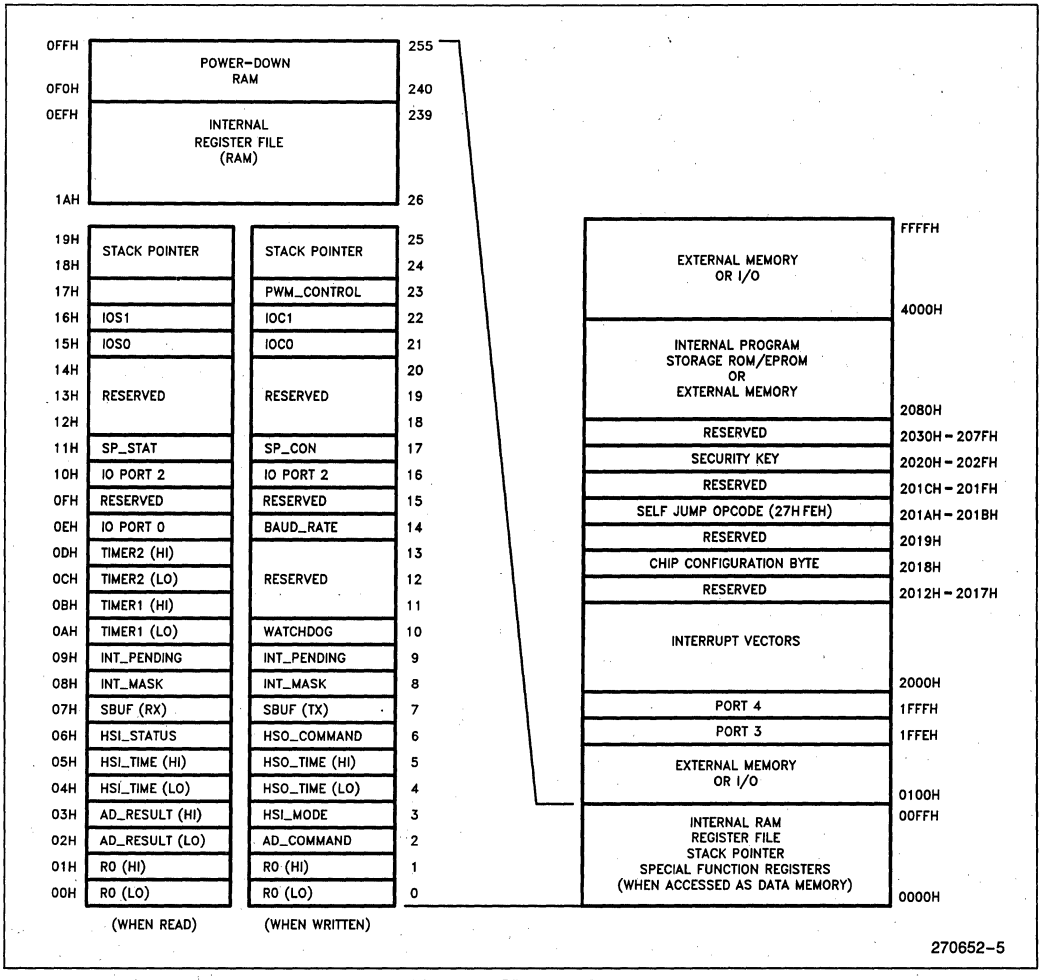


Figure 5. Memory Map

2.2 Special Function Registers

All of the I/O on the 8098 is controlled through the SFRs. Many of these registers serve two functions; one if they are read from, the other if they are written to. Figure 5 shows the locations and names of these registers. A summary of the capabilities of each of these registers is shown in Figure 6, with complete descriptions reserved for later sections.

Within the SFR space are several registers labeled "RESERVED". These registers are reserved for future expansion and test purposes. Operations should not be performed with these registers as reads from them and writes to them may produce unexpected results.

2.3 Power Down

The upper 16 RAM locations (0F0H through 0FFH) receive their power from the V_{PD} pin. If it is desired to keep the memory in these locations alive during a power down situation, one need only keep voltage on the pin. The current required to keep the RAM alive is approximately 1 mA (refer to the datasheet for the exact specification). Both V_{CC} and V_{PD} must have power applied for normal operation.

To place the 8098 into a power down mode, the $\overline{\text{RESET}}$ pin is pulled low. Two state times later the part will be in reset. This is necessary to prevent the part from writing into RAM as the power goes down. The power may now be removed from the V_{CC} pin, the V_{PD} pin must remain within specifications. The 8098 can remain in this state for any amount of time and the 16 RAM bytes will retain their values.

To bring the 8098 out of power down, $\overline{\text{RESET}}$ is held low while V_{CC} is applied. The 8098 should have stable power when the clock circuit starts. Two state times after the oscillator has stabilized, the $\overline{\text{RESET}}$ pin can be pulled high. The 8098 will begin to execute code at location 02080H 10 state times after $\overline{\text{RESET}}$ is pulled high. Figure 7 shows a timing diagram of the power down sequence. To ensure that the 2 state time minimum reset time (synchronous with CLKOUT) is met, it is recommended that 10 XTAL1 cycles be used.

Register	Description	Section
R0	**Zero Register:** Always reads as a zero, useful for a base when indexing and as a constant for calculations and compares.	3
AD_RESULT	**A/D Result Hi/Low:** Low and high order Results of the A/D converter (byte read only)	8
AD_COMMAND	**A/D Command Register:** Controls the A/D	8
HSI_MODE	**HSI Mode Register:** Sets the mode of the High Speed Input unit.	6
HSI_TIME	**HSI Time Hi/Lo:** Contains the time at which the High Speed Input unit was triggered. (word read only)	6
HSO_TIME	**HSO Time Hi/Lo:** Sets the time or count for the High Speed Output to execute the command in the Command Register. (word write only)	7
HSO_COMMAND	**HSO Command Register:** Determines what will happen at the time loaded into the HSO Time Registers.	7
HSI_STATUS	**HSI Status Registers:** Indicates which HSI pins were detected at the time in the HSI Time registers and the current state of the pins.	6
SBUF (TX)	Transmit buffer for the serial port, holds contents to be outputted.	9
SBUF (RX)	Receive buffer for the serial port, holds the byte just received by the serial port.	9
INT_MASK	**Interrupt Mask Register:** Enables or disables the individual interrupts.	4
INT_PENDING	**Interrupt Pending Register:** Indicates that an interrupt signal has occurred on one of the sources and has not been serviced.	4
WATCHDOG	**Watchdog Timer Register:** Written to periodically to hold off automatic reset every 64K state times.	12
TIMER1	**Timer 1 Hi/Lo:** Timer 1 high and low bytes. (word read only)	5
TIMER2	**Timer 2 Hi/Lo:** Timer 2 high and low bytes. (word read only)	5
IOPORT0	**Port 0 Register:** Levels on pins of port 0.	10
BAUD_RATE	Register which determines the baud rate, this register is loaded sequentially.	9
IOPORT1	**Port 1 Register:** Used to read or write to Port 1.	10
IOPORT2	**Port 2 Register:** Used to read or write to Port 2.	10
SP_STAT	**Serial Port Status:** Indicates the status of the serial port.	9
SP_CON	**Serial Port Control:** Used to set the mode of the serial port.	9
IOS0	**I/O Status Register 0:** Contains information on the HSO status.	11
IOS1	**I/O Status Register 1:** Contains information on the status of the timers and of the HSI.	11
IOC0	**I/O Control Register 0:** Controls alternate functions of HSI pins, Timer 2 reset sources and Timer 2 clock sources.	11
IOC1	**I/O Control Register 1:** Controls alternate functions of Port 2 pins, timer interrupts and HSI interrupts.	11
PWM_CONTROL	**Pulse Width Modulation Control Register:** Sets the duration of the PWM pulse.	8

Figure 6. SFR Summary

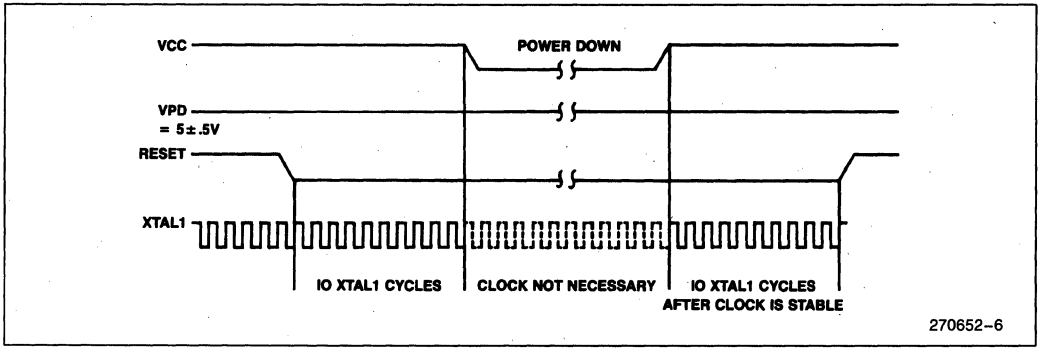


Figure 7. Power Down Timing

2.4 Reserved Memory Spaces

A listing of locations with special significance is shown in Figure 8. The locations marked Reserved are reserved by Intel for use in testing or future products. They must be filled with the Hex value FFH to insure compatibility with future parts.

Resetting the 8098 causes instructions to be fetched starting from location 2080H. This location was chosen to allow a system to have up to 8K of RAM continuous with the register file. Further information on reset can be found in Section 13.

0000H-	0017H	Register Mapped I/O (SFRs)
0018H-	0019H	Stack Pointer
1FFEH-	1FFFH	Ports 3 and 4
2000H-	2011H	Interrupt Vectors
2012H-	2017H	Reserved
2018H		Chip Configuration Byte
2019H		Reserved
201AH-	201BH	"Jump to Self" Opcode (27H FEH)
201CH-	201FH	Reserved
2020H-	202FH	Security Key
2030H-	207FH	Reserved
2080H		Reset Location

Figure 8. Registers with Special Significance

2.5 Internal ROM and EPROM

When an 8398 ROM part is ordered, or an 8795BH EPROM part is programmed, the internal memory locations 2080H through 3FFFH are used specified, as are the interrupt vectors, Chip Configuration Register and Security Key in locations 2000H through 202FH.

Instruction and data fetches from the internal ROM or EPROM occur only if $\overline{EA}$ is tied high and the address is between 2000H and 3FFFH. At all other times data is accessed from either the internal RAM space or external memory and instructions are fetched from external memory. The $\overline{EA}$ pin is latched on $\overline{RESET}$ rising. Information on programming EPROMs can be found in the datasheet.

2.6 Memory Controller

The RALU talks to the memory (except for the locations in the register file and SFR space) through the memory controller which is connected to the RALU by the A-Bus and several control lines. Since the A-Bus is eight bits wide, the memory controller uses a Slave Program Counter to avoid having to always get the instruction location from the RALU. This slave PC is incremented after each fetch. When a jump or call occurs, the slave PC must be loaded from the A-Bus before instruction fetches can continue.

In addition to holding a slave PC, the memory controller contains a 4 byte queue to held speed execution. This queue is transparent to the RALU and to the user unless wait states are forced during external bus cycles.

2.7 System Bus

There are several operating modes on the 8098. The standard bus mode uses a 16-bit multiplexed address and 8-bit data bus. In addition, there are several options available on the type of control signals used by the bus. In the standard mode, external memory is addressed through lines AD0 through AD15 which form a 16-bit address bus multiplexed with an 8-bit data bus. These lines share pins with I/O Ports 3 and 4. The falling edge of the Address Latch Enable (ALE) line is used to provide a signal to a transparent latch (i.e. 74LS373) to hold the lower eight address bits while data is placed on the bus.

To avoid confusion during the explanation of the memory system it is reasonable to give names to the demultiplexed address/data signals. The address signals will be called MA0 through MA15 (Memory Address), and the data signals will be called MD0 through MD7 (Memory Data).

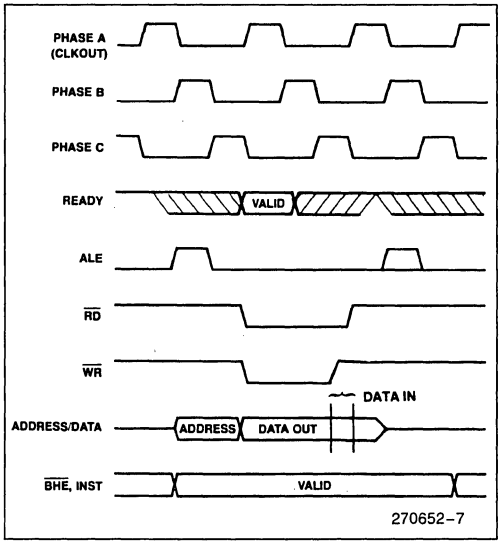


Figure 9. External Memory Timings

TIMINGS

Figure 9 shows the idealized waveforms related to the following description of external memory manipulations. For exact timing specifications please refer to the latest datasheet. When an external memory fetch begins, the Address Latch Enable (ALE) line rises, and the address is put on AD0–AD15. ALE then falls, the address is taken off the pins, and the $\overline{RD}$ (Read) signal goes low. When $\overline{RD}$ falls, external memory should present its data to the 8098.

BUS PERFORMANCE

When using an 8-bit external bus with 16-bit data, some performance degradation is to be expected when compare to fetching the same data internally. On the 8098, execution times will slow down with an 8-bit bus if any of three conditions occur. First, word writes to external memory will cause the executing instruction to take two extra state times to complete. Second, word reads from external memory will cause a one state time extension of instruction execution time. Finally, if the prefetch queue is empty when an instruction fetch is requested, instruction execution is lengthened by one state time for each byte that must be externally acquired (worst case is the number of bytes in the instruction minus one).

READ

The data from the external memory must be on the bus and stable for a minimum of the specified setup time before the rising edge of $\overline{RD}$. The rising edge of $\overline{RD}$ latches the information into the 8098.

WRITE

Writing to external memory requires timings that are similar to those required when reading from it. The main difference is that the write ($\overline{WR}$) signal is used instead of the $\overline{RD}$ signal. The timings are the same until the falling edge of the $\overline{WR}$ line. At this point the 8098 removes the address and places the data on the bus. When the $\overline{WR}$ line goes high the data should be latched to the external memory. A unique feature of the 8098 is its ability to use the Chip Configuration Register value (CCR described later) to configure the timing placement of the falling edge of $\overline{WR}$. The exact timing specifications for memory accesses can be found in the datasheet, but Figure 9 contains a conceptual diagram of this signal placement.

READY

A ready line is available on the 8098 to extend the width of the $\overline{RD}$ and $\overline{WR}$ pulses in order to allow access of slow memories or for DMA purposes. If the READY line is low by the specified time after ALE falls, the 8098 will hold the bus lines to their values at the falling edge of CLKOUT. When the READY line rises the bus cycle will continue with the next falling edge of CLKOUT.

The 8098 has the ability to internally limit the number of wait states to 1, 2 or 3 as determined by the value in the Chip Configuration Register (CCR). Using the CCR for ready timing is discussed at the end of this section. If a ready limit is set, the TLLYH MAX specification is not used. If READY is held low at reset, three wait-states will be inserted into the CCR fetch cycle.

OPERATING MODES

The 8098 supports a variety of options to simplify memory systems, interfacing requirements and ready control. Bus flexibility is provided by allowing selection of bus control signal definitions. In addition, several ready control modes are available to simplify the external hardware requirements for accessing slow devices. The Chip Configuration Register (CCR) is used to store the operating mode information.

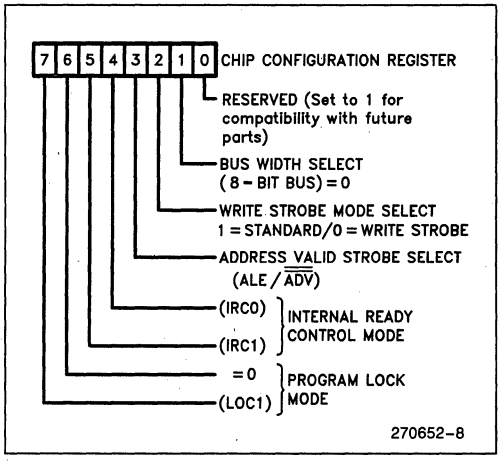


Figure 10. Chip Configuration Register

CHIP CONFIGURATION REGISTER (CCR)

Configuration information is stored in the Chip Configuration Register (CCR). Three of the bits in the register specify the bus control mode and ready control mode. Two bits also govern the level of ROM/EPROM protection. The CCR bit map is shown in Figure 10. The functions associated with each bit are described in this section.

The CCR is loaded on reset with the Chip Configuration Byte, located at address 2018H. The CCR register is a non-memory mapped location that can only be written to during the reset sequence; once it is loaded it cannot be changed until the next reset occurs. The 8098 will correctly read this location in every bus mode.

If the $\overline{EA}$ pin is set to a logical 0, the access to 2018H comes from external memory. If $\overline{EA}$ is a logical 1, the access comes from internal ROM/EPROM. If $\overline{EA}$ is = 12.5V, the CCR is loaded with a byte from a separate non-memory-mapped location called PCCB (Programming CCB). The Programming mode is described in the datasheet.

BUS CONTROL

Using the CCR, the 8098 can be made to provide bus control signals of several types. Two control lines have dual functions designed to reduce external hardware. Bits 2 and 3 of the CCR specify the functions performed by these control lines.

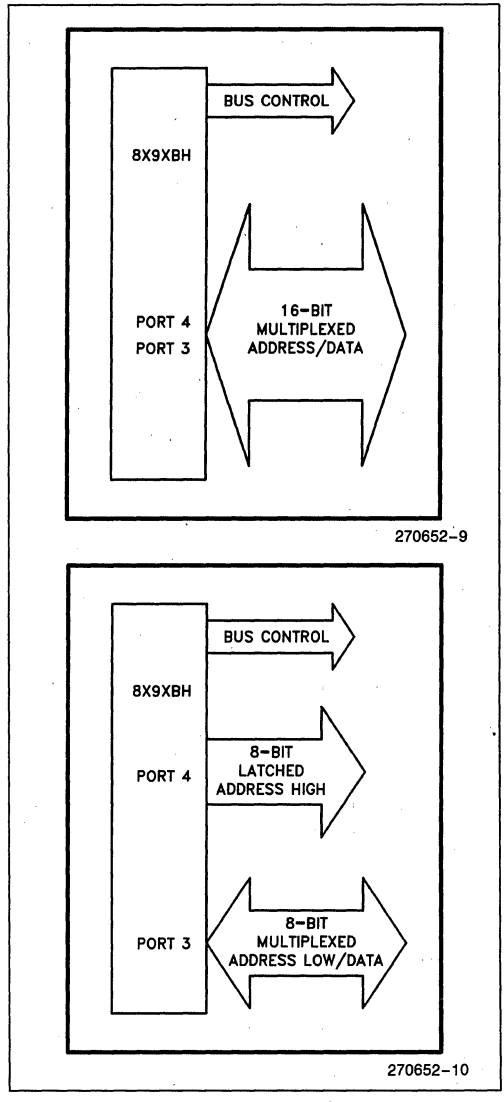


Figure 11. Bus Width Options

Standard Bus Control

If CCR bits 2 and 3 are 1s, then the standard 8098 control signals $\overline{WR}$ and ALE are provided (Figure 12) $\overline{WR}$ will come out for every write. ALE will rise as the address starts to come out, and will fall to provide the signal to externally latch the address.

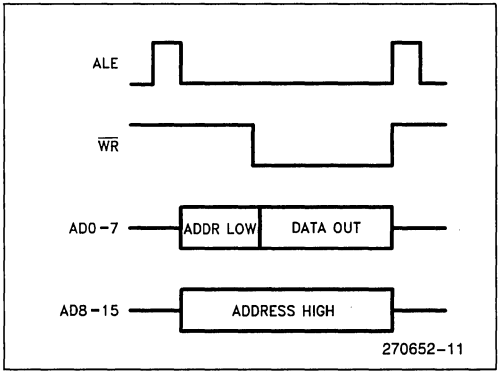


Figure 12. Standard Bus Control

Address Valid Strobe Mode

If CCR bit 3 is a 0, then an Address Valid strobe is provided in the place of ALE (Figure 14). When the address valid mode is selected, ADV will go low after an external address is setup. It will stay low until the end of the bus cycle, where it will go inactive high. This can be used to provide a chip select for external memory.

Write Strobe Mode

A unique ability of the Bus Controller is to utilize the Chip Configuration Register to select at reset time the width of the WR signal by changing the position of the falling edge relative to the memory cycle. Clearing bit 2 of the CCR to 0 will enable a shorter WR width. This is useful when interfacing to device that latch data on the falling edge of the WR signal.

Address Valid with Write Strobe

If both CCR bits 2 and 3 are 0s, both the Address Valid strobe and the shortened Write Strobe timing will be provided for bus control. Figure 15 shows these signals.

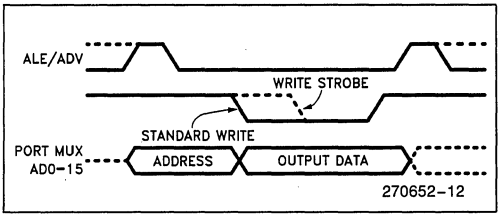


Figure 13. Write Strobe Mode

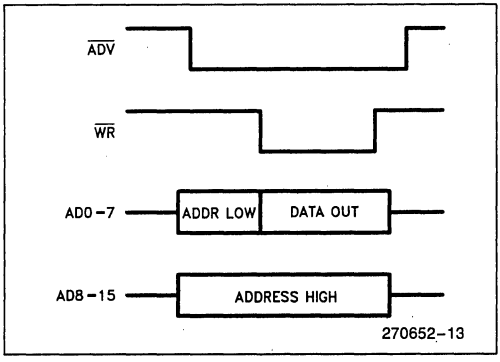


Figure 14. Address Valid Strobe Mode

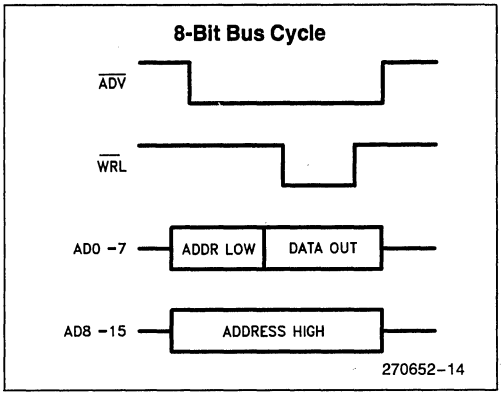


Figure 15. Write Strobe with Address Valid Strobe

READY CONTROL

To simplify read control, four modes of internal ready control logic have been provided. The modes are chosen by properly configuring bits 4 and 5 of the CCR.

The internal ready control logic can be used to limit the number of wait states that slow devices can insert into the bus cycle. When the READY pin is pulled low, wait states will be inserted into the bus cycle until the READY pin goes high, or the number of wait states equals the number specified by CCR bits 4 and 5, whichever comes first. Figure 16 shows the number of wait states that can be selected. Internal Ready control can be disabled by loading 11 into bits 4 and 5 of the CCR.

IRC1	IRC0	Description
0	0	Limit to 1 Wait State
0	1	Limit to 2 Wait States
1	0	Limit to 3 Wait States
1	1	Disable Internal Ready Control

FIGURE 16. Internal Ready Control

This feature provides for simple ready control. For example, every slow memory chip select line could be ORed together and be connected to the READY pin with CCR bits 4 and 5 programmed to give the desired number of wait states to the slow devices.

ROM/EPROM LOCK

Several modes of program memory lock are available on the 8398/9795BH. CCR bits 6 and 7 (LOC0, LOC1) select whether internal program memory can be read (or written in EPROM parts) by a program executing from external memory. The modes are shown in Figure 17. Internal ROM/EPROM addresses 2020H through 3FFFH are protected from reads while 2000H through 3FFFH are protected from writes, as set by the CCR.

LOC1	Protection
0	Read Protected
1	No Protection

Figure 17. Program Lock Modes

Only code executing from internal memory can read protected internal memory, while a write protected memory cannot be written to, even from internal execution. As a result of 8098 prefetching of instructions, however, accesses to protected memory are not allowed for instructions located above 3FFAH. This is because the lock protection mechanism is gated off of the Memory Controller's slave program counter and not the CPU program counter. If the bus controller receives a request to perform a read of protected memory, the read sequence occurs with indeterminate data being returned to the CPU. Note that the interrupt vectors and the CCR are not protected.

To provide verification and testing when the program lock feature is enabled, the 8398 and 8795BH verify the security key before programming or test modes are allowed to read from protected memory. Before protected memory can be read, the chip reads external memory locations 4020H through 402FH and compares the values found to the internal security key located from 2020H through 202FH. Only when the values exactly match will accesses to protected memory be allowed.

The details of ROM/EPROM accessing are discussed in the datasheets.

3.0 SOFTWARE OVERVIEW

This section provides information on writing programs to execute in the 8098. Additional information can be found in the following documents:

INTEL EMBEDDED CONTROLLER HANDBOOK 1989 16-BIT
Order Number 270646

MCS-96 MACRO ASSEMBLER USER'S GUIDE
Order Number I86 ASM96 (Intel Systems)
Order Number D86 ASM96NL (DOS Systems)

PL/M-96 USER'S GUIDE
Order Number I86 PLM96 (Intel Systems)
Order Number D86 PLM96NL (DOS Systems)

C96 USER'S GUIDE
Order Number D86 C96NL (DOS Systems)

Throughout this section, short sections of code are used to illustrate the operation of the device. For these sections it has been assumed that a set of temporary registers have been predeclared. The names of these registers have been chosen as follows:

AX, BX, CX and DX are 16-bit registers.
AL is the low byte of AX, AH is the high byte.
BL is the low byte of BX
CL is the low byte of CX
DL is the low byte of DX

These are the same as the names for the general data registers used in the 80186. It is important to note, however, that in the 8098, these are not dedicated register, but merely the symbolic names assigned by the programmer to an eight byte region within the onboard register file.

3.1 Operand Types

The 8098 architecture provides support for a variety of data types which are likely to be useful in an 8098 control application. In the discussion of these operand types that follows, the names adopted by the PLM-96 programming language will be used where appropriate. To avoid confusion, the name of an operand type will be capitalized. A "BYTE" is an unsigned eight bit variable; a "byte" is an eight bit unit of data of any type.

BYTES

BYTES are unsigned 8-bit variables which can take on the values between 0 and 255 Arithmetic and relational operators can be applied to BYTE operands but the result must be interpreted in modulo 256 arithmetic. Logical operations on BYTES are applied bitwise. Bits within BYTES are labeled from 0 to 7, with 0 being the least significant bit. There are no alignment restrictions for BYTES, so they may be placed anywhere in the MCS-96 address space.

WORDS

WORDS are unsigned 16-bit variables which can take on the values between 0 and 65535. Arithmetic and relational operators can be applied to WORD operands but the result must be interpreted modulo 65536. Logical operations on WORDS are applied bitwise. Bits within words are labeled from 0 to 15 with 0 being the least significant bit. WORDS must be aligned at even byte boundaries in the MCS-96 address space. The least significant byte of the WORD is in the even byte address and the most significant byte is in the next higher (odd) address. The address of a word is the address of its least significant byte. Word operations to odd addresses are not guaranteed to operate in a consistent manner.

SHORT-INTEGERS

SHORT-INTEGERS are 8-bit signed variables which can take on the values between -128 and $+127$. Arithmetic operations which generate results outside of the range of a SHORT-INTEGER will set the overflow indicators in the program status word. The actual numeric result returned will be the same as the equivalent operation on BYTE variables. There are no alignment restrictions on SHORT-INTEGERS so they may be placed anywhere in the MCS-96 address space.

INTEGERS

INTEGERS are 16-bit signed variables which can take on the values between $-32,768$ and 32,767. Arithmetic operations which generate results outside of the range of an INTEGER will set the overflow indicators in the program status word. The actual numeric result returned will be the same as the equivalent operation on WORD variables. INTEGERS conform to the same alignment and addressing rules as do WORDS.

BITS

BITS are single-bit operands which can take on the Boolean values of true and false. In addition to the normal support for bits as components of BYTE and WORD operands, the 8098 provides for the direct testing of any bit in the internal register file. The MCS-96 architecture requires that bits be addressed as components of BYTES or WORDS, it does not support the direct addressing of bits that can occur in the MCS-51 architecture.

DOUBLE-WORDS

DOUBLE-WORDS are unsigned 32-bit variables which can take on the values between 0 and 4,294,967,295. The MCS-96 architecture provides direct support for this operand type only for shifts and as the dividend in a 32 by 16 divide and the product of a 16 by 16 multiply. For these operations a DOUBLE-WORD variable must reside in the on-board register file of the 8098 and be aligned at an address which is evenly divisible by 4. A DOUBLE-WORD operand is addressed by the address of its least significant byte. DOUBLE-WORD operations which are not directly supported can be easily implemented with two WORD operations. For consistency with Intel provided software the user should adopt the conventions for addressing DOUBLE-WORD operands which are discussed in Section 3.5.

LONG-INTEGERS

LONG-INTEGERS are 32-bit signed variables which can take on the values between 2,147,483,648 and 2,147,483,647. The 8098 architecture provides direct support for this data type only for shifts and as the dividend in a 32 by 16 divide and the product of a 16 by 16 multiply.

LONG-INTEGERS can also be normalized. For these operations a LONG-INTEGER variable must reside in the onboard register file of the 8098 and be aligned at an address which is evenly divisible by 4. A LONG-INTEGER is addressed by the address of its least significant byte.

LONG-INTEGER operations which are not directly supported can be easily implemented with two INTEGER operations. For consistency with Intel provided software, the user should adopt the conventions for addressing LONG operands which are discussed in Section 3.5.

3.2 Operand Addressing

Operands are accessed within the address space of the 8098 with one of six basic addressing modes. Some of the details of how these addressing modes work are hidden by the assembly language. If the programmer is to take full advantage of the architecture, it is important that these details be understood. This section will describe the addressing modes as they are handled by the hardware. At the end of this section the addressing modes will be described as they are seen through the assembly language. The six basic address modes which will be described are termed register-direct, indirect, indirect with auto-increment, immediate, short-indexed and long-indexed. Several other useful addressing operations can be achieved by combining these basic addressing modes with specific registers such as the ZERO register or the stack pointer.

REGISTER-DIRECT REFERENCES

The register-direct mode is used to directly access a register from the 256 byte on-board register file. The register is selected by an 8-bit field within the instruction and register address and must conform to the alignment rules for the operand type. Depending on the instruction, up to three registers can take part in the calculation.

INDIRECT REFERENCES

The indirect mode is used to access an operand by placing its address in a WORD variable in the register file. The calculated address must conform to the alignment rules for the operand type. Note that the indirect address can refer to an operand anywhere within the address space of the 8098, including the register file. The register which contains the indirect address is selected by an eight bit field within the instruction. An instruction can contain only one indirect reference and the remaining operands of the instruction (if any) must be register-direct references.

INDIRECT WITH AUTO-INCREMENT REFERENCES

This addressing mode is the same as the indirect mode except that the WORD variable which contains the indirect address is incremented after it is used to address the operand. If the instruction operates on BYTES or SHORT-INTEGERS the indirect address variable will be incremented by one, if the instruction operates on WORDS or INTEGERS the indirect address variable will be incremented by two.

IMMEDIATE REFERENCES

This addressing mode allows an operand to be taken directly from a field in the instruction. For operations on BYTE or SHORT-INTEGER operands this field is eight bits wide, for operations on WORD or INTEGER operands the field is 16 bits wide. An instruction can contain only one immediate reference and the remaining operand(s) must be register-direct references.

SHORT-INDEXED REFERENCES

In this addressing mode an eight bit field in the instruction selects a WORD variable in the register file which is assumed to contain an address. A second eight bit field in the instruction stream is sign-extended and summed with the WORD variable to form the address of the operand which will take part in the calculation. Since the eight bit field is sign-extended, the effective address can be up to 128 bytes before the address in the WORD variable and up to 127 bytes after it. An instruction can contain only one short-indexed reference and the remaining operand(s) must be register-direct references.

LONG-INDEXED REFERENCES

This addressing mode is like the short-indexed mode except that a 16-bit field is taken from the instruction and added to the WORD variable to form the address of the operand. No sign extension is necessary. An instruction can contain only one long-indexed reference and the remaining operand(s) must be register-direct references.

ZERO REGISTER ADDRESSING

The first two bytes in the register file are fixed at zero by the 8098 hardware. In addition to providing a fixed source of the constant zero for calculations and comparisons, this register can be used as the WORD variable in a long-indexed reference. This combination of register selection and address mode allows any location in memory to be addressed directly.

STACK POINTER REGISTER ADDRESSING

The system stack pointer in the 8098 can be accessed as register 18H of the internal register file. In addition to providing for convenient manipulation of the stack pointer, this also facilitates the accessing of operands in the stack. The top of the stack, for example, can be accessed by using the stack pointer as the WORD variable in an indirect reference. In a similar fashion, the stack pointer can be used in the short-indexed mode to access data within the stack.

ASSEMBLY LANGUAGE ADDRESSING MODES

The ASM96 language simplifies the choice of addressing modes to be used in several respects:

Direct Addressing. The assembly language will choose between register-direct addressing and long-indexed with the ZERO register depending on where the operand is in memory. The user can simply refer to an operand by its symbolic name; if the operand is in the register file, a register-direct reference will be used, if the operand is elsewhere in memory, a long-indexed reference will be generated.

Indexed Addressing. The assembly language will choose between short and long indexing depending on the value of the index expression. If the value can be expressed in eight bits then short indexing will be used, if it cannot be expressed in eight bits then long indexing will be used.

The use of these features of the assembly language simplifies the programming task and should be used wherever possible.

3.3 Program Status Word

The program status word (PSW) is a collection of Boolean flags which retain information concerning the state of the user's program. The format of the PSW is shown in Figure 18. The information in the PSW can be broken down into two basic categories; interrupt control and condition flags. The PSW can be saved in the system stack with a single operation (PUSHF) and restored in a like manner (POPF).

INTERRUPT FLAGS

The lower eight bits of the PSW are used to individually mask the various sources of interrupt to the 8098. A logical "1" in these bit positions enables the servicing of the corresponding interrupt. These mask bits can be accessed as an eight bit byte (INT_MASK_address 8) in the on-board register file. Bit 9 in the PSW is the global interrupt disable. If this bit is cleared then all interrupts will be locked out. Note that the various interrupts are collected in the INT_PENDING register even if they are locked out. Execution of the corresponding service routines will procede according to their priority when they become enabled. Further information on the interrupt structure of the 8098 can be found in Section 4.

CONDITION FLAGS

The remaining bits in the PSW are set as side effects of instruction execution and can be tested by the conditional jump instructions.

Z The Z (Zero) flag is set to indicate that the operation generated a result equal to zero. For the add-with-carry (ADDC) and subtract-with-borrow (SUBC) operations the Z flag is cleared if the result in non-zero but is never set. These two instructions are normally used in conjunction with the ADD and SUB instructions to perform multiple precision arithmetic. The operation of the Z flag for these instructions leaves it indicating the proper result for the entire multiple precision calculation.

N The N (Negative) flag is set to indicate that the operation generated a negative result. Note that the N flag will be set to the algebraically correct state even if the calculation overflows.

V The V (overflow) flag is set to indicate that the operation generated a result which is outside the range that can be expressed in the destination data type. For the SHL, SHLB and SHLL instructions, the V flag will be set if the most significant bit of the operand changes at any time during the shift.

BIT	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FLAG	Z	N	V	VT	C	—	I	ST	<Interrupt Mask Reg>							

Figure 18. PSW Register

VT The VT (oVerflow Trap) flag is set whenever the V flag is set but can only be cleared by an instruction which explicitly operates on it such as the CLRVT or JVT instructions. The operation of the VT flag allows for the testing for a possible overflow condition at the end of a sequence of related arithmetic operations. This is normally more efficient than testing the V flag after each instruction.

C The C (Carry) flag is set to indicate the state of the arithmetic carry from the most significant bit of the ALU for an arithmetic operation or the state of the last bit shifted out of the operand for a shift. Arithmetic Borrow after a subtract operation is the complement of the C flag. (i.e. if the operation generated a borrow then C = 0).

ST The ST (STicky bit) flag is set to indicate that during a right shift a 1 has been shifted first into the C flag and then been shifted out. The ST flag is undefined after a multiply operation. The ST flag can be used along with the C flag to control rounding after a right shift. Consider multiplying two eight bit quantities and then scaling the result down to 12 bits:

```
MULUB AX, CL, DL   ;AX: = CL*DL
SHR   AX, #4       ;Shift Right 4 Places
```

If the C flag is set after the shift, it indicates that the bits shifted off the end of the operand were greater-than or equal-to one half the least significant bit (LSB) of the result. If the C flag is clear after the shift, it indicates that the bits shifted off the end of the operand were less than half the LSB of the result. Without the ST flag, the rounding decision must be made on the basis of this information alone. (Normally the result would be rounded up if the C flag is set.) The ST flag allows a finer resolution in the rounding decision:

C ST	Value of the Bits Shifted Off
0 0	Value = 0
0 1	0 < Value < ½ LSB
1 0	Value = ½ LSB
1 1	Value > ½ LSB

Figure 19. Rounding Alternatives

Imprecise rounding can be a major source of error in a numerical calculation; use of the ST flag improves the options available to the programmer.

3.4 Instruction Set

The 8098 instruction set contains a full set of arithmetic and logical operations for the 8-bit data types BYTE and SHORT INTEGER and for the 16-bit data types WORD and INTEGER. The DOUBLE-WORD and LONG data types (32 bits) are supported for the products of 16 by 16 multiplies and the dividends of 32 by 16 divides and for shift operations. The remaining operations on 32-bit variables can be implemented by combinations of 16-bit operations. As an example the sequence:

```
ADD  AX, CX
ADDC BX, DX
```

performs a 32-bit addition, and the sequence

```
SUB  AX, CX
SUBC BX, DX
```

performs a 32-bit subtraction. Operations on REAL (i.e. floating-point) variables are not supported directly by the hardware but are supported by the floating-point library for the 8098 (FPAL-96) which implements a single precision subset of the proposed IEEE standard for floating-point operations. The performance of this software is significantly improved by the 8098 NORML instruction which normalizes a 32-bit variable and by the existence of the ST flag in the PSW.

In addition to the operations on the various data types, the 8098 supports conversions between these types. LDBZE (load byte zero extended) converts a BYTE to a WORD and LDBSE (load byte sign extended) converts a SHORT-INTEGER into an INTEGER. WORDS can be converted to DOUBLE-WORDS by simply clearing the upper WORD of the DOUBLE-WORD (CLR) and INTEGERS can be converted to LONGS with the EXT (sign extend) instruction.

The 8098 instructions for addition, subtraction, and comparison do not distinguish between unsigned words and signed integers. Conditional jumps are provided to allow the user to treat the results of these operations as either signed or unsigned quantities. As an example, the CMPB (compare byte) instruction is used to compare both signed and unsigned eight bit quantities. A JH (jump if higher) could be used following the compare if unsigned operands were involved or a JGT (jump if greater-than) if signed operands were involved.

Section 14.7 summarizes the operation of each of the instructions. Complete descriptions of each instruction and its timings can be found in the Instruction Set chapter. Examples of using the instruction set of the MCS-96 family can be found in Application Note AP-248 "Using the 8096".

3.5 Software Standards and Conventions

For a software project of any size it is a good idea to modularize the program and to establish standards which control the communication between these modules. The nature of these standards will vary with the needs of the final application. A common component of all of these standards, however, must be the mechanism for passing parameters to procedures and returning results from procedures. In the absence of some overriding consideration which prevents their use, it is suggested that the user conform to the conventions adopted by the PLM-96 programming language for procedure linkage. It is a very usable standard for both the assembly language and PLM-96 environment and it offers compatibility between these environments. Another advantage is that it allows the user access to the same floating point arithmetics library that PLM-96 uses to operate on REAL variables.

REGISTER UTILIZATION

PLM-96 adopts the simple and effective strategy of allocating the eight bytes between addresses 1CH and 23H as temporary storage. The starting address of this region is called PLMREG. The remaining area in the register file is treated as a segment of memory which is allocated as required.

ADDRESSING 32-BIT OPERANDS

These operands are formed from two adjacent 16-bit words in memory. The least significant word of the double word is always in lower address, even when the data is in the stack (which means that the most significant word must be pushed into the stack first). A double word is addressed by the address of its least significant byte. Note that the hardware supports some operations on double words (e.g. normalize and divide). For these operations the double word must be in the internal register file and must have an address which is evenly divisible by four.

SUBROUTINE LINKAGE

Parameters are passed to subroutines in the stack. Parameters are pushed into the stack in the order that they are encountered in the scanning of the source text. Eight-bit parameters (BYTES or SHORT-INTEGERS) are pushed into the stack with the high order byte undefined. Thirty-two bit parameters (LONG-INTEGERS, DOUBLE-WORDS and REALS) are pushed into the stack as two 16-bit values; the most significant half of the parameter is pushed into the stack first.

As an example, consider the following PLM-96 procedure:

```
example_procedure: PROCEDURE
(param1, param2, param3)

DECLARE param1 BYTE,

  param2 DWORD,
  param3 WORD;
```

When this procedure is entered at run time the stack will contain the parameters in the following order:

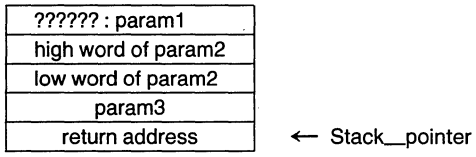


Figure 20. Stack Image

If a procedure returns a value to the calling code (as opposed to modifying more global variables) then the result is returned in the variable PLMREG.PLMREG is viewed as either an 8-, 16- or 32-bit variable depending on the type of the procedure.

The standard calling convention adopted by PLM-96 has several key features: Procedures can always assume that the eight bytes of register file memory starting at PLMREG can be used as temporaries within the body of the procedure. Code which calls a procedure must assume that the eight bytes of register file memory starting at PLMREG are modified by the procedure. The Program Status Word (PSW_see Section 3.3) is not saved and restored by procedures so the calling code must assume that the condition flags (Z, N, V, VT, C and ST) are modified by the procedure. Function results from procedures are always returned in the variable PLMREG.

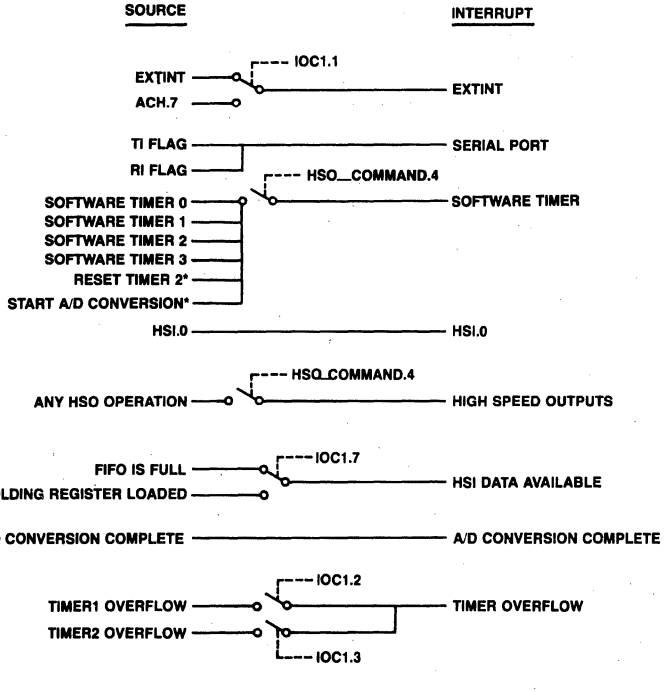
PLM-96 allows the definition of INTERRUPT procedures which are executed when a predefined interrupt occurs. These procedures do not conform to the rules of a normal procedure. Parameters cannot be passed to these procedures and they cannot return results. Since they can execute essentially at any time (hence the term interrupt), these procedures must save the PSW and PLMREG when they are entered and restore these values before they exit.

4.0 INTERRUPT STRUCTURE

There are 21 sources of interrupts on the 8098. These sources are gathered into 8 interrupt types as indicated in Figure 21. The I/O control registers which control some of the sources are indicated in the figure. Each of the eight types of interrupts has its own interrupt vector as listed in Figure 22. In addition to the 8 standard interrupts, there is a TRAP instruction which acts as a software generated interrupt. This instruction is not currently supported by the MCS-96 Assembler and is reserved for use in Intel development systems.

The programmer must initialize the interrupt vector table with the starting address of the appropriate interrupt service routine. It is suggested that any unused interrupts be vectored to an error handling routine. The error routine should contain recovery code that will not further corrupt an already erroneous situation. In a debug environment, it may be desirable to have the routine lock into a jump to self loop which would be easily traceable with emulation tools. More sophisticated routines may be appropriate for production code recoveries.

Three registers control the operation of the interrupt system: Interrupt Pending, Interrupt Mask, and the PSW which contains a global disable bit. A block diagram of the system is shown in Figure 23. The transition detector looks for 0 to 1 transitions on any of the sources. External sources have a maximum transition speed of one edge every state time. If this is exceeded the interrupt may not be detected. This means that a signal that transitions in one state time may not transition again until a subsequent state time.



NOTE:
*Only when initiated by the HSO unit.

Figure 21. All Possible Interrupt Sources

Vector	Vector Location (High Byte)	Vector Location (Low Byte)	Priority
Software	2011H	2010H	Not Applicable
Extint	200FH	200EH	7 (Highest)
Serial Port	200DH	200CH	6
Software Timers	200BH	200AH	5
HSI.0	2009H	2008H	4
High Speed Outputs	2007H	2006H	3
HSI Data Available	2005H	2004H	2
A/D Conversion Complete	2003H	2002H	1
Timer Overflow	2001H	2000H	0 (Lowest)

Figure 22. Interrupt Vector Locations

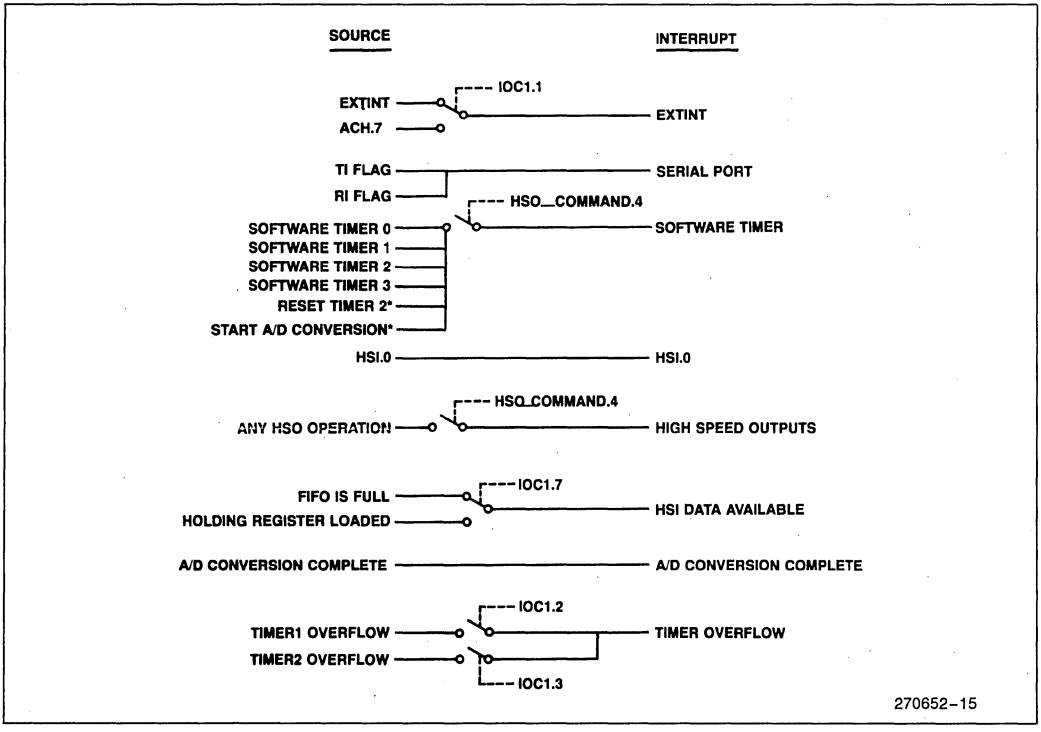


Figure 22. Block Diagram of Interrupt System

4.1 Interrupt Control

Interrupt Pending Register

When the hardware detects one of the eight interrupts it sets the corresponding bit in the pending interrupt register (INT_PENDING-09H). When the interrupt vector is taken, the pending bit is cleared. This register, the format of which is shown in Figure 24, can be read or modified as a byte register. It can be read to determine which of the interrupts are pending at any given time or modified to either clear pending interrupts or generate interrupts under software control. Any software which modifies the INT_PENDING register should ensure that the entire operation is indivisible. The easiest way to do this is to use the logical instructions in the two or three operand format, for example:

```
ANDB INT_PENDING, #11111101B
     ;Clears the A/D Interrupt

ORB INT_PENDING, #00000010B
    ;Sets the A/D Interrupt
```

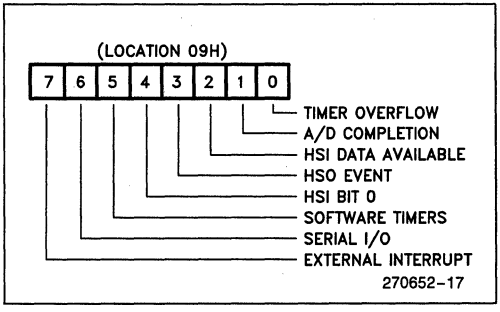


Figure 24. Interrupt Pending Register

Interrupt Mask Register

Individual interrupts can be enabled or disabled by setting or clearing bits in the interrupt mask register (INT_MASK-08H). The format of this register is the same as that of the Interrupt Pending Register shown in Figure 24.

The INT_MASK register can be read or written as byte register. A one in any bit position will enable the corresponding interrupt source and a zero will disable the source. The hardware will save any interrupts that occur by setting bits in the pending register, even if the interrupt mask bit is cleared. The INT_MASK register also can be accessed as the lower eight bits of the PSW so the PUSHF and POPF instructions save and restore the INT_MASK register as well as the global interrupt lockout and the arithmetic flags.

GLOBAL DISABLE

The processing of all interrupts can be disabled by clearing the I bit in the PSW. Setting the I bit will enable interrupts that have mask register bits which are set. The I bit is controlled by the EI (Enable Interrupts) and DI (Disable Interrupts) instructions. Note that the I bit only controls the actual servicing of interrupts. Interrupts that occur during periods of lockout will be held in the pending register and serviced on a prioritized basis when the lockout period ends.

4.2 Interrupt Priorities

The priority encoder looks at all of the interrupts which are both pending and enabled, and selects the one with the highest priority. The priorities are shown in Figure 22 (7 is highest, 0 is lowest). The interrupt generator then forces a call to the location in the indicated vector location. This location would be the starting location of the Interrupt Service Routine (ISR).

Interrupt service routines must share some data with other routines. Whenever the programmer is coding those sections of code which access these shared pieces of data, great care must be taken to ensure that the integrity of the data is maintained. Consider clearing a bit in the interrupt pending register as part of a non-interrupt routine:

```
LDB   AL,INT_PENDING
ANDB AL, #bitmask
STB   AL,INT_PENDING
```

This code works if no other routines are operating concurrently, but will cause occasional but serious problems if used in a concurrent environment. (All programs which make use of interrupts must be considered to be part of a concurrent environment). These problems can be avoided by assuring mutual exclusion which basically means that if more than one routine can change a variable, then the programmer must ensure exclusive access to the variable during the entire operation on the variable.

In many cases the instruction set of the 8098 allows the variable to be modified with a single instruction. The code in the above example can be implemented with a single instruction.

```
ANDB   INT_PENDING, #bitmask
```

4.4 Interrupt Timing

Interrupts are not always acknowledged immediately. If the interrupt signal does not occur prior to 4 state-times before the end of an instruction, the interrupt will not be acknowledged until after the next instruction has been executed. This is because an instruction is fetched and prepared for execution a few state times before it is actually executed.

There are 6 instructions which always inhibit interrupts from being acknowledged until after the next instruction has been executed. These instructions are:

EI, DI	Enable and Disable Interrupts
POPF, PUSHF	Pop and Push Flags
SIGND	Prefix to perform signed multiply and divide (Note that this is note an ASM-96 Mnemonic, but is used for signed multiply and divide)
TRAP	Software Interrupt

When an interrupt is acknowledged, the interrupt pending bit is cleared, and a call is forced to the location indicated by the specified interrupt vector. This call occurs after the completion of the instruction in process, except as noted above. The procedure of getting the vector and forcing the call requires 21 state times. If the stack is in external RAM an additional 3 state times are required.

The maximum number of state times required from the time an interrupt is generated (not acknowledged) until the 8098 begins executing code at the desired location is the time of the longest instruction, NORML (Normalize—42 state times), plus the 4 state times prior to the end of the previous instruction, plus the response time (21 to 24 state times). Therefore, the maximum response time is 70 (42 + 4 + 24) state times. This does not include the 12 state times required for PUSHF if it is used as the first instruction in the interrupt routine or additional latency caused by having the interrupt masked or disabled. Refer to Figure 25, Interrupt Response Time, to visualize an example of worst case scenario.

Interrupt latency time can be reduced by careful selection of instructions in areas of code where interrupts are expected. Using "EI" followed immediately by a long instruction (e.g. MUL, NORML, etc.) will increase the maximum latency by 4 state times, as an interrupt cannot occur between EI and the instruction following EI. The "DI", "PUSHF", "POPF" and "TRAP" instructions will also cause the same situation. Typically the PUSHF, POPF and TRAP instructions would only effect latency when one interrupt routine is already in process, as these instructions are seldom used at other times.

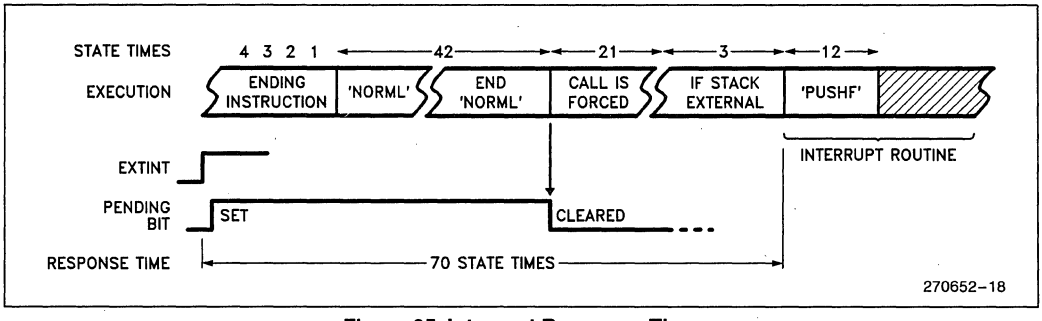


Figure 25. Interrupt Response Time

5.0 TIMERS

Two 16-bit timers are available for use on the 8098. The first is designated "Timer 1", the second, "Timer 2". Timer 1 is used to synchronize events to real time.

5.1 Timer 1

Timer 1 is clocked once every eight state times and can be cleared only by executing a reset. The only other way to change its value is by writing to 000CH but this is a test mode which sets both timers to 0FFFXH and should not be used in programs.

5.2 Timer 2

Timer 2 can be incremented by transitions (one count each transition, rising and falling) on HSI.1. The functionality of the timer is determined by the state of I/O Control Register 0, bit 7 (IOC0.7). To ensure that all CAM entries are checked each count of Timer 2, the maximum transition speed is limited to once per eight state times. Timer 2 can be cleared by: executing a reset by setting IOC0.1, by triggering HSO channel 0EH, or by pulling HSI.0 high. The HSO and CAM are described in Section 7 and 8. IOC0.3 and ICO0.5 control the resetting of Timer 2. Figure 26 shows the different ways of manipulating Timer 2.

5.3 Timer Interrupts

Both Timer 1 and Timer 2 can be used to trigger a timer overflow interrupt and set a flag in the I/O Status Register 1 (IOS1). The interrupts are controlled by IOC1.2 and IOC1.3 respectively. The flags are set in IOS1.5 and IOS1.4, respectively.

Caution must be used when examining the flags, as any access (including Compare and Jump on Bit) of IOS1 clears bits 0 through 5 including the software timer flags. It is, therefore, recommended to write the byte to a temporary register before testing bits. The general enabling and disabling of the timer interrupts are controlled by the Interrupt Mask Register bit 0. In all cases, setting a bit enables a function, while clearing a bit disables it.

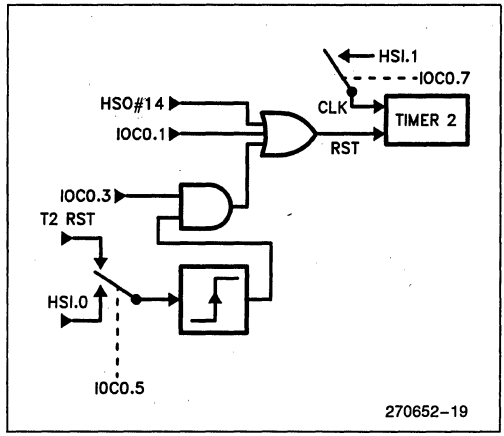


Figure 26. Timer 2 Clock and Reset Options

5.4 Timer Related Sections

The High Speed I/O unit is coupled to the timers in that the HSI records the value on Timer 1 when transitions occur and the HSO causes transitions to occur based on values of either Timer 1 or Timer 2.

A complete listing of the functions of IOS1, IOC0 and IOC1 are in Section 11.

6.0 HIGH SPEED INPUTS

The High Speed Input Unit (HSI), can be used to record the time at which an event occurs with respect to Timer 1. There are 4 lines (HSI.0 through HSI.3) which can be used in this mode and up to a total of 8 events can be recorded. HSI.2 and HSI.3 are bidirectional pins which can also be used as HSO.4 and HSO.5. The I/O Control Registers (IOC0 and IOC1) are used to determine the functions of these pins. A block diagram of the HSI unit is shown in Figure 27.

HSI Trigger Options

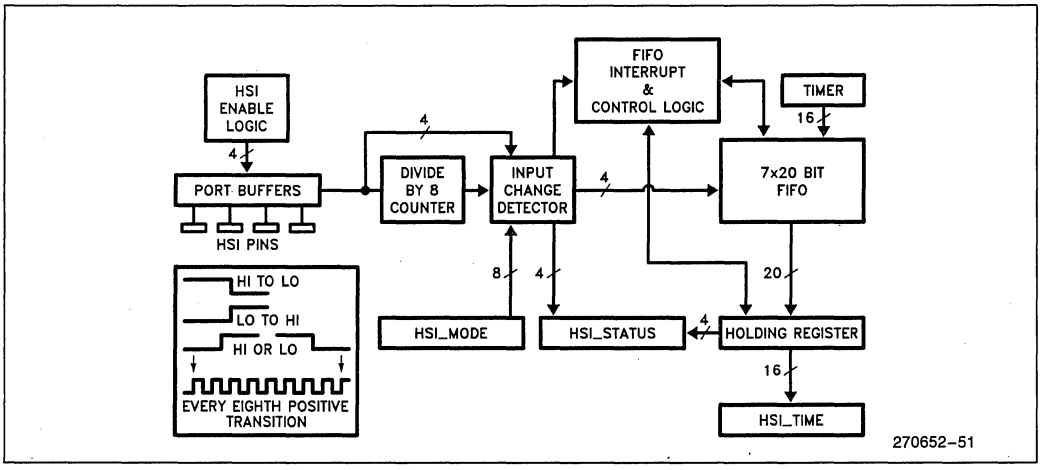


Figure 27. High Speed Input Unit

6.1 HSI Modes

There are 4 possible modes of operation for each of the HSI pins. The HSI mode register is used to control which pins will look for what type of events. The 8-bit register is setup as shown in Figure 28.

High and low levels each need to be held for at least 1 state time to ensure proper operation. The maximum input speed is 1 event every 8 state times except when the 8 transition mode is used, in which case it is 1 transition per state time. The divide by eight counter can only be zeroed in mid-count by performing a hardware reset on the 8098.

Ths HSI lines can be individually enabled and disabled using bits in IOC0, at location 0015H. Figure 29 shows the bit locations which control the HSI pins. If the pin is disabled, transitions will not be entered in the FIFO.

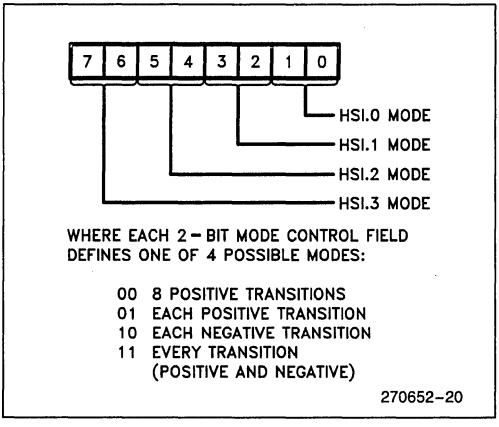


Figure 28. HSI Mode Register Diagram

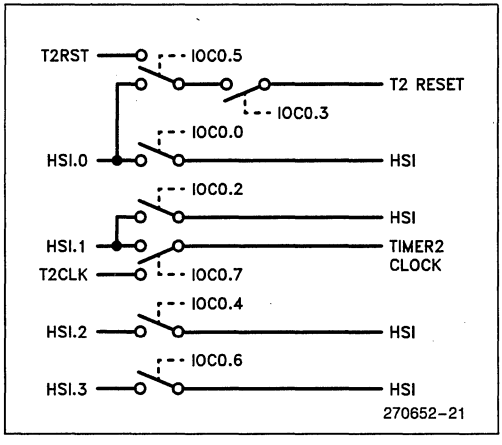


Figure 29. IOC0 Control of HSI Pin Functions

6.2 HSI FIFO

When an HSI event occurs, a 7 × 20 bit FIFO stores the 16 bits of Timer 1 and the 4 bits indicating which pins had events. In can take up to 8 state times for this information to reach the holding register. For this reason, 8 state times must be allowed between consecutive reads of HSI_TIME. When the FIFO is full, one additional event, for a total of 8 events, can be stored by considering the holding register part of the FIFO. If the FIFO and holding register are full, any additional events will not be recorded.

6.3 HSI Interrupts

Interrupts can be generated by the HSI unit in three ways; two FIFO related interrupts and 0 to 1 transitions on the HSI.0 pin. The HSI.0 pin can generate interrupts even if it is not enabled to the HSI FIFO. Interrupts generated by this pin cause a vector through location 2008H. The FIFO related interrupts are controlled by bit 7 of I/O Control Register 1, (IOC1.7). If the bit is a 0, then an interrupt will be generated every time a value is loaded into the holding register. If it is a 1, an interrupt will only be generated when the FIFO, (independent of the holding register), has six entries in it. Since all interrupts are rising edge triggered, if IOC1.7 = 1, the processor will not be re-interrupted until the FIFO first contains 5 or less records, then contains six or more.

6.4 HSI Status

Bits 6 and 7 of the I/O Status register 1 (IOS1) indicate the status of the HSI FIFO. If bit 6 is a 1, the FIFO contains at least six entries. If bit 7 is a 1, the FIFO contains at least 1 entry and the HSI holding register has data available to be read. The FIFO may be read after verifying that it contains valid data. Caution must be used when reading or testing bits in IOS1, as this action clears bits 0–5, including the software and hardware timer overflow flags. It is best to store the byte and then test the stored value. See Section 11.

Reading the HSI is done in two steps. First, the HSI Status register is read to obtain the current state of the HSI pins and which pins had changed at the recorded time. The format of the HSI_STATUS Register is shown in Figure 30. Second, the HSI Time register is read. Reading the Time register unloads one level of the FIFO, so if the Time register is read before the Status register, the event information in the Status register will be lost. The HSI Status register is at location 06H and the HSI Time registers are in locations 04H and 05H.

If the HSI_Time register is read without the holding register being loaded, the returned value will be indeterminate. Under the same conditions, the four bits in HSI_STATUS indicating which events have occurred will also be indeterminate. The four HSI_STATUS bit which indicate the current state of the pins will always return the correct value.

It should be noted that many of the Status register conditions are changed by a reset, see Section 13. A complete listing of the functions of IOS0, IOS1 and IOC1 can be found in Section 11.

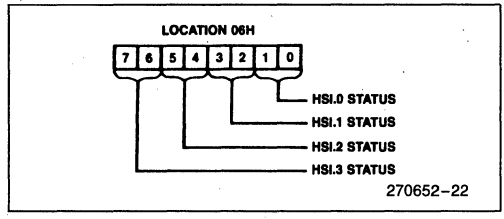


Figure 30. HSI Status Register Diagram

7.0 HIGH SPEED OUTPUTS

The High Speed Output unit, (HSO), is used to trigger events at specific times with minimal CPU overhead. These events include: starting an A to D conversion, resetting Timer 2, setting 4 software flags, and switching 6 output lines (HSO.0 through HSO.5). Up to eight events can be pending at one time and interrupts can be generated whenever any of these events are triggered HSO.4 and HSO.5 are bidirectional pins which can also be used as HSI.2 and HSI.3 respectively. Bits 4 and 6 of I/O Control Register 1, (IOC1.4, IOC1.6), enable HSO.4 and HSO.5 as outputs.

The HSO unit can generate two types of interrupts. The HSO execution interrupt (vector = 2006H) is generated (if enabled) for HSO commands which operate one or more of the six output pins. The other HSO interrupt is the software timer interrupt (vector = (200BH) which is generated (if enabled) by any other HSO command, (e.g. triggering the A/D, resetting Timer 2 or generating a software time delay.)

7.1 HSO CAM

A block diagram of the HSO unit is shown in Figure 31. The Content Addressable Memory (CAM) file is the center of control. One CAM register is compared with the timer values every state time, taking 8 state times to compare all CAM registers with the timers. This defines the time resolution of the HSO to be 8 state times (2.0 μs at an oscillator frequency of 12 MHz).

Each CAM register is 23 bits wide. Sixteen bits specify the time at which the action is to be carried out and 7 bits specify both the nature of the action and whether Timer 1 or Timer 2 is the reference.

The format of the command to the HSO unit is shown in Figure 32. Note that bit 5 is ignored for command channels 8 through 0FH.

To enter a command into the CAM file, write the 7-bit "Command Tag" into location 0006H followed by the time at which the action is to be carried out into word address 0004H. The typical code would be:

```
LDB   HSO_COMMAND, #what_to_do
ADD   HSO_TIME, TIMER1, #when_to_do_it
```

Writing the time value loads the HSO Holding Register with both the time and the last written command tag. The command does not actually enter the CAM file until an empty CAM register becomes available.

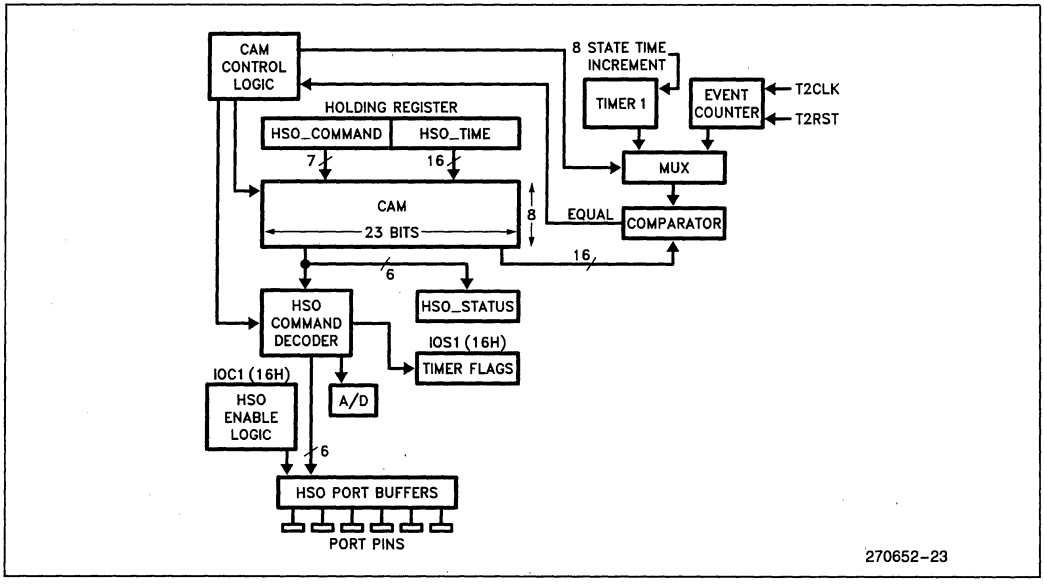


Figure 31. High Speed Output Unit

Commands in the holding register will not execute even if their time tag is reached. Commands must be in the CAM for this to occur. Commands in the holding register can also be overwritten. Since it can take up to 8 state times for a command to move from the holding register to the CAM, 8 states must be allowed between successive writes to the CAM.

To provide proper synchronization, the minimum time that should be loaded to Timer 1 is Timer 1 + 2. Smaller values may cause the Timer match to occur 65,636 counts later than expected. A similar restriction applies if Timer 2 is used.

Care must be taken when writing the command tag for the HSO. If an interrupt occurs during the time between writing the command tag and loading the time value, and the interrupt service routine writes to the HSO time register, the command tag used in the interrupt routine will be written to the CAM at both the time specified by the interrupt routine and the time specified by the main program. The command tag from the main program will not be executed. One way of avoiding this problem would be to disable interrupts when writing commands and times to the HSO unit. See also Section 4.5.

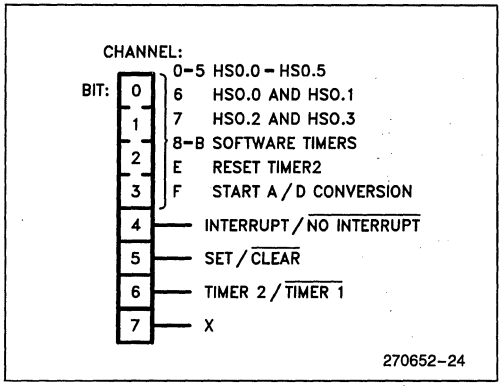


Figure 32. HSO Command Tag Format

7.2 HSO Status

Before writing to the HSO, it is desirable to ensure that the Holding Register is empty. If it is not, writing to the HSO will overwrite the value in the Holding Register. I/O Status Register 0 (IOS0) bits 6 and 7 indicate the status of the HSO unit. This register is described in Section 11. IF ISO0.6 equals 0, the holding register is empty and at least one CAM register is empty. If IOS0.7 equals 0, the holding register is empty.

The programmer should carefully decide which of these two flags is the best to use for each application.

7.3 Clearing the HSO

All 8 CAM locations of the HSO are compared before any action is taken. This allows a pending external event to be cancelled by simply writing the opposite event to the CAM. However, once an entry is placed in the CAM, it cannot be removed until either the specified timer matches the written value or the chip is reset. If, as an example, a command has been issued to set HSO.1 when TIMER 1 = 1234, then entering a second command which clears HSO.1 when TIMER 1 = 1234 will result in no operation on HSO.1. Both commands will remain in the CAM until TIMER 1 = 1234.

Internal events are not synchronized to Timer 1, and therefore cannot be cleared. This includes events on HSO channels 8 through F and all interrupts.

7.4 Using Timers with the HSO

Timer 1 is incremented only once every 8 state-times. When it is being used as the reference timer for an HSO action, the comparator has a chance to look at all 8 CAM registers before Timer 1 changes its value. Following the same reasoning, Timer 2 has been synchronized to allow it to change at a maximum rate of once per 8 state-times. Timer 2 increments on both edges of the input signal.

When using Timer 2 as the HSO reference, caution must be taken that Timer 2 is not reset prior to the highest value for a Timer 2 match in the CAM. This is because the HSO CAM will hold an event pending until a time match occurs, if that match is to a time value on Timer 2 which is never reached, the event will remain pending in the CAM until the part is reset.

Additional caution must be used when Timer 2 is being reset using the HSO unit, since resetting Timer 2 using the HSO is an internal event and can therefore happen at any time within the eight-state-time window. This situation arises when the event is set to occur when Timer 2 is equal to zero. If HSI.0 or the T2RST pin is used to clear Timer 2, and Timer 2 equal to zero triggers the event, then the event may not occur. This is because HSI.0 and T2RST clear Timer 2 asynchronously, and Timer 2 may then be incremented to one before the HSO CAM entry can be read and acted upon. This can be avoided by setting the event to occur when Timer 2 is equal to one. This method will ensure that there is enough time for the CAM entry recognition.

The same asynchronous nature can affect scheduled to occur at the same time as an internal Timer 2 reset. These events should be logged into the CAM with a Timer 2 value of zero. When using this method to make a programmable modulo counter, the count will stay at

the maximum Timer 2 value only until the Reset T2 command is recognized. The count will stay at zero for the transition which would have changed the count from "N" to zero, and then changed to a one on the next transition.

7.5 Software Timers

The HSO can be programmed to generate interrupts at preset times. Up to four such "Software Timers" can be in operation at a time. As each preprogrammed time is reached, the HSO unit sets a Software Timer Flag. If the interrupt bit in the command tag was set then a Software Timer Interrupt will also be generated. The interrupt service routine can then examine I/O Status register 1 (IOS1) to determine which software timer expired and caused the interrupt. When the HSO resets Timer 2 or starts an A to D conversion, it can also be programmed to generate a software timer interrupt but there is no flag to indicate that this has occurred.

Each read or test of any bit in IOS1 will clear bits 0 through 5. Be certain to save the byte before testing it unless you are only concerned with 1 bit. See also Section 11.5.

A complete listing of the functions of IOS0, IOS1 and IOC1 can be found in Section 11. The Timers are described in Section 5 and the HSI is described in Section 6.

8.0 ANALOG INTERFACE

The 8098 can easily interface to analog signals using its Analog to Digital Converter and its Pulse-Width-Modulated (PWM) output and HSO Unit. Analog inputs are accepted by the 8-input, 10-bit A to D converter. The PWM and HSO units provide digital signals which can be filtered for use as analog outputs.

8.1 Analog Inputs

The on-chip analog-to-digital aquisition system is a monotonic successive approximation converter with the sample and hold, multiplexer, and D/A ladder circuits built into the silicon. This system can multiplex up to eight channels of conversion to 10 bits of resolution (1024 unique codes). It has a fixed conversion time of 88 state times which includes the 4 state time sample window. With a 12 MHz clock the conversion would take 22 μs, of which one microsecond was the sample window. The sample window period begins 4 state times after the conversion is triggered. A 2 pF capacitance is charged from the input signal during this sample window period. The D/A is comprised of a 256 resistor ladder which provides ± 4 analog LSB's (± 20 mV) of absolute error, and uses ratioed capacitors to capacitively interpolate the result to 10 digital LSB's (5 mV) of resolution. This result is the ratio of the input voltage and the analog supply voltage (V_{REF}). If the ratio of this comparison is 1.00, then the result will be a 10-bit value will all bits set to logical one. This is particularly advantageous when used with ratiometric sensors which output proportional signals based on the V_{REF}.

In many applications it is less critical to record the absolute accuracy of an input, than it is to resolve that some determinable change has occurred. This is an acceptable approach as long as the converter is guaranteed to be monotonic and has no missing codes, as is the case for the 8098. This means that increasing input voltages produce adjacent and unique output codes that are also increasing. Decreasing input voltages are guaranteed to produce adjacent and unique output codes that are also decreasing. There exists on the 8098 for each 10 bit output code a unique input voltage range that produces that code only, with a repeatability of typically ± 0.25 LSB's (1.5 mV).

The MSC-96 datasheet guarantees that the maximum Differential Non-Linearity will be 2 LSB, or 10 mV (the minimum is zero). Differential non-linearity specifies the **maximum difference** between the actual code widths seen in a converter and what those code widths would be in an ideal (perfect) converter. In the MCS-96 10 bit converter, the code widths are ideally 5 mV (5.12 $V_{REF}/1024$). If such a converter is specified to have a maximum Differential Non-Linearity of 10 mV, then the maximum code width will be no greater than 10 mV larger than ideal, or 15 mV. This indicates to the user how much the input voltage may have changed under worst case conditions to produce a one count change in a particular 10-bit conversion. Due to the fact that the 8098 converter has no missing codes, the minimum code width will always be greater than zero. The differential non-linearity error on a particular code width is compensated for by other code widths in the transfer function such that 1024 unique steps occur. The actual code widths in the 8098 converter typically vary from about 2.5 mV to 7.5 mV.

The analog input must be in the range of zero to V_{REF} (nominally, $V_{REF} = 5V$). This input can be selected from 8 analog inputs which connect to the same pins as PORT 0. A conversion can be initiated either by setting the control bit in the A/D Command Register (Address 02Hex), or by programming the High Speed Output CAM to trigger the conversion at some specified time with sampling intervals occurring accurate to ± 50 ns. (See AP 406 "MCS-96 Analog Aquisition Primer" and the datasheet for further information).

8.2 A/D Commands

Analog signals can be sampled by any one of the 8 analog input pins (ACH0 through ACH7) which are shared with Port 0. ACH7 can also be used as an external interrupt if IOC1.1 is set (see Sections 4 and 11). The A/D Command Register, at location 02H, selects which channel is to be converted and whether the conversion should start immediately or when the HSO triggers it. The A/D command register must be written to for each conversion, even if the HSO is used as the trigger. A to D commands are formatted as shown in Figure 33.

The command register is double buffered so it is possible to write a command to start a conversion triggered by the HSO while one is still in progress. Care must be taken when this is done since if a new conversion is started while one is already in progress, the conversion in progress is cancelled and the new one is started. When a conversion is started, the result register is cleared. For this reason the result register must be read before a new conversion is started or data will be lost.

8.3 A/D Results

Results of the analog conversions are read from the A/D Result Register at locations 02H and 03H. Although these addresses are on a word boundary, they must be read as individual bytes. Information in the A/D Result register is formatted as shown in Figure 34. Note that the status bit may not be set until 8 state times after the go command, so it is necessary to wait 8 state times before testing it. Information on using the HSO is in Section 7.

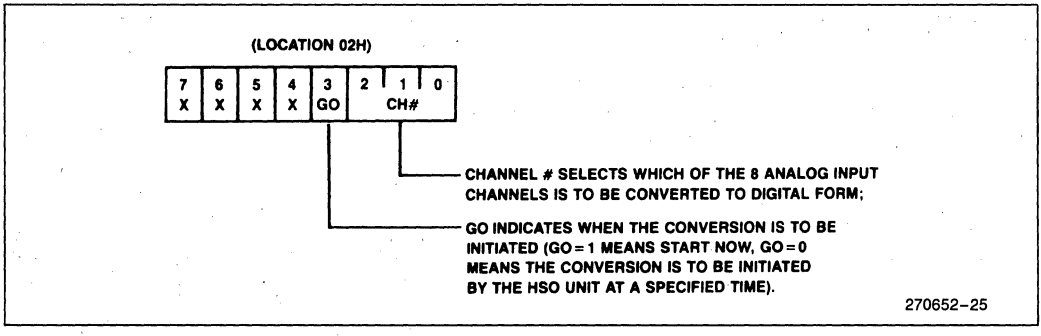


Figure 33. A/D Command Register

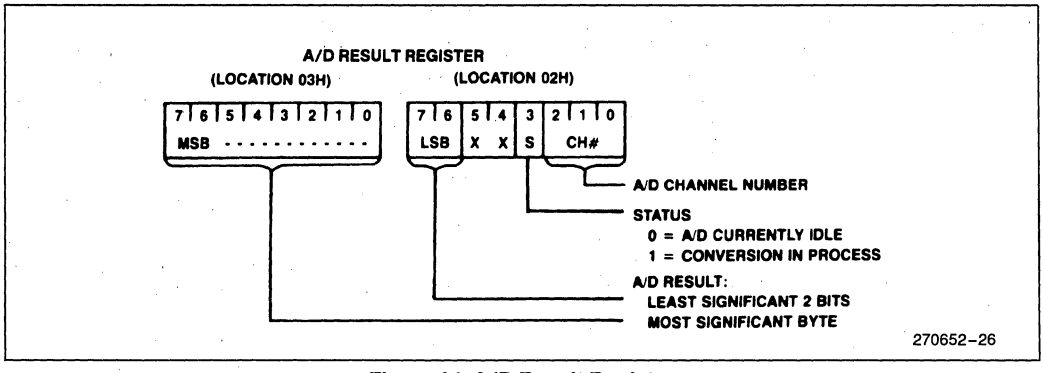


Figure 34. A/D Result Register

8.4 Pulse Width Modulation Output (D/A)

Digital to analog conversion can be done with the Pulse Width Modulation output; a block diagram of the circuit is shown in Figure 35. The 8-bit counter is incremented every state time. When it equals 0, the PWM output is set to a one. When the counter matches the value in the PWM register, the output is switched low. When the counter overflows, the output is once again switched high. A typical output waveform is shown in Figure 36. Note that when the PWM register equals 00, the output is always low. Additionally, the PWM register will only be reloaded from the temporary latch when the counter overflows. This means that the compare circuit will not recognize a new value to compare against until the counter has expired the remainder of the current 8-bit count.

The output waveform is a variable duty cycle pulse which repeats every 256 state times (64 μs at 12 MHz). Changes in the duty cycle are made by writing to the PWM register at location 17H. There are several types of motors which require a PWM waveform for most efficient operation. Additionally, if this waveform is integrated it will produce a DC level which can be changed in 256 steps by varying the duty cycle.

The PWM output shares a pin with Port 2, pin 5 so that these two features cannot be used at the same time. IOC1.0 equal to 1 selects the PWM function instead of the standard port function. More information on IOC1 is in Section 11.

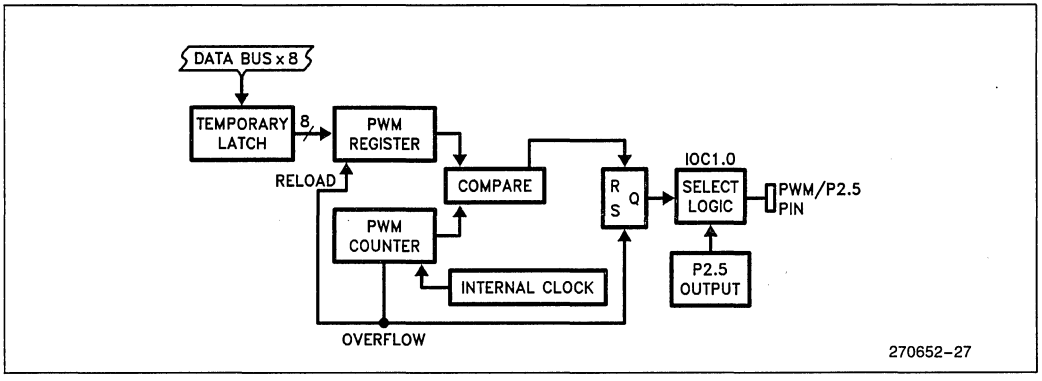


Figure 35. Pulse Width Modulated (D/A) Output

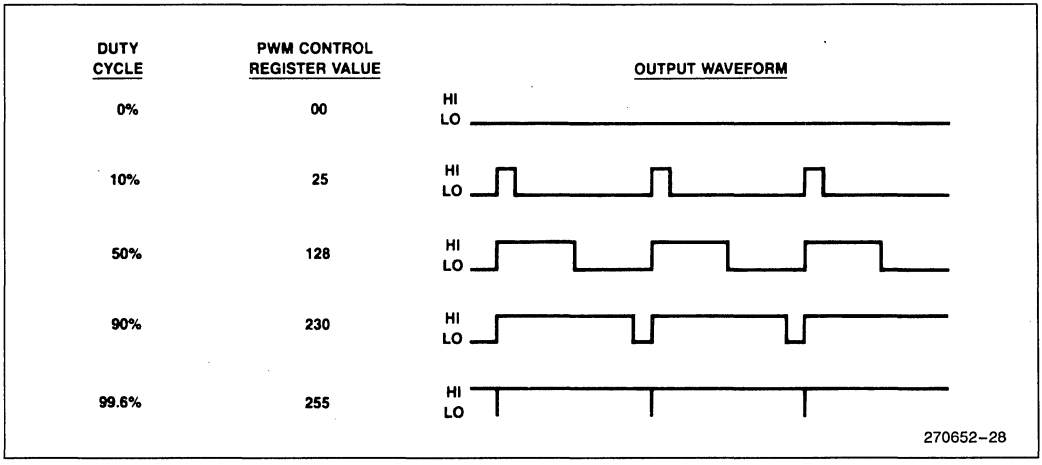


Figure 36. Typical PWM Outputs

8.5 PWM Using the HSO

The HSO unit can be used to generate PWM waveforms with very little CPU overhead. If the HSO is not being used for other purposes, a 4 line PWM unit can be made by loading the on and off times into the CAM in sets of 4. The CAM would then always be loaded and only 2 interrupts per PWM period would be needed. An example of using the HSO in their manner can be found AP-248, "Using the 8096".

9.0 SERIAL PORT

The serial port on the 8098 has 3 asynchronous and one synchronous mode. The asynchronous modes are full duplex, meaning they can transmit and receive at the same time. The receiver is double buffered so that the reception of a second byte can begin before the first byte has been read. The port is functionally compatible with the serial port on the MCS-51 family of microcontrollers, although the software used to control the ports is different.

Control of the serial port is handled through the Serial Port Control/Status Register at location 11 Hex. Figure 40 shows the layout of this register.

Data to and from the serial port is transferred through SBUF (rx) and SBUF (tx), both located at 07H. Although these registers share the same address, they are physically separate, with SBUF (rx) containing the data received by the serial port and SBUF (tx) used to hold data ready for transmission. The program cannot write to SBUF (rx) or read from SBUF (tx).

The baud rate at which the serial port operates is controlled by an independent baud rate generator. The input to this generator is from the XTAL1 pin. Details on setting up the baud rate are given in Section 9.3.

9.1 Serial Port Modes

MODE 0

Mode 0 is a synchronous mode which is commonly used for shift register based I/O expansion. In this mode the TXD pin outputs a set of 8 pulses while the RXD pin either transmits or receives data. Data is transferred 8 bits at a time with the LSB first. A diagram of the relative timing of these signals is shown in Figure 37. Note that this is the only mode which uses RXD as an output.

Although it is not possible to transmit and receive at the same time using this mode, two external gates and a port pin can be used to time-multiplex the two functions.

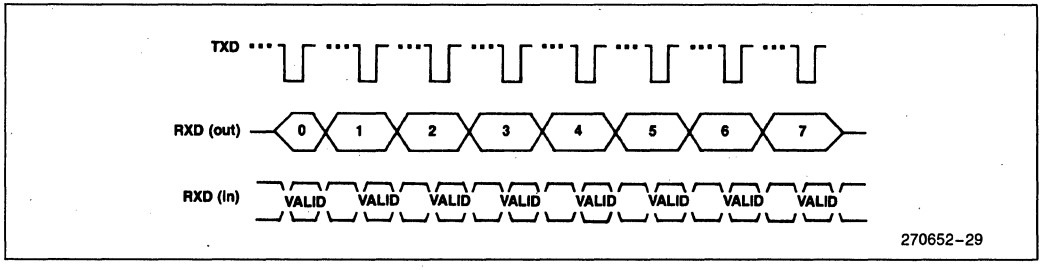


Figure 37. Serial Port Mode 0 Timing

MODE 1

Mode 1 is the standard asynchronous communications mode. The data frame used in this mode is shown in Figure 38. It consists of 10 bits; a start bit (0), 8 data bits (LSB first), and a stop bit (1). If parity is enabled, (the PEN bit is set to a 1), an even parity bit is sent instead of the 8th data bit and parity is checked on reception.

MODE 2

Mode 2 is the asynchronous 9th bit recognition mode. This mode is commonly used with Mode 3 for multiprocessor communications. Figure 39 shows the data frame used in this mode. It consists of a start bit (0), 9 data bits (LSB first), and a stop bit (1). When transmitting, the 9th bit can be set to a one by setting the TB8 bit in the control register before writing to SBUF (tx). The TB8 bit is cleared on every transmission, so it must be set prior to writing to SBUF (tx) each time it is desired. During reception, the serial port interrupt and the Receive Interrupt (RI) bit will not be set unless the 9th bit being received is set. This provides an easy way to have selective reception on a data link. Parity cannot be enabled in this mode.

MODE 3

Mode 3 is the asynchronous 9th bit mode. The data frame for this mode is identical to that of Mode 2. The transmission differences between Mode 3 and Mode 2 are that parity can be enabled (PEN = 1) and cause the 9th data bit to take the even parity value. The TB8 bit can still be used if parity is not enabled (PEN = 0). When in Mode 3, a reception always causes an interrupt regardless of the state of the 9th bit. The 9th bit is stored if PEN = 0 and can be read in bit RB8. If PEN = 1 then RB8 becomes the Receive Parity Error (RPE) flag.

9.2 Controlling the Serial Port

Control of the serial port is done through the Serial Port Control (SP_CON) and Serial Port Status (SP_STAT) registers shown in Figure 40. Writing to location 11H accesses SP_CON while reading it access SP_STAT. Note that reads of SP_STAT will return indeterminate data in the lower 5 bits and writing to the upper 3 bits of SP_CON has no effect on chip functionality. The TB8 bit is cleared after each transmission and both TI and RI are cleared whenever SP_STAT (not SP_CON) is accessed. Whenever the TXD pin is used for the serial port it must be enabled by setting IOC1.5 to a 1. IOC1 is discussed further in Section 11.3.

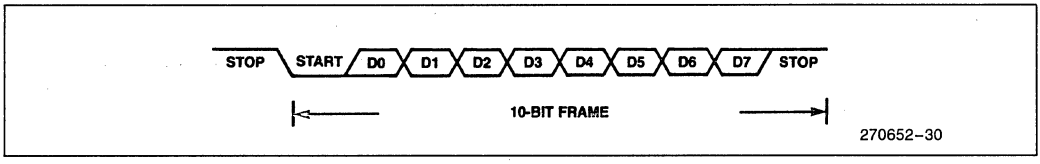


Figure 38. Serial Port Frame_Mode 1

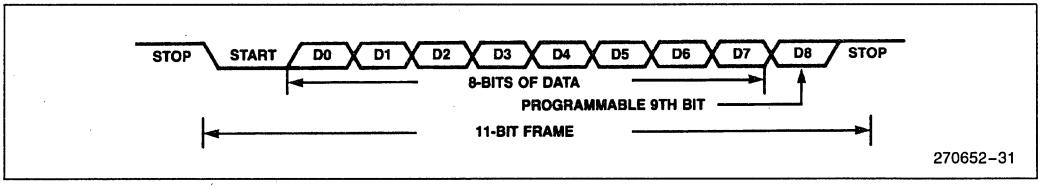


Figure 39. Serial Port Frame Modes 2 and 3

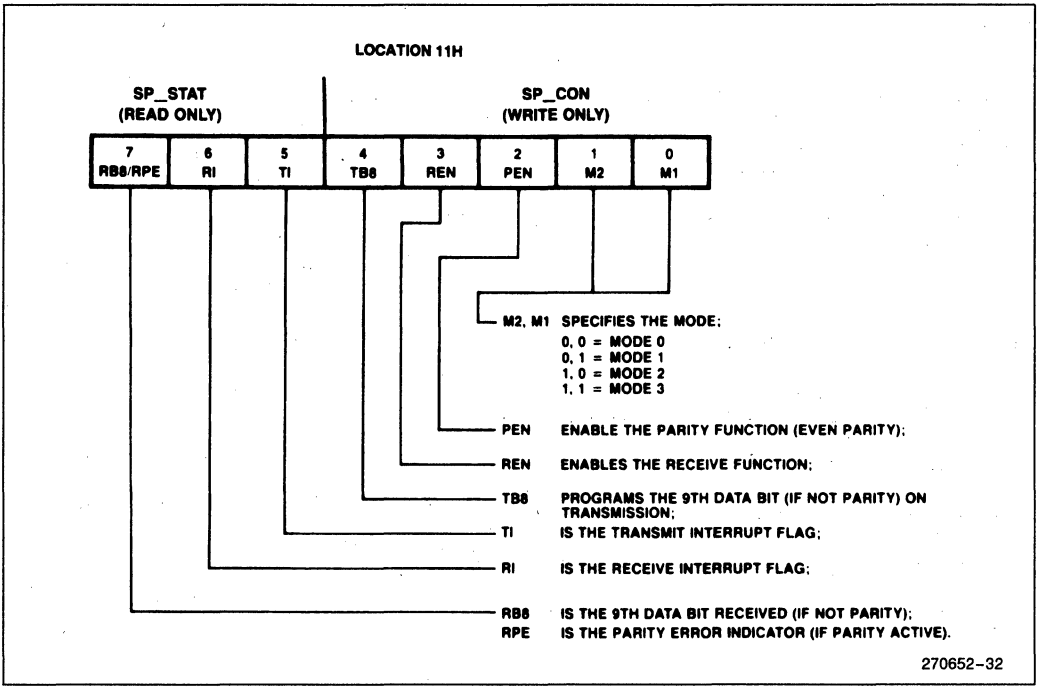


Figure 40. Serial Port Control/Status Register

In Mode 0, if REN = 1, writing to SBUF (tx) will start a transmission. Causing a rising edge on REN, or clearing RI with REN = 1, will start a reception. Setting REN = 0 will stop a reception in progress and inhibit further receptions. To avoid a partial or complete undersired reception, REN must be set to zero before RI is cleared. This can be handled in an interrupt environment by using software flags or in straight-line code by using the Interrupt Pending register to signal the completion of a reception.

In the asynchronous modes, writing to SBUF (tx) starts a transmission. A falling edge on RXD will begin a reception if REN is set to 1. New data placed in SBUF (tx) is held and will not be transmitted until the end of the stop bit has been sent.

In all modes, the RI flag is set after the last data bit is sampled approximately in the middle of the bit time. Also for all modes, the TI flag is set after the last data bit (either 8th or 9th) is sent, also in the middle of the bit time. The flags clear when SP_STAT is read, but do not have to be clear for the port to receive or transmit. The serial port interrupt bit is set as a logical OR of the RI and TI bits. Note that changing modes will reset the Serial Port and abort any transmission or reception in progress on the channel.

9.3 Determining Baud Rates

Baud rates in all modes are determined by the contents of a 16-bit register at location 000EH. This register must be loaded sequentially with 2 bytes (least significant byte first). The serial port will not function between the loading of the first and second bytes. The MSB of this register equal to a logic one selects the source (XTAL 1) for the input frequency to the baud rate generator.

The unsigned integer represented by the lower 15 bits of the baud rate register defines a number B, where B has a maximum value of 32767. The baud rate for the four serial modes using XTAL1 is given by Figure 41.

Using XTAL1:

$$\text{Mode 0: } \frac{\text{Baud}}{\text{Rate}} = \frac{\text{XTAL1 Frequency}}{4*(B+1)};\ B \neq 0$$

$$\text{Others: } \frac{\text{Baud}}{\text{Rate}} = \frac{\text{XTAL1 Frequency}}{64*(B+1)}$$

Note that B cannot equal 0, except when using XTAL1 in other than Mode 0.

Figure 41. Baud Rate Calculations

Common baud rate values,using XTAL1 at 12 MHz, are shown below.

Baud Rate	Baud Register Value	
	Mode 0	Others
9600	8137H	8013H
4800	8270H	8026H
2400	84E1H	804DH
1200	89C3H	809BH
300	A70FH	8270H

Figure 42. Common Baud Rates

The maximum baud rates are 1.5 Mbaud synchronous and 187.5 Kbaud asynchronous with 12 MHz on XTAL1.

9.4 Multiprocessor Communications

Mode 2 and 3 are provided for multiprocessor communications. In Mode 2 if the received 9th data bit is not 1, the serial port interrupt is not activated. The way to use this feature in multiprocessor systems is described below.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address frame which identifies the target slave. An address frame wil differ from a data frame in that the 9th data bit is 1 in an address frame and 0 in a data frame. No salve in Mode 2 will be interrupted by a data frame. An address frame, however, will interrupt all slaves so that each slave can examine the received byte and see if it is being addressed. The addressed slave switches to Mode 3 to receive the coming data frames, while the slaves that were not addressed stay in Mode 2 and go on about their business.

10.0 I/O PORTS

There are 32 I/O port pins on the 8098. Some of these ports are input only, some are output only, some are bidirectional and some have alternate functions. In addition to these ports, the HSI/O unit can be used to provide extra I/O lines if the timer related features of these lines are not needed.

Input ports connect to the internal bus through an input buffer. Output ports connect through an output buffer to an internal register that hold the bits to be output. Bidirectional ports consist of an internal register, an input buffer and an output buffer.

Port 0 is an input port which is also used as the analog input for the A to D converter. Port 2 contains input and output. The input and output lines are shared with other functions in the 8098 as shown in Figure 43. Ports 3 and 4 are open-drain bidirectional ports which share their pins with the address/data bus.

Port	Function	Alternate Function
P2.0	Output	TXD (Serial Port Transmit)
P2.1	Input	RXD (Serial Port Receive)
P2.2	Input	EXTINT (External Interrupt)
P2.5	Output	PWM (Pulse Width Modulation)

Figure 43. Port 2 Alternate Functions

11.0 STATUS AND CONTROL REGISTERS

There are two I/O Control registers, IOC0 and IOC1. IOC0 controls Timer 2 and the HSI lines. IOC1 controls some pin functions, interrupt sources and 2 HSO pins.

Whenever input lines are switched between two sources, or enabled, it is possible to generate transitions on these lines. This could cause problems with respect to edge sensitive lines such as the HSI lines, Interrupt line and Timer 2 control lines.

11.1 I/O Control Register 0 (IOC0)

IOC0 is located at 0015H. The four HSI lines can be enabled or disabled to the HSI unit by setting or clearing bits in IOC0. Timer 2 functions including clock and reset sources are also determined by IOC0. The control bit locations are shown in Figure 44.

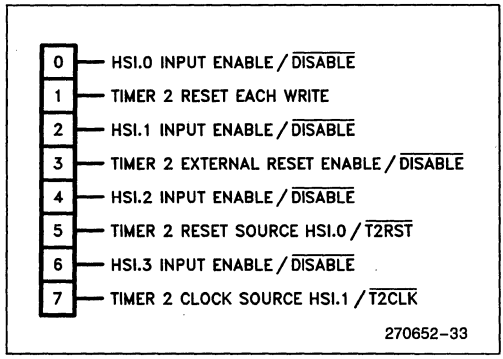


Figure 44. I/O Control Register 0 (IOC0)

11.2 I/O Control Register 1 (IOC1)

IOC1 is used to select some pin functions and enable or disable some interrupt sources. Its location is 0016H. Port pin P2.5 can be selected to be the PWM output instead of a standard output. The external interrupt source can be selected to be either EXTINT (same pin as P2.2) or Analog Channel 7 (ACH7, same pin as P0.7). Timer 1 and Timer 2 overflow interrupts can be individually enabled or disabled. The HSI interrupt can be selected to activate either when there is 1 FIFO entry or 7. Port pin P2.0 can be selected to be the TXD output. HSO.4 and HSO.5 can be enabled or disabled to the HSO unit. More information on interrupts is available in Section 4. The positions of the IOC1 control bits are shown in Figure 45.

11.3 I/O Status Register 0 (IOS0)

There are two I/O Status registers, IOS0 and IOS1. IOS0, located at 0015H, holds the current status of the HSO lines and CAM. The status bits of IOS0 are shown in Figure 46.

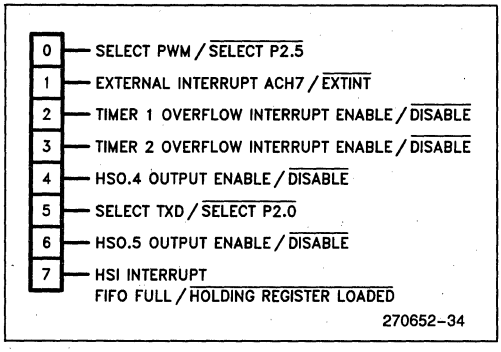


Figure 45. I/O Control Register 1 (IOC1)

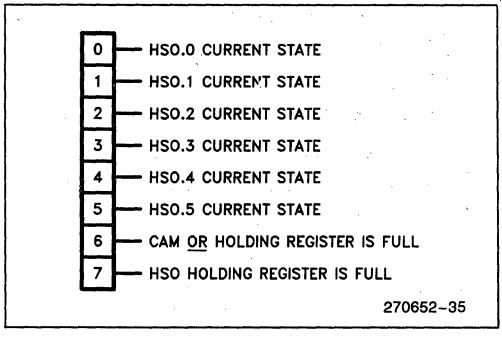


Figure 46. I/O Status Register 0 (IOS0)

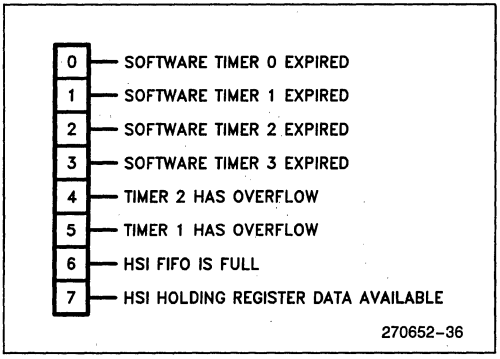


Figure 47. HSIO Status Register 1 (IOS1)

11.4 I/O Status Register 1 (IOS1)

IOS1 is located at 016H. It contains status bits for the timers and the HSI/O. The positions of these bits are shown in Figure 47.

Whenever the processor reads this register all of the time-related flags (bits 5 through 0) are cleared. This applies not only to explicit reads such as:

```
LDB   AL, IOS1
```

but also to implicit reads such as:

```
JB    IOS1.3, somewhere_else
```

which jumps to somewhere else if bit 3 of IOS1 is set. In most cases this situation can best be handled by having a byte in the register file which is used to maintain an image of lower five bits of the register. Any time a hardware timer interrupt or a HSO software timer interrupt occurs the byte can be updated:

```
ORB   IOS1_image, IOS1
```

leaving IOS1 image containing all the flags that were set before plus all the new flags that were read and cleared from IOS1. Any other routine which needs to sample the flags can safely check IOS1 image. Note that if these routines need to clear the flags that they have acted on, then the modification of IOS1 image must be done from inside a critical region (see Section 4.4).

12.0 WATCHDOG TIMER

The WatchDog Timer (WDT) provides a means to recover gracefully from a software upset. When the watchdog is enabled it will initiate a hardware reset unless the software clears it every 64K state times.

The WDT is implemented as an 8-bit timer with an 8-bit prescaler. The prescaler is not synchronized, so the timer will overflow between 65280 and 65535 state times after being reset.

When the timer overflows it pulls down the RESET pin for at least two state times, resetting the 8098 and any other devices tied to the RESET line. If a large capacitor is connected to the line, the pin may take a long time to go low. This will effect the length of time the pin is low and the voltage on the pin when it is finished falling. The datasheet contains more information about reset hardware connections.

The WDT is enabled the first time it is cleared. Once it is enabled, it can only be disabled by resetting the 8098. The internal bit which controls the watchdog can typically maintain its state through power glitches as low as V_{SS} and as high as 7.0V for up to 1 ms.

Enabling and clearing the WDT is done by writing a "01EH" followed by a "0E1H" to the WDT register at location 0AH. This double write is used to help prevent accidental clearing of the timer.

12.1 Software Protection Hints

Glitches and noise on the PC board can cause software upsets, typically by changing either memory locations or the program counter. These changes can be internal to the chip or be caused by bad data returning to the chip.

There are both hardware and software solutions to noise problems, but the best solution is good design practice and a few ounces of prevention. The software can be designed so that the WatchDog times out if the program does not progress properly. The WatchDog will also time-out if the software error was due to ESD (Electrostatic Discharge) or other hardware related problems. This prevents the controller from having a malfunction for longer than 16 ms if a 12 MHz oscillator is used.

When using the WDT to protect software it is desirable to reset if from only one place in code. This will lessen the chance that an undesired WDT reset will occur. The section of code that resets the WDT should monitor the other code sections for proper operation. This can be done by checking variables to make sure they are within reasonable values. Simply using a software timer to reset the WDT every 15 ms will not provide much protection against minor problems.

It is also recommended that unused areas of code be filled with NOPs and periodic jumps to an error routine or RST (reset chip) instructions. This is particularly important in the code around lookup tables, since if lookup tables are executed undesired results will occur. Wherever space allows, each table should be surrounded by 7 NOPs (the longest 8098 instruction has 7 bytes) and a RST or jump to error routine instruction. Since RST is a one-byte instruction, the NOPs are not needed if RSTs are used instead of jumps to an error routine. This will help to ensure a speedy recovery should the processor have a glitch in the program flow. Since RST instruction has an opcode of 0FFH, pulling the data lines high with resistors will cause an RST to be executed if unimplemented memory is addressed.

12.2 Disabling the WatchDog

The WatchDog should be disabled by software not initializing it. If this is not possible, such as during program development, the WatchDog can be disabled by holding the RESET pin at 2.0V to 2.5V. Voltages over 2.5V on the pin could quickly damage the part. Even at 2.5V, using this technique for other than debugging purposes is not recommended, as it may effect long term reliability. It is further recommended that any part used in this way for more than several seconds, not be used in production versions of products. The datasheet has more information on disabling the WatchDog Timer.

13.0 RESET

13.1 Reset Signal

As with all processors, the 8098 must be reset each time the power is turned on. This is done by holding the RESET pin low for at least 2 state times after the power supply is within tolerance and the oscillator has stabilized.

After the RESET pin is brought high, a ten state reset sequence is executed. During this time, the Chip Configuration Byte (CCB) is read from location 2018H and written to the 8098 Chip Configuration Register (CCR). If the voltage on the EA pin selects the internal/external execution mode the CCB is read from internal ROM/EPROM. If the voltage on the EA pin selects the external execution only mode the CCB is read from external memory.

The 8098 can be reset using a capacitor, 1-shot, or any other method capable of providing a pulse of at least 2 state times longer than required for V_{CC} and the oscillator to stabilize.

For best functionality, it is suggested that the reset pin be pulled low with an open collector device. In this way, several reset sources can be wired ORed together. Remember, the RESET pin itself can be a reset source when the RST instruction is executed or when the WatchDog Timer overflows. Details of hardware suggestions for reset can be found in the datasheet.

13.2 Reset Status

The I/O lines and control lines of the 8098 will be in their reset state within 2 state times after reset is low, with V_{CC} and the oscillator stabilized. Prior to that time, the status of the I/O lines is indeterminate. After the 10 state time reset sequence, the Special Function Registers will be set as follows:

Register	Reset Value
Port 2	XX0XXXX1B
Port 3	11111111B
Port 4	11111111B
PWM Control	00H
Serial Port (Transmit)	Undefined
Serial Port (Receive)	Undefined
Baud Rate Register	Undefined
Serial Port Control	XXXX0XXXB
Serial Port Status	X00XXXXXB
A/D Command	Undefined
A/D Result	Undefined
Interrupt Pending	Undefined
Interrupt Mask	00000000B
Timer 1	0000H
Timer 2	0000H
WatchDog Timer	0000H
HSI Mode	11111111B
HSI Status	Undefined
IOS0	00000000B
IOS1	00000000B
IOC0	X0X0X0X0B
IOC1	X0X0XXX1B
HSI FIFO	Empty
HSO CAM	Empty
HSO SFR	000000B
PSW	0000H
Stack Pointer	Undefined
Program Counter	2080H

Figure 48. Register Reset Status

Other conditions following a reset are:

Pin	Reset Value
$\overline{RD}$	High
$\overline{WR}/\overline{WRL}$	High
ALE/$\overline{ADV}$	High
INST	High

Figure 49. Bus Control Pins Reset Status

It is important to note that the Stack Pointer and Interrupt Pending Register are undefined, and need to be initialized in software. The Interrupts are disabled by both the mask register and PSW.9 after a reset.

13.3 Reset Sync Mode

The RESET line can be used to start the 8098 at an exact state time to provide for synchronization of test equipment and multiple chip systems. RESET is active low. To synchronize parts, RESET is brought high on the rising edge of XTAL1.

It is very possible that parts which start in sync may not stay that way. The best example of this would be when a "jump on I/O bit" is being used to hold the processor in a loop. If the line changes during the time it is being tested, one processor may see it as a one, while the other sees its as a zero. The result is that one processor will do an extra loop, thus putting it several states out of sync with the other.

14.0 QUICK REFERENCE

14.1 Pin Description

On the 8098/8398 the following pins of the 8096BH are not bonded out: Port1, Port0 (Analog In) bits 0–3, T2CLK (P2.3), T2RST (P2.4), P2.6, P2.7, CLKOUT, INST, NMI, BUSWIDTH, $\overline{BHE}/\overline{WRH}$.

PIN DESCRIPTIONS

Symbol	Name and Function
V_{CC}	Main supply voltage (5V).
V_{SS}	**Digital Circuit Ground (0V).** There are two V_{SS} pins, both of which must be connected.
V_{PD}	**RAM Standby Supply Voltage (5V).** This voltage must be present during normal operation. In a Power Down condition (i.e. V_{CC} drops to zero), if $\overline{RESET}$ is activated before V_{CC} drops below spec and V_{PD} continues to be held within spec., the top 16 bytes in the Register File will retain their contents. $\overline{RESET}$ must be held low during the Power Down and should not be brought high until V_{CC} is within spec and the oscillator has stabilized.
V_{REF}	**Reference Voltage for the A/D Converter (5V).** V_{REF} is also the supply voltage to the analog portion of the A/D converter and the logic used to read Port 0. Must be connected for A/D and Port 0 to function.
ANGND	**Reference Gound for the A/D Converter.** Must be held at nominally the same potential as V_{SS}.
V_{PP}	Programming voltage for the future EPROM parts.
XTAL1	Input of the oscillator inverter and of the internal clock generator.
XTAL2	Output of the oscillator inverter.
$\overline{RESET}$	**Reset Input to the Chip.** Input low for at least 2 state times to reset the chip. The subsequent low-to-high transition re-synchronizes CLKOUT and commences a 10-state-time sequence in which the PSW is cleared, a byte read from 2018H loads CCR, and a jump to location 2080H is executed. Input high for normal operation. $\overline{RESET}$ has an internal pullup.
$\overline{EA}$	**Input for Memory Select (External Access).** $\overline{EA}$ equal to a TTL-high causes memory accesses to locations 2000H through 3FFFH to be directed to on-chip ROM. $\overline{EA}$ equal to a TTL-low causes accesses to these locations to be directed to off-chip memory. $\overline{EA}$ has an internal pulldown, so it goes to 0 unless driven otherwise. $\overline{EA}$ is latched at reset.
ALE/$\overline{ADV}$	**Address Latch Enable or Address Valid Output**, as selected by CCR. Both pin options provide a latch to demultiplex the address from the address/data bus. When the pin is $\overline{ADV}$, it goes inactive high at the end of the bus cycle. $\overline{ADV}$ can be used as a chip select for external memory. ALE/$\overline{ADV}$ is activated only during external memory accesses.
$\overline{RD}$	**Read Signal Output to External Memory.** $\overline{RD}$ is activated only during external memory reads.
$\overline{WR}$	**Write Output to External Memory.** $\overline{WR}$ is activated only during external memory writes.
READY	**Ready Input to Lengthen External Memory Cycles**, for interfacing to slow or dynamic memory, or for bus sharing. The bus cycle can be lengthened by up to 1 μs. When the external memory is not being used, READY has no effect. Internal control of the number of wait states inserted into a bus cycle held not ready is available through configuration of CCR. READY has a weak internal pullup, so it goes to 1 unless externally pulled low.
HSI	**Inputs to High Speed Input Unit.** Four HSI pins are available: HSI.0, HSI.1, HSI.2 and HSI.3. Two of them (HSI.2 and HSI.3) are shared with the HSO Unit.
HSO	**Outputs from High Speed Output Unit.** Six HSO pins are available: HSO.0, HSO.1, HSO.2, HSO.3, HSO.4 and HSO.5. Two of them (HSO.4 and HSO.5) are shared with the HSI Unit.
Port 0	**4-Bit High Impedance Input-Only Port.** These pins can be used as digital inputs and/or as analog inputs to the on-chip A/D converter.
Port 2	**4-Bit Multi-Functional Port.** Its pins are shared with other functions in the 8098.
Ports 3 and 4	8-Bit Bi-Directional I/O Ports with Open Drain Outputs. These pins are shared with the multiplexed address/data bus which has strong internal pullups.

14.3 Packaging

The 8098 products are available in 48-pin packages, with and without on-chip ROM. The 8795BH EPROM is to be used as the prototype vechicle (see the MCS-96 datasheets for additional information on the 8795BH). Section 14.4 shows the pinouts for the 48-pin packages. The 48-pin version is offered in a Dual-In-Line Package.

Transistor Count

Device Type	#MOS Gates
839X/879X	120,000
809X	50,000

MTBF Calculations*

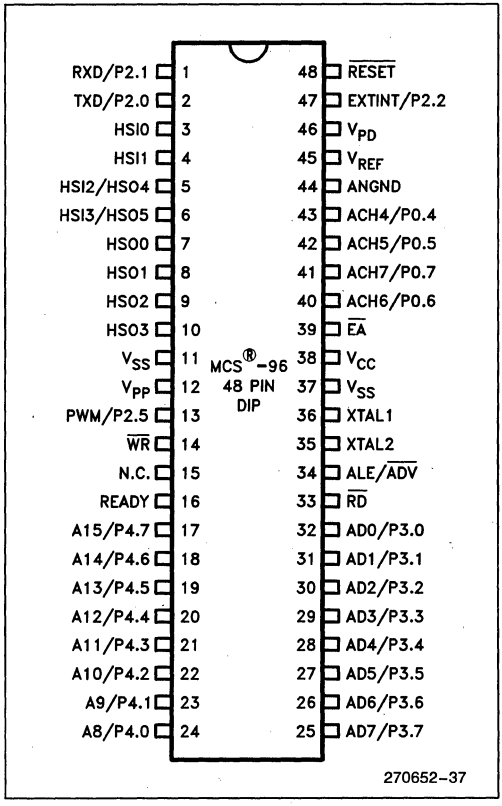
3.8×10^7 Device Hours @ 55°C
1.7×10^7 Device Hours @ 70°C

*MTBF data was obtained through calculations based upon the actual average junction temperatures under stress at 55°C and 70°C ambient.

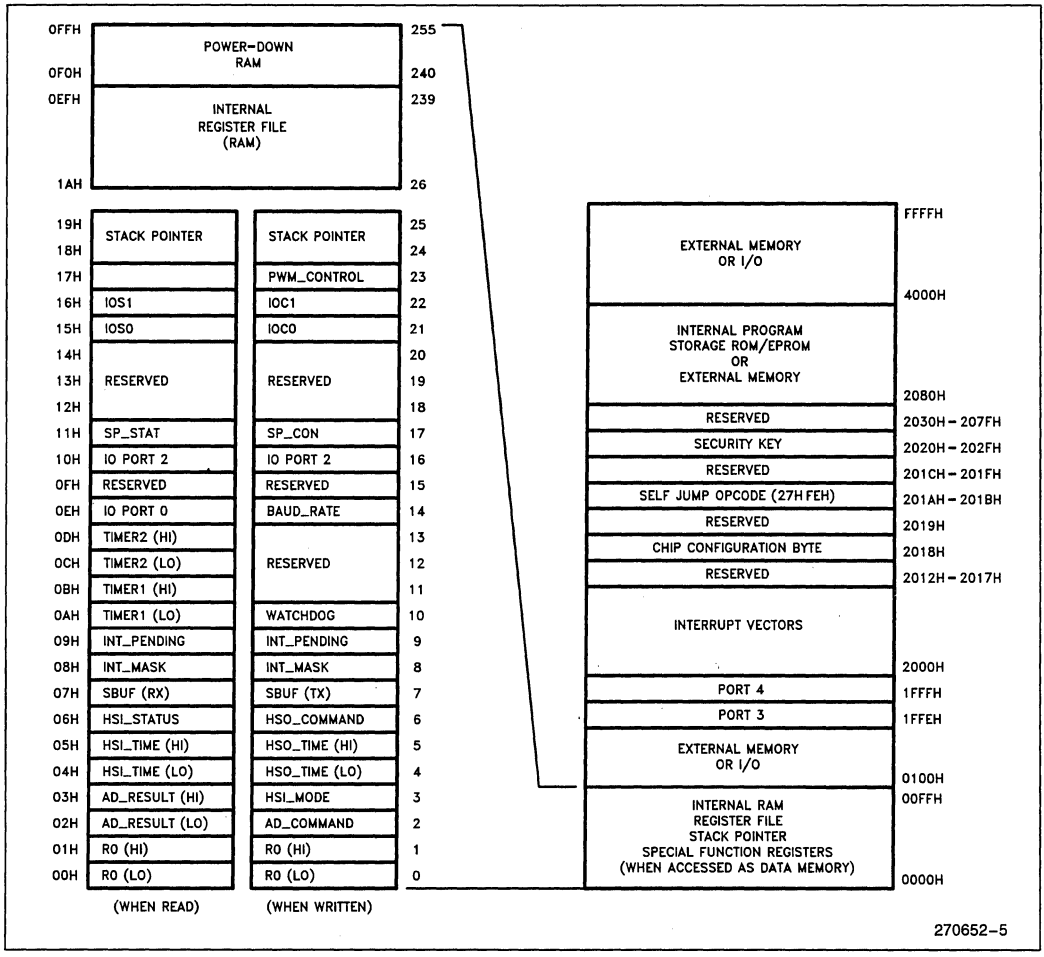
Thermal Characteristics

T_{CASE}		Package Type	θ_{JA}	θ_{JC}
COMM'L	EXPRESS			
		Plastic DIP	38°C/W	
79.75°C	94.75°C	Ceramic DIP	26°C/W	6.5°C/W

14.4 Package



14.5 Memory Map



14.7 Instruction Summary

Mnemonic	Operands	Operation (Note 1)	Flags						Notes
			Z	N	C	V	VT	ST	
ADD/ADDB	2	D ← D + A	↕	↕	↕	↕	↑	—	
ADD/ADDB	3	D ← B + A	↕	↕	↕	↕	↑	—	
ADDC/ADDCB	2	D ← D + A + C	↓	↕	↕	↕	↑	—	
SUB/SUBB	2	D ← D − A	↕	↕	↕	↕	↑	—	
SUB/SUBB	3	D ← B − A	↕	↕	↕	↕	↑	—	
SUBC/SUBCB	2	D ← D − A + C − 1	↓	↕	↕	↕	↑	—	
CMP/CMPB	2	D − A	↕	↕	↕	↕	↑	—	
MUL/MULU	2	D, D + 2 ← D * A	—	—	—	—	—	?	2
MUL/MULU	3	D, D + 2 ← B * A	—	—	—	—	—	?	2
MULB/MULUB	2	D, D + 1 ← D * A	—	—	—	—	—	?	3
MULB/MULUB	3	D, D + 1 ← B * A	—	—	—	—	—	?	3
DIVU	2	D ← (D, D + 2)/A, D + 2 ← remainder	—	—	—	↕	↑	—	2
DIVUB	2	D ← (D, D + 1)/A, D + 1 ← remainder	—	—	—	↕	↑	—	3
DIV	2	D ← (D, D + 2)/A, D + 2 ← remainder	—	—	—	?	↑	—	
DIVB	2	D ← (D, D + 1)/A, D + 1 ← remainder	—	—	—	?	↑	—	
AND/ANDB	2	D ← D and A	↕	↕	0	0	—	—	
AND/ANDB	3	D ← B and A	↕	↕	0	0	—	—	
OR/ORB	2	D ← D or A	↕	↕	0	0	—	—	
XOR/XORB	2	D ← D (excl. or) A	↕	↕	0	0	—	—	
LD/LDB	2	D ← A	—	—	—	—	—	—	
ST/STB	2	A ← D	—	—	—	—	—	—	
LDBSE	2	D ← A; D + 1 ← SIGN(A)	—	—	—	—	—	—	3, 4
LDBZE	2	D ← A; D + 1 ← 0	—	—	—	—	—	—	3, 4
PUSH	1	SP ← SP − 2; (SP) ← A	—	—	—	—	—	—	
POP	1	A ← (SP); SP ← SP2	—	—	—	—	—	—	
PUSHF	0	SP ← SP − 2; (SP) ← PSW; PSW ← 0000H I ← 0	0	0	0	0	0	0	
POPF	0	PSW ← (SP); SP ← SP + 2; I ← ↕	↕	↕	↕	↕	↕	↕	
SJMP	1	PC ← PC + 11-bit offset	—	—	—	—	—	—	5
LJMP	1	PC ← PC + 16-bit offset	—	—	—	—	—	—	5
BR [indirect]	1	PC ← (A)	—	—	—	—	—	—	
SCALL	1	SP ← SP − 2; (SP) ← PC; PC ← PC + 11-bit offset	—	—	—	—	—	—	5
LCALL	1	SP ← SP − 2; (SP) ← PC; PC ← PC + 16-bit offset	—	—	—	—	—	—	5
RET	0	PC ← (SP); SP ← SP + 2	—	—	—	—	—	—	
J (conditional)	1	PC ← PC + 8-bit offset (if taken)	—	—	—	—	—	—	5
JC	1	Jump if C = 1	—	—	—	—	—	—	5
JNC	1	Junp if C = 0	—	—	—	—	—	—	5
JE	1	Jump if Z = 1	—	—	—	—	—	—	5

NOTES:
1. If the nmemonic ends in "B", a byte operation is performed, otherwise a word operation is done. Operands D, B, and A must confrom to the alignment rules for the required operand type. D and B are locations in the Register File; a can be located anywhere in memory.
2. D, D + 2 are consecutive WORDS in memory; D is DOUBLE-WORD aligned.
3. D, D + 1 are consecutive BYTES in memory; D is WORD aligned.
4. Changes a byte to a word.
5. Offset is a 2's complement number.

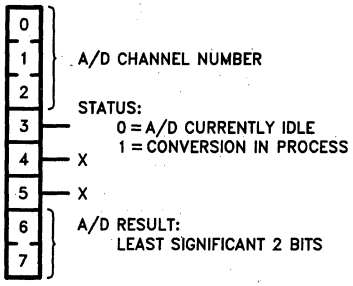
14.7 Instruction Summary (Continued)

Mnemonic	Oper-ands	Operation (Note 1)	Flags						Notes
			Z	N	C	V	VT	ST	
JNE	1	Jump if Z = 0	—	—	—	—	—	—	5
JGE	1	Jump if N = 0	—	—	—	—	—	—	5
JLT	1	Jump if N = 1	—	—	—	—	—	—	5
JGT	1	Jump if N = 0 and Z = 0	—	—	—	—	—	—	5
JLE	1	Jump if N = 1 or Z = 1	—	—	—	—	—	—	5
JH	1	Jump if C = 1 and Z = 0	—	—	—	—	—	—	5
JNH	1	Jump if C = 0 or Z = 1	—	—	—	—	—	—	5
JV	1	Jump if V = 1	—	—	—	—	—	—	5
JNV	1	Jump if V = 0	—	—	—	—	—	—	5
JVT	1	Jump if VT = 1; Clear VT	—	—	—	—	0	—	5
JNVT	1	Jump if VT = 0; Clear VT	—	—	—	—	0	—	5
JST	1	Jump if ST = 1	—	—	—	—	—	—	5
JNST	1	Jump if ST = 0	—	—	—	—	—	—	5
JBS	3	Jump if Specified Bit = 1	—	—	—	—	—	—	5, 6
JBC	3	Jump if Specified Bit = 0	—	—	—	—	—	—	5, 6
DJNZ	1	D ← D − 1; if D ≠ 0 then PC ← PC + 8-bit offset	—	—	—	—	—	—	5
DEC/DECB	1	D ← D − 1	↕	↕	↕	↕	↑	—	
NEG/NEGB	1	D ← 0 − D	↕	↕	↕	↕	↑	—	
INC/INCB	1	D ← D + 1	↕	↕	↕	↕	↑	—	
EXT	1	D ← D; D + 2 ← Sign (D)	↕	↕	0	0	—	—	2
EXTB	1	D ← D; D + 1 ← Sign (D)	↕	↕	0	0	—	—	3
NOT/NOTB	1	D ← Logical Not (D)	↕	↕	0	0	—	—	
CLR/CLRB	1	D ← 0	1	0	0	0	—	—	
SHL/SHLB/SHLL	2	C ← msb_____lsb ← 0	↕	?	↕	↕	↑	—	7
SHR/SHRB/SHRL	2	0 → msb_____lsb → C	↕	?	↕	0	—	↕	7
SHRA/SHRAB/SHRAL	2	msb → msb_____lsb → C	↕	↕	↕	0	—	↕	7
SETC	0	C ← 1	—	—	1	—	—	—	
CLRC	0	C ← 0	—	—	0	—	—	—	
CLRVT	0	VT ← 0	—	—	—	—	0	—	
RST	0	PC ← 2080H	0	0	0	0	0	0	8
DI	0	Disable All Interrupts (I ← 0)	—	—	—	—	—	—	
EI	0	Enable All Interrupts (I ← 1)	—	—	—	—	—	—	
NOP	0	PC ← PC + 1	—	—	—	—	—	—	
SKIP	0	PC ← PC + 2	—	—	—	—	—	—	
NORML	2	Left shift till msb = 1; D ← shift count	↕	?	0	—	—	—	7
TRAP	0	SP ← SP −2; (SP) ← PC PC ← (2010H)	—	—	—	—	—	—	9

NOTES:
1. If the mnemonic ends in "B", a byte operation is performed, otherwise a word operation is done. Operands D, B and A must conform to the alignment rules for the required operand type. D and B are locations in the Register File; A can be located anywhere in memory.
5. Offset is a 2's complement number.
6. Specified bit is one of the 2048 bits in the register file.
7. The "L" (Long) suffix indicates double-word operation.
8. Initiates a Reset by pulling $\overline{\text{RESET}}$ low. Software should re-initialize all the necessary registers with code starting at 2080H.
9. The assembler will not accept this mnemonic.

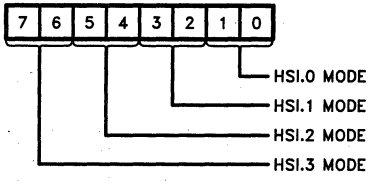
14.8 SFR Summary

A/D Result LO (02H)



270652-39

HSI_Mode (03H)

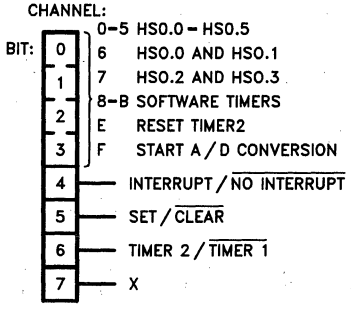


WHERE EACH 2-BIT MODE CONTROL FIELD DEFINES ONE OF 4 POSSIBLE MODES:

00 8 POSITIVE TRANSITIONS
01 EACH POSITIVE TRANSITION
10 EACH NEGATIVE TRANSITION
11 EVERY TRANSITION (POSITIVE AND NEGATIVE)

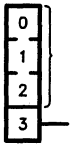
270652-40

HSO Command (06H)



270652-41

A/D Command (02H)

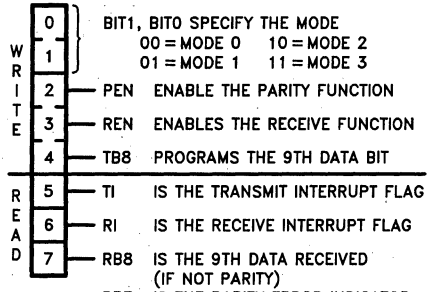


CHANNEL # SELECTS WHICH OF THE 8 ANALOG INPUT CHANNELS IS TO BE CONVERTED TO DIGITAL FORM.

GO INDICATES WHEN THE CONVERSION IS TO BE INITIATED (GO = 1 MEANS START NOW, GO = 0 MEANS THE CONVERSION IS TO BE INITIATED BY THE HSO UNIT AT A SPECIFIED TIME).

270652-42

SPCON/SPSTAT (11H)



270652-43

Baud Rate Calculations

Using XTAL:

$$\text{Mode 0: } \frac{\text{Baud}}{\text{Rate}} = \frac{\text{XTAL1 frequency}}{4^*(B+1)};\ B \neq 0$$

$$\text{Others: } \frac{\text{Baud}}{\text{Rate}} = \frac{\text{XTAL1 Frequency}}{64^*(B+1)}$$

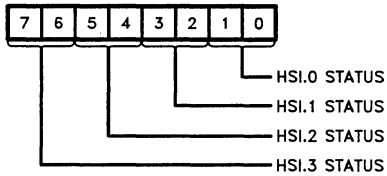
Using T2CLK:

$$\text{Mode 0: } \frac{\text{Baud}}{\text{Rate}} = \frac{\text{T2CLK frequency}}{B};\ B \neq 0$$

$$\text{Others: } \frac{\text{Baud}}{\text{Rate}} = \frac{\text{T2CLK Frequency}}{16^*B};\ B \neq 0$$

Note that B cannot equal 0, except when using XTAL1 in other than Mode 0.

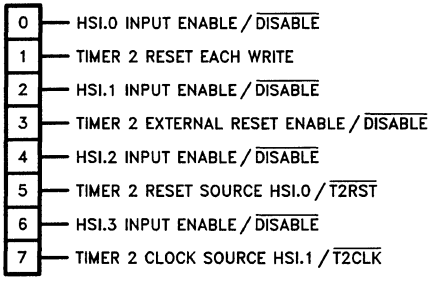
HSI Status (06H)



WHERE FOR EACH 2-BIT STATUS FIELD THE LOWER BIT INDICATES WHETHER OR NOT AN EVENT HAS OCCURED ON THIS PIN AND THE UPPER BIT INDICATES THE CURRENT STATUS OF THE PIN.

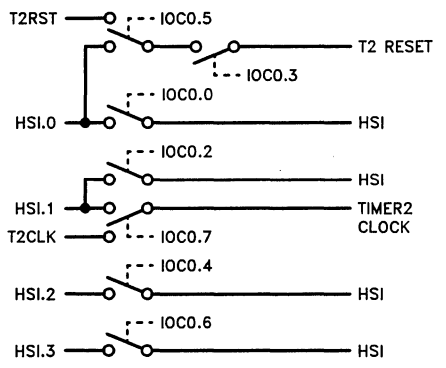
270652-44

IOC0 (15H)



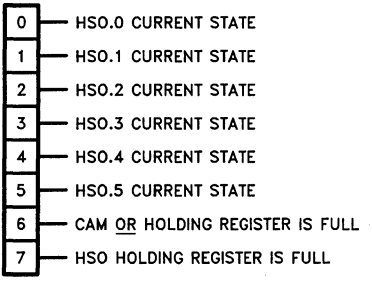
270652-45

IOC0 (15H)



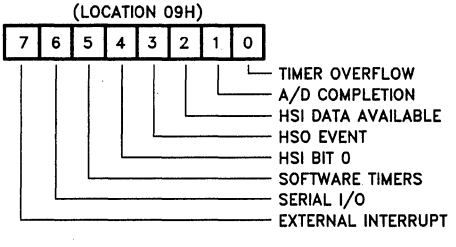
270652-46

IOS0 (15H)



270652-47

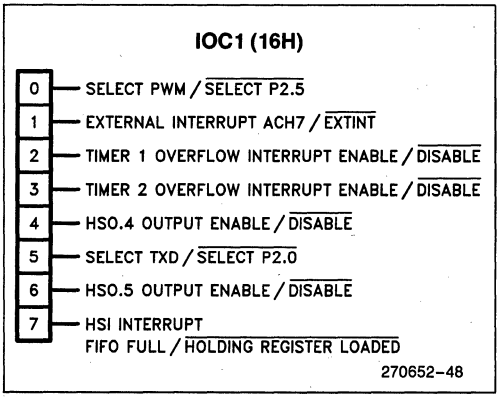
Interrupt Pending Register

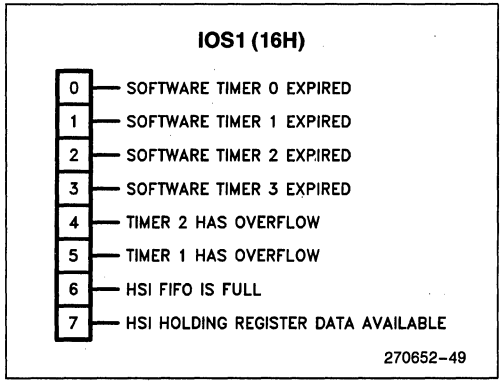


270652-17

PSW Register

BIT	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FLAG	Z	N	V	VT	C	—	I	ST	<Interrupt Mask Reg>							





Vector	Vector Location		Priority
	(High Byte)	(Low Byte)	
Software	2011H	2010H	Not Applicable
Extint	200FH	200EH	7 (Highest)
Serial Port	200DH	200CH	6
Software Timers	200BH	200AH	5
HSI.0	2009H	2008H	4
High Speed Outputs	2007H	2006H	3
HSI Data Available	2005H	2004H	2
A/D Conversion Complete	2003H	2002H	1
Timer Overflow	2001H	2000H	0 (Lowest)

MCS®-96
8098/8398
ADVANCED 8-BIT MICROCONTROLLER WITH 16-BIT CPU

- **8398: an 8098 with 8K Bytes of On-Chip ROM**
- **232 Byte Register File**
- **Register-to-Register Architecture**
- **10-Bit A/D Converter with S/H**
- **Two 8-Bit and Two 4-Bit I/O Ports**
- **20 Interrupt Sources**
- **Pulse-Width Modulated Output**
- **ROM Lock**
- **High Speed I/O Subsystem**
- **Full Duplex Serial Port**
- **Dedicated Baud Rate Generator**
- **6.25 μs 16 x 16 Multiply**
- **6.25 μs 32/16 Divide**
- **16-Bit Watchdog Timer**
- **Four 16-Bit Software Timers**
- **Two 16-Bit Counter/Timers**

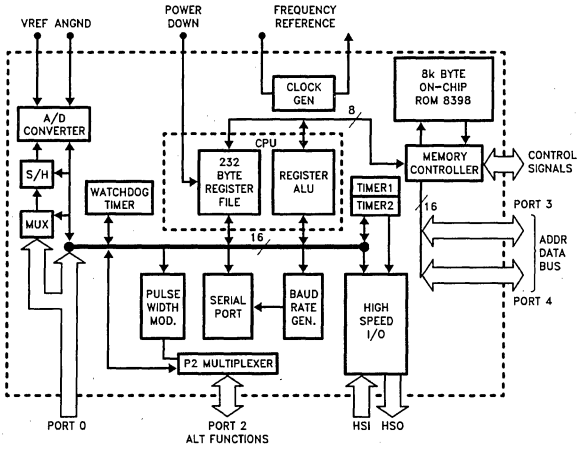
The MCS®-96 family of 16-bit microcontrollers consists of many members, all of which are designed for high-speed control functions. The 8X98 members were designed specifically for those applications that require the speed of a 16-bit microcontroller but are limited by board space and cost requirements to an 8-bit external bus. The 8X98 members are produced using Intel's HMOS-III process.

The CPU supports bit, byte, and word operations. Thirty-two bit double-words are supported for a subset of the instruction set. With a 12 MHz input frequency the 8098 can do a 16-bit addition in 1.0 μs and a 16 x 16-bit multiply or 32/16 divide in 6.25 μs. Instruction execution times average 1 to 2 μs in typical applications.

Four high-speed trigger inputs are provided to record the times at which external events occur. Six high-speed pulse generator outputs are provided to trigger external events at preset times. The high-speed output unit can simultaneously perform software timer functions. Up to four 16-bit software timers can be in operation at once.

The on-chip A/D converter includes a Sample and Hold, and converts up to 4 multiplexed analog input channels to 10-bit digital values. With a 12 MHz crystal, each conversion takes 22 μs.

Also provided on-chip are a serial port, a Watchdog Timer, and a pulse-width modulated output signal.



270532-1

Figure 1. MCS®-96 Block Diagram

October 1988
Order Number: 270532-002

FUNCTIONAL OVERVIEW

The following section is an overview of the 8X98 devices, generally referred to as the 8098. Additional information is available in the Embedded Controller Handbook, order number 210918.

CPU Architecture

The 8098 uses the same address space for both program and data memory, except in the address range from 00H through 0FFH. Data fetches in this range are always to the Register File, while instruction fetches from these locations are directed to external memory. (Locations 00H through 0FFH in external memory are reserved for Intel development systems).

Within the Register File, locations 00H through 17H are register mapped I/O control registers, also referred to as Special Function Registers (SFRs). The rest of the Register File (018H through 0FFH) contains 232 bytes of RAM, which can be referenced as bytes, words, or double-words. This register space allows the user to keep the most frequently-used variables in on-chip RAM, which can be accessed faster than external memory. Locations 0F0H through 0FFH can be preserved during power down via a separate power down pin (V_{PD}).

Outside of the Register File, program memory, data memory, and peripherals can be intermixed. The addresses with special significance are:

0000H-	0017H	Register Mapped I/O (SFRs)
0018H-	0019H	Stack Pointer
1FFEH-	1FFFH	Ports 3 and 4
2000H-	2011H	Interrupt Vectors
2012H-	2017H	Reserved
2018H		Chip Configuration Byte
2019H		Reserved
201AH-	201BH	"Jump to Self" Opcode (27 FE)
201CH-	201FH	Reserved
2020H-	202FH	Security Key
2030H-	207FH	Reserved
2080H		Reset Location

The 8398 carries 8K bytes of ROM. Internal program memory occupies addresses 2000H through 3FFFH. Instruction or data fetches from these addresses access the on-chip memory if the $\overline{EA}$ pin is externally held at 5V. If the $\overline{EA}$ pin is at 0V, these addresses access off-chip memory.

A memory map for the 8098 version of the MCS-96 product family is shown in Figure 3.

Reserved location warning: Intel Reserved addresses can not be used by applications which use 8X98 internal ROM. The data read from a reserved location is not guaranteed, and a write to any reserved location could cause unpredictable results. When attempting to program Intel Reserved addresses, the data must be 0FFFFH to ensure a harmless result. A memory map indicating reserved locations on the 8X98 is shown in Figure 2.

Intel Reserved locations, when mapped to external memory, must be filled with 0FFFFH to ensure compatibility with future parts.

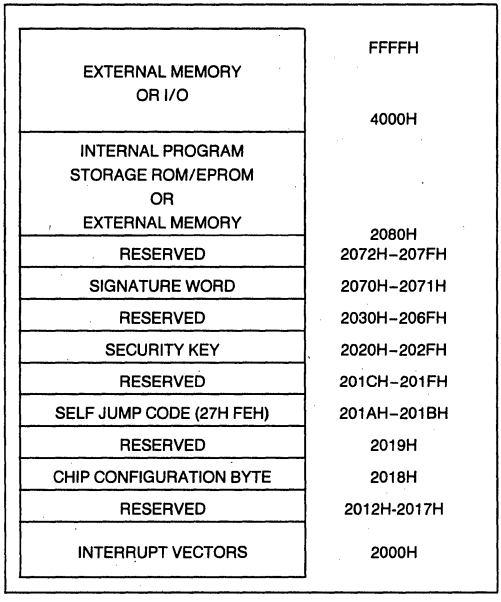


Figure 2. Reserved Locations

The RALU (Register/ALU) section consists of a 17-bit ALU, the Program Status Word, the Program Counter, and several temporary registers. A key feature of the 8098 is that it does not use an accumulator. Rather, it operates directly on any register in the Register File. Being able to operate directly on data in the Register File without having to move it into and out of an accumulator results in a significant improvement in execution speed.

In addition to the normal arithmetic and logical functions, the MCS-96 instruction set provides the following special features:

- 6.25 μs Multiply and Divide
- Multiple Shift Instruction
- 3 Operand Instructions
- Normalize Instruction
- Software Reset Instruction

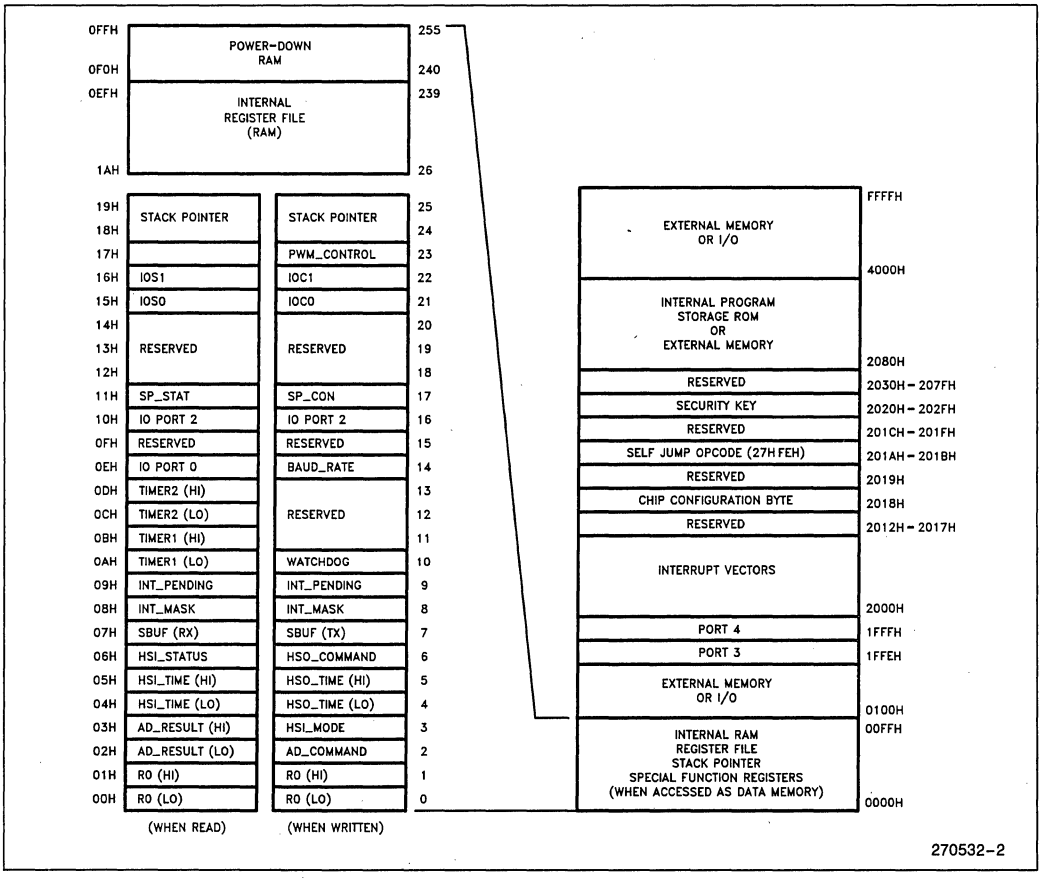


Figure 3. Memory Map

All operations on the 8098 take place in a set number of "State Times." The 8098 uses a three phase internal clock, so each state time is 3 oscillator periods. With a 12 MHz clock, each state time requires 0.25 μs, based on a T_{osc} of 83 ns.

Operating Modes

The 8098 supports a variety of options to simplify memory systems. Several ready control modes are available to simplify the external hardware requirements for accessing slow devices. The Chip Configuration Register is used to store the operating mode information.

CHIP CONFIGURATION REGISTER (CCR)

Configuration information is stored in the Chip Configuration Register (CCR). Three of the bits in the register specify the bus control mode and ready control mode. One bit also governs the level of ROM protection. The CCR bit map is shown in Figure 4, and the functions associated with each bit are described later.

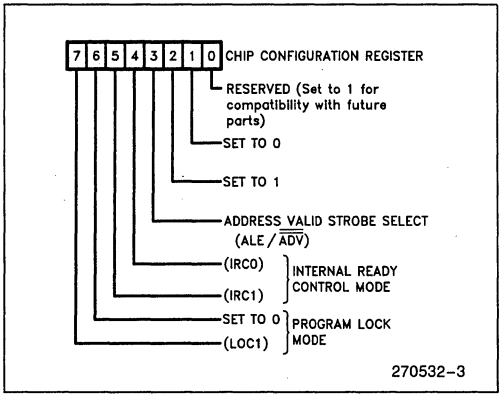


Figure 4. Chip Configuration Register

The CCR is loaded on reset with the Chip Configuration Byte, located at address 2018H. The CCR register is a non-memory mapped location that can only be written to during the reset sequence; once it is loaded it cannot be changed until the next reset occurs.

In order to work properly with an 8-bit only system, it is necessary to hold the upper address byte on the address bus throughout the CCB read cycle since an address latch may not be present.

If the $\overline{EA}$ pin is set to a logical 0, the access to 2018H comes from external memory. If $\overline{EA}$ is a logical 1, the access comes from internal ROM.

BUS CONTROL

The 8098 can be made to provide two types of bus control signals. ALE has a dual function designed to reduce external hardware. Bit 3 of the CCR specifies the function performed by these control lines.

Standard Bus Control

If CCR bit 3 is a 1, then the standard 8098 control signal ALE is provided (Figure 5). ALE will rise as the address starts to come out, and will fall to provide the signal to externally latch the address.

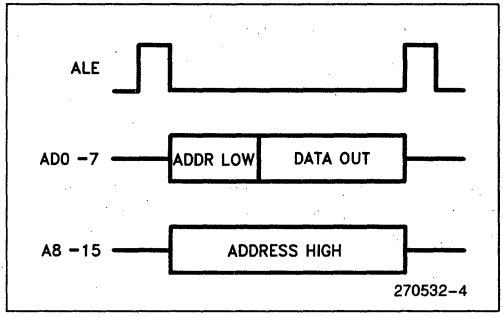


Figure 5. Standard Bus Control

Address Valid Strobe Mode

If CCR bit 3 is a 0, then an Address Valid Strobe is provided in the place of ALE (Figure 6). When the Address Valid Mode is selected, $\overline{ADV}$ will go low after an external address is set up. It will stay low until the end of the bus cycle, where it will go inactive high. This can be used to provide a chip select for external memory.

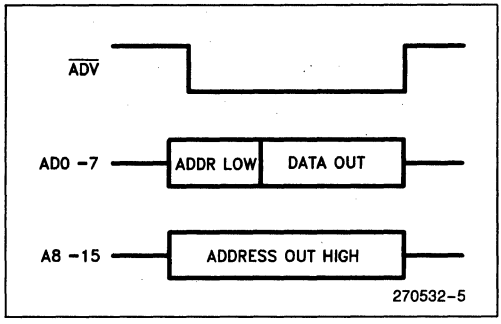


Figure 6. Address Valid Strobe Mode

READY CONTROL

To simplify ready control, four modes of internal ready control logic have been provided. The modes are chosen by properly configuring bits 4 and 5 of the CCR.

The internal ready control logic can be used to limit the number of wait states that slow devices can insert into the bus cycle. When the READY pin is pulled low, wait states will be inserted into the bus cycle until the READY pin goes high, or the number of wait states equals the number specified by CCR bits 4 and 5, whichever comes first. Table 1 shows the number of wait states that can be selected. Internal ready control can be disabled by loading 11 into bits 4 and 5 of the CCR.

Table 1. Internal Ready Control

IRC1	IRC0	Description
0	0	Limit to 1 Wait State
0	1	Limit to 2 Wait States
1	0	Limit to 3 Wait States
1	1	Disable Internal Ready Control

This feature provides for simple ready control. For example, every slow memory chip select line could be ORed together and be connected to the READY pin with CCR bits 4 and 5 programmed to give the proper number of wait states to the slow devices.

ROM LOCK

Program memory lock is available on the 8398 part. CCR bit 7 (LOC1) selects whether internal program memory can be read by a program executing from external memory. The modes are shown in Table 2. Internal ROM addresses 2020H through 3FFFH are protected from reads as set by the CCR when LOC1 is cleared to 0.

Table 2. Program Lock Modes

LOC1	Protection
0	Read Protected
1	No Protection

Only code executing from internal memory can read protected internal memory. As a result of 8098 prefetching of instructions, however, accesses to protected memory are not allowed for instructions located above 3FFAH. Note that the interrupt vectors and the CCR are not protected.

To provide ROM lock while allowing verification and testing, the 8398 requires security key verification before programming or test modes are allowed to read protected memory.

High Speed I/O Unit (HSIO)

The HSIO unit consists of the High Speed Input Unit (HSI), the High Speed Output Unit (HSO), one counter and one timer. "High Speed" denotes that the units can perform functions related to the timers without CPU intervention. The HSI records times when events occur and the HSO triggers events at pre-programmed times.

All actions within the HSIO unit are synchronized to the timers. The two 16-bit timer/counter registers in the HSIO unit are cleared on chip reset and can be programmed to generate an interrupt on overflow. The Timer 1 register is automatically incremented every 8 state times (every 2.0 μs, with a 12 MHz clock). The Timer 2 register counts transitions on the HSI.1 pin. It is incremented on both positive and negative edges of the input line. In addition to being cleared by reset, Timer 2 can also be cleared in software or by signals from input pin HSI.0. Neither of these timers is required for either the Watchdog Timer or the serial port.

The High Speed Input (HSI) unit can detect transitions on any of its 4 input lines. When one occurs it records the time (from Timer 1) and which input lines made the transition. This information is recorded with 2 μs (12 MHz system) resolution and stored in an 8-level FIFO. The unit can be programmed to look for four types of events, as shown in Figure 7. It can activate the HSI Data Available interrupt either when the Holding Register is loaded or the 6th FIFO entry has been made. Each input line can be individually enabled or disabled to the HSI unit by software.

The High Speed Output (HSO) unit is shown in Figure 8. It can be programmed to set or clear any of its 6 output lines, reset Timer 2, trigger an A/D conversion, or set one of 4 Software Timer flags at a programmed time. An interrupt can be enabled for any of these events. Either Timer 1 or Timer 2 can be referenced for the programmed time value and up to 8 commands for preset actions can be stored in the CAM (Content Addressable Memory) file at any one time. As each action is carried out at its preset time that command is removed from the CAM making space for another command. HSO.4 and HSO.5 are shared with the HSI unit as HSI.2 and HSI.3, and can be individually enabled or disabled as outputs.

Standard I/O Ports

There are 2 8-bit and 2 4-bit I/O ports on the 8098 in addition to the High Speed I/O lines.

Port 0 is an input-only port which shares its pins with the analog inputs to the A/D converter. The port can be read digitally and/or, by writing to the A/D Command Register, one of the lines can be selected as the input to the A/D converter.

Port 2 is a multi-functional port. The 4 pins are shared with other functions in the 8098, as shown in Table 3.

Table 3. Port 2 Pin Functions

Port	Function	Alternate Function
P2.0	Output	TXD (Serial Port Transmit)
P2.1	Input	RXD (Serial Port Receive)
P2.2	Input	EXTINT (External Interrupt)
P2.5	Output	PWM (Pulse Width Modulation)

Ports 3 and 4 are bi-directional I/O ports with open drain outputs. These pins are also used as the multiplexed address/data bus when accessing external memory, in which case they have strong internal pullups. The internal pullups are only used during external memory read or write cycles when the pins

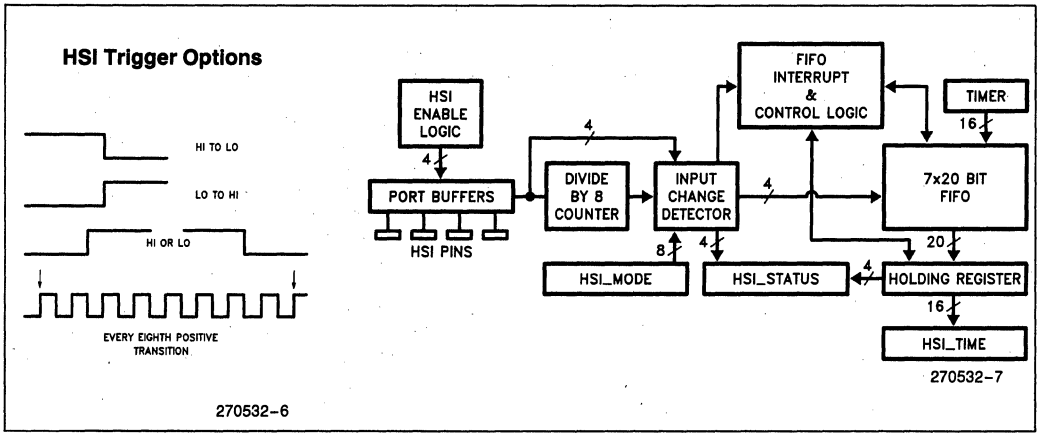


Figure 7. High Speed Input Unit

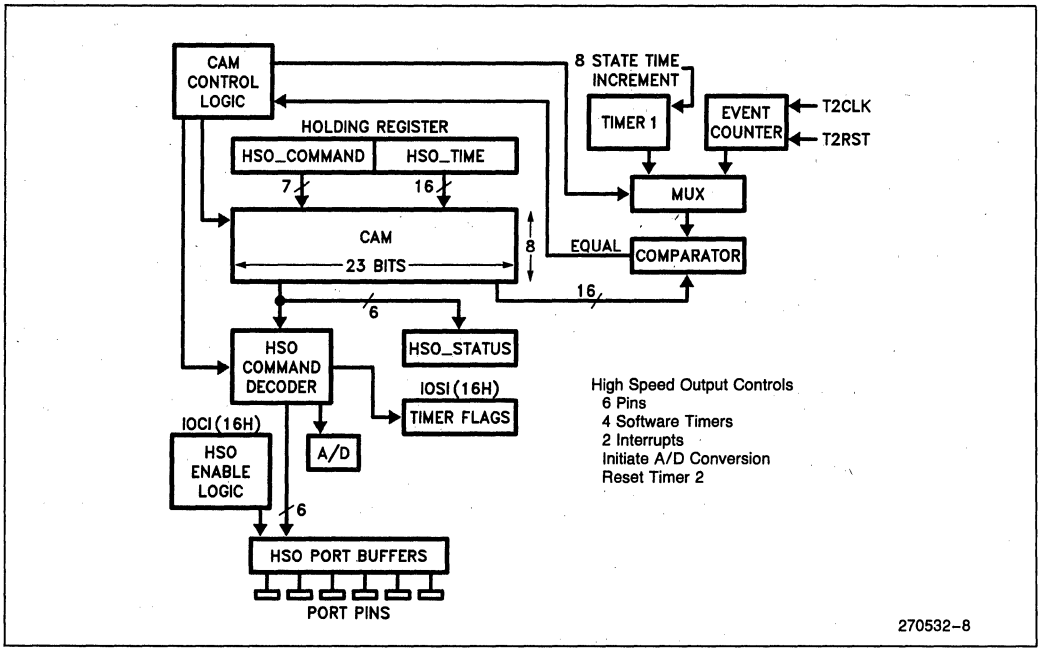


Figure 8. High Speed Output Unit

are outputting address or data bits. At any other time, the internal pullups are disabled. When used as a system bus, Ports 3 and 4 are a multiplexed 16-bit address/ 8-bit data bus.

Serial Port

The serial port is compatible with the MCS-51 family, (8051, 8031 etc.), serial port. It is full duplex, and double-buffered on receive. There are 3 asynchronous modes and 1 synchronous mode of operation for the serial port. The asynchronous modes allow for 8 or 9 bits of data with even parity optionally inserted for one of the data bits. Selective interrupts based on the 9th data bit are available to support interprocessor communication.

Baud rates in all modes are determined by an independent 16-bit on-chip baud rate generator. The XTAL1 pin is used as the input to the baud rate generator. The maximum baud rate in the asynchronous mode is 187.5 KBaud. The maximum baud rate in the synchronous mode is 1.5 MBaud.

Pulse Width Modulator (PWM)

The PWM output shares a pin with port bit P2.5. When the PWM output is selected, this pin outputs a pulse train having a fixed period of 256 state times, and a programmable width of 0 to 255 state times. The width is programmed by loading the desired value, in state times, to the PWM Control Register.

A/D Converter with Sample and Hold

The analog-to-digital converter is a 10-bit, successive approximation converter with internal sample and hold. It has a fixed conversion time of 88 state times which includes the 4 state acquisition time of the internal Sample/Hold. With a 12 MHz clock, the conversion takes 22 μs, including the 1 μs sample for the Sample and Hold. The Sample acquisition begins 4 state times after the conversion is triggered. A 2 pF capacitance is charged from the input signal during acquisition.

The analog input must be in the range of 0 to V_{REF} (nominally, V_{REF} = 5V). This input can be selected from 4 analog input lines, which connect to the same pins as Port 0. A conversion can be initiated either by setting a control bit in the A/D Command register, or by programming the HSO unit to trigger the conversion at some specified time.

Interrupts

The 8098 has 20 interrupt sources which vector through 8 interrupt vectors. A 0-to-1 transition from any of the sources sets a corresponding bit in the Interrupt Pending register. The content of the Interrupt Mask register determines if a pending interrupt will be serviced or not. If it is to be serviced, the CPU pushes the current Program Counter onto the stack and reloads it with the vector corresponding to the desired interrupt. The interrupt vectors are located in addresses 2000H through 2011H, as shown in Figure 9.

Vector	Vector Location		Priority
	(High Byte)	(Low Byte)	
Software	2011H	2010H	Not Applicable
Extint	200FH	200EH	7 (Highest)
Serial Port	200DH	200CH	6
Software Timers	200BH	200AH	5
HSI.0	2009H	2008H	4
High Speed Outputs	2007H	2006H	3
HSI Data Available	2005H	2004H	2
A/D Conversion Complete	2003H	2002H	1
Timer Overflow	2001H	2000H	0 (Lowest)

Figure 9. Interrupt Vectors

At the end of the interrupt routine the RET instruction pops the program counter from the stack and execution continues where it left off. It is not necessary to store and replace registers during interrupt routines as each routine can be set up to use a different section of the Register File. This feature of the architecture provides for very fast context switching. While the 8098 has a single priority level in the sense that any interrupt may itself be interrupted, a priority structure exists for resolving simultaneously pending interrupts, as indicated in Figure 9. Since the interrupt pending and interrupt mask registers can be manipulated in software, it is possible to dynamically alter the interrupt priorities to suit the users software.

Watchdog Timer

The Watchdog Timer is a 16-bit counter which, once started, is incremented every state time. If not cleared before it overflows, the $\overline{\text{RESET}}$ pin will be pulled down for two state times, causing the system to be reinitialized. In a 12 MHz system, the Watchdog Timer overflows after 16 ms.

This feature is provided as a means of graceful recovery from a software upset. The counter must be cleared by the software before it overflows, or else the system assumes an upset has occurred and activates $\overline{\text{RESET}}$. Once the Watchdog Timer is started it cannot be turned off by software. The flip-flop which enables the Watchdog Timer has been designed to maintain its state through V_{CC} glitches to as low as 0V or as high as 7V for 1 μs to 1 ms.

To start the Watchdog Timer, or to clear it, one writes 1EH followed by 0E1H to the WDT address (000AH). The Watchdog cannot be stopped once it is started unless the system is reset.

PACKAGING

The 8098 is available in a 48-pin package with and without on-chip ROM. The MCS-96 numbering system for the 8X98 parts is shown in Figure 10. Figure 11 shows the pinout for the 48-pin package. The 48-pin version is offered in a Dual-In-Line package.

		With A/D
ROMless	48 Pin	P8098 - Plastic DIP
ROM	48 Pin	P8398 - Plastic DIP

Figure 10. The MCS®-96 Family Nomenclature

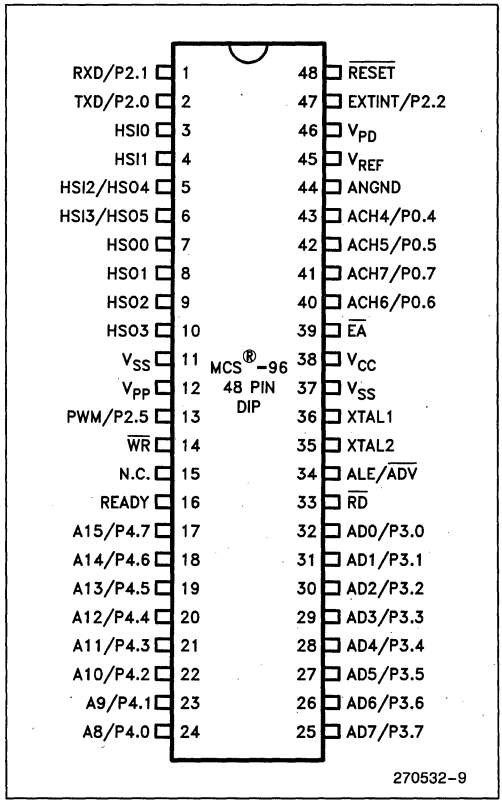


Figure 11. 48-Pin Package

PIN DESCRIPTIONS

Symbol	Name and Function
V_{CC}	Main supply voltage (5V).
V_{SS}	Digital circuit ground (0V). There are two V_{SS} pins, both of which must be connected.
V_{PD}	RAM standby supply voltage (5V). This voltage must be present during normal operation. In a Power Down condition (i.e. V_{CC} drops to zero), if $\overline{RESET}$ is activated before V_{CC} drops below spec and V_{PD} continues to be held within spec., the top 16 bytes in the Register File will retain their contents. $\overline{RESET}$ must be held low during the Power Down and should not be brought high until V_{CC} is within spec and the oscillator has stabilized.
V_{REF}	Reference voltage for the A/D converter (5V). V_{REF} is also the supply voltage to the analog portion of the A/D converter and the logic used to read Port 0. Must be connected for A/D and Port 0 to function.
ANGND	Reference ground for the A/D converter. Must be held at nominally the same potential as V_{SS}.
V_{PP}	Programming voltage for the future EPROM parts.
XTAL1	Input of the oscillator inverter and of the internal clock generator.
XTAL2	Output of the oscillator inverter.
$\overline{RESET}$	Reset input to the chip. Input low for at least 2 state times to reset the chip. The subsequent low-to-high transition re-synchronizes CLKOUT and commences a 10-state-time sequence in which the PSW is cleared, a byte read from 2018H loads CCR, and a jump to location 2080H is executed. Input high for normal operation. $\overline{RESET}$ has an internal pullup.
NMI	A positive transition causes a vector to external memory location 0000H. External memory from 00H through 0FFH is reserved for Intel development systems.
$\overline{EA}$	Input for memory select (External Access). $\overline{EA}$ equal to a TTL-high causes memory accesses to locations 2000H through 3FFFH to be directed to on-chip ROM. $\overline{EA}$ equal to a TTL-low causes accesses to these locations to be directed to off-chip memory. $\overline{EA}$ has an internal pulldown, so it goes to 0 unless driven otherwise. $\overline{EA}$ is latched at reset.
ALE/$\overline{ADV}$	Address Latch Enable or Address Valid output, as selected by CCR. Both pin options provide a latch to demultiplex the address from the address/data bus. When the pin is $\overline{ADV}$, it goes inactive high at the end of the bus cycle. $\overline{ADV}$ can be used as a chip select for external memory. ALE/$\overline{ADV}$ is activated only during external memory accesses.
$\overline{RD}$	Read signal output to external memory. $\overline{RD}$ is activated only during external memory reads.
$\overline{WR}$	Write output to external memory. $\overline{WR}$ is activated only during external memory writes.
READY	Ready input to lengthen external memory cycles, for interfacing to slow or dynamic memory, or for bus sharing. The bus cycle can be lengthened by up to 1 μs. When the external memory is not being used, READY has no effect. Internal control of the number of wait states inserted into a bus cycle held not ready is available through configuration of CCR. READY has a weak internal pullup, so it goes to 1 unless externally pulled low.
HSI	Inputs to High Speed Input Unit. Four HSI pins are available: HSI.0, HSI.1, HSI.2, and HSI.3. Two of them (HSI.2 and HSI.3) are shared with the HSO Unit.
HSO	Outputs from High Speed Output Unit. Six HSO pins are available: HSO.0, HSO.1, HSO.2, HSO.3, HSO.4, and HSO.5. Two of them (HSO.4 and HSO.5) are shared with the HSI Unit.
Port 0	4-bit high impedance input-only port. These pins can be used as digital inputs and/or as analog inputs to the on-chip A/D converter.
Port 2	4-bit multi-functional port. Its pins are shared with other functions in the 8098.
Ports 3 and 4	8-bit bi-directional I/O ports with open drain outputs. These pins are shared with the multiplexed address/data bus which has strong internal pullups.

INSTRUCTION SET

The 8098 instruction set makes use of six addressing modes as described below:

DIRECT—The operand is specified by an 8-bit address field in the instruction. The operand must be in the Register File or SFR space (locations 0000H through 00FFH).

IMMEDIATE—The operand itself follows the opcode in the instruction stream as immediate data. The immediate data can be either 8-bits or 16-bits as required by the opcode.

INDIRECT—An 8-bit address field in the instruction gives the word address of a word register in the Register File which contains the 16-bit address of the operand. The operand can be anywhere in memory.

INDIRECT WITH AUTO-INCREMENT—Same as Indirect, except that, after the operand is referenced, the word register that contains the operand's address is incremented by 1 if the operand is a byte, or by 2 if the operand is a word.

INDEXED (LONG AND SHORT)—The instruction contains an 8-bit address field and either an 8-bit or a 16-bit displacement field. The 8-bit address field gives the word address of a word register in the Register File which contains a 16-bit base address. The 8- or 16-bit displacement field contains a signed displacement that will be added to the base address to produce the address of the operand. The operand can be anywhere in memory.

The 8098 contains a zero register at word address 0000H (and which contains 0000H). This register is available for performing comparisons and for use as a base register in indexed addressing. This effectively provides direct addressing to all 64K of memory.

In the 8098 the Stack Pointer is at word address 0018H in the Register File. If the 8-bit address field contains 18H, the Stack Pointer becomes the base register. This allows direct accessing of variables in the stack.

The following tables list the MCS-96 instructions, their opcodes, and execution times.

A Word on Instruction Execution Times

Some performance degradation is to be expected when using the 8098 with external program memory. The reason for this is that the bus controller has a difficult time keeping the instruction prefetch queue full when the CPU is executing short instructions. The CPU then has to wait while the bus controller fetches the next instruction and its operands. The percentage that this degradation affects a particular string of code is dependent on the instructions used and in the sequencing of those instructions. Typically, add 20% to the state times calculated from the following tables for a given sequence of code that uses long state time instructions, and add 40% for a given sequence that uses short state time instructions.

Instruction Summary

Mnemonic	Operands	Operation (Note 1)	Flags						Notes
			Z	N	C	V	VT	ST	
ADD/ADDB	2	D ← D + A	↗	↗	↗	↗	↑	—	
ADD/ADDB	3	D ← B + A	↗	↗	↗	↗	↑	—	
ADDC/ADDCB	2	D ← D + A + C	↓	↗	↗	↗	↑	—	
SUB/SUBB	2	D ← D − A	↗	↗	↗	↗	↑	—	
SUB/SUBB	3	D ← B − A	↗	↗	↗	↗	↑	—	
SUBC/SUBCB	2	D ← D − A + C − 1	↓	↗	↗	↗	↑	—	
CMP/CMPB	2	D − A	↗	↗	↗	↗	↑	—	
MUL/MULU	2	D, D + 2 ← D * A	—	—	—	—	—	?	2
MUL/MULU	3	D, D + 2 ← B * A	—	—	—	—	—	?	2
MULB/MULUB	2	D, D + 1 ← D * A	—	—	—	—	—	?	3
MULB/MULUB	3	D, D + 1 ← B * A	—	—	—	—	—	?	3
DIVU	2	D ← (D, D + 2)/A, D + 2 ← remainder	—	—	—	↗	↑	—	2
DIVUB	2	D ← (D, D + 1)/A, D + 1 ← remainder	—	—	—	↗	↑	—	3
DIV	2	D ← (D, D + 2)/A, D + 2 ← remainder	—	—	—	?	↑	—	
DIVB	2	D ← (D, D + 1)/A, D + 1 ← remainder	—	—	—	?	↑	—	
AND/ANDB	2	D ← D and A	↗	↗	0	0	—	—	
AND/ANDB	3	D ← B and A	↗	↗	0	0	—	—	
OR/ORB	2	D ← D or A	↗	↗	0	0	—	—	
XOR/XORB	2	D ← D (excl. or) A	↗	↗	0	0	—	—	
LD/LDB	2	D ← A	—	—	—	—	—	—	
ST/STB	2	A ← D	—	—	—	—	—	—	
LDBSE	2	D ← A; D + 1 ← SIGN(A)	—	—	—	—	—	—	3, 4
LDBZE	2	D ← A; D + 1 ← 0	—	—	—	—	—	—	3, 4
PUSH	1	SP ← SP − 2; (SP) ← A	—	—	—	—	—	—	
POP	1	A ← (SP); SP ← SP + 2	—	—	—	—	—	—	
PUSHF	0	SP ← SP − 2; (SP) ← PSW; PSW ← 0000H I ← 0	0	0	0	0	0	0	
POPF	0	PSW ← (SP); SP ← SP + 2; I ← ↗	↗	↗	↗	↗	↗	↗	
SJMP	1	PC ← PC + 11-bit offset	—	—	—	—	—	—	5
LJMP	1	PC ← PC + 16-bit offset	—	—	—	—	—	—	5
BR [indirect]	1	PC ← (A)	—	—	—	—	—	—	
SCALL	1	SP ← SP − 2; (SP) ← PC; PC ← PC + 11-bit offset	—	—	—	—	—	—	5
LCALL	1	SP ← SP − 2; (SP) ← PC; PC ← PC + 16-bit offset	—	—	—	—	—	—	5
RET	0	PC ← (SP); SP ← SP + 2	—	—	—	—	—	—	
J (conditional)	1	PC ← PC + 8-bit offset (if taken)	—	—	—	—	—	—	5
JC	1	Jump if C = 1	—	—	—	—	—	—	5
JNC	1	Jump if C = 0	—	—	—	—	—	—	5
JE	1	Jump if Z = 1	—	—	—	—	—	—	5

NOTES:
1. If the mnemonic ends in "B", a byte operation is performed, otherwise a word operation is done. Operands D, B, and A must conform to the alignment rules for the required operand type. D and B are locations in the Register File; A can be located anywhere in memory.
2. D, D + 2 are consecutive WORDS in memory; D is DOUBLE-WORD aligned.
3. D, D + 1 are consecutive BYTES in memory; D is WORD aligned.
4. Changes a byte to a word.
5. Offset is a 2's complement number.

Instruction Summary (Continued)

Mnemonic	Operands	Operation (Note 1)	Flags						Notes
			Z	N	C	V	VT	ST	
JNE	1	Jump if Z = 0	—	—	—	—	—	—	5
JGE	1	Jump if N = 0	—	—	—	—	—	—	5
JLT	1	Jump if N = 1	—	—	—	—	—	—	5
JGT	1	Jump if N = 0 and Z = 0	—	—	—	—	—	—	5
JLE	1	Jump if N = 1 or Z = 1	—	—	—	—	—	—	5
JH	1	Jump if C = 1 and Z = 0	—	—	—	—	—	—	5
JNH	1	Jump if C = 0 or Z = 1	—	—	—	—	—	—	5
JV	1	Jump if V = 1	—	—	—	—	—	—	5
JNV	1	Jump if V = 0	—	—	—	—	—	—	5
JVT	1	Jump if VT = 1; Clear VT	—	—	—	—	0	—	5
JNVT	1	Jump if VT = 0; Clear VT	—	—	—	—	0	—	5
JST	1	Jump if ST = 1	—	—	—	—	—	—	5
JNST	1	Jump if ST = 0	—	—	—	—	—	—	5
JBS	3	Jump if Specified Bit = 1	—	—	—	—	—	—	5, 6
JBC	3	Jump if Specified Bit = 0	—	—	—	—	—	—	5, 6
DJNZ	1	D ← D − 1; if D ≠ 0 then PC ← PC + 8-bit offset	—	—	—	—	—	—	5
DEC/DECB	1	D ← D − 1	↕	↕	↕	↕	↑	—	
NEG/NEGB	1	D ← 0 − D	↕	↕	↕	↕	↑	—	
INC/INCB	1	D ← D + 1	↕	↕	↕	↕	↑	—	
EXT	1	D ← D; D + 2 ← Sign (D)	↕	↕	0	0	—	—	2
EXTB	1	D ← D; D + 1 ← Sign(D)	↕	↕	0	0	—	—	3
NOT/NOTB	1	D ← Logical Not (D)	↕	↕	0	0	—	—	
CLR/CLRB	1	D ← 0	1	0	0	0	—	—	
SHL/SHLB/SHLL	2	C ← msb — — — — — — lsb ← 0	↕	?	↕	↕	↑	—	7
SHR/SHRB/SHRL	2	0 → msb — — — — — — lsb → C	↕	?	↕	0	—	↕	7
SHRA/SHRAB/SHRAL	2	msb → msb — — — — — — lsb → C	↕	↕	↕	0	—	↕	7
SETC	0	C ← 1	—	—	1	—	—	—	
CLRC	0	C ← 0	—	—	0	—	—	—	
CLRVT	0	VT ← 0	—	—	—	—	0	—	
RST	0	PC ← 2080H	0	0	0	0	0	0	8
DI	0	Disable All Interrupts (I ← 0)	—	—	—	—	—	—	
EI	0	Enable All Interrupts (I ← 1)	—	—	—	—	—	—	
NOP	0	PC ← PC + 1	—	—	—	—	—	—	
SKIP	0	PC ← PC + 2	—	—	—	—	—	—	
NORML	2	Left shift till msb = 1; D ← shift count	↕	?	0	—	—	—	7
TRAP	0	SP ← SP − 2; (SP) ← PC PC ← (2010H)	—	—	—	—	—	—	9

NOTES:
1. If the mnemonic ends in "B", a byte operation is performed, otherwise a word operation is done. Operands D, B and A must conform to the alignment rules for the required operand type. D and B are locations in the Register File; A can be located anywhere in memory.
5. Offset is a 2's complement number.
6. Specified bit is one of the 2048 bits in the register file.
7. The "L" (Long) suffix indicates double-word operation.
8. Initiates a Reset by pulling $\overline{\text{RESET}}$ low. Software should re-initialize all the necessary registers with code starting at 2080H.
9. The assembler will not accept this mnemonic.

Opcode and State Time Listing

MNEMONIC	OPERANDS	DIRECT			IMMEDIATE			INDIRECT*					INDEXED*				
								NORMAL			AUTO-INC.		SHORT			LONG	
		OPCODE	BYTES	STATE TIMES	OPCODE	BYTES	STATE TIMES	OPCODE	BYTES	STATE ① TIMES	BYTES	STATE ① TIMES	OPCODE	BYTES	STATE ① TIMES	BYTES	STATE ① TIMES
ARITHMETIC INSTRUCTIONS																	
ADD	2	64	3	4	65	4	5	66	3	6/12	3	7/13	67	4	6/12	5	7/13
ADD	3	44	4	5	45	5	6	46	4	7/13	4	8/14	47	5	7/13	6	8/14
ADDB	2	74	3	4	75	3	4	76	3	6/11	3	7/12	77	4	6/11	5	7/12
ADDB	3	54	4	5	55	4	5	56	4	7/12	4	8/13	57	5	7/12	6	8/13
ADDC	2	A4	3	4	A5	4	5	A6	3	6/12	3	7/13	A7	4	6/12	5	7/13
ADDCB	2	B4	3	4	B5	3	4	B6	3	6/11	3	7/12	B7	4	6/11	5	7/12
SUB	2	68	3	4	69	4	5	6A	3	6/12	3	7/13	6B	4	6/12	5	7/13
SUB	3	48	4	5	49	5	6	4A	4	7/13	4	8/14	4B	5	7/13	6	8/14
SUBB	2	78	3	4	79	3	4	7A	3	6/11	3	7/12	7B	4	6/11	5	7/12
SUBB	3	58	4	5	59	4	5	5A	4	7/12	4	8/13	5B	5	7/12	6	8/13
SUBC	2	A8	3	4	A9	4	5	AA	3	6/12	3	7/13	AB	4	6/12	5	7/13
SUBCB	2	B8	3	4	B9	3	4	BA	3	6/11	3	7/12	BB	4	6/11	5	7/12
CMP	2	88	3	4	89	4	5	8A	3	6/12	3	7/13	8B	4	6/12	5	7/13
CMPB	2	98	3	4	99	3	4	9A	3	6/11	3	7/12	9B	4	6/11	5	7/12
MULU	2	6C	3	25	6D	4	26	6E	3	27/33	3	28/34	6F	4	27/33	5	28/34
MULU	3	4C	4	26	4D	5	27	4E	4	28/34	4	29/35	4F	5	28/34	6	29/35
MULUB	2	7C	3	17	7D	3	17	7E	3	19/24	3	20/25	7F	4	19/24	5	20/25
MULUB	3	5C	4	18	5D	4	18	5E	4	20/25	4	21/26	5F	5	20/25	6	21/26
MUL	2	②	4	29	②	5	30	②	4	31/37	4	32/38	②	5	31/37	6	32/38
MUL	3	②	5	30	②	6	31	②	5	32/38	5	33/39	②	6	32/38	7	33/39
MULB	2	②	4	21	②	4	21	②	4	23/28	4	24/29	②	5	23/28	6	24/29
MULB	3	②	5	22	②	5	22	②	5	24/29	5	25/30	②	6	24/29	7	25/30
DIVU	2	8C	3	25	8D	4	26	8E	3	28/33	3	29/34	8F	4	28/33	5	29/34
DIVUB	2	9C	3	17	9D	3	17	9E	3	20/24	3	21/25	9F	4	20/24	5	21/25
DIV	2	②	4	29	②	5	30	②	4	32/37	4	33/38	②	5	32/37	6	33/38
DIVB	2	②	4	21	②	4	21	②	4	24/28	4	25/29	②	5	24/28	6	25/29

NOTES:

*Long indexed and Indirect + instructions have identical opcodes with Short indexed and Indirect modes, respectively. The second byte of instructions using any Indirect or indexed addressing mode specifies the exact mode used. If the second byte is even, use Indirect or Short indexed. If it is odd, use Indirect + or Long indexed. In all cases the second byte of the instruction always specifies an even (word) location for the address referenced.

① Number of state times shown for internal/external operands.

② The opcodes for signed multiply and divide are the opcodes for the unsigned functions with an "FE" appended as a prefix.

Opcode and State Time Listing (Continued)

MNEMONIC	OPERANDS	DIRECT			IMMEDIATE			INDIRECT*					INDEXED*				
								NORMAL			AUTO-INC.		SHORT			LONG	
		OPCODE	BYTES	STATE TIMES	OPCODE	BYTES	STATE TIMES	OPCODE	BYTES	STATE① TIMES	BYTES	STATE① TIMES	OPCODE	BYTES	STATE① TIMES	BYTES	STATE① TIMES
LOGICAL INSTRUCTIONS																	
AND	2	60	3	4	61	4	5	62	3	6/12	3	7/13	63	4	6/12	5	7/13
AND	3	40	4	5	41	5	6	42	4	7/13	4	8/14	43	5	7/13	6	8/14
ANDB	2	70	3	4	71	3	4	72	3	6/11	3	7/12	73	4	6/11	5	7/12
ANDB	3	50	4	5	51	4	5	52	4	7/12	4	8/13	53	5	7/12	6	8/13
OR	2	80	3	4	81	4	5	82	3	6/12	3	7/13	83	4	6/12	5	7/13
ORB	2	90	3	4	91	3	4	92	3	6/11	3	7/12	93	4	6/11	5	7/12
XOR	2	84	3	4	85	4	5	86	3	6/12	3	7/13	87	4	6/12	5	7/13
XORB	2	94	3	4	95	3	4	96	3	6/11	3	7/12	97	4	6/11	5	7/12
DATA TRANSFER INSTRUCTIONS																	
LD	2	A0	3	4	A1	4	5	A2	3	6/12	3	7/13	A3	4	6/12	5	7/13
LDB	2	B0	3	4	B1	3	4	B2	3	6/11	3	7/12	B3	4	6/11	5	7/12
ST	2	C0	3	4	—	—	—	C2	3	7/13	3	8/14	C3	4	7/13	5	8/14
STB	2	C4	3	4	—	—	—	C6	3	7/11	3	8/12	C7	4	7/11	5	8/12
LDBSE	2	BC	3	4	BD	3	4	BE	3	6/12	3	7/13	BF	4	6/12	5	7/13
LDBZE	2	AC	3	4	AD	3	4	AE	3	6/12	3	7/13	AF	4	6/12	5	7/13
STACK OPERATIONS (internal stack)																	
PUSH	1	C8	2	8	C9	3	8	CA	2	11/16	2	12/17	CB	3	11/16	4	12/17
POP	1	CC	2	12	—	—	—	CE	2	14/20	2	14/20	CF	3	14/20	4	14/20
PUSHF	0	F2	1	8	—	—	—	—	—	—	—	—	—	—	—	—	—
POPF	0	F3	1	9	—	—	—	—	—	—	—	—	—	—	—	—	—
STACK OPERATIONS (external stack)																	
PUSH	1	C8	2	14	C9	3	14	CA	2	17/22	2	18/23	CB	3	17/22	4	18/23
POP	1	CC	2	15	—	—	—	CE	2	17/23	2	17/23	CF	3	17/23	4	17/23
PUSHF	0	F2	1	14	—	—	—	—	—	—	—	—	—	—	—	—	—
POPF	0	F3	1	14	—	—	—	—	—	—	—	—	—	—	—	—	—

JUMP AND CALLS							
MNEMONIC	OPCODE	BYTES	STATES	MNEMONIC	OPCODE	BYTES	STATES
LJMP	E7	3	4	LCALL	EF	3	13/18⑤
SJMP	20–27④	2	8	SCALL	28–2F④	2	13/18⑤
BR[ ]	E3	2	8	RET	F0	1	12/18⑤
				TRAP③	F7	1	21/26

NOTES:
*Long indexed and Indirect + instructions have identical opcodes with Short indexed and Indirect modes, respectively. The second byte of instructions using any Indirect or indexed addressing mode specifies the exact mode used. If the second byte is even, use Indirect or Short indexed. If it is odd, use Indirect + or Long indexed. In all cases the second byte of the instruction always specifies an even (word) location for the address referenced.
① Number of state times shown for internal/external operands.
③ The assembler does not accept this mnemonic.
④ The least significant 3 bits of the opcode are concatenated with the following 8 bits to form an 11-bit, 2's complement, offset for the relative call or jump.
⑤ State times for stack located internal/external.

CONDITIONAL JUMPS

All conditional jumps are 2 byte instructions. They require 8 state times if the jump is taken, 4 if it is not.							
MNEMONIC	**OPCODE**	**MNEMONIC**	**OPCODE**	**MNEMONIC**	**OPCODE**	**MNEMONIC**	**OPCODE**
JC	DB	JE	DF	JGE	D6	JGT	D2
JNC	D3	JNE	D7	JLT	DE	JLE	DA
JH	D9	JV	DD	JVT	DC	JST	D8
JNH	D1	JNV	D5	JNVT	D4	JNST	D0

JUMP ON BIT CLEAR OR BIT SET

These instructions are 3-byte instructions. They require 9 state times if the jump is taken, 5 if it is not.								
	BIT NUMBER							
MNEMONIC	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
JBC	30	31	32	33	34	35	36	37
JBS	38	39	3A	3B	3C	3D	3E	3F

LOOP CONTROL

MNEMONIC	**OPCODE**	**BYTES**	**STATE TIMES**
DJNZ	EO	3	5/9 STATE TIME (NOT TAKEN/TAKEN)

SINGLE REGISTER INSTRUCTIONS

MNEMONIC	**OPCODE**	**BYTES**	**STATES**	**MNEMONIC**	**OPCODE**	**BYTES**	**STATES**
DEC	05	2	4	EXT	06	2	4
DECB	15	2	4	EXTB	16	2	4
NEG	03	2	4	NOT	02	2	4
NEGB	13	2	4	NOTB	12	2	4
INC	07	2	4	CLR	01	2	4
INCB	17	2	4	CLRB	11	2	4

SHIFT INSTRUCTIONS

INSTR MNEMONIC	**WORD**		**INSTR MNEMONIC**	**BYTE**		**INSTR MNEMONIC**	**DBL WD**		**STATE TIMES**
	OP	**B**		**OP**	**B**		**OP**	**B**	
SHL	09	3	SHLB	19	3	SHLL	0D	3	7 + 1 PER SHIFT(7)
SHR	08	3	SHRB	18	3	SHRL	0C	3	7 + 1 PER SHIFT(7)
SHRA	0A	3	SHRAB	1A	3	SHRAL	0E	3	7 + 1 PER SHIFT(7)

SPECIAL CONTROL INSTRUCTIONS

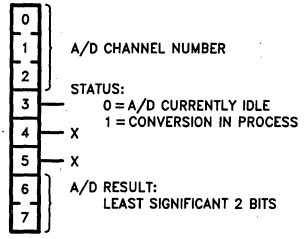
MNEMONIC	**OPCODE**	**BYTES**	**STATES**	**MNEMONIC**	**OPCODE**	**BYTES**	**STATES**
SETC	F9	1	4	DI	FA	1	4
CLRC	F8	1	4	EI	FB	1	4
CLRVT	FC	1	4	NOP	FD	1	4
RST(6)	FF	1	166	SKIP	00	2	4

NORMALIZE

MNEMONIC	OPCODE	BYTES	STATE TIMES
NORML	0F	3	11 + 1 PER SHIFT

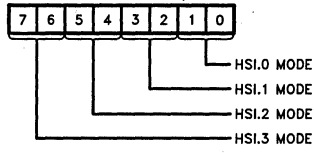
NOTES:
6. This instruction takes 2 states to pull $\overline{\text{RESET}}$ low, then holds it low for 2 states to initiate a reset. The reset takes 12 states, at which time the program restarts at location 2080H. If a capacitor is tied to $\overline{\text{RESET}}$, the pin may take longer to go low and may never reach the V_{OL} specification.
7. Execution will take at least 8 states, even for 0 shift.

A/D Result LO (02H)



270532-10

HSI_Mode (03H)

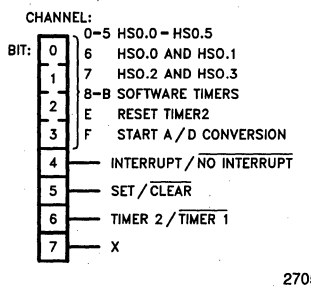


WHERE EACH 2-BIT MODE CONTROL FIELD DEFINES ONE OF 4 POSSIBLE MODES:

00 8 POSITIVE TRANSITIONS
01 EACH POSITIVE TRANSITION
10 EACH NEGATIVE TRANSITION
11 EVERY TRANSITION (POSITIVE AND NEGATIVE)

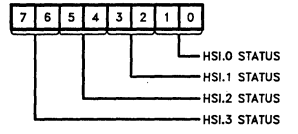
270532-11

HSO Command (06H)



270532-12

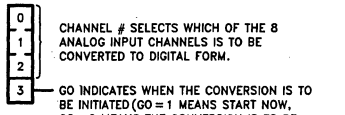
HSI_Status (06H)



WHERE FOR EACH 2-BIT STATUS FIELD THE LOWER BIT INDICATES WHETHER OR NOT AN EVENT HAS OCCURRED ON THIS PIN AND THE UPPER BIT INDICATES THE CURRENT STATUS OF THE PIN.

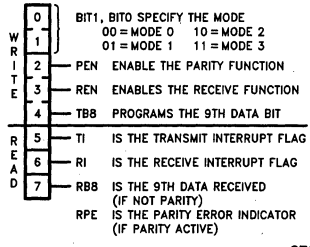
270532-13

A/D Command (02H)



270532-14

SPCON/SPSTAT (11H)



270532-15

Baud Rate Calculations

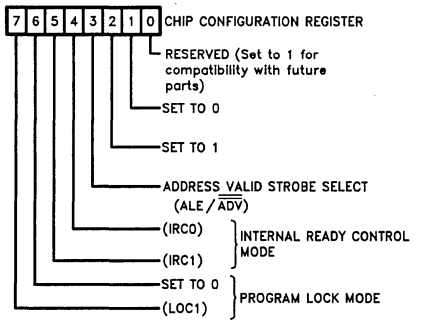
Using XTAL1:

Mode 0: $\text{Baud Rate} = \frac{\text{XTAL1 frequency}}{4*(B+1)}$; $B \neq 0$

Others: $\text{Baud Rate} = \frac{\text{XTAL1 frequency}}{64*(B+1)}$

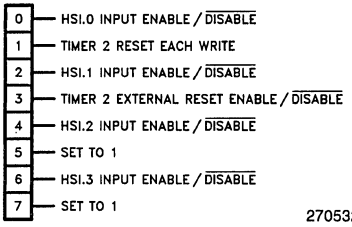
Note that B cannot equal 0, except when using XTAL1 in other than Mode 0.

Chip Configuration



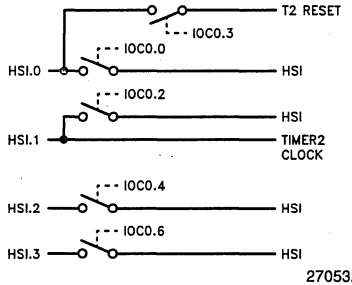
270532-16

IOC0 (15H)



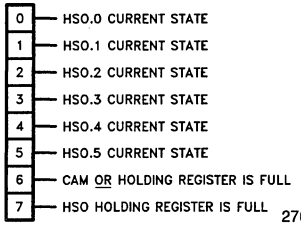
270532-17

IOC0 (15H)



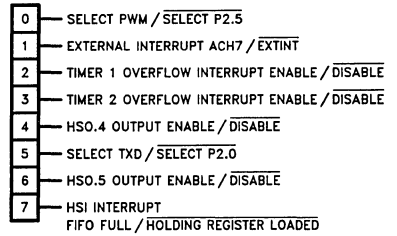
270532-18

IOS0 (15H)



270532-19

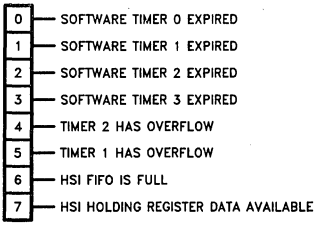
IOC1 (16H)



270532-20

Vector	Vector Location (High Byte)	Vector Location (Low Byte)	Priority
Software	2011H	2010H	Not Applicable
Extint	200FH	200EH	7 (Highest)
Serial Port	200DH	200CH	6
Software Timers	200BH	200AH	5
HSI.0	2009H	2008H	4
High Speed Outputs	2007H	2006H	3
HSI Data Available	2005H	2004H	2
A/D Conversion Complete	2003H	2002H	1
Timer Overflow	2001H	2000H	0 (Lowest)

IOS1 (16H)



270532-21

ELECTRICAL CHARACTERISTICS
ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−40°C to +150°C
Voltage from $\overline{EA}$ or V_{PP} to V_{SS} or ANGND .....−0.3V to +13.0V
Voltage from Any Other Pin to V_{SS} or ANGND .....−0.3V to +7.0V*
Average Output Current from Any Pin .....10 mA
Power Dissipation.....1.5W
*This includes V_{PP} on ROM and CPU only devices.

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Units
T_A	Ambient Temperature Under Bias	0	+70	C
V_{CC}	Digital Supply Voltage	4.50	5.50	V
V_{REF}	Analog Supply Voltage	4.50	5.50	V
f_{OSC}	Oscillator Frequency	6.0	12	MHz
V_{PD}	Power-Down Supply Voltage	4.50	5.50	V

NOTE:
ANGND and V_{SS} should be nominally at the same potential.

D.C. CHARACTERISTICS

(Test Conditions: V_{CC}, V_{REF}, V_{PD}, V_{PP}, V_{EA} = 5.0V ±0.5V; F_{OSC} = 6.0 MHz; T_A = 0°C to 70°C; V_{SS}, ANGND = 0V)

Symbol	Parameter	Min	Max	Units	Test Conditions
I_{CC}	V_{CC} Supply Current (0°C ≤ T_A ≤ 70°C)		240	mA	All Outputs Disconnected.
I_{CC1}	V_{CC} Supply Current (T_A = 70°C)		185	mA	
I_{PD}	V_{PD} Supply Current		1	mA	Normal operation and Power-Down.
I_{REF}	V_{REF} Supply Current		8	mA	
V_{IL}	Input Low Voltage (Except $\overline{RESET}$)	−0.3	+0.8	V	
V_{IL1}	Input Low Voltage, $\overline{RESET}$	−0.3	+0.7	V	
V_{IH}	Input High Voltage (Except $\overline{RESET}$, NMI, XTAL1)	2.0	V_{CC} +0.5	V	
V_{IH1}	Input High Voltage, $\overline{RESET}$ Rising	2.4	V_{CC} +0.5	V	
V_{IH2}	Input High Voltage, $\overline{RESET}$ Falling Hysteresis	2.1	V_{CC} +0.5	V	
V_{IH3}	Input High Voltage, NMI, XTAL1	2.2	V_{CC} +0.5	V	
I_{LI}	Input Leakage Current to each pin of HSI, P3, P4, and to P2.1.		±10	µA	V_{in} = 0 to V_{CC}
I_{LI1}	D.C. Input Leakage Current to each pin of P0		+3	µA	V_{in} = 0 to V_{CC}
I_{IH}	Input High Current to $\overline{EA}$		100	µA	V_{IH} = 2.4V

D.C. CHARACTERISTICS

(Test Conditions: V_{CC}, V_{REF}, V_{PD}, V_{PP}, V_{EA} = 5.0V ±0.5V; F_{OSC} = 6.0 MHz; T_A = 0°C to 70°C; V_{SS}, ANGND = 0V) (Continued)

Symbol	Parameter	Min	Max	Units	Test Conditions
I_{IL1}	Input Low Current to $\overline{RESET}$	−0.25	−2	mA	V_{IL} = 0.45V
I_{IL2}	Input Low Current P2.2		−50	μA	V_{IL} = 0.45V
V_{OL}	Output Low Voltage on P3, P4 when used as ports		0.45	V	I_{OL} = 0.8 mA (Note 1)
V_{OL1}	Output Low Voltage on P3, P4 when used as ports		0.75	V	I_{OL} = 2.0 mA (Notes 1, 2, 3)
V_{OL2}	Output Low Voltage on Standard Output pins, $\overline{RESET}$ and Bus/Control Pins		0.45	V	I_{OL} = 2.0 mA (Notes 1, 2, 3)
V_{OH1}	Output High Voltage on Standard Output pins and Bus/Control pins	2.4		V	I_{OH} = −200 μA (Note 1)
I_{OH3}	Output High Current on $\overline{RESET}$	−50		μA	V_{OH} = 2.4V
C_S	Pin Capacitance (Any Pin to V_{SS})		10	pF	fTEST = 1.0 MHz

NOTES:
1. Standard Output Pins include TXD, RXD (Mode 0 only), PWM, and HSO pins. Bus/Control pins include ALE, $\overline{RD}$, $\overline{WR}$, AD0–AD7, and A8–A15.
2. Maximum current per pin must be externally limited to the following values if V_{OL} is held above 0.45V.
   I_{OL} on Ports 3 and 4 when used as ports: 4.0 mA
   I_{OL} on standard output pins and $\overline{RESET}$: 8.0 mA
   I_{OL} on Bus/Control pins: 2.0 mA
3. During normal (non-transient) operation the following limits apply:
   Total I_{OL} on P2.0, $\overline{RESET}$ and all HSO pins must not exceed 15 mA.
   Total I_{OL} on Port 3 must not exceed 10 mA.
   Total I_{OL} on P2.5 and Port 4 must not exceed 20 mA.
4. I_{OL} on HSO.0, HSO.4, HSO.5, @ 0.5V = 1.6 mA.

A.C. CHARACTERISTICS

V_{CC}, V_{PD} = 4.5V to 5.5V; T_A = 0°C to 70°C; f_{OSC} = 6.0 MHz to 10.0 MHz
Test Conditions: Load Capacitance on Output Pins = 80 pF
Oscillator Frequency = 10 MHz

TIMING REQUIREMENTS (Other system components must meet these specs.)

Symbol	Parameter	Min	Max	Units
T_{LLYV}	End of ALE/$\overline{ADV}$ to READY Valid		2Tosc − 70	ns
T_{LLYH}	End of ALE/$\overline{ADV}$ to READY High	2Tosc + 40	4Tosc − 80	ns
T_{YLYH}	Non-Ready Time		1000	ns
T_{AVDV}(1)	Address Valid to Input Data Valid		5Tosc − 120	ns
T_{RLDV}	$\overline{RD}$ Active to Input Data Valid		3Tosc − 100	ns
T_{RHDX}	Data Hold after $\overline{RD}$ Inactive	0		ns
T_{RHDZ}	$\overline{RD}$ Inactive to Input Data Float	0	Tosc − 25	ns

NOTE:
1. The term "Address Valid" applies to A0–A15.

A.C. CHARACTERISTICS V_{CC}, V_{PD} = 4.5V to 5.5V; T_A = 0°C to 70°C; f_{OSC} = 6.0 MHz to 12.0 MHz (Continued)

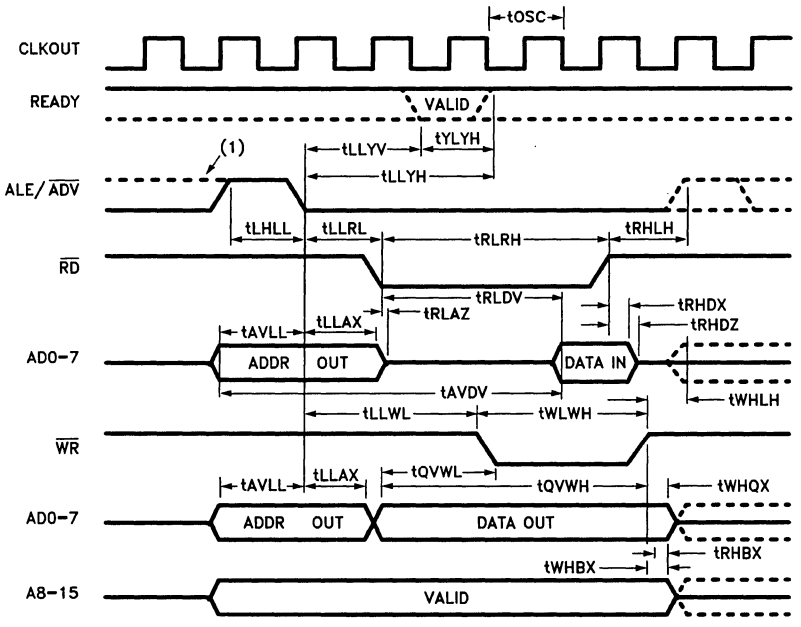
Test Conditions: Load Capacitance on Output Pins = 80 pF
Oscillator Frequency = 10 MHz

TIMING RESPONSES (MCS-96 parts meet these specs.)

Symbol	Parameter	Min	Max	Units
F_{XTAL}	Oscillator Frequency	6.0	12.0	MHz
T_{OSC}	Oscillator Period	83	166	ns
T_{LHLL}	ALE/$\overline{ADV}$ High Time	Tosc − 30	Tosc + 35(3)	ns
T_{AVLL}(4)	Address Setup to End of ALE/$\overline{ADV}$	Tosc − 50		ns
T_{RLAZ}(5)	$\overline{RD}$ or $\overline{WR}$ Low to Address Float	Typ. = 0	10	ns
T_{LLRL}	End of ALE/$\overline{ADV}$ to $\overline{RD}$ Active	Tosc − 40		ns
T_{LLAX}(5)	Address Hold after End of ALE/$\overline{ADV}$	Tosc − 40		ns
T_{WLWH}	$\overline{WR}$ Pulse Width	2Tosc − 35	2Tosc + 40	ns
T_{QVWH}	Output Data Valid to End of $\overline{WR}$	3Tosc − 60		ns
T_{WHQX}	Output Data Hold after $\overline{WR}$	Tosc − 50		ns
T_{WHLH}	End of $\overline{WR}$ to ALE/$\overline{ADV}$ High	Tosc − 75		ns
T_{RLRH}	$\overline{RD}$ Pulse Width	3Tosc − 30		ns
T_{RHLH}	End of $\overline{RD}$ to ALE/$\overline{ADV}$ High	Tosc − 45		ns
T_{RHBX}	$\overline{RD}$ High to A8–A15 Inactive	Tosc − 25	Tosc + 30	ns
T_{WHBX}	$\overline{WR}$ High to A8–A15 Inactive	Tosc − 25	Tosc + 100	ns
T_{LLWL}	ALE/$\overline{ADV}$ Low to $\overline{WR}$ Low	2Tosc − 30	2Tosc + 55	ns
T_{QVWL}	Output Data Valid to $\overline{WR}$ Low	Tosc − 30		ns

NOTES:
2. If more than one wait state is desired, add 3Tosc for each additional wait state.
3. Max spec applies only to ALE. Min spec applies to both ALE and $\overline{ADV}$.
4. The term "Address Valid" applies to AD0–AD7, A8–A15.
5. The term " Address" in this definition applies to AD0–AD7.

WAVEFORM DIAGRAM



NOTE:
1. When $\overline{ADV}$ selected.

A.C. CHARACTERISTICS—SERIAL PORT—SHIFT REGISTER MODE

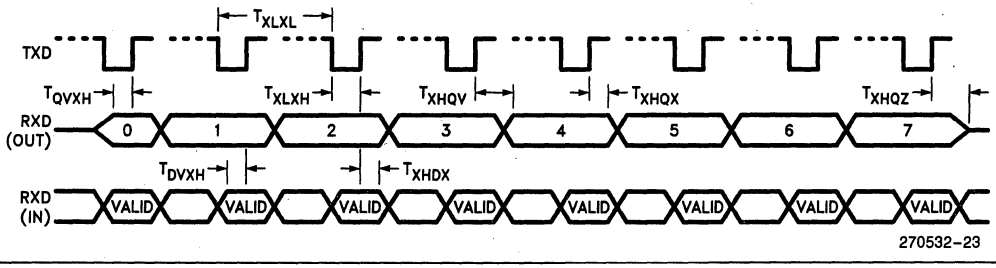
SERIAL PORT TIMING—SHIFT REGISTER MODE

Test Conditions: T_A = 0°C to +70°C; V_{CC} = 5V ±10%; V_{SS} = 0V; Load Capacitance = 80 pF

Symbol	Parameter	Min	Max	Units
T_{XLXL}	Serial Port Clock Period	$8T_{OSC}$		ns
T_{XLXH}	Serial Port Clock Falling Edge to Rising Edge	$4T_{OSC} - 50$	$4T_{OSC} + 50$	ns
T_{QVXH}	Output Data Setup to Clock Rising Edge	$3T_{OSC}$		ns
T_{XHQX}	Output Data Hold After Clock Rising Edge	$2T_{OSC} - 50$		ns
T_{XHQV}	Next Output Data Valid After Clock Rising Edge		$2T_{OSC} + 50$	ns
T_{DVXH}	Input Data Setup to Clock Rising Edge	$2T_{OSC} + 200$		ns
T_{XHDX}	Input Data Hold After Clock Rising Edge	0		ns
T_{XHQZ}	Last Clock Rising to Output Float		$5T_{OSC}$	ns

WAVEFORM—SERIAL PORT—SHIFT REGISTER MODE

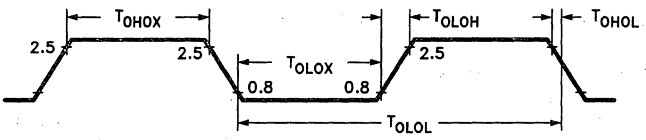
SERIAL PORT WAVEFORM—SHIFT REGISTER MODE



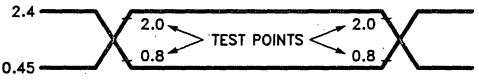
EXTERNAL CLOCK DRIVE

Symbol	Parameter	Min	Max	Units
$1/T_{OLOL}$	Oscillator Frequency	6	12	MHz
T_{OHOX}	High Time	25		ns
T_{OLOX}	Low Time	25		ns
T_{OLOH}	Rise Time		15	ns
T_{OHOL}	Fall Time		15	ns

EXTERNAL CLOCK DRIVE WAVEFORMS



A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

FLOAT WAVEFORM

For Timing Purposes a Port Pin is no Longer Floating when a 100 mV change from Load Voltage Occurs, and Begins to Float when a 100 mV change from the Loaded V_{OH}/V_{OL} Level occurs $I_{OL}/I_{OH} \geq \pm 15$ mA.

A/D CONVERTER SPECIFICATIONS

A/D Converter operation is verified only on the 8097BH, 8397BH, 8095BH, 8395BH, 8797BH, 8795BH, 8098, 8398.

The absolute conversion accuracy is dependent on the accuracy of V_{REF}. The specifications given below assume adherence to the Operating Conditions section of these data sheets. Testing is done at V_{REF} = 5.120V.

OPERATING CONDITIONS

V_{CC}, V_{PD}, V_{REF} ..... 4.5V to 5.5V
V_{SS}, ANGND 0.0V
T_A ..... 0°C to 70°C
F_{OSC} ..... 6.0 MHz to 12.0 MHz
Test Conditions:
V_{REF} 5.120V

Parameter	Typical*(1)	Minimum	Maximum	Units**	Notes
Resolution		1024 10	1024 10	Levels Bits	
Absolute Error		0	±4	LSBs	
Full Scale Error	−0.5 ±0.5			LSBs	
Zero Offset Error	±0.5			LSBs	
Non-Linearity		0	±4	LSBs	
Differential Non-Linearity		0	+2	LSBs	
Channel-to-Channel Matching		0	±1	LSBs	
Repeatability	±0.25			LSBs	1
Temperature Coefficients: Offset Full Scale Differential Non-Linearity	 0.009 0.009 0.009			 LSB/°C LSB/°C LSB/°C	 1 1 1
Off Isolation		−60		dB	1, 2, 4
Feedthrough	−60			dB	1, 2
V_{CC} Power Supply Rejection	−60			dB	1, 2
Input Resistance		1K	5K	Ω	1
D.C. Input Leakage		0	3.0	μA	
Sample Delay		$3T_{OSC}$ − 50	$3T_{OSC}$ + 50	ns	1, 3
Sample Time		$12T_{OSC}$ − 50	$12T_{OSC}$ + 50	ns	1
Sampling Capacitor			2	pF	

NOTES:
* These values are expected for most parts at 25°C.
** An "LSB", as used here, is defined in the glossary which follows and has a value of approximately 5 mV.
1. These values are not tested in production and are based on theoretical estimates and laboratory tests.
2. DC to 100 KHz.
3. For starting the A/D with an HSO Command.
4. Multiplexer Break-Before-Make Guaranteed.

A/D GLOSSARY OF TERMS

ABSOLUTE ERROR—The maximum difference between corresponding actual and ideal code transitions. Absolute Error accounts for all deviations of an actual converter from an ideal converter.

ACTUAL CHARACTERISTIC—The characteristic of an actual converter. The characteristic of a given converter may vary over temperature, supply voltage, and frequency conditions. An actual characteristic rarely has ideal first and last transition locations or ideal code widths. It may even vary over multiple conversions under the same conditions.

BREAK-BEFORE-MAKE—The property of a multiplexer which guarantees that a previously selected channel will be deselected before a new channel is selected. (e.g. the converter will not short inputs together.)

CHANNEL-TO-CHANNEL MATCHING—The difference between corresponding code transitions of actual characteristics taken from different channels under the same temperature, voltage and frequency conditions.

CHARACTERISTIC—A graph of input voltage versus the resultant output code for an A/D converter. It describes the transfer function of the A/D converter.

CODE—The digital value output by the converter.

CODE CENTER—The voltage corresponding to the midpoint between two adjacent code transitions.

CODE TRANSITION—The point at which the converter changes from an output code of Q, to a code of Q + 1. The input voltage corresponding to a code transition is defined to be that voltage which is equally likely to produce either of two adjacent codes.

CODE WIDTH—The voltage corresponding to the difference between two adjacent code transitions.

CROSSTALK—See "Off-Isolation".

D.C. INPUT LEAKAGE—Leakage current to ground from an analog input pin.

DIFFERENTIAL NON-LINEARITY—The difference between the ideal and actual code widths of the terminal based characteristic.

FEEDTHROUGH—Attenuation of a voltage applied on the selected channel of the A/D Converter after the sample window closes.

FULL SCALE ERROR—The difference between the expected and actual input voltage corresponding to the full scale code transition.

IDEAL CHARACTERISTIC—A characteristic with its first code transition at V_{IN} = 0.5 LSB, its last code transition at V_{IN} = (V_{REF} − 1.5 LSB) and all code widths equal to one LSB.

INPUT RESISTANCE—The effective series resistance from the analog input pin to the sample capacitor.

LSB—Least Significant Bit: The voltage corresponding to the full scale voltage divided by 2^n, where n is the number of bits of resolution of the converter. For a 10-bit converter with a reference voltage of 5.12V, one LSB is 5.0 mV. Note that this is different than digital LSBs, since an uncertainty of two LSB, when referring to an A/D converter, equals 10 mV. (This has been confused with an uncertainty of two digital bits, which would mean four counts, or 20 mV.)

MONOTONIC—The property of successive approximation converters which guarantees that increasing input voltages produce adjacent codes of increasing value, and that decreasing input voltages produce adjacent codes of decreasing value.

NO MISSED CODES—For each and every output code, there exists a unique input voltage range which produces that code only.

NON-LINEARITY—The maximum deviation of code transitions of the terminal-based characteristic from the corresponding code transitions of the ideal characteristic.

OFF-ISOLATION—Attenuation of a voltage applied on a deselected channel of the A/D converter. (Also referred to as Crosstalk.)

REPEATABILITY—The difference between corresponding code transitions from different actual characteristics taken from the same converter on the same channel at the same temperature, voltage and frequency conditions.

RESOLUTION—The number of input voltage levels that the converter can unambiguously distinguish between. Also defines the number of useful bits of information which the converter can return.

SAMPLE DELAY—The delay from receiving the start conversion signal to when the sample window opens.

SAMPLE DELAY UNCERTAINTY—The variation in the sample delay.

SAMPLE TIME—The time that the sample window is open.

SAMPLE TIME UNCERTAINTY—The variation in the sample time.

SAMPLE WINDOW—Begins when the sample capacitor is attached to a selected channel and ends when the sample capacitor is disconnected from the selected channel.

SUCCESSIVE APPROXIMATION—An A/D conversion method which uses a binary search to arrive at the best digital representation of an analog input.

TEMPERATURE COEFFICIENTS—Change in the stated variable per degree centigrade temperature change. Temperature coefficients are added to the typical values of a specification to see the effect of temperature drift.

TERMINAL BASED CHARACTERISTIC—An actual characteristic which has been rotated and translated to remove zero offset and full scale error.

V_{CC} REJECTION—Attenuation of noise on the V_{CC} line to the A/D converter.

ZERO OFFSET—The difference between the expected and actual input voltage corresponding to the first code transition.

The RUPI™-44 Family

18

THE RUPI™-44 FAMILY: MICROCONTROLLER WITH ON-CHIP COMMUNICATION CONTROLLER

INTRODUCTION

The RUPI-44 family is designed for applications requiring local intelligence at remote nodes, and communication capability among these distributed nodes. The RUPI-44 integrates onto a single chip Intel's highest performance microcontroller, the 8051-core, with an intelligent and high performance Serial communication controller, called the Serial Interface Unit, or SIU. See Figure 1. This dual controller architecture allows complex control and high speed data communication functions to be realized cost effectively.

The RUPI-44 family consists of three pin compatible parts:

- 8344—8051 Microcontroller with SIU
- 8044—An 8344 with 4K bytes of on-chip ROM program memory
- 8744—An 8344 with 4K bytes of on-chip EPROM program memory

1.0 ARCHITECTURE OVERVIEW

The 8044's dual controller architecture enables the RUPI to perform complex control tasks and high speed communication in a distributed network environment.

The 8044 microcontroller is the 8051-core, and maintains complete software compatibility with it. The microcontroller contains a powerful CPU with on-chip peripherals, making it capable of serving sophisticated real-time control applications such as instrumentation, industrial control, and intelligent computer peripherals. The microcontroller features on-chip peripherals such as two 16-bit timer/counters and 5 source interrupt capability with programmable priority levels. The microcontroller's high performance CPU executes most instructions in 1 microsecond, and can perform an 8 × 8 multiply in 4 microseconds. The CPU features a Boolean processor that can perform operations on 256 directly addressable bits. 192 bytes of on-chip data RAM can be extended to 64K bytes externally. 4K bytes of on-chip program ROM can be extended to 64K bytes externally. The CPU and SIU run concurrently. See Figure 2.

The SIU is designed to perform serial communications with little or no CPU involvement. The SIU supports data rates up to 2.4 Mbps, externally clocked, and 375 Kbps self clocked (i.e., the data clock is recovered by an on-chip digital phase locked loop). SIU hardware supports the HDLC/SDLC protocol: zero bit insertion/deletion, address recognition, cyclic redundancy check, and frame number sequence check are automatically performed.

The SIU's Auto mode greatly reduces communication software overhead. The AUTO mode supports the SDLC Normal Response Mode, by performing secondary station responses in hardware without any CPU involvement. The Auto mode's interrupt control and frame sequence numbering capability eliminates software overhead normally required in conventional systems. By using the Auto mode, the CPU is free to concentrate on real time control of the application.

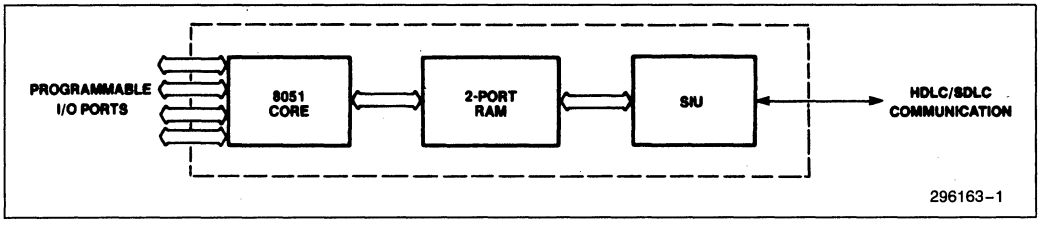


Figure 1. RUPI™-44 Dual Controller Architecture

Order Number: 296163-001

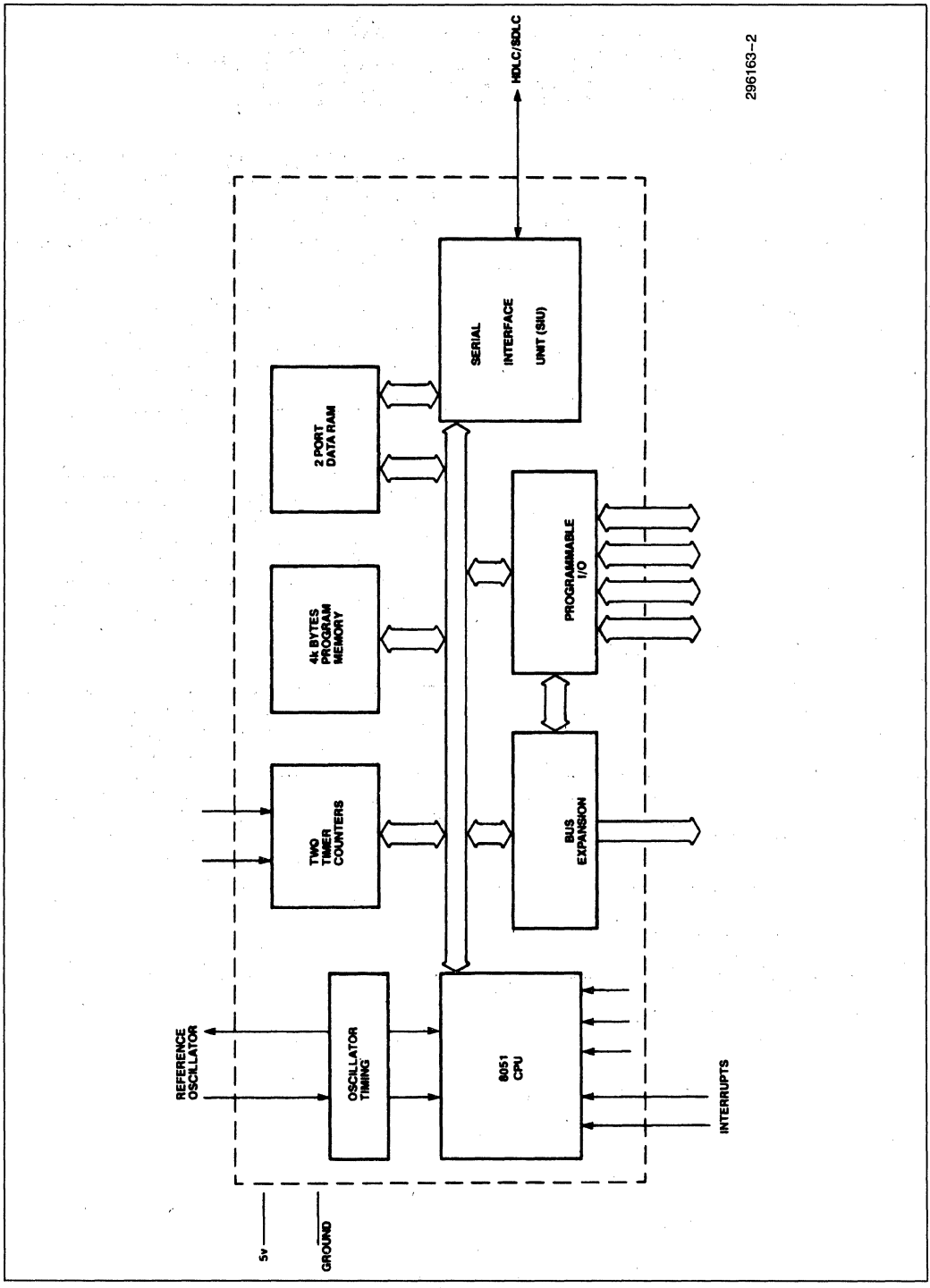


Figure 2. Simplified 8044 Block Diagram

2.0 THE HDLC/SDLC PROTOCOLS

2.1 HDLC/SDLC Advantages over Async

The High Level Data Link Control, HDLC, is a standard communication link control established by the International Standards Organization (ISO). SDLC is a subset of HDLC.

HDLC and SDLC are both well recognized standard serial protocols. The Synchronous Data Link Control, SDLC, is an IBM standard communication protocol. IBM originally developed SDLC to provide efficient, reliable and simple communication between terminals and computers.

The major advantages of SDLC/HDLC over Asynchronous communications protocol (Async):

- SIMPLE: Data Transparency
- EFFICIENT: Well Defined Message-Level Operation
- RELIABLE: Frame Check Sequence and Frame Numbering

The SDLC reduces system complexity. HDLC/SDLC are "data transparent" protocols. Data transparency means that an arbitrary data stream can be sent without concern that some of the data could be mistaken for a protocol controller. Data transparency relieves the communication controller having to detect special characters.

SDLC/HDLC provides more data throughout than Async. SDLC/HDLC runs at Message-level Operation which transmits multiple bytes within the frame, whereas Async is based on character-level operation. Async transmits or receives a character at a time. Since Async requires start and stop bits in every transmission, there is a considerable waste of overhead compared to SDLC/HDLC.

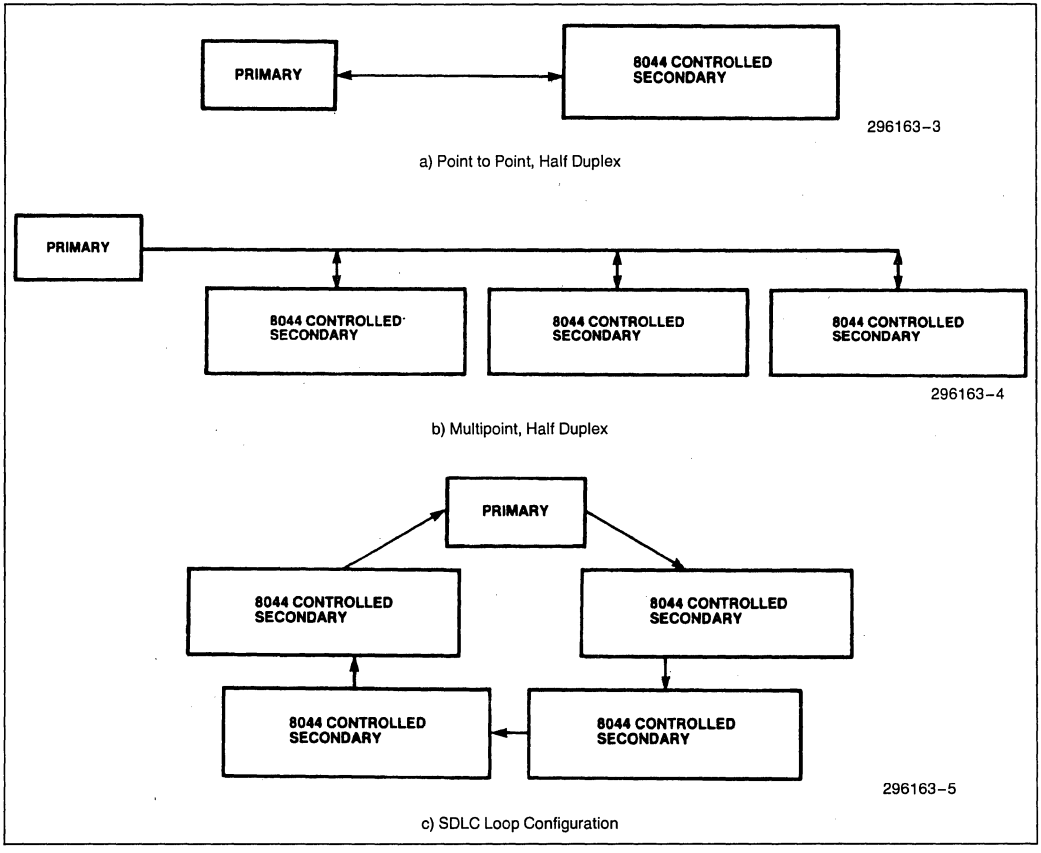


Figure 3. RUPI™-44 Supported Network Configurations

Due to SDLC/HDLC's well delineated field (see Figure 4) the CPU does not have to interpret character by character to determine control field and information field. In the case of Async, CPU must look at each character to interpret what it means. The practical advantage of such feature is straight forward use of DMA for information transfer.

In addition, SDLC/HDLC further improves Data throughput using implied Acknowledgement of transferred information. A station using SDLC/HDLC may acknowledge previously received information while transmitting different information in the same frame. In addition, up to 7 messages may be outstanding before an acknowledgement is required.

The HDLC/SDLC protocol can be used to realize reliable data links. Reliable Data transmission is ensured at the bit level by sending a frame check sequence, cyclic redudancy checking, within the frame. Reliable frame transmission is ensured by sending a frame number identification with each frame. This means that a receiver can sequentially count received frames and at any time infer what the number of the next frame to be received should be. More important, it provides a means for the receiver to identify to the sender some particular frame that it wishes to have resent because of errors.

2.2 HDLC/SDLC Networks

In both the HDLC and SDLC line protocols a (Master) primary station controls the overall network (data link) and issues commands to the secondary (Slave) stations. The latter complies with instructions and responds by sending appropriate responses. Whenever a transmitting station must end transmission prematurely, it sends an abort character. Upon detecting an abort character, a receiving station ignores the transmission block called a frame.

RUPI-44 supported HDLC/SDLC network configurations are point to point (half duplex) multipoint (half duplex), and loop. In the loop configuration the stations themselves act as repeaters, so that long links can be easily realized, see Figure 3.

2.3 Frames

An HDLC/SDLC frame consists of five basic fields: Flag, Address, Control, Data and Error Detection. A frame is bounded by flags—opening and closing flags. An address field is 8 bits wide in SDLC, extendable to 2 or more bytes in HDLC. The control field is also 8 bits wide, extendable to two bytes in HDLC. The SDLC data field or information field may be any number of bytes. The HDLC data field may or may not be on an 8 bit boundary. A powerful error detection code called Frame Check Sequence contains the calculated CRC (Cycle Redundancy Code) for all the bits between the flags. See Figure 4.

In HDLC and SDLC are three types of frames; an Information Frame is used to transfer data, a Supervisory Frame is used for control purposes, and a Nonsequenced Frame is used for initialization and control of the secondary stations.

For a more detailed discussion of higher level protocol functions interested readers may refer to the references listed in Section 2.6.

2.4 Zero Bit Insertion

In data communications, it is desirable to transmit data which can be of arbitrary content. Arbitrary data transmission requires that the data field cannot contain characters which are defined to assist the transmission protocol (like opening flag in HDLC/SDLC communications). This property is referred to as "data transparency". In HDLC/SDLC, this code transparency is made possible by Zero Bit Insertion (ZBI).

The flag has a unique bit pattern: 01111110 (7E HEX). To eliminate the possibility of the data field containing a 7E HEX pattern, a bit stuffing technique called Zero Bit Insertion is used. This technique specifies that during transmission, a binary 0 be inserted by the transmitter after any succession of five contiguous binary 1's. This will ensure that no pattern of 0 1 1 1 1 1 1 0 is ever transmitted between flags. On the receiving side, after receiving the flag, the receiver hardware automatically deletes any 0 following five consecutive 1's. The 8044 performs zero bit insertion and deletion automatically.

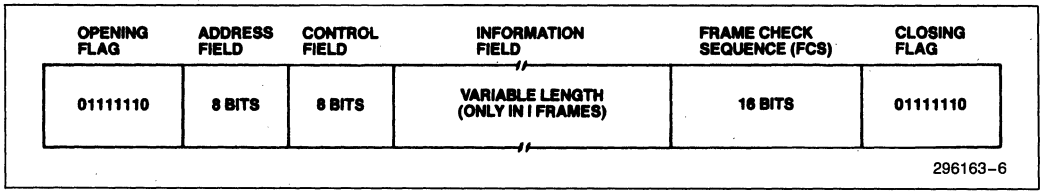


Figure 4. Frame Format

2.5 Non-return to Zero Inverted (NR21)

NRZI is a method of clock and data encoding that is well suited to the HDLC/SDLC protocol. It allows HDLC/SDLC protocols to be used with low cost asynchronous modems. NRZI coding is done at the transmitter to enable clock recovery from the data at the receiver terminal by using standard digital phase locked loop (DPLL) techniques. NRZI coding specifies that the signal condition does not change for transmitting a 1, while a 0 causes a change of state. NRZI coding ensures that an active data line will have a transition at least every 5-bit times (recall Zero Bit Insertion), while contiguous 0's will cause a change of state. Thus, ZBI and NRZI encoding makes it possible for the 8044's on-chip DPLL to recover a receive clock (from received data) synchronized to the received data and at the same time ensure data transparency.

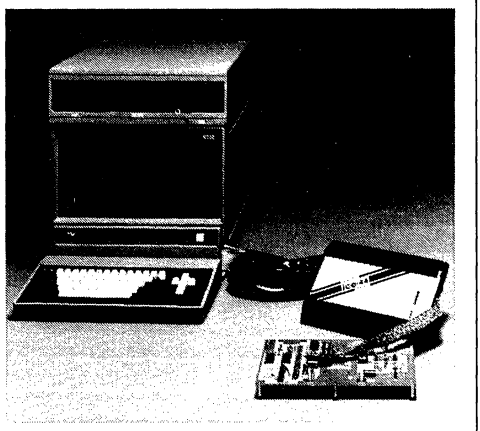
2.6 References

1. *IBM Synchronous Data Link Control General Information GA27-3093-2 File No. GENL-09.*
2. *Standard Network Access Protocol Specification, DATAPAC Trans-Canada Telephone System CCG111.*
3. *IBM 3650 Retail Store System Loop Interface OEM Information, IBM, GA27-3098-0.*
4. *Guidebook to Data Communications, Training Manual, Hewlett-Packard 5955-1715.*
5. *"Serial Backplane Suits Multiprocessor Architectures", Mike Webb, Computer Design, July 1984, pp. 85–96.*
6. *"Serial Bus Simplifies Distributed Control", P.D. MacWilliams, Control Engineering, June 1984, pp. 101–104.*
7. *"Chips Support Two Local Area Networks", Bob Dahlberg, Computer Design, May 1984, pp. 107–114.*
8. *"Build a VLSI-based Workstation for the Ethernet Environment", Mike Webb, EDN, 23 February 1984, pp. 297–307.*
9. *"Networking With the 8044", Young Sohn & Charles Gopen, Digital Design, May 1984, pp. 136–137.*

3.0 RUPI™-44 DESIGN SUPPORT

3.1 Design Tool Support

A critical design consideration is time to market. Intel provides a sophisticated set of design tools to speed hardware and software development time of 8044 based products. These include ICE-44, ASM-51, PL/M-51, and EMV-44.



296163-7

Figure 5. RUPI™-44 Development Support Configuration Intellec® System, ICE™-44 Buffer Box, and ICE-44 Module Plugged into a User Prototype Board

A primary tool is the 8044 In Circuit Emulator, called ICE-44. See Figure 5. In conjunction with Intel's Intellec® Microprocessor Development System, the ICE-44 emulator allows hardware and software development to proceed interactively. This approach is more effective than the traditional method of independent hardware and software development followed by system integration. With the ICE-44 module, prototype hardware can be added to the system as it is designed. Software and hardware integration occurs while the product is being developed.

The ICE-44 emulator assists four stages of development:

1) Software Debugging

   It can be operated without being connected to the user's system before any of the user's hardware is available. In this stage ICE-44 debugging capabilities can be used in conjunction with the Intellec text editor and 8044 macroassembler to facilitate program development.

2) Hardware Development

   The ICE-44 module's precise emulation characteristics and full-speed program RAM make it a valuable tool for debugging hardware, including the time-critical SDLC serial port, parallel port, and timer interfaces.

3) System Integration

Integration of software and hardware can begin when any functional element of the user system hardware is connected to the 8044 socket. As each section of the user's hardware is completed, it is added to the prototype. Thus, each section of the hardware and software is system tested in real-time operation as it becomes available.

4) System Test

When the user's prototype is complete, it is tested with the final version of the user system software. The ICE-44 module is then used for real-time emulation of the 8044 to debug the system as a completed unit.

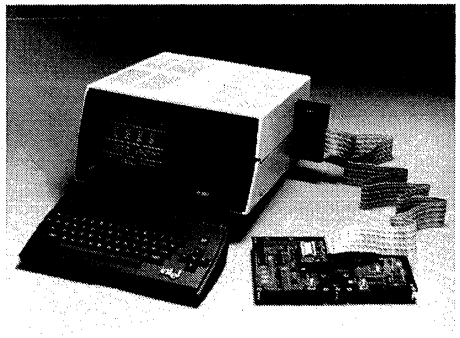
The final product verification test may be performed using the 8744 EPROM version of the 8044 microcomputer. Thus, the ICE-44 module provides the user with the ability to debug a prototype or production system at any stage in its development.

A conversion kit, ICE-44 CON, is available to upgrade an ICE-51 module to ICE-44.

Intel's ASM-51 Assembler supports the 8044 special function registers and assembly program development. PL/M-51 provides designers with a high level language for the 8044. Programming in PL/M can greatly reduce development time, and ensure quick time to market.

These tools have recently been expanded with the addition of the EMV-44CON. This conversion kit allows you to convert an EMV-51 into an EMV-44 emulation vehicle. The resultant low cost emulator is designed for use with an iPDS Personal Development System, which also supports the ASM-51 assembler and PL/M-51. See Figure 6.

Emulation support is similar to the ICE-44 with support for Software and Hardware Development, System Integration, and System Test. The iPDS's rugged portability and ease of use also make it an ideal system for production tests and field service of your finished design. In addition, the iPDS offers EPROM programming module for the 8744, and direct communications with the 8044-based BITBUS via an optional iSBX-344 distributed control module.



296163-8

Figure 6. RUPI-44 iPDS Personal Development System, EMV-44 Buffer Box, and EMV-44 Module Plugged into a User Prototype Board

3.2 8051 Workshop

Intel provides 8051 training to its customers through the 5-day 8051 workshop. Familiarity with the 8051 and 8044 is achieved through a combination of lecture and laboratory exercises.

For designers not familiar with the 8051, the workshop is an effective way to become proficient with the 8051 architecture and capabilities.

8044 Architecture

19

8044 ARCHITECTURE

GENERAL

The 8044 is based on the 8051 core. The 8044 replaces the 8051's serial port with an intelligent HDLC/SDLC controller called the Serial Interface or SIU. Thus the differences between the two result from the 8044's increased on-chip RAM (192 bytes) and additional special function registers necessary to control the SIU. Aside from the increased memory, the SIU itself, and differences in 5 pins (for the serial port), the 8044 and 8051 are compatible.

This chapter describes the differences between the 8044 and 8051. Information pertaining to the 8051 core, eg. instruction set, port operation, EPROM programming, etc. is located in the 8051 sections of this manual.

A block diagram of the 8044 is shown in Figure 1. The pinpoint is shown on the inside front cover.

1.0 MEMORY ORGANIZATION OVERVIEW

The 8044 maintains separate address spaces for Program Memory and Data Memory. The Program Memory can be up to 64K bytes long, of which the lowest 4K bytes are in the on-chip ROM.

If the $\overline{EA}$ pin is held high, the 8044 executes out of internal ROM unlwess the Program Counter exceeds 0FFFH. Fetches from locations 1000H through FFFFH are directed to external Program Memory.

If the $\overline{EA}$ pin is held low, the 8044 fetches all instructions from external Program Memory.

The Data Memory consists of 192 bytes of on-chip RAM, plus 35 Special Function Registers, in addition to which the device is capable of accessing up to 64K bytes of external data memory.

The Program Memory uses 16-bit addresses. The external Data Memory can use erither 8-bit or 16-bit addresses. The internal Data Memory uses 8-bit addresses, which provide a 256-location address space. The lower 192 addresses access the on-chip RAM. The Special Function Registers occupy various locations in the upper 128 bytes of the same address space.

The lowest 32 bytes in the internal RAM (locations 00 through 1FH) are divided into 4 banks of registers, each bank consisting of 8 bytes. Any one of these banks can be selected to be the "working registers" of the CPU, and can be accessed by a 3-bit address in the same byte as the opcode of an instruction. Thus, a large number of instructions are one-byte instructions.

The next higher 16 bytes of the internal RAM (locations 20H through 2FH) have individually addressable bits. These are provided for use as software flags or for one-bit (Boolean) processing. This bit-addressing capability is an important feature of the 8044. In addition to the 128 individually addressable bits in RAM, twelve of the Special Function Registers also have individually addressable bits.

A memory map is shown in Figure 2.

1.1 Special Function Registers

The Special Function Registers are as follows:

*	ACC	Accumulator (A Register)
*	B	B Register
*	PSW	Program Status Word
	SP	Stack Pointer
	DPTR	Data Pointer (consisting of DPH AND DPL)
*	P0	Port 0
*	P1	Port 1
*	P2	Port 2
*	P3	Port 3
*	IP	Interrupt Priority
*	IE	Interrupt Enable
	TMOD	Timer/Counter Mode
*	TCON	Timer/Counter Control
	TH0	Timer/Counter 0 (high byte)
	TL0	Timer/Counter 0 (low byte)
	TH1	Timer/Counter 1 (high byte)
	TL1	Timer/Counter 1 (low byte)
	SMD	Serial Mode
*	STS	Status/Command
*	NSNR	Send/Receive Count
	STAD	Station Address
	TBS	Transmit Buffer Start Address
	TBL	Transmit Buffer Length
	TCB	Transmit Control Byte
	RBS	Receive Buffer Start Address
	RBL	Receive Buffer Length
	RFL	Received Field Length
	RCB	Received Control Byte
	DMA CNT	DMA Count
	FIFO	FIFO (three bytes)
	SIUST	SIU State Counter
	PCON	Power Control

The registers marked with * are both byte- and bit-addressable.

Order Number: 296164-001

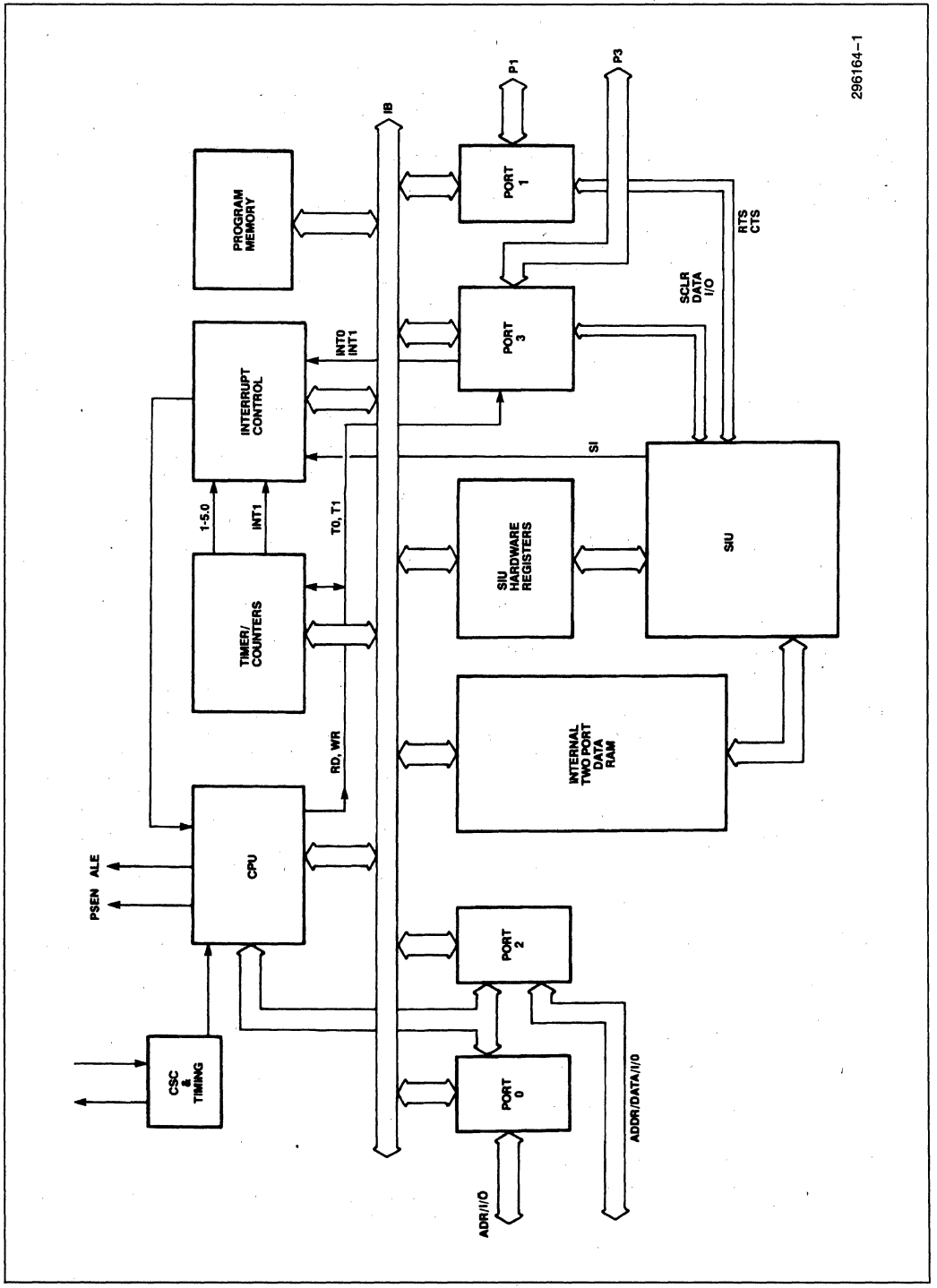


Figure 1. RUPI™ Block Diagram

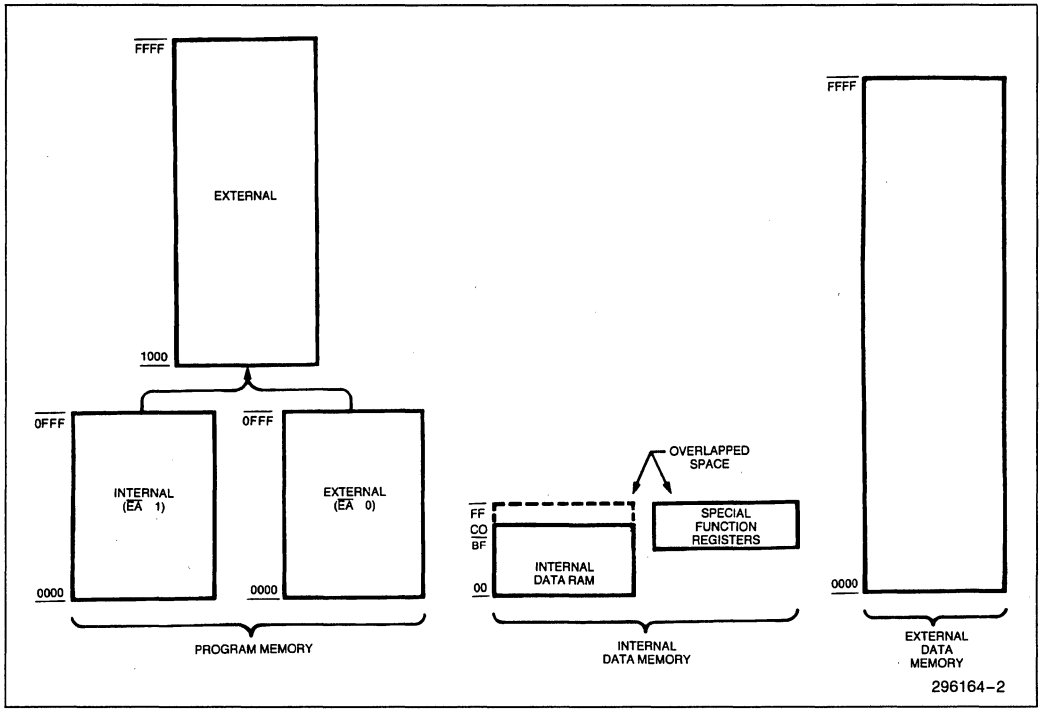


Figure 2. RUPI™-44 Memory Map

Stack Pointer

The Stack Pointer is 8 bits wide. The stack can reside anywhere in the 192 bytes of on-chip RAM. When the 8044 is reset, the stack pointer is initialized to 07H. When executing a PUSH or a CALL, the stack pointer is incremented before data is stored, so the stack would begin at location 08H.

1.2 Interrupt Control Registers

The Interrupt Request Flags are as listed below:

Source	*Request Flag*	*Location*
External Interrupt 0	$\overline{INT0}$, if IT0 = 0	P3.2
	IE0, if IT0 = 1	TCON.1
Timer 0 Overflow	TF0	TCON.5
External Interrupt 1	$\overline{INT1}$, if IT1 = 0	P3.3
	IE1, if IT1 = 1	TCON.3
Timer 1 Overflow	TF1	TCON.7
Serial Interface Unit	SI	STS.4

External Interrupt control bits IT0 and IT1 are in TCON.0 and TCON.2, respectively. Reset leaves all flags inactive, with IT0 and IT1 cleared.

All the interrupt flags can be set or cleared by software, with the same effect as by hardware.

The Enable and Priority Control Registers are shown below. All of these control bits are set or cleared by software. All are cleared by reset.

IE: Interrupt Enable Register (bit-addressable)

Bit:	7	6	5	4	3	2	1	0
	EA	X	X	ES	ET1	EX1	ET0	EX0

where:

- EA disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
- ES enables or disables the Serial Interface Unit interrupt. If ES = 0, the Serial Interface Unit interrupt is disabled.
- ET1 enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.

- EX1 enables or disables External Interrupt 1. If EX1 = 0, External Interrupt 1 is disabled.
- ET0 enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.

IP: Interrupt Priority Register (bit-addressable)

Bit:	7	6	5	4	3	2	1	0
	X	X	X	PS	PT1	PX1	PT0	PX0

where:

- PS defines the Serial Interface Unit interrupt priority level. PS = 1 programs it to the higher priority level.
- PT1 defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
- PX1 defines the External Interrupt priority level. PX1 = 1 programs it to the higher priority level.
- PT0 defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.
- PX0 defines the External Interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

2.0 MEMORY ORGANIZATION DETAILS

In the 8044 family the memory is organized over three address spaces and the program counter. The memory spaces shown in Figure 2 are the:

- 64K-byte Program Memory address space
- 64K-byte External Data Memory address space
- 320-byte Internal Data Memory address space

The 16-bit Program Counter register provides the 8044 with its 64K addressing capabilities. The Program Counter allows the user to execute calls and branches to any location within the Program Memory space. There are no instructions that permit program execution to move from the Program Memory space to any of the data memory spaces.

In the 8044 and 8744 the lower 4K of the 64K Program Memory address space is filled by internal ROM and EPROM, respectively. By tying the $\overline{EA}$ pin high, the processor can be forced to fetch from the internal ROM/EPROM for Program Memory addresses 0 through 4K. Bus expansion for accessing Program Memory beyond 4K is automatic since external instruction fetches occur automatically when the Program Counter increases above 4095. If the $\overline{EA}$ pin is tied low all Program Memory fetches are from external memory. The execution speed of the 8044 is the same regardless of whether fetches are from internal or external Program Memory. If all program storage is on-chip, byte location 4095 should be left vacant to prevent an undesired prefetch from external Program Memory address 4096.

Certain locations in Program Memory are reserved for specific programs. Locations 0000 through 0002 are reserved for the initialization program. Following reset, the CPU always begins execution at location 0000. Locations 0003 through 0042 are reserved for the five interrupt-request service programs. Each resource that can request an interrupt requires that its service program be stored at its reserved location.

The 64K-byte External Data Memory address space is automatically accessed when the MOVX instruction is executed.

Functionally the Internal Data Memory is the most flexible of the address spaces. The Internal Data Memory space is subdivided into a 256-byte Internal Data RAM address space and a 128-byte Special Function Register address space as shown in Figure 3.

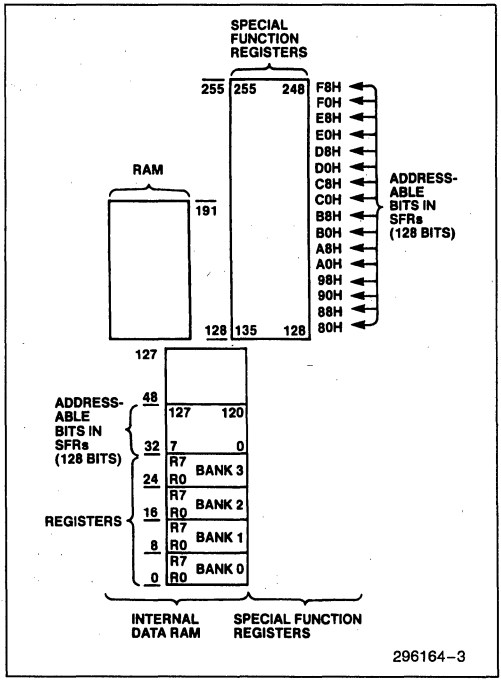


Figure 3. Internal Data Memory Address Space

The Internal Data RAM address space is 0 to 255. Four 8-Register Banks occupy locations 0 through 31. The stack can be located anywhere in the Internal Data RAM address space. In addition, 128 bit locations of the on-chip RAM are accessible through Direct Addressing. These bits reside in Internal Data RAM at byte locations 32 through 47. Currently locations 0 through 191 of the Internal Data RAM address space are filled with on-chip RAM.

The stack depth is limited only by the available Internal Data RAM, thanks to an 8-bit reloadable Stack Pointer. The stack is used for storing the Program Counter during subroutine calls and may be used for passing parameters. Any byte of Internal Data RAM or Special Function Register accessible though Direct Addressing can be pushed/popped.

The Special Function Register address space is 128 to 255. All registers except the Program Counter and the four 8-Register Banks reside here. Memory mapping the Special Function Registers allows them to be accessed as easily as internal RAM. As such, they can be operated on by most instructions. In the overlapping memory space (address 128-191), indirect addressing is used to access RAM, and direct addressing is used to access the SFR's. The SFR's at addresses 192–255 are also accessed using direct addressing. The Special Function Registers are listed in Figure 4. Their mapping in the Special Function Register address space is shown in Figures 5 and 6.

Performing a read from a location of the Internal Data memory where neither a byte of Internal Data RAM (i.e., RAM addresses 192-255) nor a Special Function Register exists will access data of indeterminable value.

Architecturally, each memory space is a linear sequence of 8-bit wide bytes. By Intel convention the storage of multi-byte address and data operands in program and data memories is the least significant byte at the low-order address and the most significant byte at the high-order address. Within byte X, the most significant bit is represented by X.7 while the least significant bit is X.0. Any deviation from these conventions will be explicitly stated in the text.

ARITHMETIC REGISTERS:
Accumulator*, B register*,
Program Status Word*
POINTERS:
Stack Pointer, Data Pointer (high & low)
PARALLEL I/O PORTS:
Port 3*, Port 2*, Port 1*, Port 0*
INTERRUPT SYSTEM:
Interrupt Priority Control*,
Interrupt Enable Control*
TIMERS:
Timer Mode, Timer Control*, Timer 1 (high & low), Timer 0 (high & low)
SERIAL INTERFACE UNIT:
Transmit Buffer Start,
Transmit Buffer Length,
Transmit Control Byte,
Send Count Receive Count*,
DMA Count,
Station Address
Receive Field Length
Receive Buffer Start
Receive Buffer Length
Receive Control Byte,
Serial Mode,
Status Register.*

***Bits in these registers are bit addressable.**

Figure 4. Special Function Registers

2.1 Operand Addressing

There are five methods of addressing source operands. They are Register Addressing, Direct Addressing, Register-Indirect Addressing, Immediate Addressing

ARITHMETIC REGISTERS:
Accumulator*, B register*,
Program Status Word*
POINTERS:
Stack Pointer, Data Pointer (high & low)
PARALLEL I/O PORTS:
Port 3*, Port 2*, Port 1*, Port 0*
INTERRUPT SYSTEM:
Interrupt Priority Control*,
Interrupt Enable Control*
TIMERS:
Timer Mode, Timer Control*, Timer 1 (high & low), Timer 0 (high & low)
SERIAL INTERFACE UNIT:
Serial Mode, Status/Command*,
Send/Receive Count*, Station Address,
Transmit Buffer Start Address,
Transmit Buffer Length,
Transmit Control Byte,
Receive Buffer Start Address,
Receive Buffer Length,
Receive Field Length,
Receive Control Byte,
DMA Count,
FIFO (three bytes),
SIU Controller State Counter

***Bits in these registers are bit-addressable.**

Figure 5. Mapping of Special Function Registers

and Base-Register-plus Index-Register-Indirect Addressing. The first three of these methods can also be used to address a destination operand. Since operations in the 8044 require 0 (NOP only), 1, 2, 3 or 4 operands, these five addressing methods are used in combinations to provide the 8044 with its 21 addressing modes.

Most instructions have a "destination, source" field that specifies the data type, addressing methods and operands involved. For operations other than moves, the destination operand is also a source operand. For example, in "subtract-with-borrow A,#5" the A register receives the result of the value in register A minus 5, minus C.

Most operations involve operands that are located in Internal Data Memory. The selection of the Program Memory space or External Data Memory space for a second operand is determined by the operation mnemonic unless it is an immediate operand. The subset of the Internal Data Memory being addressed is determined by the addressing method and address value. For example, the Special Function Registers can be accessed only through Direct Addressing with an address of 128-255. A summary of the operand addressing methods is shown in Figure 6. The following paragraphs describe the five addressing methods.

2.2 Register Addressing

Register Addressing permits access to the eight registers (R7-R0) of the selected Register Bank (RB). One of the four 8-Register Banks is selected by a two-bit field in the PSW. The registers may also be accessed through Direct Addressing and Register-Indirect Addressing, since the four Register Banks are mapped into the lowest 32 bytes of internal Data RAM as shown in Figures 9 and 10. Other Internal Data Memory locations that are addressed as registers are A, B, C, AB and DPTR.

2.3 Direct Addressing

Direct Addressing provides the only means of accessing the memory-mapped byte-wide Special Function Registers and memory mapped bits within the Special Function Registers and Internal Data RAM. Direct Addressing of bytes may also be used to access the lower 128 bytes of Internal Data RAM. Direct Addressing of bits gains access to a 128 bit subset of the Special Function Registers as shown in Figures 5, 6, 9, and 10.

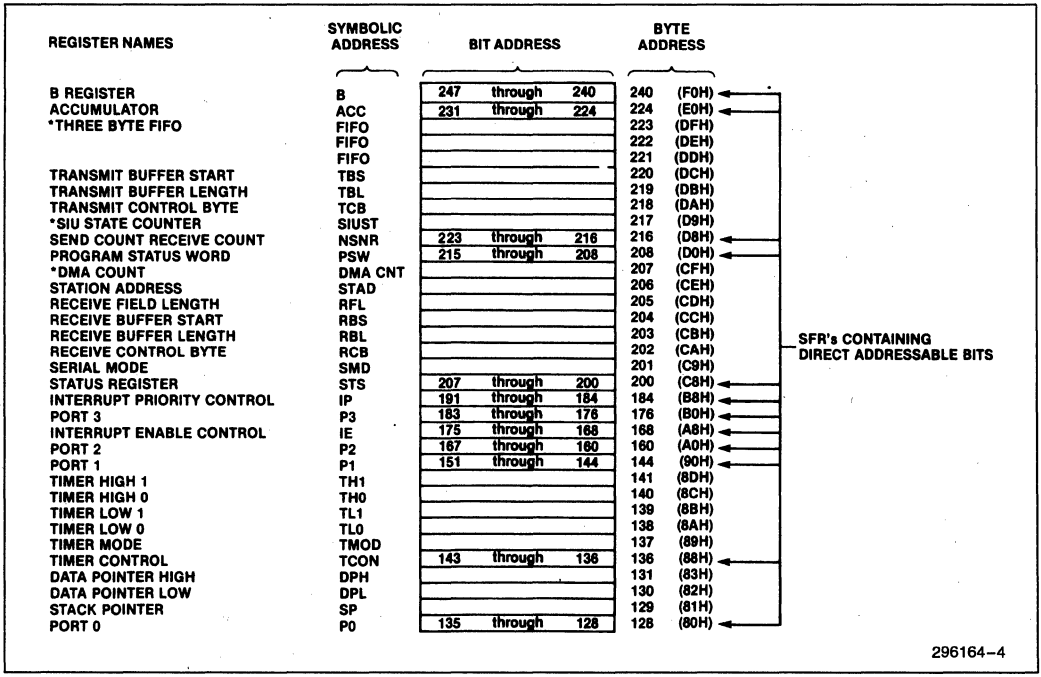


Figure 6. Mapping of Special Function Registers

Direct Byte Address	Bit Address (MSB)							(LSB)	Hardware Register Symbol
240	F7	F6	F5	F4	F3	F2	F1	F0	8
224	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	NS2	NS1	NS0	SES	NR2	NR1	NR0	SER	
216	DF	DE	DD	DC	DB	DA	D9	D8	NSNR
	CY	AC	FO	RS1	RSO	OV		P	
204	D7	D6	D5	D4	D3	D2	D1	D0	PSW
	TBF	RE	RTS	SI	BV	CPB	AM	RBP	
200	CF	CE	CD	CC	CB	CA	C9	C8	STS
				PS	PT1	PX1	PT0	PX0	
184	—	—	—	BC	BB	BA	B9	B8	1P
176	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA			E5	ET1	EX1	ET0	EX0	
168	AF	—	—	AC	AB	AA	A9	A8	1E
160	A7	A6	A5	A4	A3	A2	A1	A0	P2
144	97	96	95	94	93	92	91	90	P1
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
136	8F	8E	8D	8C	8B	8A	89	88	TCON
128	87	86	85	84	83	82	81	80	P0

Figure 7. Special Function Register Bit Address

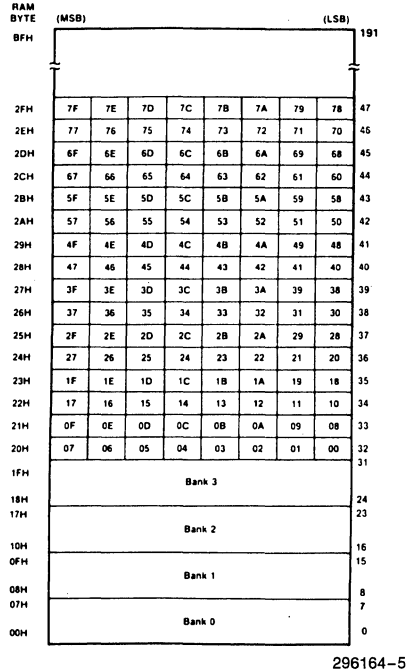


Figure 9. RAM Bit Address

- **Register Addressing**
  - — **R7-R0**
  - — **A, B, C (bit), AB (two bytes), DPTR (double byte)**
- **Direct Addressing**
  - — **Lower 128 bytes of Internal Data RAM**
  - — **Special Function Registers**
  - — **128 bits in subset of Special Function Register address space**
- **Register-Indirect Addressing**
  - — **Internal Data RAM [@R1, @R0, @SP (PUSH and POP only)]**
  - — **Least Significant Nibbles in Internal Data RAM (@R1, @R0)**
  - — **External Data Memory (@R1, @R0, @DPTR)**
- **Immediate Addressing**
  - — **Program Memory (in-code constant)**
- **Base-Register-plus Index-Register-Indirect Addressing**
  - — **Program Memory (@ DPTR + A, @ PC + A)**

Figure 8. Operand Addressing Methods

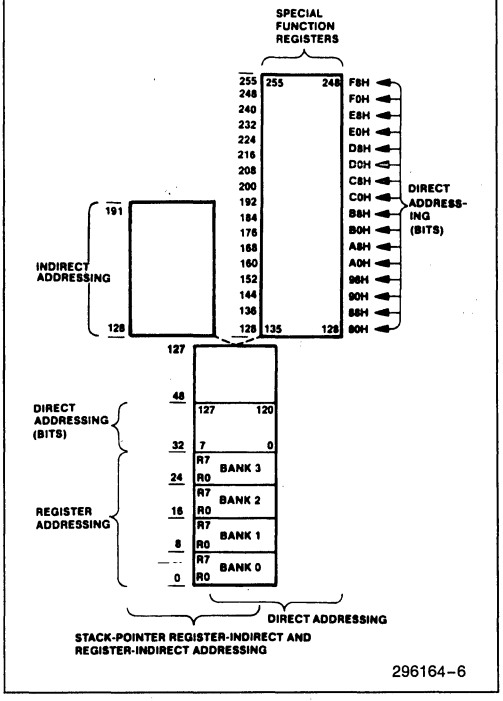


Figure 10. Addressing Operands in Internal Data Memory

Register-Indirect Addressing using the content of R1 or R0 in the selected Register Bank, or using the content of the Stack Pointer (PUSH and POP only), addresses the Internal Data RAM. Register-Indirect Addressing is also used for accessing the External Data Memory. In this case, either R1 or R0 in the selected Register Bank may be used for accessing locations within a 256-byte block. The block number can be preselected by the contents of a port. The 16-bit Data Pointer may be used for accessing any location within the full 64K external address space.

3.0 RESET

Reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods) *while the oscillator is running.* The CPU responds by executing an internal reset. It also configures the ALE and PSEN pins as inputs. (They are quasi-bidirectional.) The internal reset is executed during the second cycle in which RST is high and is repeated every cycle until RST goes low. It leaves the internal registers as follows:

Register	Content
PC	0000H
A	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	0FFH
IP	(XXX00000)
IE	(0XX00000)
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SMD	00H
STS	00H
NSNR	00H
STAD	00H
TBS	00H
TBL	00H
TCB	00H
RBS	00H
RBL	00H
RFL	00H
RCB	00H
DMA CNT	00H
FIFO1	00H
FIFO2	00H
FIFO3	00H
SIUST	01H
PCON	(0XXXXXXX)

The internal RAM is not affected by reset. When VCC is turned on, the RAM content is indeterminate unless VPD was applied prior to VCC being turned off (see Power Down Operation.)

4.0 RUPI™-44 FAMILY PIN DESCRIPTION

VSS: Circuit ground potential.

VCC: Supply voltage during programming (of the 8744), verification (of the 8044 or 8744), and normal operation.

Port 0: Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus during accessses to external memory (during which accesses it activates internal pullups). It also outputs instruction bytes during program verification. (External pullups are required during program verification.) Port 0 can sink eight LS TTL inputs.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. It receives the low-order address byte during program verification in the 8044 or 8744. Port 1 can sink/source four LS TTL inputs, It can drive MOS inputs without external pullups.

Two of the Port 1 pins serve alternate functions, as listed below:

Port Pin	*Alternate Function*
P1.6	$\overline{RTS}$ (Request to Send). In a non-loop configuration, $\overline{RTS}$ signals that the 8044 is ready to transmit data.

8044 Serial Interface

20

THE RUPI™-44 SERIAL INTERFACE UNIT

SERIAL INTERFACE

The serial interface provides a high-performance communication link. The protocol used for this communication is based on the IBM Synchronous Data Link Control (SDLC). The serial interface also supports a subset of the ISO HDLC (International Standards Organization High-Level Data Link Control) protocol.

The SDLC/HDLC protocols have been accepted as standard protocols for many high-level teleprocessing systems. The serial interface performs many of the functions required to service the data link without intervention from the 8044's own CPU. The programmer is free to concentrate on the 8044's function as a peripheral controller, rather than having to deal with the details of the communication process.

Five pins on the 8044 are involved with the serial interface:

Pin 7	$\overline{RTS}$/P16
Pin 8	$\overline{CTS}$/P17
Pin 10	I/$\overline{O}$/RXD/P30
Pin 11	DATA/TXD/P31
Pin 15	SCLK/T1//P35

Figure 1 is a functional block diagram of the serial interface unit (SIU). More details on the SIU hardware are given later in this chapter.

1.0 DATA LINK CONFIGURATIONS

The serial interface is capable of operating in three serial data link configurations:

1) Half-Duplex, point-to-point
2) Half-Duplex, multipoint (with a half-duplex or full-duplex primary)
3) Loop

Figure 2 shows these three configurations. The RTS (Request to Send) and CTS (Clear to Send) hand-shaking signals are available in the point-to-point and multipoint configurations.

2.0 DATA CLOCKING OPTIONS

The serial interface can operate in an externally clocked mode or in a self clocked mode.

Externally Clocked Mode

In the externally clocked mode, a common Serial Data Clock (SCLK on pin 15) synchronizes the serial bit stream. This clock signal may come from the master CPU or primary station, or from an external phase-locked loop local to the 8044. Figure 3 illustrates the timing relationships for the serial interface signals when the externally clocked mode is used in point-to-point and multipoint data link configurations.

Incoming data is sampled at the rising edge of SCLK, and outgoing data is shifted out at the falling edge of SCLK. More detailed timing information is given in the 8044 data sheet.

Self Clocked (Asynchronous) Mode

The self clocked mode allows data transfer without a common system data clock. Using an on-chip DPLL (digital phase locked loop) the serial interface recovers the data clock from the data stream itself. The DPLL requires a reference clock equal to either 16 times or 32 times the data rate. This reference clock may be externally supplied or internally generated. When the serial interface generates this clock internally, it uses either the 8044's internal logic clock (half the crystal frequency's PH2) or the "timer 1" overflow. Figure 4 shows the serial interface signal timing relationships for the loop configuration, when the unclocked mode is used.

The DPLL monitors the received data in order to derive a data clock that is centered on the received bits. Centering is achieved by detecting all transitions of the received data, and then adjusting the clock transition (in increments of $1/16$ bit period) toward the center of the received bit. The DPLL converges to the nominal bit center within eight bit transitions, worst case.

To aid in the phase locked loop capture process, the 8044 has a NRZI (non-return-to-zero inverted) data encoding and decoding option. NRZI coding specifies that a signal does not change state for a transmitted binary 1, but does change state for a binary 0. Using the NRZI coding with zero-bit insertion, it can be guaranteed that an active signal line undergoes a transition at least every six bit times.

3.0 DATA RATES

The maximum data rate in the externally clocked mode is 2.4M bits per second (bps) a half-duplex configuration, and 1.0M in a loop configuration.

Order Number: 296165-001

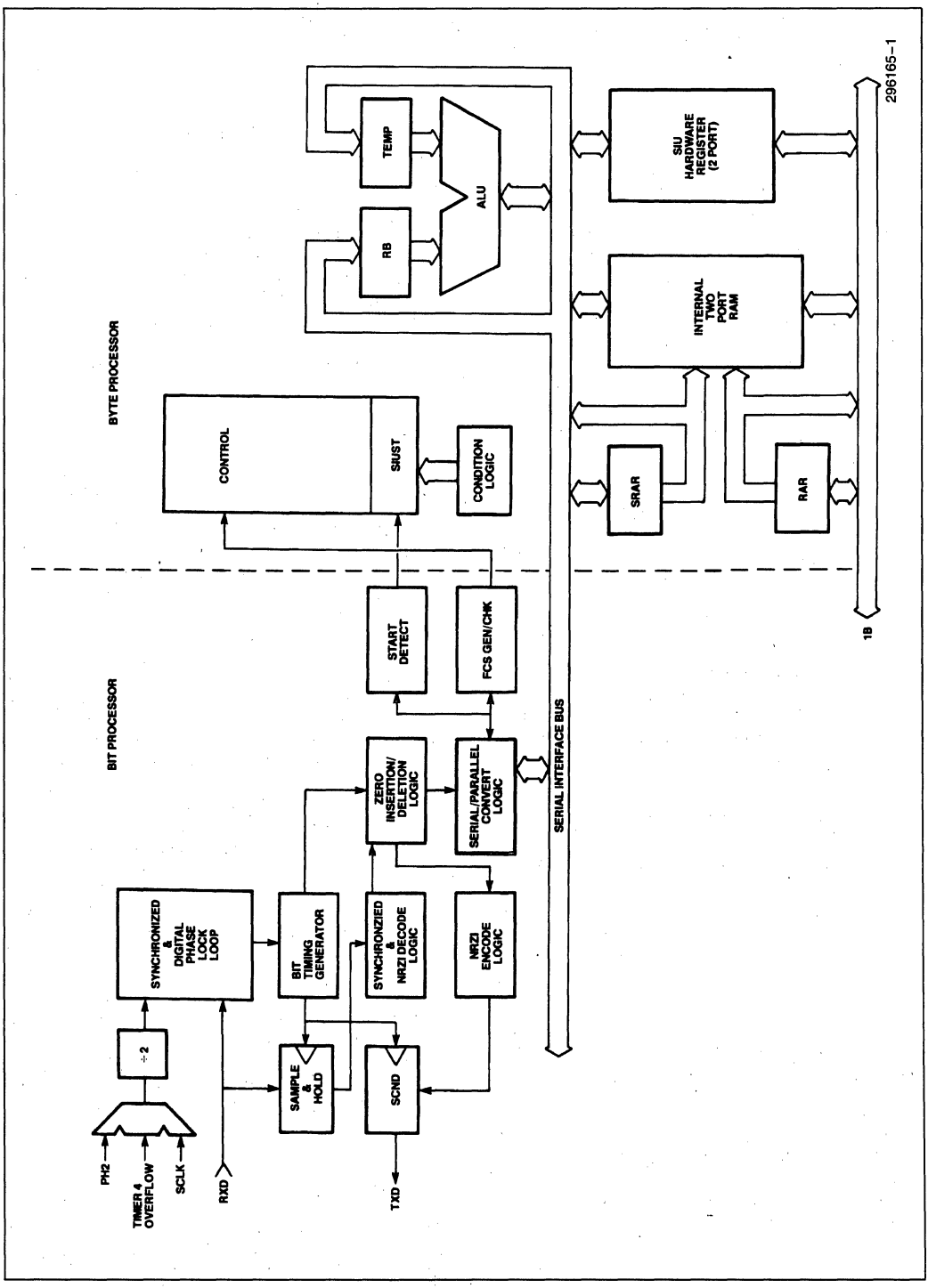


Figure 1. SIU Block Diagram

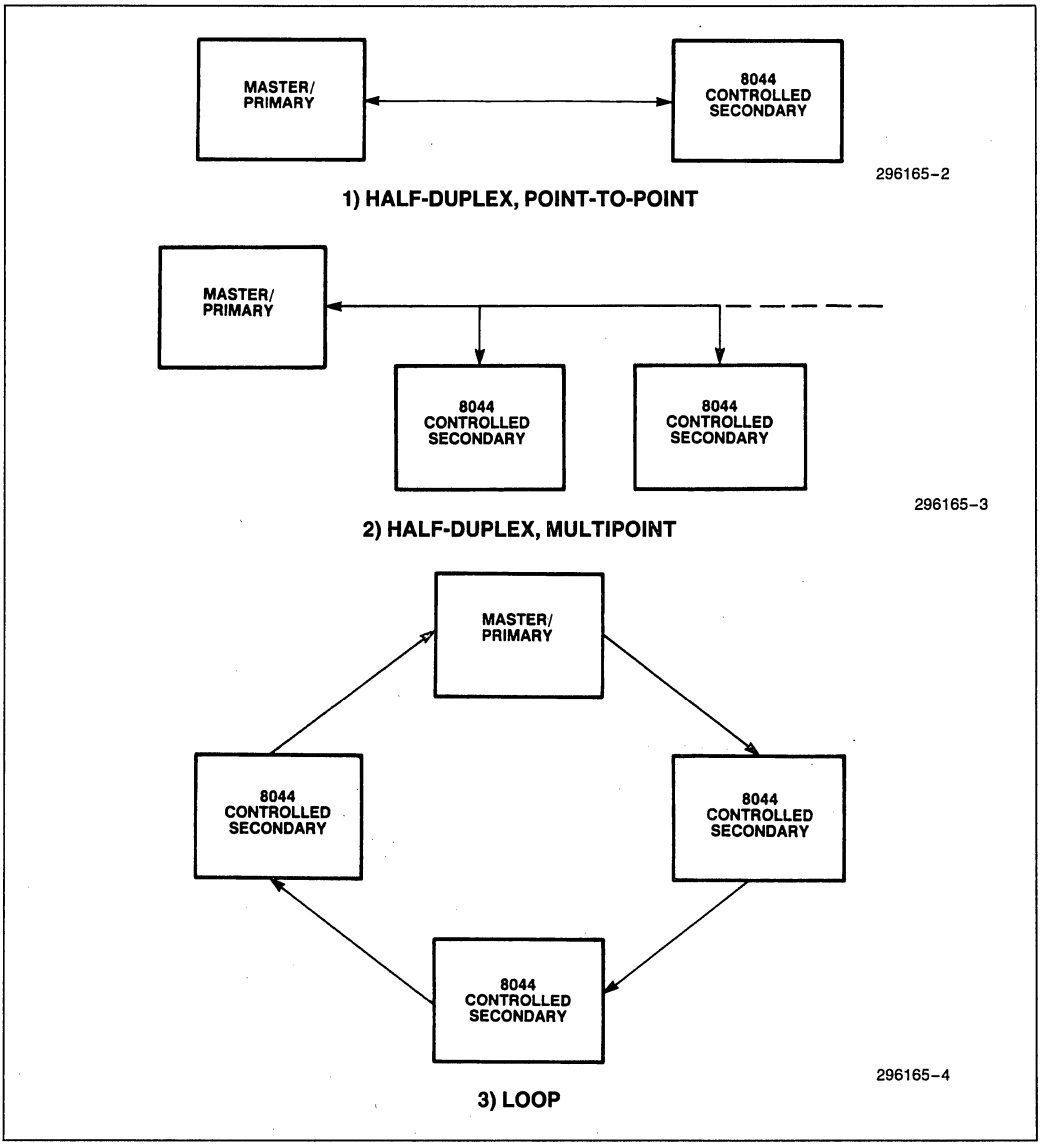


Figure 2. RUPI-44 Data Link Configurations

In the self clocked mode with an external reference clock, the maximum data rate is 375K bps.

In the self clocked mode with an internally generated reference clock, and the 8044 operating with a 12 MHz crystal, the available data rates are 244 bps to 62.5K bps, 187.5K bps and 375K bps.

For more details see the table in the SMD register description, below.

4.0 OPERATIONAL MODES

The Serial Interface Unit (SIU) can operate in either of two response modes:

1) AUTO mode
2) FLEXIBLE (NON-AUTO) mode

In the AUTO mode, the SIU performs in hardware a subset of the SDLC protocol called the normal response mode. The AUTO mode enables the SIU to recognize and respond to certain kinds of SDLC frames without intervention from the 8044's CPU. AUTO mode provides a faster turnaround time and a simplified software interface, whereas NON-AUTO mode provides a greater flexibility with regard to the kinds of operation permitted.

In AUTO mode, the 8044 can act only as a normal response mode secondary station—that is, it can transmit only when instructed to do so by the primary station. All such AUTO mode responses adhere strictly to IBM's SDLC definitions.

In the FLEXIBLE mode, reception or transmission of each frame by the SIU is performed under the control of the CPU. In this mode the 8044 can be either a primary station or a secondary station.

In both AUTO and FLEXIBLE modes, short frames, aborted frames, or frames which have had CRC's are ignored by the SIU.

The basic format of an SDLC frame is as follows:

Flag	Address	Control	Information	FCS	Flag

Format variations consist of omitting one or more of the fields in the SDLC frame. For example, a supervisory frame is formed by omitting the information field. Supervisory frames are used to confirm received frames, indicate ready or busy conditions, and to report errors. More details on frame formats are given in the SDLC Frame Format Options section, below.

4.1 AUTO Mode

To enable the SIU to receive a frame in AUTO mode, the 8044 CPU sets up a receive buffer. This is done by writing two registers—Receive Buffer Start (RBS) Address and Receive Buffer Length (RBL).

The SIU receives the frame, examines the control byte, and takes the appropriate action. If the frame is an information frame, the SIU will load the receive buffer, interrupt the CPU (to have the receive buffer read), and make the required acknowledgement to the primary station. Details on these processes are given in the Operation section, below.

In addition to receiving the information frames, the SIU in AUTO mode is capable of responding to the following commands (found in the control field of supervisory frames) from the primary station:

RR (Receive Ready): Acknowledges that the Primary station has correctly received numbered frames up through $N_R - 1$, and that it is ready to receive frame N_R.

RNR (Receive Not Ready): Indicates a temporary busy condition (at the primary station) due to buffering or other internal constraints. The quantity N_R in the control field indicates the number of the frame expected after the busy condition ends, and may be used to acknowledge the correct reception of the frames up through $N_R - 1$.

REJ (Reject): Acknowledges the correct reception of frames up through $N_R - 1$, and requests transmission or retransmission starting at frame N_R. The 8044 is capable of retransmitting at most the previous frame, and then only if it is still available in the transmit buffer.

UP (Unnumbered Poll): Also called NSP (Non-Sequenced Poll) or ORP (Optional Response Poll). This command is used in the loop configuration.

To enable the SIU to transmit an information frame in AUTO mode, the CPU sets up a transmit buffer. This is done by writing two registers—Transmit Buffer Start (TBS) Address and Transmit Buffer Length (TBL), and filling the transmit buffer with the information to be transmitted.

When the transmit buffer is full, the SIU can automatically (without CPU intervention) send an information frame (I-frame) with the appropriate sequence numbers, when the data link becomes available (when the 8044 is polled for information). After the SIU has transmitted the I-frame, it waits for acknowledgement from the receiving station. If the acknowledgement is

negative, the SIU retransmits the frame. If the acknowledgement is positive, the SIU interrupts the CPU, to indicate that the transmit buffer may be reloaded with new information.

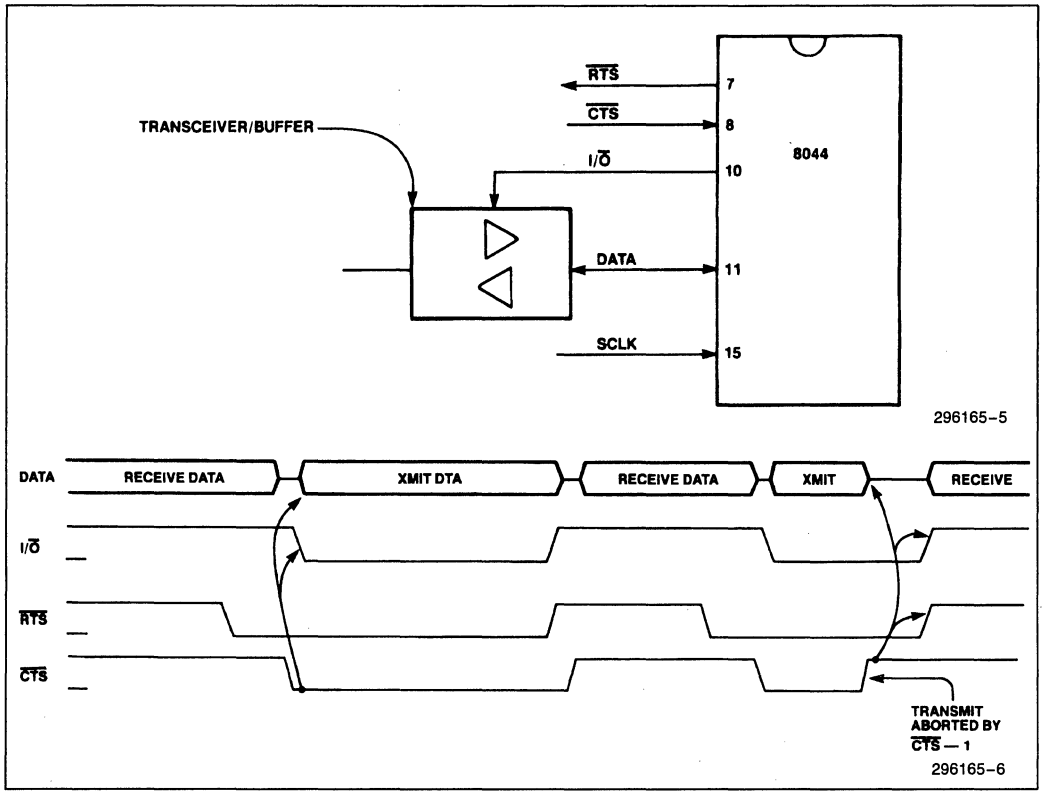


Figure 3. Serial Interface Timing—Clocked Mode

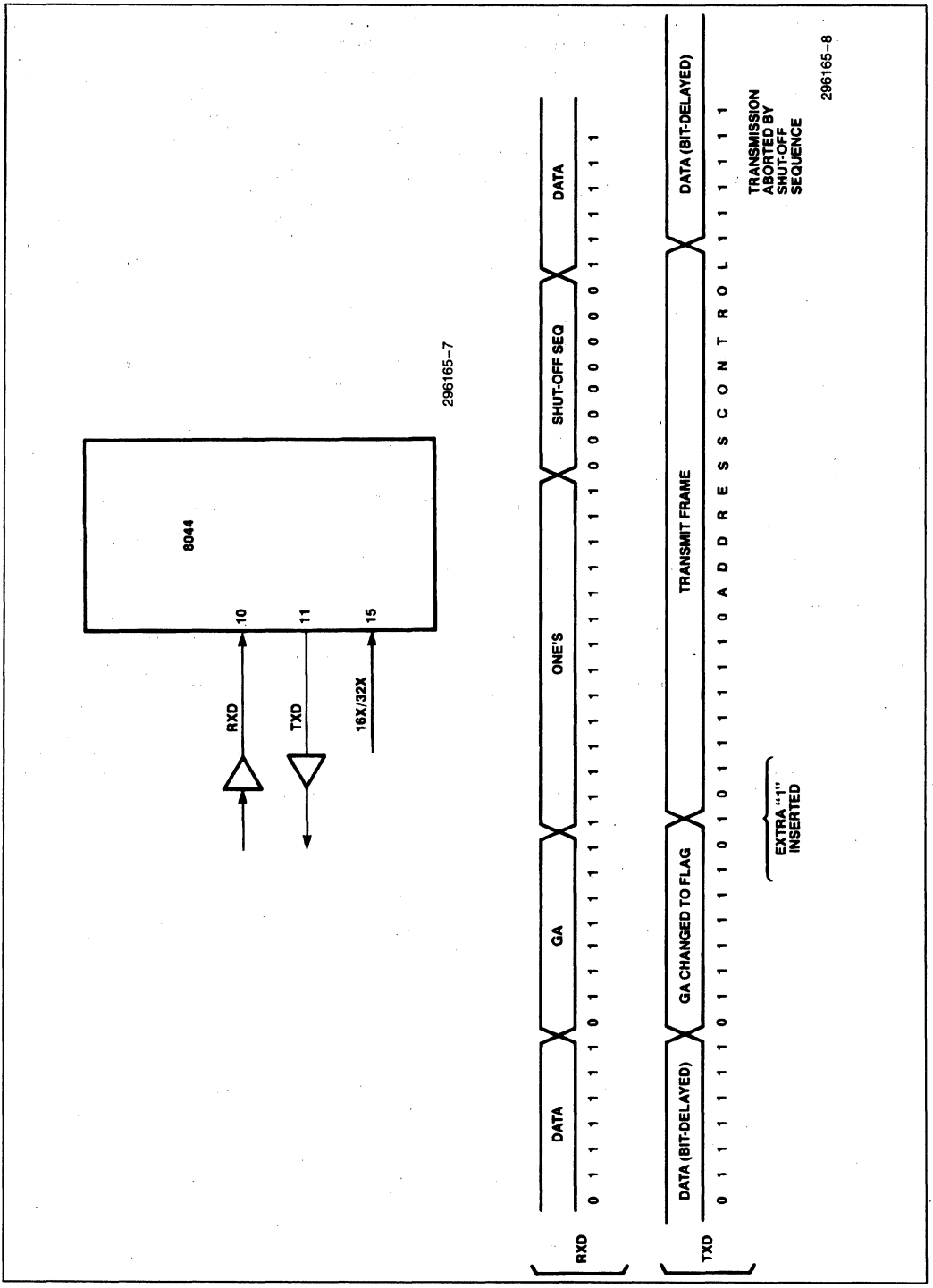


Figure 4. Serial Interface Timing—Self Clocked Mode

In addition to transmitting the information frames, the SIU in AUTO mode is capable of sending the following responses to the primary station:

RR (Receive Ready): Acknowledges that the 8044 has correctly received numbered frames up through $N_R - 1$, and that it is ready to receive frame N_R.

RNR (Receive Not Ready): Indicates a temporary busy condition (at the 8044) due to buffering or other internal constraints. The quantity N_R in the control field indicates the number of the frame expected after the busy condition ends, and acknowledges the correct reception of the frames up through $N_R - 1$.

4.2 FLEXIBLE Mode

In the FLEXIBLE (or non-auto) mode, all reception and transmission is under the control of the CPU. The full SDLC and HDLC protocols can be implemented, as well as any bit-synchronous variants of these protocols.

FLEXIBLE mode provides more flexibility than AUTO mode, but it requires more CPU overhead, and much longer recognition and response times. This is especially true when the CPU is servicing an interrupt that has higher priority than the interrupts from the SIU.

In FLEXIBLE mode, when the SIU receives a frame, it interrupts the CPU. The CPU then reads the control byte from the Receive Control Byte (RCB) register. If the received frame is an information frame, the CPU also reads the information from the receive buffer, according to the values in the Receive Buffer Start (RBS) address register and the Received Field Length (RFL) register.

In FLEXIBLE mode, the 8044 can initiate transmissions without being polled, and thus it can act as the primary station. To initiate transmission or to generate a response, the CPU sets up and enables the SIU. The SIU then formats and transmits the desired frame. Upon completion of the transmission, without waiting for a positive acknowledgement from the receiving station, the SIU interrupts the CPU.

5.0 8044 FRAME FORMAT OPTIONS

As mentioned above, variations on the basic SDLC frame consist of omitting one or more of the fields. The choice of which fields to omit, as well as the selection of AUTO mode versus FLEXIBLE mode, is specified by the settings of the following three bits in the Serial Mode Register (SMD) and the Status/Control Register (STS):

SMD Bit 0: NFCS (No Frame Check Sequence)

SMD Bit 1: NB (Non-Buffered Mode—No Control Field)

STS Bit 1: AM (AUTO Mode or Addressed Mode)

Figure 5 shows how these three bits control the frame format.

The following paragraphs discuss some properties of the standard SDLC format, and the significance of omitting some of the fields.

5.1 Standard SDLC Format

The standard SDLC format consists of an opening flag, an 8-bit address field, an 8-bit control field, an n-byte information field, a 16-bit Frame Check Sequence (FCS), and a closing flag. The FCS is based on the CCITT-CRC polynominal ($X^{16} + X^{12} + X^5 + 1$). The address and control fields may not be extended. Within the 8044, the address field is held in the Station Address (STAD) register, and the control field is held in the Receive Control Byte (RCB) or Transmit Control Byte (TCB) register. The standard SDLC format may be used in either AUTO mode or FLEXIBLE mode.

5.2 No Control Field (Non-Buffered Mode)

When the control field is not present, the RCB and TCB registers are not used. The information field begins immediately after the address field, or, if the address field is also absent, immediately after the opening flag. The entire information field is stored in the 8044's on-chip RAM. If there is no control field, FLEXIBLE mode must be used. Control information may, of course, be present in the information field, and in this manner the No Control Field option may be used for implementing extended control fields.

5.3 No Control Field and No Address Field

The No Address Field option is available only in conjuction with the No Control Field option. The STAD, RCB, and TCB registers are not used. When both these fields are absent, the information field begins immediately after the opening flag. The entire information field is stored in on-chip RAM. FLEXIBLE mode must be used. Formats without an address field have the following applications:

Point-to-point data links (where no addressing is necessary)

Monitoring line activity (receiving all messages regardless of the address field)

Extended addressing

FRAME OPTION	NFCS	NB	AM	FRAME FORMAT
Standard SDLC FLEXIBLE Mode	0	0	0	F A C I FCS F
Standard SDLC AUTO Mode	0	0	1	F A C I FCS F
No Control Field FLEXIBLE Mode	0	1	1	F A I FCS F
No Control Field No Address Field FLEXIBLE Mode	0	1	0	F I FCS F
No FCS Field FLEXIBLE Mode	1	0	0	F A C I F
No FCS Field AUTO Mode	1	0	1	F A C I F
No FCS Field No Control Field FLEXIBLE Mode	1	1	1	F A I F
No FCS Field No Control Field No Address Field FLEXIBLE Mode	1	1	0	F I F

Key to Abbreviations:

F = Flag (01111110)
A = Address Field
C = Control Field
I = Information Field
FCS = Frame Check Sequence

NOTE:
The AM bit is AUTO mode control bit when NB = 0, and Address Mode control bit when NB = 1.

Figure 5. Frame Format Options

5.4 No FCS Field

In the normal case (NFCS = 0), the last 16 bits before the closing flag are the Frame Check Sequence (FCS) field. These bits are not stored in the 8044's RAM. Rather, they are used to compute a cyclic redundancy check (CRC) on the data in the rest of the frame. A received frame with a CRC error (incorrect FCS) is ignored. In transmission, the FCS field is automatically computed by the SIU, and placed in the transmitted frame just prior to the closing flag.

The NFCS bit (SMD Bit 0) gives the user the capability of overriding this automatic feature. When this bit is set (NFCS = 1), all bits from the beginning of the information field to the beginning of the closing flag are treated as part of the information field, and are stored in the on-chip RAM. No FCS checking is done on the received frames, and no FCS is generated for the transmitted frames. The No FCS Field option may be used in conjunction with any of the other options. It is typically used in FLEXIBLE mode, although it does not strictly include AUTO mode. Use of the No FCS Field option AUTO Mode may, however, result in SDLC protocol violations, since the data integrity is not checked by the SIU.

Formats without an FCS field have the following applications:

Receiving and transmitting frames without verifying data integrity.

Using an alternate data verification algorithm.

Using an alternate CRC-16 polynomial (such as $X^{16} + X^{15} + X^2 + 1$), or a 32-bit CRC

Performing data link diagnosis by forcing false CRCs to test error detection mechanisms

In addition to the applications mentioned above, all of the format variations are useful in the support of non-standard bit-synchronous protocols.

6.0 HDLC

In addition to its support of SDLC communications, the 8044 also supports some of the capabilities of HDLC. The following remarks indicate the principal differences between SDLC and HDLC.

HDLC permits any number of bits in the information field, whereas SDLC requires a byte structure (multiple of 8 bits). The 8044 itself operates on byte boundaries, and thus it restricts fields to multiples of 8 bits.

HDLC provides functional extensions to SDLC: an unlimited address field is allowed, and extended frame number sequencing.

HDLC does not support operation in loop configurations.

7.0 SIU SPECIAL FUNCTION REGISTERS

The 8044 CPU communicates with and controls the SIU through hardware registers. These registers are accessed using direct addressing. The SIU special function registers (SIU SFRs) are of three types:

Control and Status Registers

Parameter Registers

ICE Support Registers

7.1 Control and Status Registers

There are three SIU Control and Status Registers:

Serial Mode Register (SMD)

Status/Command Register (STS)

Send/Receive Count Register (NSNR)

The SMD, STS, and NSNR registers are all cleared by system reset. This assures that the SIU will power up in an idle state (neither receiving nor transmitting).

These registers and their bit assignments are described below (see also the More Details on Registers section).

SMD: SERIAL MODE REGISTER (BYTE-ADDRESSABLE)

Bit:	7	6	5	4	3	2	1	0
	SCM2	SCM1	SCM0	NRZI	LOOP	PFS	NB	NFCS

The Serial Mode Register (Address C9H) selects the operational modes of the SIU. The 8044 CPU can both read and write SMD. The SIU can read SMD but cannot write to it. To prevent conflict between CPU and SIU access to SMD, the CPU should write SMD only when the Request To Send (RTS) and Receive Buffer Empty (RBE) bits (in the STS register) are both false (0). Normally, SMD is accessed only during initialization.

The individual bits of the Serial Mode Register are as follows:

Bit#	Name	Description
SMD.0	NFCS	No FCS field in the SDLC frame.
SMD.1	NB	Noon-Buffered mode. No control field in the SDLC frame.
SMD.2	PFS	Pre-Frame Sync mode. In this mode, the 8044 transmits two bytes before the first flag of a frame, for DPLL synchronization. If NRZI is enabled, 00H is sent; otherwise, 55H is sent. In either case, 16 pre-frame transitions are guaranteed.
SMD.3	LOOP	Loop configuration.
SMD.4	NRZI	NRZI coding option.
SMD.5	SCM0	Select Clock Mode—Bit 0
SMD.6	SCM1	Select Clock Mode—Bit 1
SMD.7	SCM2	Select Clock Mode—Bit 2

The SCM bits decode as follows:

SCM 2 1 0	Clock Mode	Data Rate (Bits/sec)*
0 0 0	Externally clocked	0–2.4M**
0 0 1	Undefined	
0 1 0	Self clocked, timer overflow	244–62.5K
0 1 1	Undefined	
1 0 0	Self clocked, external 16x	0–375K
1 0 1	Self clocked, external 32x	0–187.5K
1 1 0	Self clocked, internal fixed	375K
1 1 1	Self clocked, internal fixed	187.5K

*Based on a 12 MHz crystal frequency
**0–1M bps in loop configuration

STS: STATUS/COMMAND REGISTER (BIT-ADDRESSABLE)

Bit:	7	6	5	4	3	2	1	0
	TBF	RBE	RTS	SI	BOV	OPB	AM	RBP

The Status/Command Register (Address C8H) provides operational control of the SIU by the 8044 CPU, and enables the SIU to post status information for the CPU's access. The SIU can read STS, and can alter certain bits, as indicated below. The CPU can both read and write STS asynchronously. However, 2-cycle instructions that access STS during both cycles ('JBC/B, REL' and 'MOV /B,C.') should not be used, since the SIU may write to STS between the two CPU accesses.

The individual bits of the Status/Command Register are as follows:

Bit#	Name	Description
STS.0	RBP	Receive Buffer Protect. Inhibits writing of data into the receive buffer. In AUTO mode, RBP forces an RNR response instead of an RR.
STS.1	AM	AUTO Mode/Addressed Mode. Selects AUTO mode where AUTO mode is allowed. If NB is true, (= 1), the AM bit selects the addressed mode. AM may be cleared by the SIU.
STS.2	OPB	Optional Poll Bit. Determines whether the SIU will generate an AUTO response to an optional poll (UP with P = 0). OPB may be set or cleared by the SIU.
STS.3	BOV	Receive Buffer Overrun. BOV may be set or cleared by the SIU.
STS.4	SI	SIU Interrupt. This is one of the five interrupt sources to the CPU. The vector location = 23H. SI may be set by the SIU. It should be cleared by the CPU before returning from an interrupt routine.
STS.5	RTS	Request To Send. Indicates that the 8044 is ready to transmit or is transmitting. RTS may be read or written by the CPU. RTS may be read by the SIU, and in AUTO mode may be written by the SIU.
STS.6	RBE	Receive Buffer Empty. RBE can be thought of as Receive Enable. RBE is set to one by the CPU when it is ready to receive a frame, or has just read the buffer, and to zero by the SIU when a frame has been received.
STS.7	TBF	Transmit Buffer Full. Written by the CPU to indicate that it has filled the transmit buffer. TBF may be cleared by the SIU.

NSNR: SEND/RECEIVE COUNT REGISTER (BIT-ADDRESSABLE)

Bit:	7	6	5	4	3	2	1	0
	NS2	NS1	NS0	SES	NR2	NR1	NR0	SER

The Send/Receive Count Register (Address D8H) contains the transmit and receive sequence numbers, plus tally error indications. The SIU can both read and write NSNR. The 8044 CPU can both read and write NSNR asynchronously. However, 2-cycle instructions that access NSNR during both cycles ('JBC /B, REL', and 'MOV /B,C') should not be used, since the SIU may write to NSNR between the two 8044 CPU accesses.

The individual bits of the Send/Receive Count Register are as follows:

Bit#	Name	Description
NSNR.0	SER	Receive Sequence Error: NS (P) ≠ NR (S)
NSNR.1	NR0	Receive Sequence Counter—Bit 0
NSNR.2	NR1	Receive Sequence Counter—Bit 1
NSNR.3	NR2	Receive Sequence Counter—Bit 2
NSNR.4	SES	Send Sequence Error: NR (P) ≠ NS (S) and NR (P) ≠ NS (S) + 1
NSNR.5	NS0	Send Sequence Counter—Bit 0
NSNR.6	NS1	Send Sequence Counter—Bit 1
NSNR.7	NS2	Send Sequence Counter—Bit 2

7.2 Parameter Registers

There are eight parameter registers that are used in connection with SIU operation. All eight registers may be read or written by the 8044 CPU. RFL and RCB are normally loaded by the SIU.

The eight parameter registers are as follows:

STAD: STATION ADDRESS REGISTER (BYTE-ADDRESSABLE)

The Station Address register (Address CEH) contains the station address. To prevent access conflict, the CPU

should access STAD only when the SIU is idle (RTS = 0 and RBE = 0). Normally, STAD is accessed only during initialization.

TBS: TRANSMIT BUFFER START ADDRESS REGISTER (BYTE-ADDRESSABLE)

The Transmit Buffer Start address register (Address DCH) points to the location in on-chip RAM for the beginning of the I-field of the frame to be transmitted. The CPU should access TBS only when the SIU is not transmitting a frame (when TBF = 0).

TBL: TRANSMIT BUFFER LENGTH REGISTER (BYTE-ADDRESSABLE)

The Transmit Buffer Length register (Address DBH) contains the length (in bytes) of the I-field to be transmitted. A blank I-field (TBL = 0) is valid. The CPU should access TBL only when the SIU is not transmitting a frame (when TBF = 0).

NOTE:
The transmit and receive buffers are not allowed to "wrap around" in the on-chip RAM. A "buffer end" is automatically generated if address 191 (BFH) is reached.

TCB: TRANSMIT CONTROL BYTE REGISTER (BYTE-ADDRESSABLE)

The Transmit Control Byte register (Address DAH) contains the byte which is to be placed in the control field of the transmitted frame, during NON-AUTO mode transmission. The CPU should access TCB only when the SIU is not transmitting a frame (when TBF = 0). The N_S and N_R counters are not used in the NON-AUTO mode.

RBS: RECEIVE BUFFER START ADDRESS REGISTER (BYTE-ADDRESSABLE)

The Receive Buffer Start address register (Address CCH) points to the location in on-chip RAM where the beginning of the I-field of the frame being received is to be stored. The CPU should write RBS only when the SIU is not receiving a frame (when RBE = 0).

RBL: RECEIVE BUFFER LENGTH REGISTER (BYTE-ADDRESSABLE)

The Receive Buffer Length register (Address CBH) contains the length (in bytes) of the area in on-chip RAM allocated for the received I-field. RBL = 0 is valid. The CPU should write RBL only when RBE = 0.

RFL: RECEIVE FIELD LENGTH REGISTER (BYTE-ADDRESSABLE)

The Received Field Length register (Address CDH) contains the length (in bytes) of the received I-field that has just been loaded into on-chip RAM. RFL is loaded by the SIU. RFL = 0 is valid. RFL should be accessed by the CPU only when RBE = 0.

RCB: RECEIVE CONTROL BYTE REGISTER (BYTE-ADDRESSABLE)

The Received Control Byte register (Address CAH) contains the control field of the frame that has just been received. RCB is loaded by the SIU. The CPU can only read RCB, and should only access RCB when RBE = 0.

7.3 ICE Support Registers

The 8044 In-Circuit Emulator (ICE-44) allows the user to exercise the 8044 application system and monitor the execution of instructions in real time.

The emulator operates with Intel's Intellec® development system. The development system interfaces with the user's 8044 system through an in-cable buffer box. The cable terminates in a 8044 pin-compatible plug, which fits into the 8044 socket in the user's system. With the emulator plug in place, the user can exercise his system in real time while collecting up to 255 instruction cycles of real-time data. In addition, he can single-step the program.

Static RAM is available (in the in-cable buffer box) to emulate the 8044 internal and external program memory and external data memory. The designer can display and alter the contents of the replacement memory in the buffer box, the internal data memory, and the internal 8044 registers, including the SFRs.

Among the SIU SFRs are the following registers that support the operation of the ICE:

DMA CNT: DMA COUNT REGISTER (BYTE-ADDRESSABLE)

The DMA Count register (Address CFH) indicates the number of bytes remaining in the information block that is currently being used.

FIFO: THREE-BYTE (BYTE-ADDRESSABLE)

The Three-Byte FIFO (Address DDH, DEH, and DFH) is used between the eight-bit shift register and the information buffer when an information block is received.

SIUST: SIU STATE COUNTER (BYTE-ADDRESSABLE)

The SIU State Counter (Address D9H) reflects the state of the internal logic which is under SIU control. Therefore, care must be taken not to write into this register.

The SIUST register can serve as a helpful aid to determine which field of a receive frame that the SIU expects next. The table below will help in debugging 8044 reception problems.

SIUST Value	Function
01H	Waiting for opening flag.
08H	Waiting for address field.
10H	Waiting for control field.
18H	Waiting for first byte of I field. This state is only entered if a FCS is expected. It pushes the received byte onto the top of the FIFO.
20H	Waiting for second byte of I field. This state always follows state 18H.
28H	Waiting for I field byte. This state can be entered from state 20H or from states 01H, 08H, or 10H depending upon the SIU's mode configuration. (Each time a byte is received, it is pushed onto the top of the FIFO and the byte at the bottom is put into memory. For no FCS formatted frames, the FIFO is collapsed into a single register).
30H	Waiting for the closing flag after having overflowed the receive buffer. Note that even if the receive frame overflows the assigned receive buffer length, the FCS is still checked.

Examples of SIUST status sequences for different frame formats are shown below. Note that status changes after acceptance of the received field byte.

Table 1. SIUST Status Sequences

										Frame Option		
										NFCS	NB	AM
Example 1:												
Frame Format	(Idle)	F	A	C	I			FCS	F	0	0	1
SIUST Value	01	01	08	10	18	20	28	28	01			
Example 2:												
Frame Format	(Idle)	F	A	I			FCS	F		0	1	1
SIUST Value	01	01	08	18	20	28	28	01				
Example 3:												
Frame Format	(Idle)	F	I			FCS	F			0	1	0
SIUST Value	01	01	18	20	28	28	01					
Example 4:												
Frame Format	(Idle)	F	A	I	F					1	1	1
SIUST Value	01	01	08	28	01							
Example 5:												
Frame Format	(Idle)	F	I		F					1	1	0
SIUST Value	01	01	28		01							
Example 6:												
Frame Format	(Idle)	F	I			I OVERFLOW		FCS	F	0	1	0
SIUST Value	01	01	18	20	28	30		30	01			

8.0 OPERATION

The SIU is initialized by a reset signal (on pin 9), followed by write operations to the SIU SFRs. Once initialized, the SIU can function in AUTO mode or NON-AUTO mode. Details are given below.

8.1 Initialization

Figure 6 is the SIU. Registers SMD, STS, and NSNR are cleared by reset. This puts the 8044 into an idle state—neither receiving nor transmitting. The following registers must be initialized before the 8044 leaves the idle state:

STAD — to establish the 8044's SDLC station address.

SMD — To configure the 8044 for the proper operating mode.

RBS, RBL — to define the area in RAM allocated for the Receive Buffer.

TBS, TBL — to define the area in RAM allocated for the Transmit Buffer.

Once these registers have been initialized, the user may write to the STS register to enable the SIU to leave the idle state, and to begin transmits and/or receives.

Setting RBE to 1 enables the SIU for receive. When RBE = 1, the SIU monitors the received data stream for a flag pattern. When a flag pattern is found, the SIU enters Receive mode and receives the frame.

Setting RTS to 1 enables the SIU for transmit. When RTS = 1, the SIU monitors the received data stream for a GA pattern (loop configuration) or waits for a CTS (non-loop configuration). When the GA or CTS arrives, the SIU enters Transmit mode and transmits a frame.

In AUTO mode, the SIU sets RTS to enable automatic transmissions of appropriate responses.

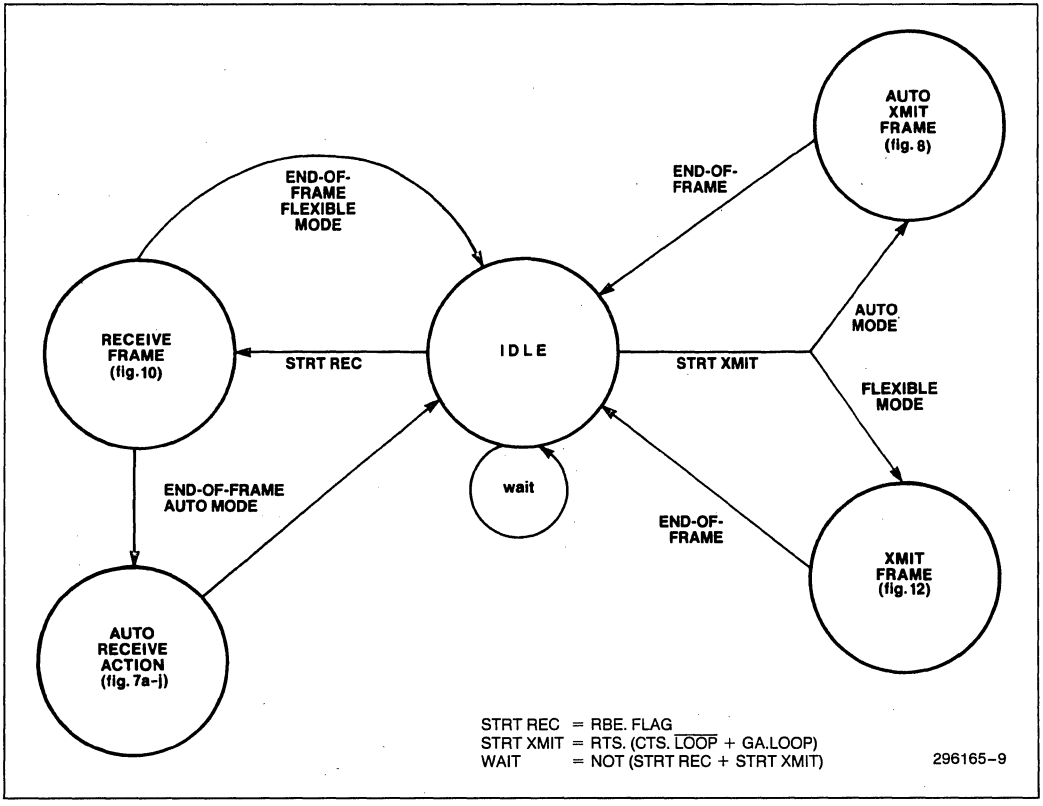


Figure 6. SIU State Diagram

8.2 AUTO Mode

Figure 7 illustrates the receive operations in AUTO mode. The overall operation is shown in Figure 7a. Particular cases are illustrated in Figures 7b through 7j. If any Unnumbered Command other than UP is received, the AM bit is cleared and the SIU responds as if in the FLEXIBLE mode, by interrupting the CPU for supervision. This will also happen if a BOV or SES condition occurs. If the received frame contains a poll, the SIU sets the RTS bit to generate a response.

Figure 8 illustrates the transmit operations in AUTO mode. When the SIU gets the opportunity to transmit, and if the transmit buffer is full, it sends an I-frame. Otherwise, it sends an RR if the buffer is free, or an RNR if the buffer is protected. The sequence counters NS and NR are used to construct the appropriate control fields.

Figure 9 shows how the CPU responds to an SI (serial interrupt) in AUTO mode. The CPU tests the AM bit (in the STS register). If AM = 1, it indicates that the SIU has received either an I-frame, or a positive response to a previously transmitted I-frame.

8.3 FLEXIBLE Mode

Figure 10 illustrates the receive operations in NON-AUTO mode. When the SIU successfully completes a task, it clears RBF and interrupts the CPU by setting SI to 1. The exact CPU response to SI is determined by software. A typical response is shown in Figure 11.

Figure 12 illustrates the transmit operations in FLEXIBLE mode. The SIU does not wait for a positive acknowledge response to the transmitted frame. Rather, it interrupts the CPU (by setting SI to 1) as soon as it finishes transmitting the frame. The exact CPU response to SI is determined by software. A typical response is shown in Figure 13. This response results in another transmit frame being set up. The sequence of operations shown in Figure 13 can also be initiated by the CPU, without an SI. Thus the CPU can initiate a transmission in FLEXIBLE mode without a poll, simply by setting the RTS bit in the STS register. The RTS bit is always used to initiate a transmission, but it is applied to the RTS pin only when a non-loop configuration is used.

8.4 8044 Data Link Particulars

The following facts should be noted:

1) In a non-loop configuration, one or two bits are transmitted before the opening flag. This is necessary for NRZI synchronization.
2) In a non-loop configuration, one to eight extra dribble bits are transmitted after the closing flag. These bits are a zero followed by ones.
3) In a loop configuration, when a GA is received and the 8044 begins transmitting, the sequence is 01111110101111110 ... (FLAG, 1, FLAG, ADDRESS, etc.). The first flag is created from the GA. The second flag begins the message.
4) CTS is sampled after the rising edge of the serial data, at about the center of the bit cell, except during a non-loop, externally clocked mode transmit, in which case it is sampled just after the falling edge.
5) The SIU does not check for illegal I-fields. In particular, if a supervisory command is received in AUTO mode, and if there is also an I-field, it will be loaded into the receive buffer (if RBP = 0), but it cannot cause a BOV.
6) In relation to the Receive Buffer Protect facility, the user should set RFL to 0 when clearing RBP, such that, if the SIU is in the process of receiving a frame, RFL will indicate the proper value when reception of the frame has been completed.

8.5 Turn Around Timing

In AUTO mode, the SIU generates an RTS immediately upon being polled. Assuming that the 8044 sends an information frame in response to the poll, the primary station sends back an acknowledgement. If, in this acknowledgement, the 8044 is polled again, a response may be generated even before the CPU gets around to processing the interrupt caused by the acknowledge. In such a case, the response would be an RR (or RNR), since TBF would have been set to 0 by the SIU, due to the acknowledge.

If the system designer does not wish to take up channel time with RR responses, but prefers to generate a new I-frame as a response, there are several ways to accomplish this:

1) Operate the 8044 in FLEXIBLE mode.
2) Specify that the master should never acknowledge and poll in one message. This is typically how a loop system operates, with the poll operation confined to the UP command. This leaves plenty of time for the 8044 to get its transmit buffer loaded with new information after an acknowledge.
3) The 8044 CPU can clear RTS. This will prevent a response from being sent, or abort it if it is already in progress. A system using external RTS/CTS handshaking could use a one-shot delay RTS or CTS, thereby giving the CPU more time to disable the response.

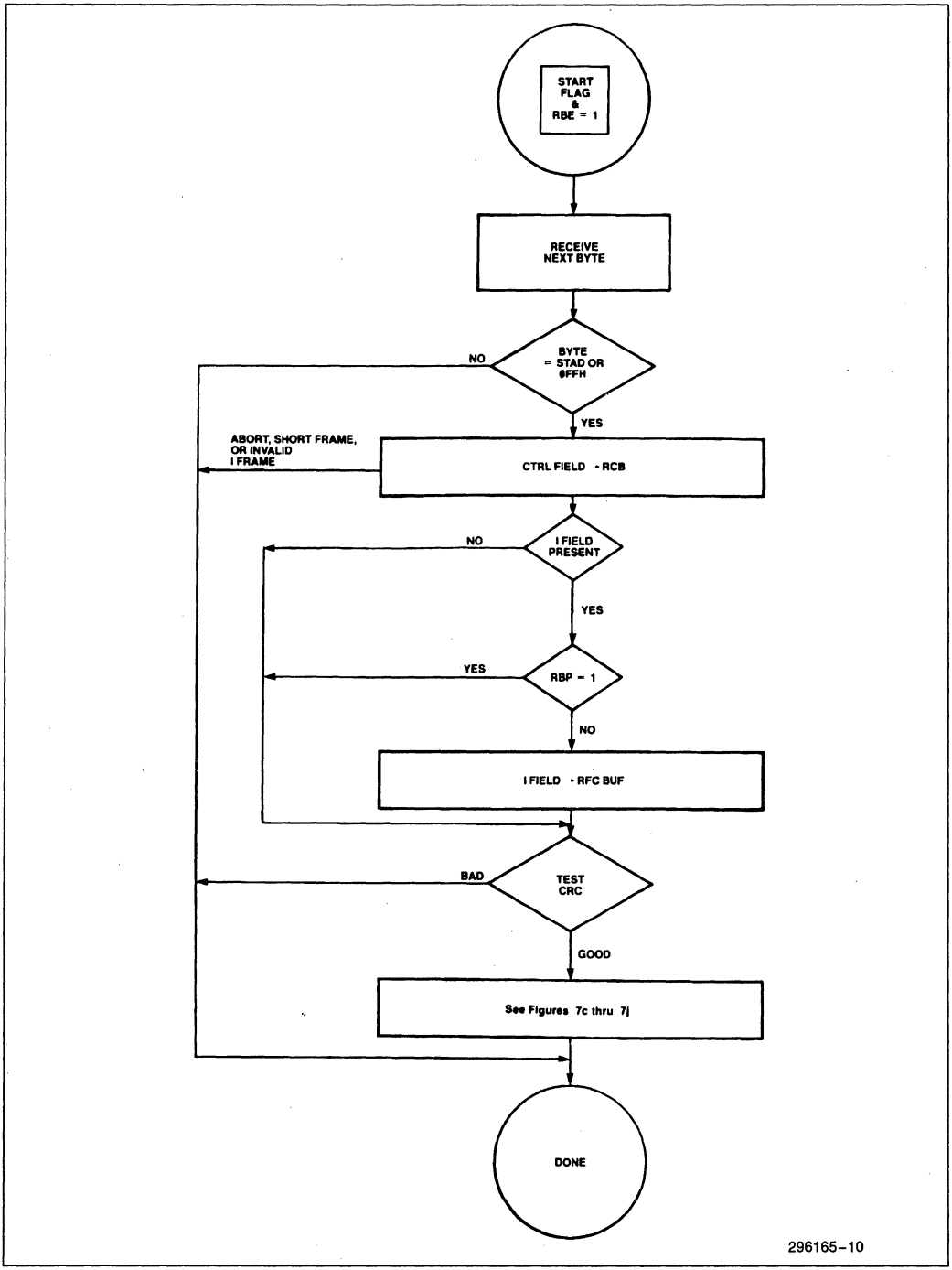


Figure 7a. SIU AUTO Mode Receive Flowchart—General

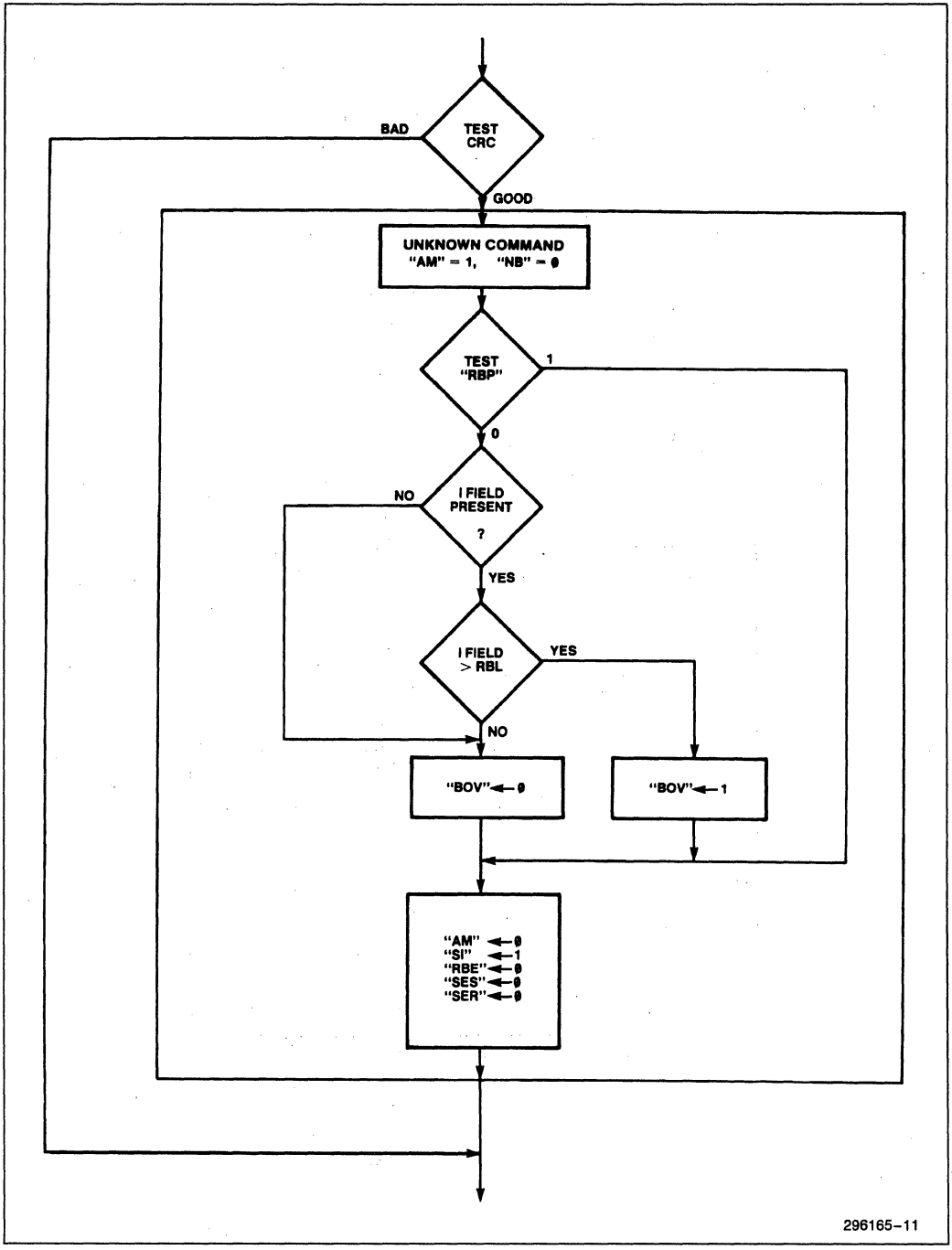


Figure 7b. SIU AUTO Mode Receive Flowchart—Unknown Command

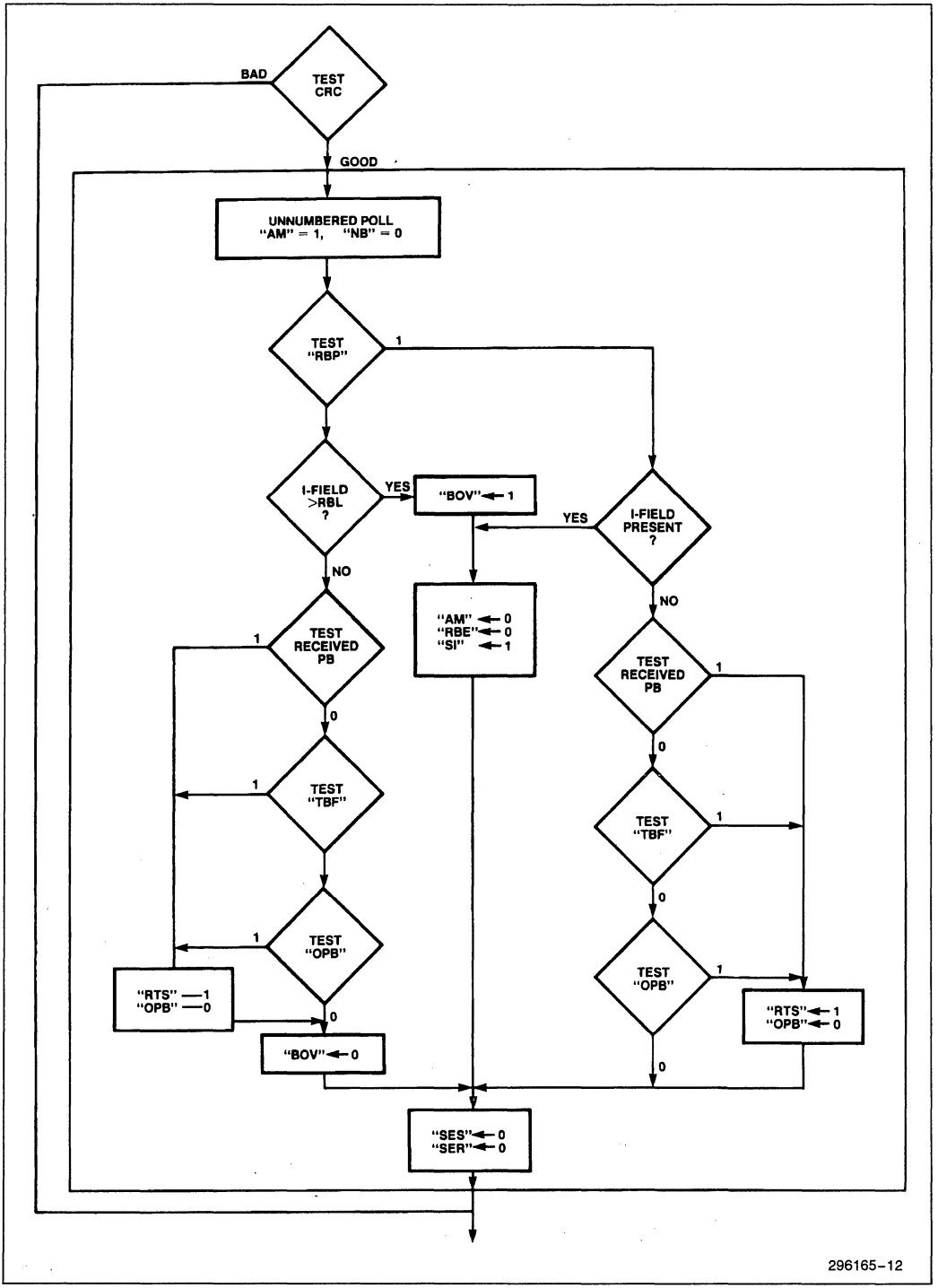


Figure 7c. SIU AUTO Mode Receive Flowchart—Unnumbered Poll

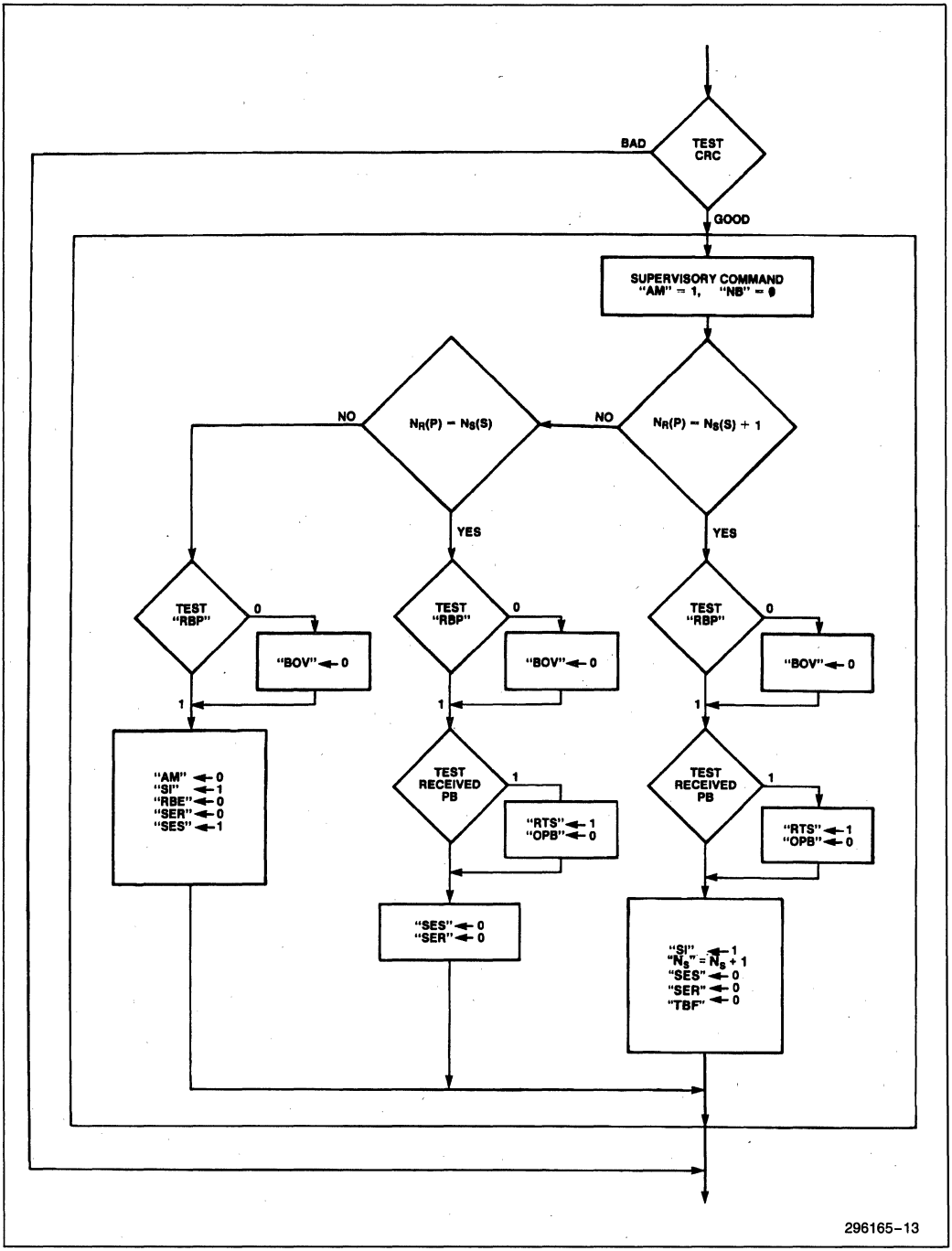


Figure 7d. SIU AUTO Mode Receive Flowchart—Supervisory Command

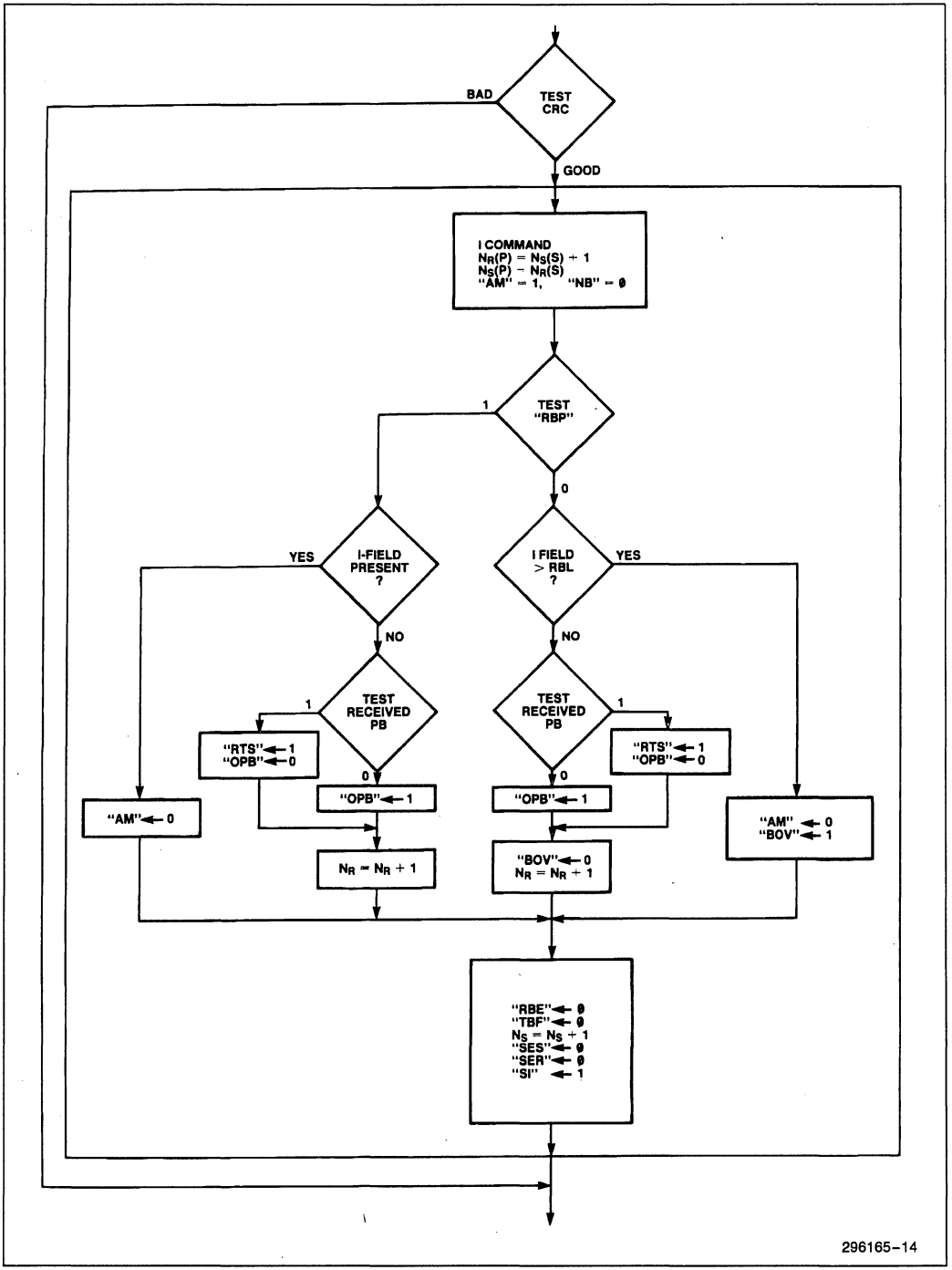


Figure 7e. SIU AUTO Mode Receive Flowchart—I Command: Prior Transmitted I-Field Confirmed, Current Received I-Field in Sequence

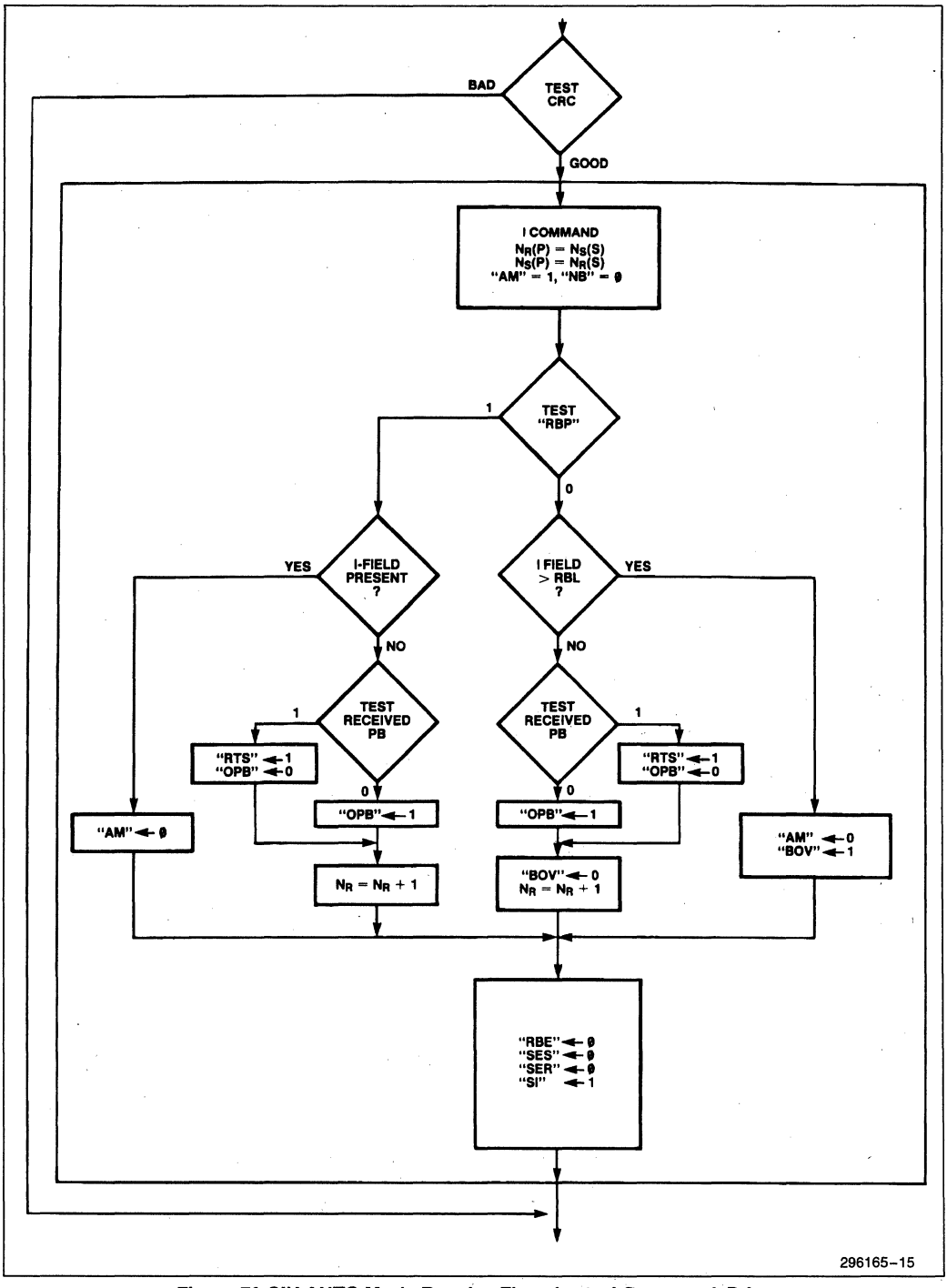


Figure 7f. SIU AUTO Mode Receive Flowchart—I Command: Prior Transmitted I-Field Not Confirmed, Current Received I-Field in Sequence

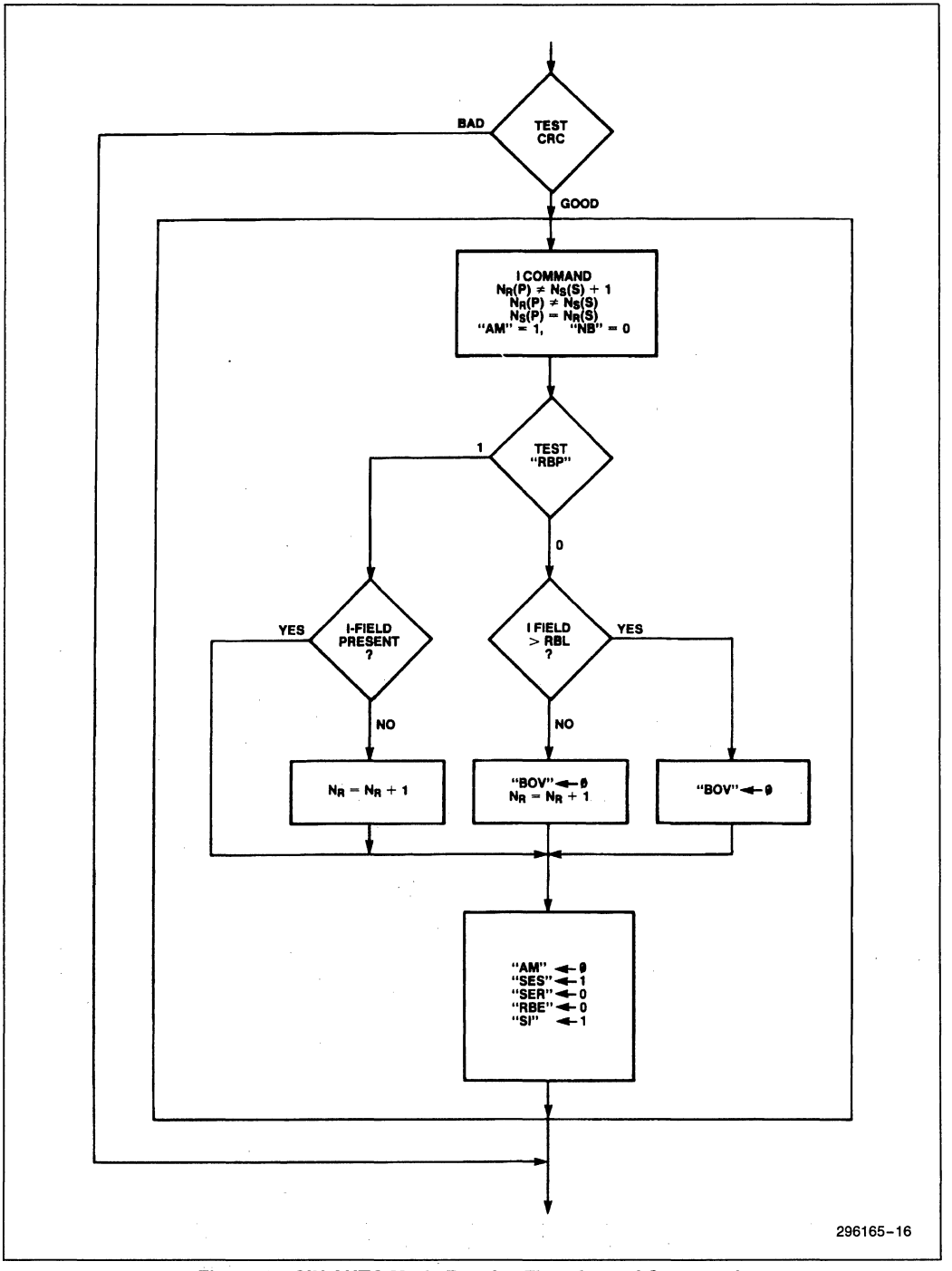


Figure 7g. SIU AUTO Mode Receive Flowchart—I Command: Sequence Error Send, Current Received I-Field in Sequence

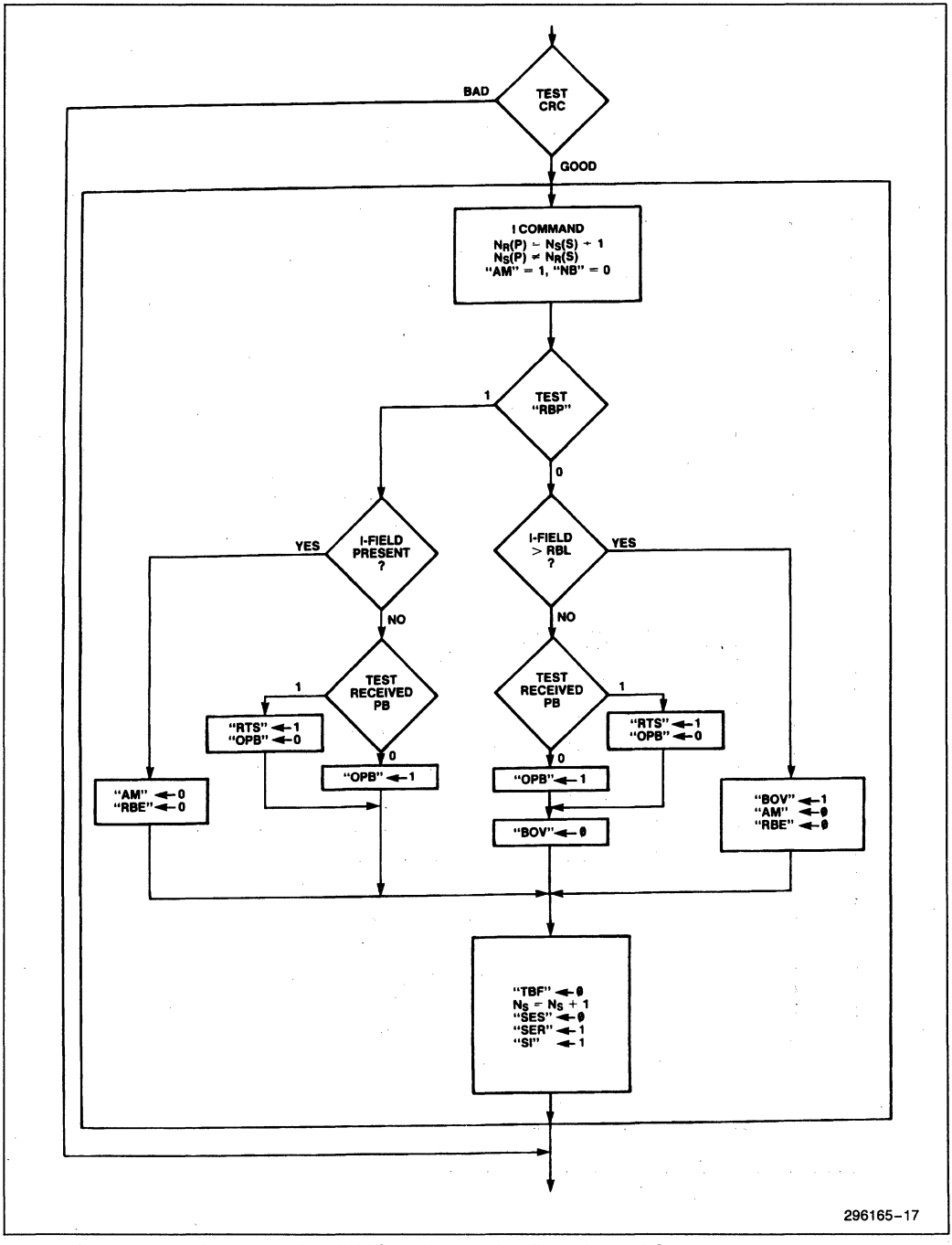


Figure 7h. SIU AUTO Mode Receive Flowchart—I Command: Prior Transmitted I-Field Confirmed Sequence Error Receive

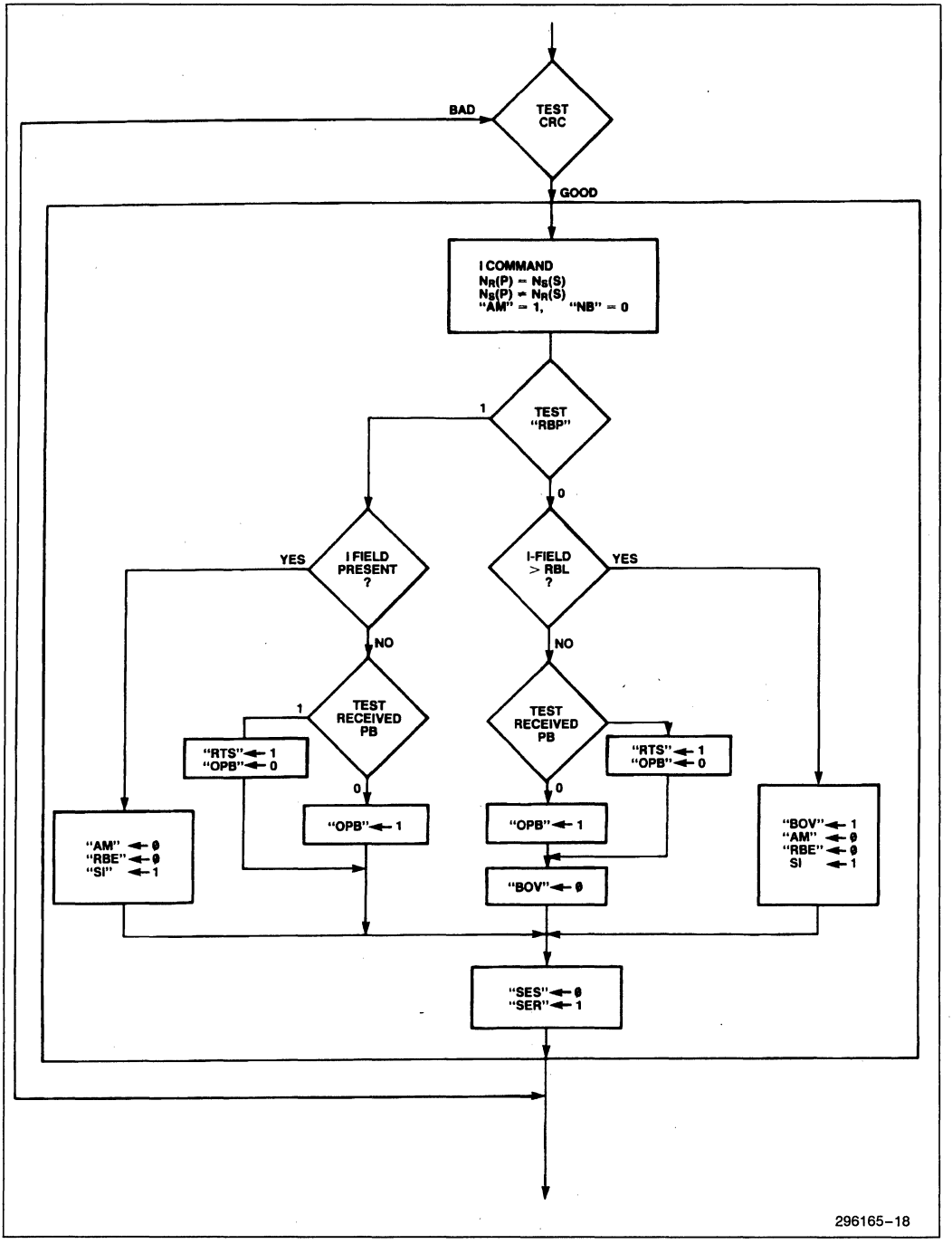


Figure 7i. SIU AUTO Mode Receive Flowchart—I Command: Prior Transmitted I-Field Not Confirmed, Sequence Error Receive

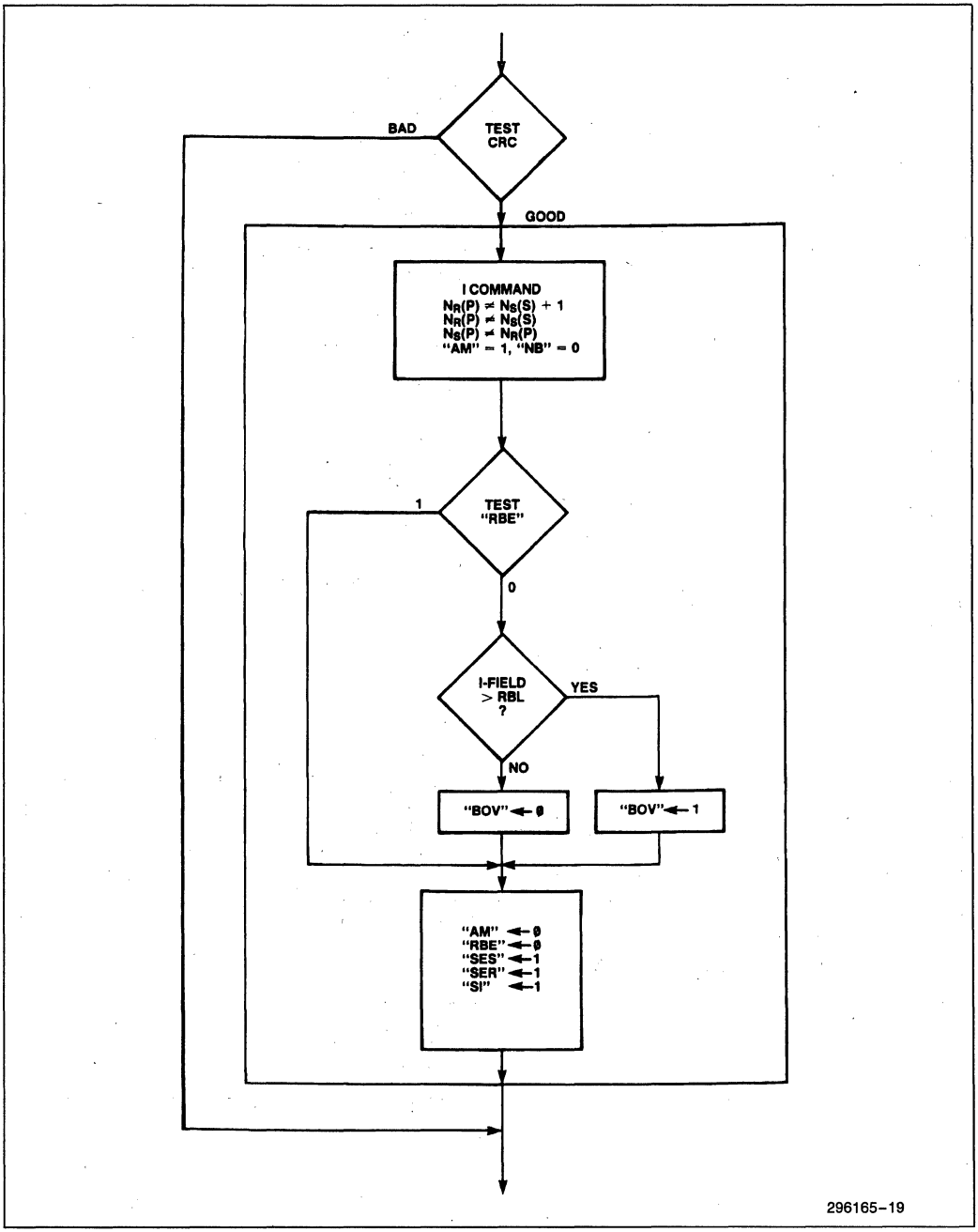


Figure 7j. SIU AUTO Mode Receive Flowchart—I Command: Sequence Error Send and Sequence Error Receive

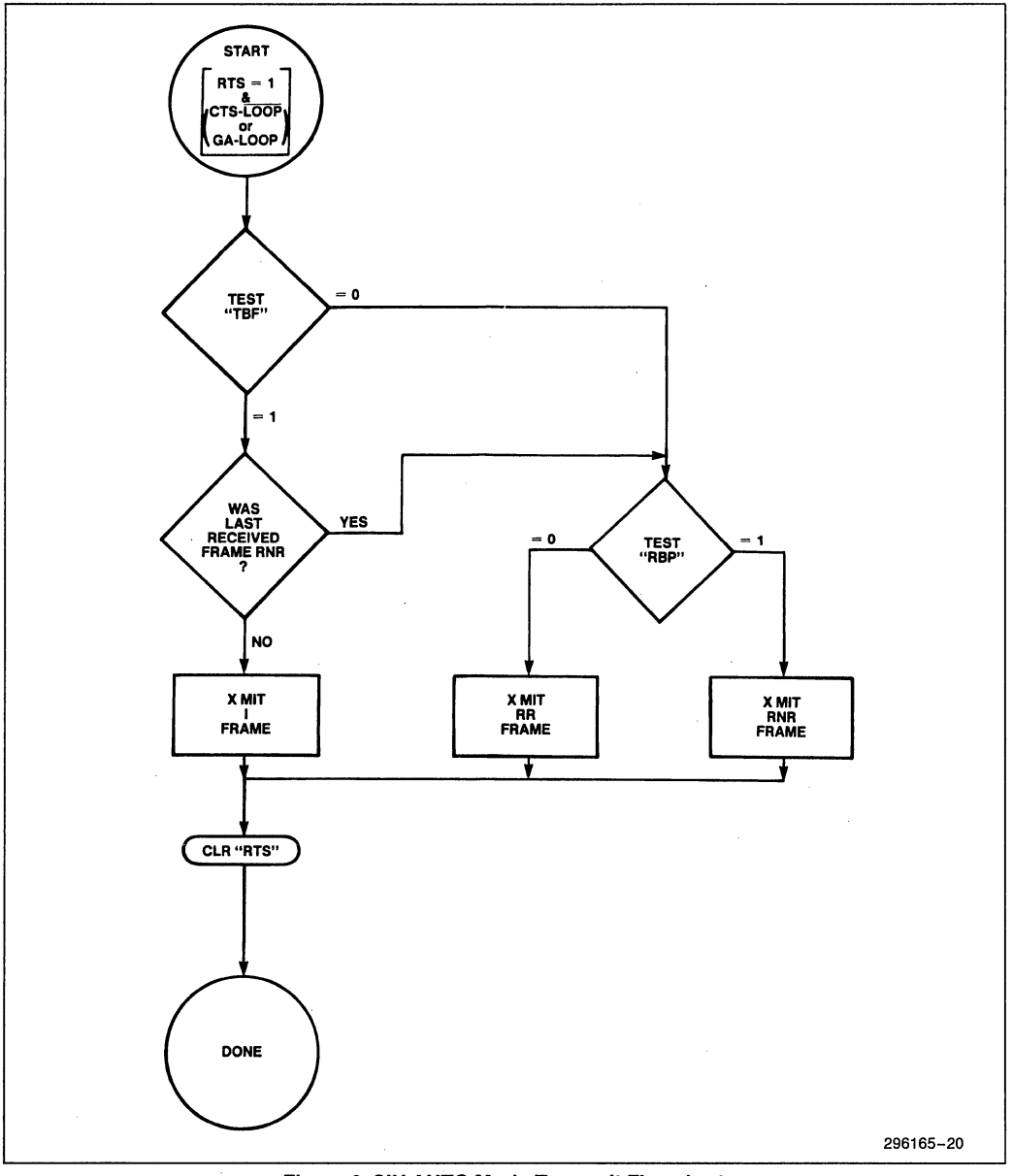


Figure 8. SIU AUTO Mode Transmit Flowchart

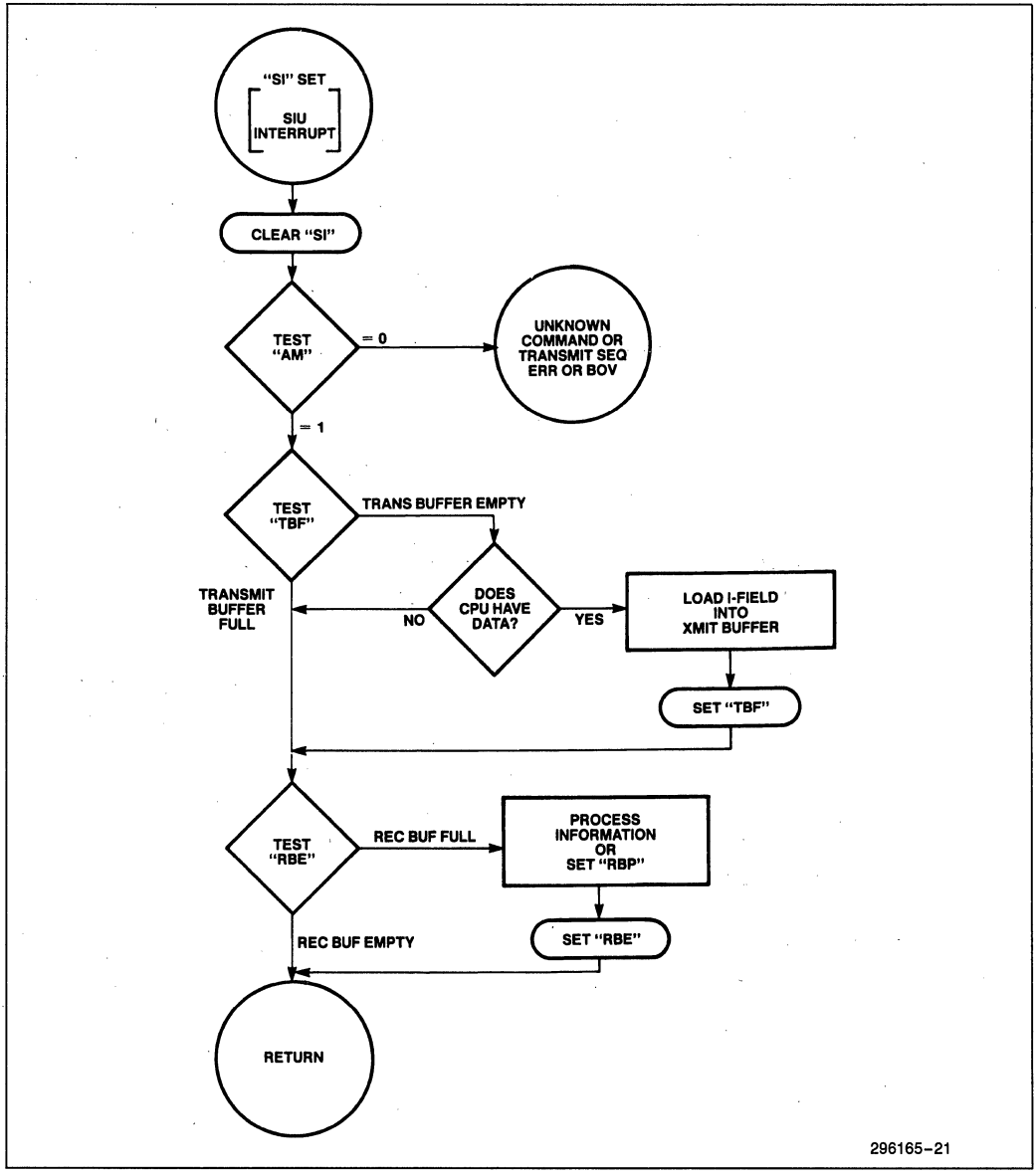


Figure 9. AUTO Mode Response to "SI"

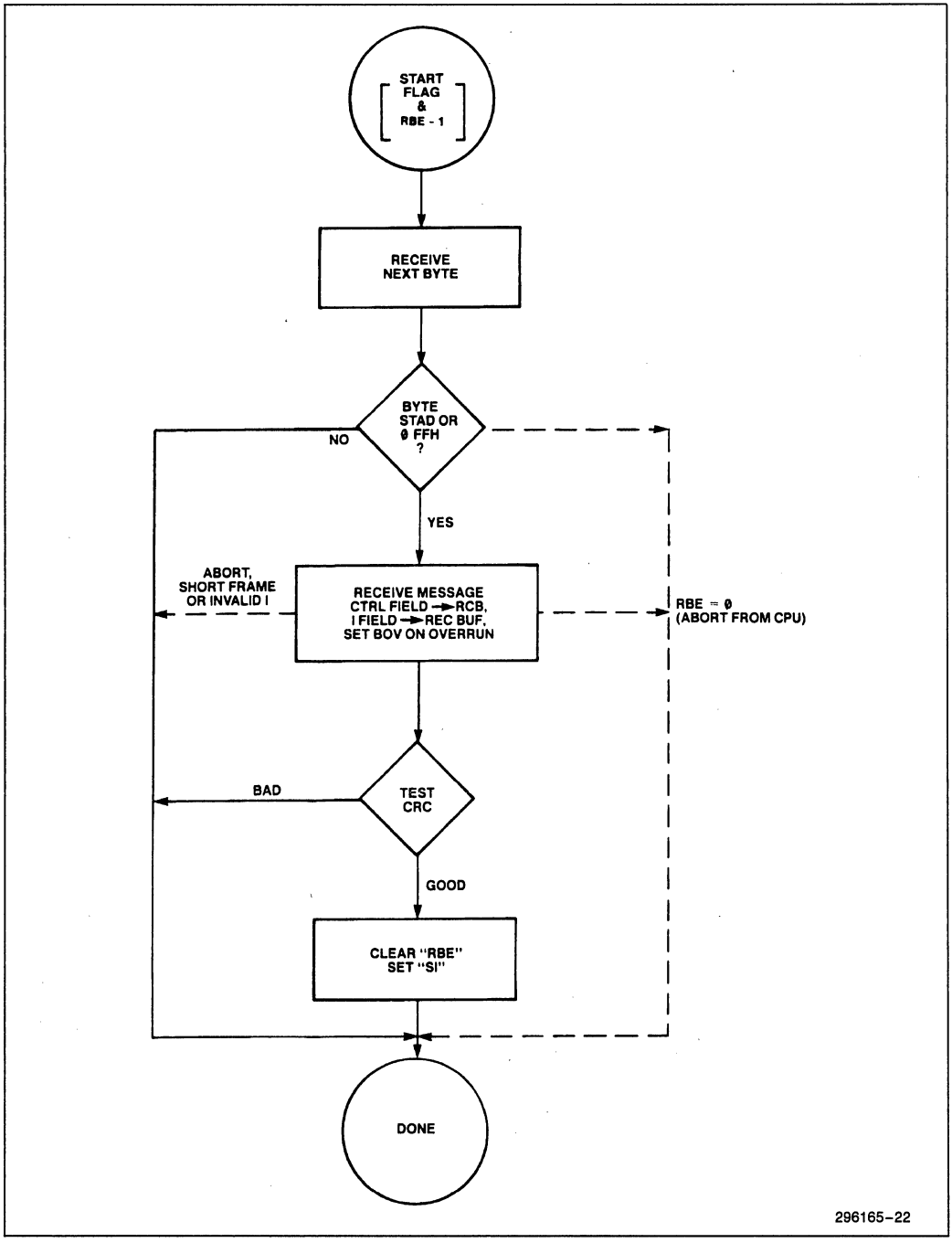


Figure 10. SIU FLEXIBLE Mode Receive Flowchart

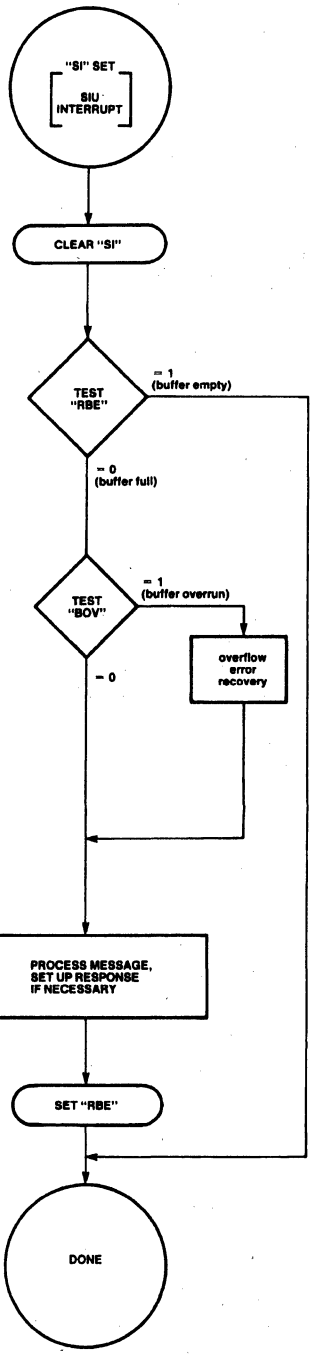


Figure 11. FLEXIBLE Mode Response to Receive "SI"

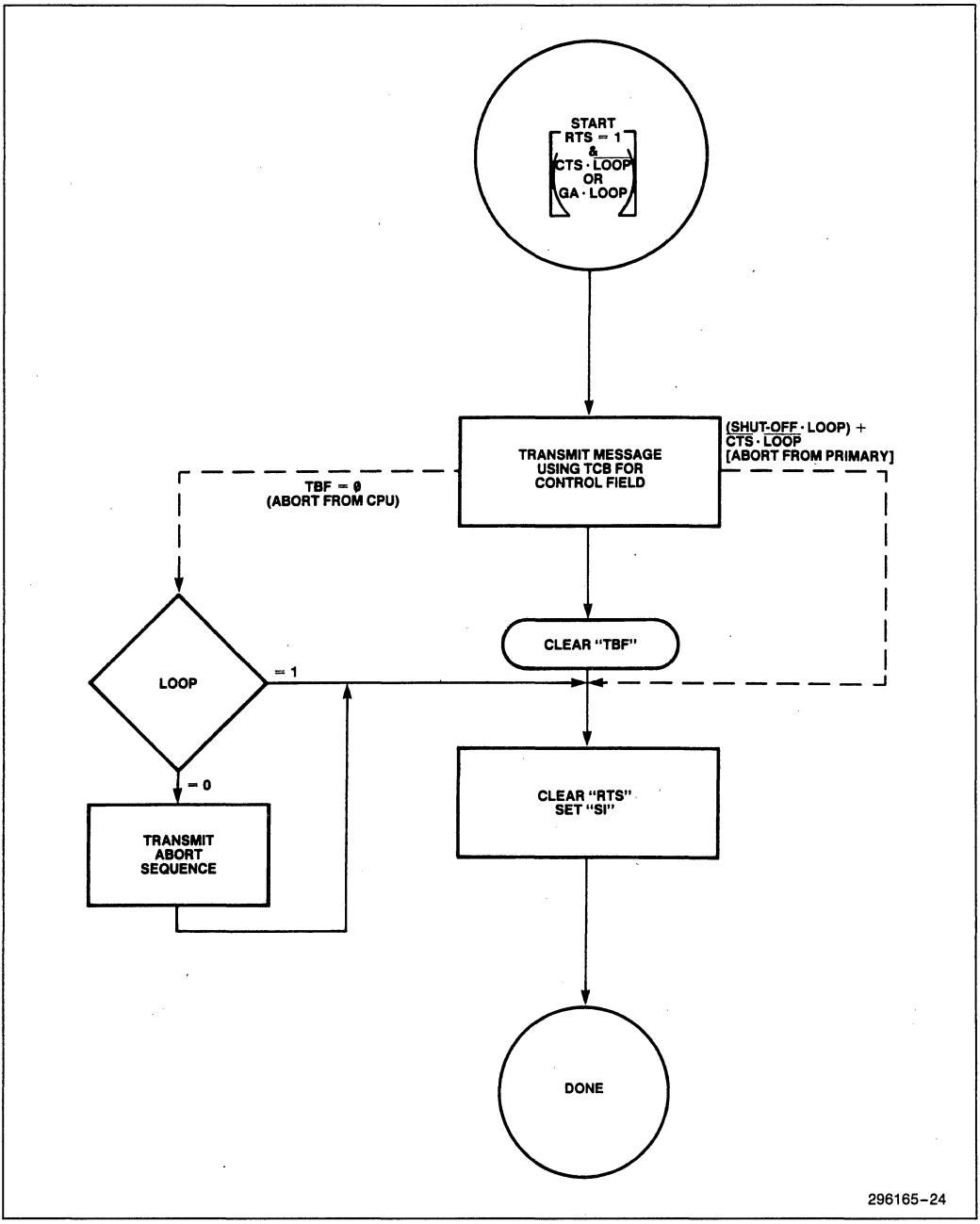


Figure 12. SIU FLEXIBLE Mode Transmit Flowchart

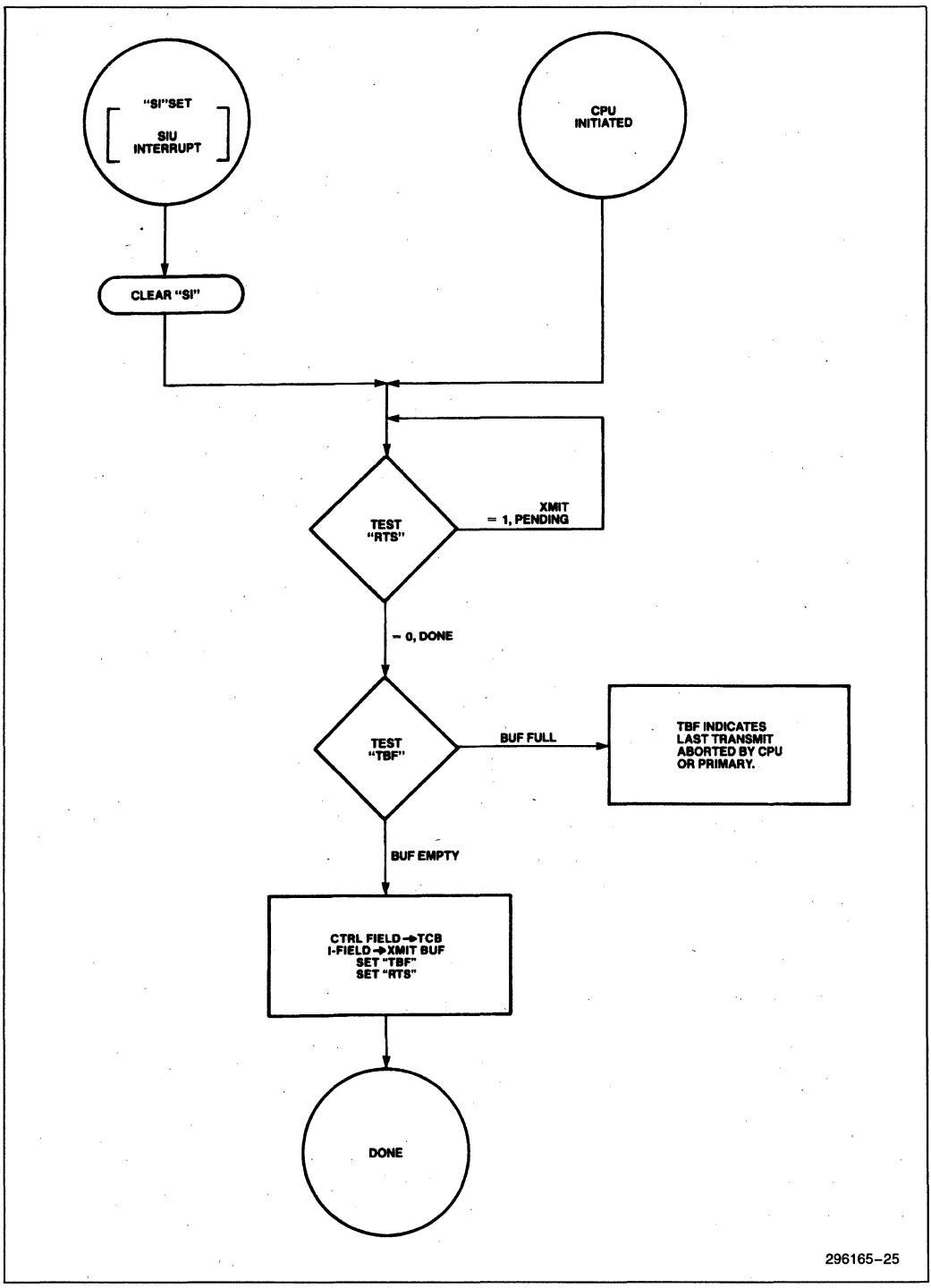


Figure 13. FLEXIBLE Mode Response to Transmit "SI"

9.0 MORE DETAILS ON SIU HARDWARE

The SIU divides functionally into two sections—a bit processor (BIP) and a byte processor (BYP)—sharing some common timing and control logic. As shown in Figure 14, the BIP operates between the serial port pins and the SIU bus, and performs all functions necessary to transmit/receive a byte of data to/from the serial data stream. These operations include shifting, NRZI encoding/decoding, zero insertion/deletion, and FCS generation/checking. The BYP manipulates bytes of data to perform message formatting, and other transmitting and receiving functions. It operates between the SIU bus (SIB) and the 8044's internal bus (IB). The interface between the SIU and the CPU involves an interrupt and some locations in on-chip RAM space which are managed by the BYP.

The maximum possible data rate for the serial port is limited to ½ the internal clock rate. This limit is imposed by both the maximum rate of DMA to the on-chip RAM, and by the requirements of synchronizing to an external clock. The internal clock rate for an 8044 running on a 12 MHz crystal is 6 MHz. Thus the maximum 8044 serial data rate is 3 MHz. This data rate drops down to 2.4 MHz when time is allowed for external clock synchronization.

9.1 The Bit Processor

In the asynchronous (self clocked) modes the clock is extracted from the data stream using the on-chip digital phase-locked-loop (DPLL). The DPLL requires a clock input at 16 times the data rate. This 16 × clock may originate from SCLK, Timer 1 Overflow, or PH2 (one half the oscillator frequency). The extra divide by-two described above allows these sources to be treated alternatively as 32 × clocks.

The DPLL is a free-running four-bit counter running off the 16 × clock. When a transition is detected in the receive data stream, a count is dropped (by suppressing the carry-in) if the current count value is greater than 8. A count is added (by injecting a carry into the second stage rather than the first) if the count is less than 8. No adjustment is made if the transition occurs at the count of 8. In this manner the counter locks in on the point at which transitions in the data stream occur at the count of 8, and a clock pulse is generated when the count overflows to 0.

In order to perform NRZI decoding, the NRZI decoder compares each bit of input data to the previous bit. There are no clock delays in going through the NRZI decoder.

The zero insert/delete circuitry (ZID) performs zero insertion/deletion, and also detects flags, GA's (Go-Ahead's), and aborts (same as GA's) in the data stream. The pattern 1111110 is detected as an early GA, so that the GA may be turned into a flag for loop mode transmission.

The shut-off detector monitors the receive data stream for a sequence of eight zeros, which is a shut-off command for loop mode transmissions. The shut-off detector is a three-bit counter which is cleared whenever a one is found in the receive data stream. Note that the ZID logic could not be used for this purpose, because the receive data must be monitored even when the ZID is being used for transmission.

As an example of the operation of the bit processor, the following sequence occurs in relation to the receive data:

1) RXD is sampled by SCLK, and then synchronized to the internal processor clock (IPC).
2) If the NRZI mode is selected, the incoming data is NRZI decoded.
3) When receiving other than the flag pattern, the ZID deletes the '0' after 5 consecutive '1's (during transmission this zero is inserted). The ZID locates the byte boundary for the rest of the circuitry. The ZID deletes the '0's by preventing the SR (shift register) from receiving a clocking pulse.
4) The FCS (which is a function of the data between the flags—not including the flags) is initialized and started at the detection of the byte boundary at the end of the opening flag. The FCS is computed each bit boundary until the closing flag is detected. Note that the received FCS has gone through the ZID during transmission.

9.2 The Byte Processor

Figure 15 is a block diagram of the byte processor (BYP). The BYP contains the registers and controllers necessary to perform the data manipulations associated with SDLC communications. The BYP registers may be read or written by the CPU over the 8044's internal bus (IB), using standard 8044 hardware register operations. The 8044 register select PLA controls these operations. Three of the BYP registers connect to the IB through the IBS, a sub-bus which also connects to the CPU interrupt control registers.

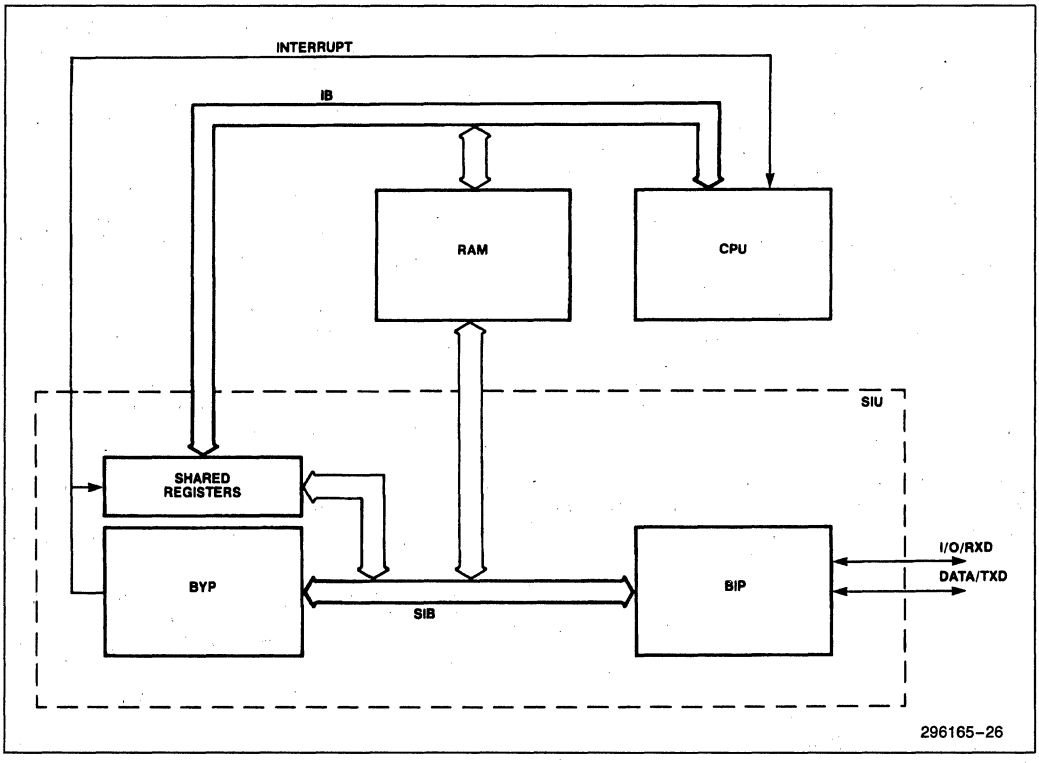


Figure 14. The Bit and Byte Processors

Simultaneous access of a register by both the IB and the SIB is prevented by timing. In particular, RAM access is restricted to alternate internal processor cycles for the CPU and the SIU, in such a way that collisions do not occur.

As an example of the operation of the byte processor, the following sequence occurs in relation to the receive data:

1) Assuming that there is an address field in the frame, the BYP takes the station address from the register file into temporary storage. After the opening flag, the next field (the address field) is compared to the station address in the temporary storage. If a match occurs, the operation continues.
2) Assuming that there is a control field in the frame, the BYP takes the next byte and loads it into the RCB register. The RCB register has the logic to update the NSNR register (increment receive count, set SES and SER flags, etc.).
3) Assuming that there is an information field, the next byte is dumped into RAM at the RBS location. The DMA CNT (RBL at the opening flag) is loaded from the DMA CNT register into the RB register and decremented. The RFL is then loaded into the RB register, incremented, and stored back into the register file.
4) This process continues until the DMA CNT reaches zero, or until a closing flag is received. Upon either event, the BYP updates the status, and, if the CRC is good, the NSNR register.

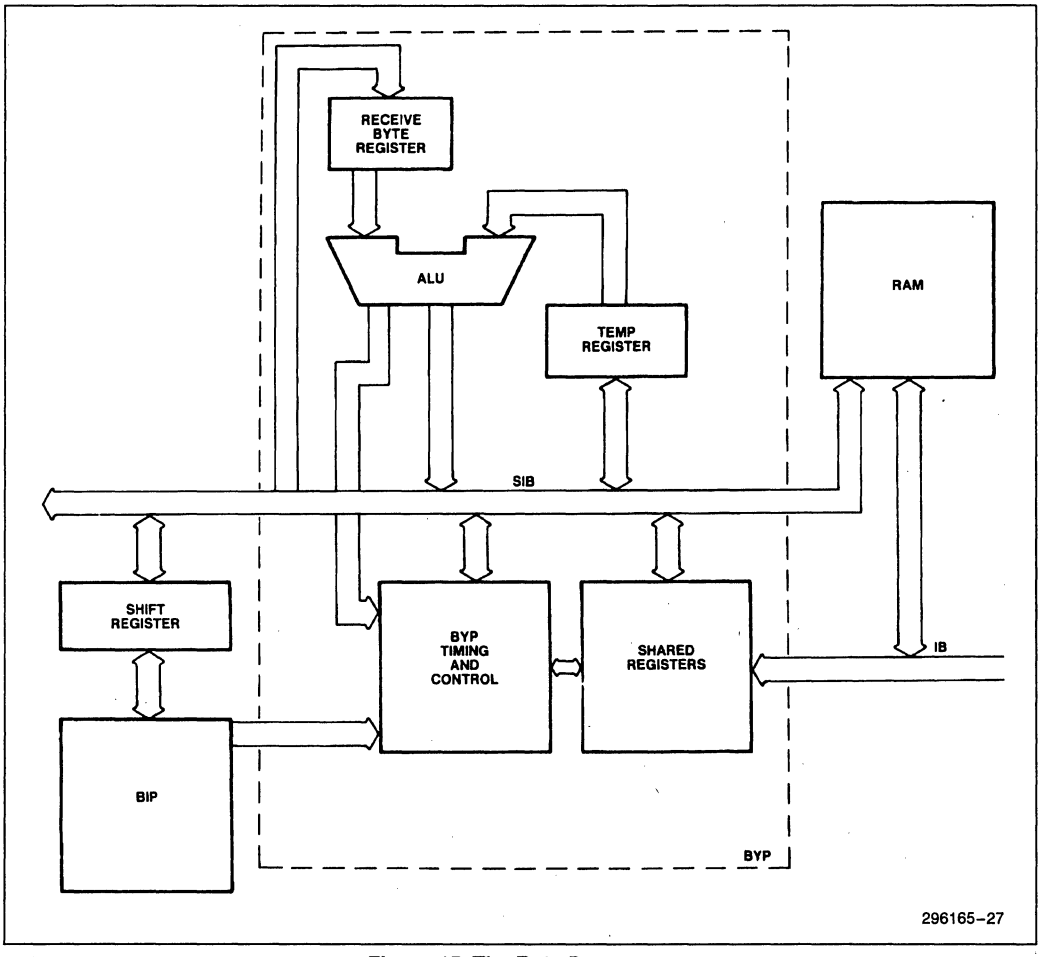


Figure 15. The Byte Processor

10.0 DIAGNOSTICS

An SIU test mode has been provided, so that the on-chip CPU can perform limited diagnostics on the SIU. The test mode utilizes the output latches for P3.0 and P3.1 (pins 10 and 11). These port 3 pins are not useful as out-put ports, since the pins are taken up by the serial port functions. Figure 16 shows the signal routing associated with the SIU test mode.

Writing a 0 to P3.1 enables the serial test mode (P3.1 is set to 1 by reset). In test mode the P3.0 bit is mapped into the received data stream, and the 'write port 3' control signal is mapped into the SCLK path in place of T1. Thus, in test mode, the CPU can send a serial data stream to the SIU by writing to P3.0. The transmit data stream can be monitored by reading P3.1. Each successive bit is transmitted from the SIU by writing to any bit in Port 3, which generates SCLK.

In test mode, the P3.0 and P3.1 pins are placed in a high voltage, high impedance state. When the CPU reads P3.0 and P3.1 the logic level applied to the pin will be returned. In the test mode, when the CPU reads 3.1, the transmit data value will be returned, not the voltage on the pin. The transmit data remains constant for a bit time. Writing to P3.0 will result in the signal being outputted for a short period of time. However, since the signal is not latched, P3.0 will quickly return to a high voltage, high impedance state.

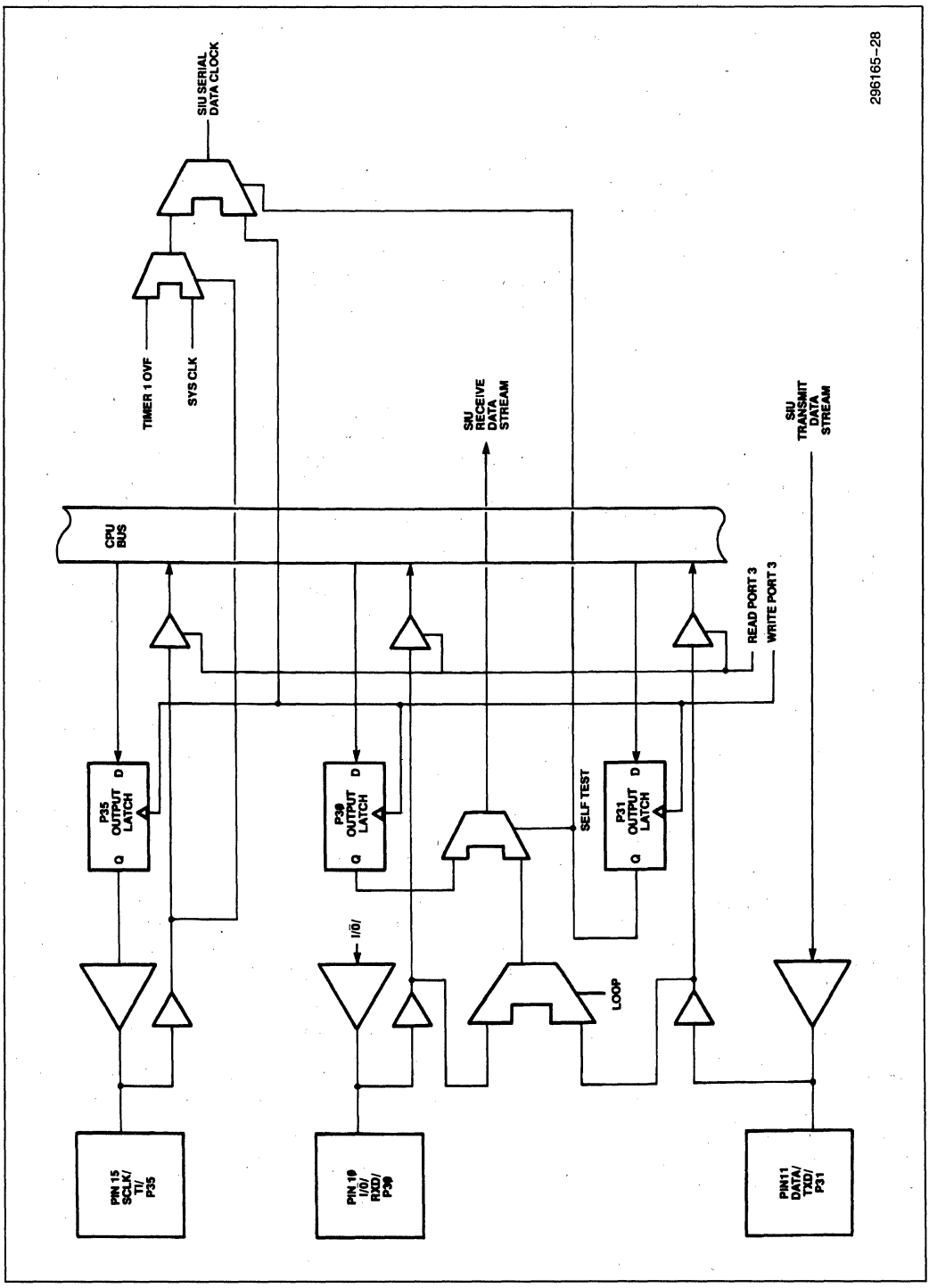


Figure 16. SIU Test Mode

The serial test mode is disabled by writing a 1 to P3.1. Care must be taken that a 0 is never written to P3.1 in the course of normal operation, since this causes the test mode to be entered.

Figure 17 is an example of a simple program segment that can be imbedded into the user's diagnostic program. That example shows how to put the 8044 into "Loop-back mode" to test the basic transmitting and receiving functions of the SIU.

Loop-back mode is functionally equivalent to a hardwire connection between pins 10 and 11 on the 8044.

In this example, the 8044 CPU plays the role of the primary station. The SIU is in the AUTO mode. The CPU sends the SIU a supervisory frame with the poll bit set and an RNR command. The SIU responds with a supervisory frame with the poll bit set and an RR command.

The operation proceeds as follows:

Interrupts are disabled, and the self test mode is enabled by writing a zero to P3.1. This establishes P3.0 as the data path from the CPU to the SIU. CTS (clear-to-send) is enabled by writing a zero to P1.7. The station address is initialized by writing 08AH into the STAD (station address register).

The SIU is configured for receive operation in the clocked mode and in AUTO mode. The CPU then transmits a supervisory frame. This frame consists of an opening flag, followed by the station address, a control field indicating that this is a supervisory frame with an RNR command, and then a closing flag.

Each byte of the frame is transmitted by writing that byte into the A register and then calling the subroutine XMIT8. Two additional SCLKs are generated to guarantee that the last bits in the frame have been clocked into the SIU. Finally the CPU reads the status register (STS). If the operation has proceeded correctly, the status will be 072H. If it is not, the program jumps to the ERROR loop and terminates.

The SIU generates an SI (SIU interrupt) to indicate that it has received a frame. The CPU clears this interrupt, and then begins to monitor the data stream that is being generated by the SIU in response to what it has received. As each bit arrives (via P3.1), it is moved into the accumulator, and the CPU compares the byte in the accumulator with 07EH, which is the opening flag. When a match occurs, the CPU identifies this as byte boundary, and thereafter processes the information byte-to-byte.

The CPU calls the RCV8 subroutine to get each byte into the accumulator. The CPU performs compare operations on (successively) the station address, the control field (which contains the RR response), and the closing flag. If any of these do not compare, the program jumps to the ERROR loop. If no error is found, the program jumps to the DONE loop.

```
MCS-51 MACRO ASSEMBLER    DATA


ISIS-II MCS-51 MACRO ASSEMBLER V2.0
OBJECT MODULE PLACED IN :F1:DATA.OBJ
ASSEMBLER INVOKED BY:  asm51 :f1:data.man device(44)


LOC  OBJ             LINE     SOURCE

                        1
                        2
0000 75C800             3     INIT:   MOV     STS,#00H
0003 C2B1               4             CLR     P3.1            ; Enable self test mode
0005 C297               5             CLR     P1.7            ; Enable CTS
0007 75CE8A             6             MOV     STAD, #8AH      ; Initialize address
                        7
                        8       ; CONFIGURE RECEIVE OPERATION
                        9
000A 75D86A            10             MOV     NSNR, #6AH      ; NS(S)=3, SES=0, NR(S)=5, SER=0
000D 75C901            11             MOV     SMD, #01H       ; NFCS=1
0010 75C8C2            12             MOV     STS, #0C2H      ; TBF=1, RBE=1, AM=1
                       13
                       14     ;    TRANSMIT A SUPERVISORY FRAME FROM THE PRIMARY STATION WITH THE POLL
                       15     ;    BIT SET AND A RNR COMMAND
                       16
0013 747E              17     SEND:   MOV     A, #7EH         ; The SIU receives a flag first
0015 120066            18             CALL    XMITB
0018 748A              19             MOV     A, #8AH         ; The address is next
001A 120066            20             CALL    XMITB
001D 7495              21             MOV     A, #095H        ; RNR SUP FRAME with P/F=1, NR(P)=4
001F 120066            22             CALL    XMITB
0022 747E              23             MOV     A, #7EH         ; Receive closing flag
0024 120066            24             CALL    XMITB
0027 D2B0              25             SETB    P3.0            ; Generate extra SCLK's to
0029 D2B0              26             SETB    P3.0            ; Initiate receive action
                       27
002B E5C8              28             MOV     A, STS          ; Check for appropriate status
002D B4722A            29             CJNE    A, #72H, ERROR
                       30
                       31     ; PREPARE TO RECEIVE RUPI'S RESPONCE TO PRIMARY'S RNR
                       32
                       33
                       34
0030 C2CC              35     RECV:   CLR     SI              ; Clear SI
0032 7400              36             MOV     A, #00H         ; Clear ACC
0034 780C              37             MOV     R3, #12         ; Try 12 times
                       38
                       39     ;  LOOK FOR THE OPENING FLAG
                       40
0036 D2B0              41     WFLAG1: SETB    P3.0            ; SCLK
0038 A2B1              42             MOV     C, P3.1         ; Transmitted data
003A 13                43             RRC     A
003B B47E03            44             CJNE    A, #07EH, WFLG1
003E 020046            45             JMP     CNTINU
0041 DBF3              46     WFLG1:  DJNZ    R3, WFLAG1
0043 02005A            47             JMP     ERROR
                       48
                       49
0046 12005C            50     CNTINU: CALL    RCVB            ; Get SIU's Transmitted address field
0049 B4BA0E            51             CJNE    A, #0BAH, ERROR
004C 12005C            52             CALL    RCVB            ; Primary expects to receive RR from SIU
004F B4B108            53             CJNE    A, #0B1H, ERROR
0052 12005C            54             CALL    RCVB            ; Receive closing flag
0055 B47E02            55             CJNE    A, #07EH, ERROR
                       56
0058 80FE              57     DONE:   JMP     DONE
                       58
005A 80FE              59     ERROR:  JMP     ERROR
                       60
                       61
005C 7808              62       RCVB: MOV     R0, #08         ; Initialize the bit counter
005E D2B0              63     GETBIT: SETB    P3.0            ; SCLK
0060 A2B1              64             MOV     C, P3.1         ; Transmitted data
0062 13                65             RRC     A
0063 DBF9              66             DJNZ    R0, GETBIT
0065 22                67             RET
                       68
                       69
                       70
0066 7809              71     XMITB:  MOV     R0, #9          ; Initialize the bit counter
0068 13                72       L3:   RRC     A               ; Put the bit to be transmitted
                       73                                     ; in the Carry
0069 D801              74             DJNZ    R0, L1          ; When all bits have been sent
006B 22                75             RET                     ; return
                       76
006C 4004              77       L1:   JC      L2              ; If the carry bit is set, set
                       78                                     ; port P3.0 else
006E C2B0              79             CLR     P3.0            ; clear port P3.0
0070 80F6              80             JMP     L3
                       81
0072 D2B0              82       L2:   SETB    P3.0
0074 80F2              83             JMP     L3
                       84     end
```

296165-29

Figure 17. Loop-Back Mode Software

8044 Application Examples 21

8044 APPLICATION EXAMPLES

1.0 INTERFACING THE 8044 TO A MICROPROCESSOR

The 8044 is designed to serve as an intelligent controller for remote peripherals. However, it can also be used as an intelligent HDLC/SDLC front end for a microprocessor, capable of extensively off-loading link control functions for the CPU. In some applications, the 8044 can even be used for communications preprocessing, in addition to data link control.

This section describes a sample hardware interface for attaching the 8044 to an 8088. It is general enough to be extended to other microprocessors such as the 8086 or the 80186.

OVERVIEW

A sample interface is shown in Figure 1. Transmission occurs when the 8088 loads a 64 byte block of memory with some known data. The 8088 then enables the 8237A to DMA this data to the 8044. When the 8044 has received all of the data from the 8237A, it sends the data in a SDLC frame. The frame is captured by the Spectron Datascope™* which displays it on a CRT in hex format.

In reception, the Datascope sends a SDLC information frame to the 8044. The 8044 receives the SDLC frame, buffers it, and sends it to the 8088's memory. In this example the 8044 is being operated in the NON-AUTO mode; therefore, it does not need to be polled by a primary station in order to transmit.

THE INTERFACE

The 8044 does not have a parallel slave port. The 8044's 32 I/O lines can be configured as a local microprocessor bus master. In this configuration, the 8044 can expand the ROM and RAM memory, control peripherals, and communicate with a microprocessor.

The 8044, like the 8051, does not have a Ready line, so there is no way to put the 8044 in wait state. The clock on the 8044 cannot be stopped. Dual port RAM could still be used, however, software arbitration would be the only way to prevent collisions. Another way to interface the 8044 with another CPU is to put a FIFO or queue between the two processors, and this was the method chosen for this design.

Figure 2 shows the schematic of the 8044/8088 interface. It involves two 8-bit tri-state latches, two SR flip-flops, and some logic gates (6 TTL packs). The circuitry implements a one byte FIFO. RS422 transceivers are used, which can be connected to a multidrop link. Figure 3 shows the 8088 and support circuitry; the memory and decoders are not shown. It is a basic 8088 Min Mode system with an 8237A DMA controller and an 8259A interrupt controller.

*Datascope is a trademark of Spectron Inc.

DMA Channel One transfers a block of memory to the tri-state latch, while Channel Zero transfers a block of data from the latch to 8088's memory. The 8044's Interrupt 0 signal vectors the CPU into a routine which reads from the internal RAM and writes to the latch. The 8044's Interrupt 1 signal causes the chip to read from the latch and write to its on-chip data RAM. Both DMA requests and acknowledges are active low.

Initially, when the power is applied, a reset pulse coming from the 8284A initializes the SR flip-flops. In this initialization state, the 8044's transmit interrupt and the 8088's transmit DMA request are active; however, the software keeps these signals disabled until either of the two processors are ready to transmit. The software leaves the receive signals enabled, unless the receive buffers are full. In this way either the 8088 or the 8044 are always ready to receive, but they must enable the transmit signal when they have prepared a block to transmit. After a block has been transmitted or received, the DMA and interrupt signals return to the initial state.

The receive and transmit buffer sizes for the blocks of data sent between the 8044 and the 8088 have a maximum fixed length. In this case the buffer size was 64 bytes. The buffer size must be less than 192 bytes to enable 8044 to buffer the data in its on-chip RAM. This design allows blocks of data that are less than 64 bytes, and accommodates networks that allow frames of varying size. The first byte transferred between the 8088 and the 8044 is the byte count to follow; thus the 8044 knows how many bytes to receive before it transmits the SDLC frame. However, when the 8044 sends data to the 8088's memory, the 8237A will not know if the 8044 will send less than the count the 8237A was programmed for. To solve this problem, the 8237A is operated in the single mode. The 8044 uses an I/O bit to generate an interrupt request to the 8259A. In the 8088's interrupt routine, the 8237A's receive DMA channel is disabled, thus allowing blocks of data less than 64 bytes to be received.

THE SOFTWARE

The software for the 8044 and the 8088 is shown in Table 1. The 8088 software was written in PL/M86, and the 8044 software was written in assembly language.

The 8044 software begins by initializing the stack, interrupt priorities, and triggering types for the interrupts. At this point, the SIU parameter registers are

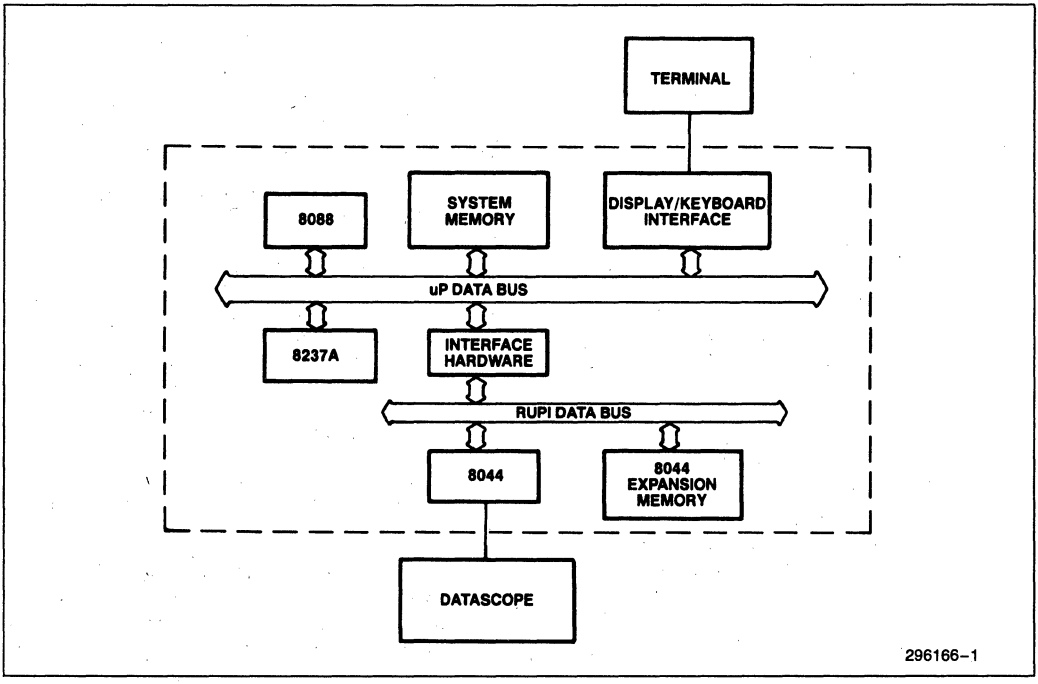


Figure 1. Block Diagram of 8088/8044 Interface Test

initialized. The receive and transmit buffer starting addresses and lengths are loaded for the on-chip DMA. This DMA is for the serial port. The serial station address and the transmit control bytes are loaded too.

Once the initialization has taken place, the SIU interrupt is enabled, and the external interrupt which receives bytes from the 8088 is enabled. Setting the 8044's Receive Buffer Empty (RBE) bit enables the receiver. If this bit is reset, no serial data can be received. The 8044 then waits in a loop for either RECEIVE DMA interrupt or the SERIAL INT interrupt.

The RECEIVE DMA interrupt occurs when the 8237A is transferring a block of data to the 8044. The first time this interrupt occurs, the 8044 reads the latch and loads the count value into the R2 register. On subsequent interrupts, the 8044 reads the latch, loads the data into the transmit buffer, and decrements R2. When R2 reaches zero, the interrupt routine sends the data in an SDLC frame, and disables the RECEIVE DMA interrupt. After the frame has been transmitted, a serial interrupt is generated. The SERIAL INT routine detects that a frame has been transmitted and re-enables the RECEIVE DMA interrupt. Thus, while the frame is being transmitted through the SIU, the 8237A is inhibited from sending data to the 8044's transmit buffer.

The TRANSMIT DMA routine sends a block of data from the 8044's receive buffer to the 8088's memory. Normally this interrupt remains disabled. However, if a serial interrupt occurs, and the SERIAL INT routine detects that a frame has been received, it calls the SEND subroutine. The SEND subroutine loads the number of bytes which were received in the frame into the receive buffer. Register R1 points to the receive buffer and R2 is loaded with the count. The TRANSMIT DMA interrupt is enabled, and immediately upon returning from the SERIAL INT routine, the interrupt is acknowledged. Each time the TRANSMIT DMA interrupt occurs, a byte is read from the receive buffer, written to the latch, and R2 is decremented. When R2 reaches 0, the TRANSMIT DMA interrupt is disabled, the SIU receiver is re-enabled, and the 8044 interrupts the 8088.

CONCLUSION

For the software shown in Table 1, the transfer rate from the 8088's memory to the 8044 was measured at 75K bytes/sec. This transfer rate largely depends upon the number of instructions in the 8044's interrupt service routine. Fewer instructions result in a higher transfer rate.

There are many ways of interfacing the 8044 locally to another microprocessor: FIFO's, dual port RAM with software arbitration, and 8255's are just a few. Alternative approaches, which may be more optimal for certain applications, are certainly possible.

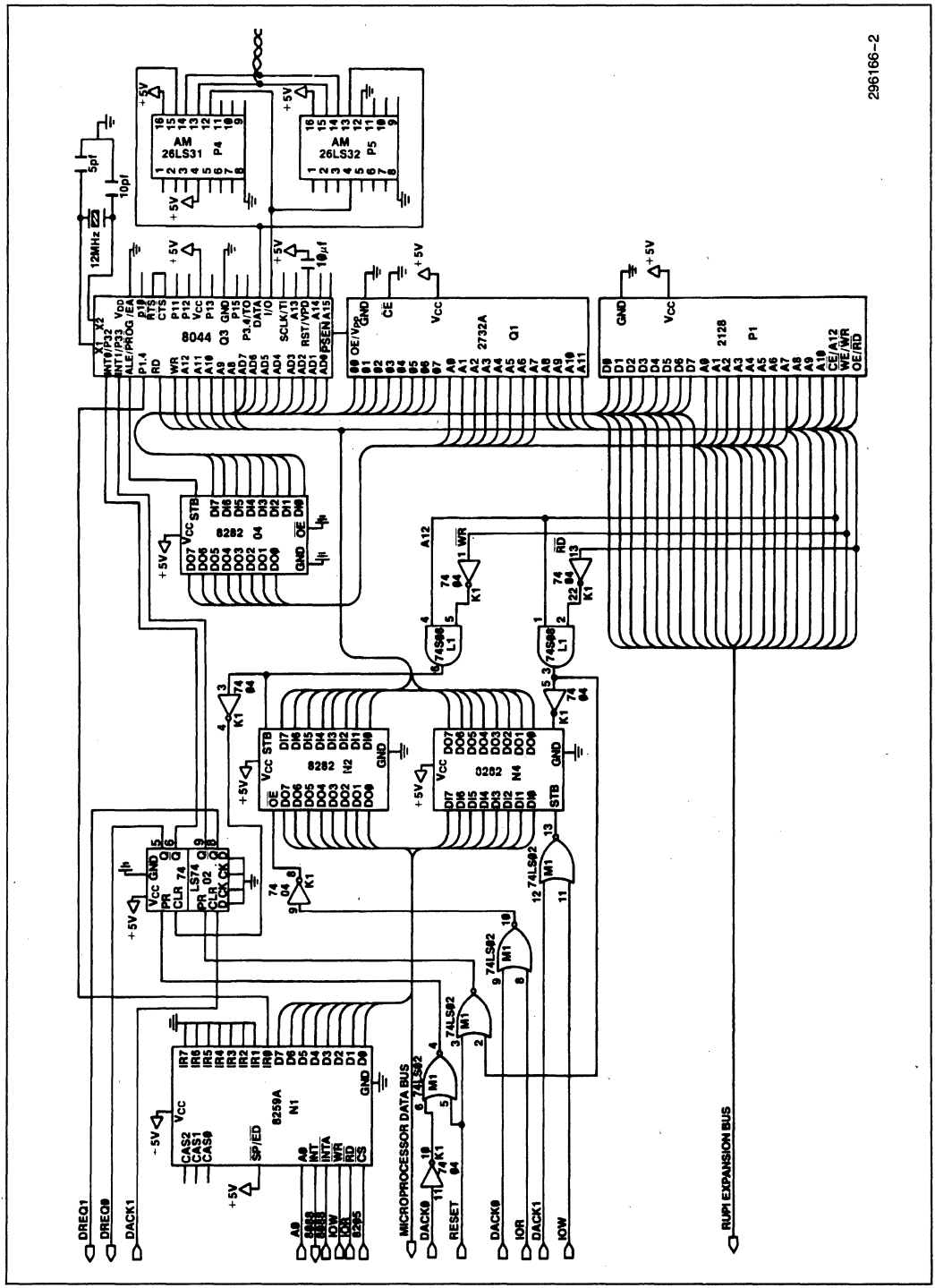


Figure 2. 8044 Interface to the 8088

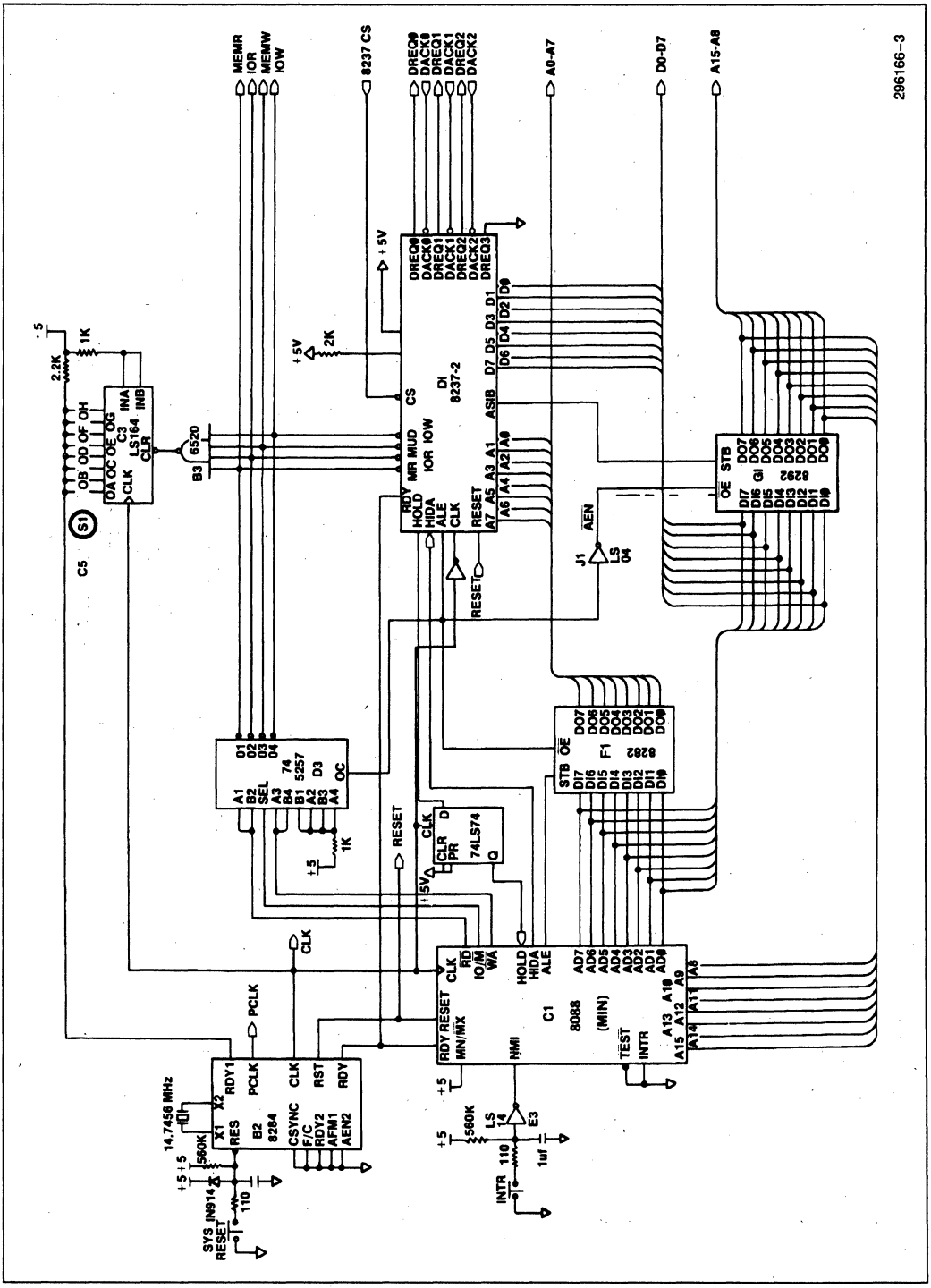


Figure 3. 8088 Min Mode System

Table 1. Transmit and Receive Software for an 8044/8088 System

```
LOC  OBJ    LINE     SOURCE
                1   $debug    title    (8044/8088 INTERFACE)
                2
                3
  0000          4   FIRST_BYTE       BIT     0       ; FLAG
                5
  0000          6            ORG     0
  0000 8024     7            SJMP    INIT
                8
  0026          9            ORG     26H
                10
  0026 7581AA   11   INIT:    MOV     SP, #170        ; INITIALIZE STACK
  0029 75B800   12            MOV     IP, #00         ; ALL INTERRUPTS ARE EQUAL PRIORITY
  002C 75C954   13            MOV     SMD, #54H       ; TIMER 1 OVERFLOW, NRZI, PRE-FRAME SYNC
  002F 758844   14            MOV     TCON, #44H      ; EDGE TRIGGERED EXTERNAL INTERRUPT 1
                15                                    ; LEVEL TRIGGERED EXTERNAL INTERRUPT 0
                16                                    ; TIMER 1 ON
  0032 758DEC   17            MOV     TH1, #0ECH      ; INITIALIZE TIMER, 3125 BPS
  0035 758920   18            MOV     TMOD, #20H      ; TIMER 1 AUTO RELOAD
                19
  0038 75DC6A   20            MOV     TBS, #106       ; SET UP SIU PARAMETER REGISTERS
  003B 75DB40   21            MOV     TBL, #64
  003E 75CC2A   22            MOV     RBS, #42
  0041 75CB40   23            MOV     RBL, #64
  0044 75CE55   24            MOV     STAD, #55H
  0047 75DA11   25            MOV     TCB, #00010001B ; RR, P/F=1
                26
  004A 901000   27            MOV     DPTR, #1000H    ; DPTR POINTS TO TRI-STATE LATCH
  004D D200     28            SETB    FIRST_BYTE      ; FLAG TO INDICATE FIRST BYTE
                29                                    ; FOR RECEIVE INTERRUPT ROUTINE
  004F D2CE     30            SETB    RBE             ; READY TO RECEIVE
  0051 75A894   31            MOV     IE, #10010100B  ; ENABLE RECEIVE DMA AND SIU INTERRUPT
                32
  0054 80FE     33            SJMP    $               ; WAIT HERE FOR INTERRUPTS
                34
  0056 80FE     35   ERROR:   SJMP    ERROR
                36 +1 $EJ
                37   ;*************************  SUBROUTINES  ********************************
                38
  0058 85CD29   39   SEND:    MOV     41, RFL         ; FIRST BYTE IN BLOCK IS COUNT
  005B 7929     40            MOV     R1, #41         ; POINT TO BLOCK OF DATA
  005D AACD     41            MOV     R2, RFL         ; LOAD COUNT
  005F 0A       42            INC     R2
  0060 D2A8     43            SETB    EX0             ; ENABLE DMA TRANSMIT INTERRUPT
  0062 22       44            RET
                45
                46
                47
                48   ;*******************  INTERRUPT SERVICE ROUTINES  ********************
                49
  0063          50   LOC_TMP  SET     $               ; SET UP INTERRUPT TABLE JUMP
  0013          51            ORG     0013H
  0013 020063   52            LJMP    RECEIVE_DMA
  0063          53            ORG     LOC_TMP
                54
                55   RECEIVE_DMA:
```

296166-69

Table 1. Transmit and Receive Software for an 8044/8088 System (Continued)

```
                        56
0063 10000E             57              JBC     FIRST_BYTE, L1 ; THE FIRST BYTE TRANSFERRED IS THE COUNT
                        58
0066 E0                 59              MOVX    A, @DPTR       ; READ THE LATCH
0067 F6                 60              MOV     @R0, A         ; PUT IT IN TRANSMIT BUFFER
0068 08                 61              INC     R0
0069 DA08               62              DJNZ    R2, L2         ; AFTER READING BYTES,
                        63
006B D2CF               64              SETB    TBF            ; SEND DATA
006D D2CD               65              SETB    RTS
006F D200               66              SETB    FIRST_BYTE
0071 C2AA               67              CLR     EX1
                        68
0073 32                 69      L2:     RETI
                        70
0074 786A               71      L1:     MOV     R0, #106       ; R0 IS A POINTER TO THE TRANSMIT
                        72                                     ; BUFFER STARTING ADDRESS
0076 E0                 73              MOVX    A, @DPTR       ; PUT THE FIRST BYTE INTO
0077 FA                 74              MOV     R2, A          ; R2 FOR THE COUNT
0078 32                 75              RETI
                        76
  0079                  77      LOC_TMP SET     $
0003                    78              ORG     0003H
0003 020079             79              LJMP    TRANSMIT_DMA
0079                    80              ORG     LOC_TMP
                        81
                        82      TRANSMIT_DMA
                        83
0079 E7                 84              MOV     A, @R1         ; READ BYTE OUT OF THE RECEIVE BUFFER
007A F0                 85              MOVX    @DPTR, A       ; WRITE IT TO THE LATCH
007B 09                 86              INC     R1
007C DA08               87              DJNZ    R2, L3         ; WHEN ALL BYTES HAVE BEEN SENT
                        88
007E C2A8               89              CLR     IE. 0          ; DISABLE INTERRUPT
0080 C294               90              CLR     P1. 4          ; CAUSE 8088 INTERRUPT TO TERMINATE DMA
0082 D294               91              SETB    P1. 4
0084 D2CE               92              SETB    RBE            ; ENABLE RECEIVER AGAIN
                        93
0086 32                 94      L3:     RETI
                        95
                        96
                        97
  0087                  98      LOC_TMP SET     $
0023                    99              ORG     0023H
0023 020087            100              LJMP    SERIAL_INT
0087                   101              ORG     LOC_TMP
                       102
                       103      SERIAL_INT:
                       104
0087 30CE06            105              JNB     RBE, RCV       ; WAS A FRAME RECEIVED
008A 30CF0B            106              JNB     TBF, XMIT      ; WAS A FRAME TRANSMITTED
008D 020056            107              LJMP    ERROR          ; IF NEITHER ERROR
                       108
0090 20CBC3            109      RCV:    JB      BOV, ERROR     ; IF BUFFER OVERRUN THEN ERROR
0093 1158              110              CALL    SEND           ; SEND THE FRAME TO THE 8088
0095 C2CC              111              CLR     SI
0097 32                112              RETI
                       113
0098 C2CC              114      XMIT:   CLR     SI
```

296166-70

Table 1. Transmit and Receive Software for an 8044/8088 System (Continued)

```
009A D2AA      115          SETB   EX1
009C 32        116          RETI
               117
               118          END

SYMBOL TABLE LISTING

NAME              TYPE      VALUE          ATTRIBUTES

BOV . . . . . .   B ADDR    00C8H.3  A
ERROR . . . . .   C ADDR    0056H    A
EX0 . . . . . .   B ADDR    00A8H.0  A
EX1 . . . . . .   B ADDR    00A8H.2  A
FIRST_BYTE . .    B ADDR    0020H.0  A
IE . . . . . .    D ADDR    00A8H    A
INIT . . . . . .  C ADDR    0026H    A
IP . . . . . .    D ADDR    00B8H    A
L1 . . . . . .    C ADDR    0074H    A
L2 . . . . . .    C ADDR    0073H    A
L3 . . . . . .    C ADDR    0086H    A
LOC_TMP . . .     C ADDR    0087H    A
P1 . . . . . .    D ADDR    0090H    A
RBE . . . . . .   B ADDR    00C8H.6  A
RBL . . . . . .   D ADDR    00CBH    A
RBS . . . . . .   D ADDR    00CCH    A
RCV . . . . . .   C ADDR    0090H    A
RECEIVE_DMA .     C ADDR    0063H    A
RFL . . . . . .   D ADDR    00CDH    A
RTS . . . . . .   B ADDR    00C8H.5  A
SEND . . . . .    C ADDR    0058H    A
SERIAL_INT . .    C ADDR    0087H    A
SI . . . . . .    B ADDR    00C8H.4  A
SMD . . . . . .   D ADDR    00C9H    A
SP . . . . . .    D ADDR    0081H    A
STAD . . . . .    D ADDR    00CEH    A
TBF . . . . . .   B ADDR    00C8H.7  A
TBL . . . . . .   D ADDR    00DBH    A
TBS . . . . . .   D ADDR    00DCH    A
TCB . . . . . .   D ADDR    00DAH    A
TCON . . . . .    D ADDR    0088H    A
TH1 . . . . . .   D ADDR    008DH    A
TMOD . . . . .    D ADDR    0089H    A
TRANSMIT_DMA .    C ADDR    0079H    A
XMIT . . . . .    C ADDR    0098H    A

REGISTER BANK(S) USED: 0, TARGET MACHINE(S): 8044

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

296166-71

Table 2. PL/M-86 Compiler RUPI/8088 Interface Example

```
SERIES-III PL/M-86 V1.0 COMPILATION OF MODULE RUPI_88
OBJECT MODULE PLACED IN :F1:R88.OBJ
COMPILER INVOKED BY:  PLM86.86 :F1:R88.SRC


            $DEBUG
            $TITLE  ('RUPI/8088 INTERFACE EXAMPLE')

  1         RUPI_88: DO;

  2  1          DECLARE

                LIT                     LITERALLY   'LITERALLY',
                TRUE                    LIT         '01H',
                FALSE                   LIT         '00H',

                RECV_BUFFER(64)         BYTE,
                XMIT_BUFFER(64)         BYTE,
                I                       BYTE,
                WAIT                    BYTE,

                                /* 8237 PORTS*/

                MASTER_CLEAR_37         LIT         'OFFDDH',
                COMMAND_37              LIT         'OFFD8H',
                ALL_MASK_37             LIT         'OFFDFH',
                SINGLE_MASK_37          LIT         'OFFDAH',
                STATUS_37               LIT         'OFFD8H',
                REQUEST_REG_37          LIT         'OFFD9H',
                MODE_REG_37             LIT         'OFFDBH',
                CLEAR_BYTE_PTR_37       LIT         'OFFDCH',

                CH0_ADDR                LIT         'OFFD0H',
                CH0_COUNT               LIT         'OFFD1H',
                CH1_ADDR                LIT         'OFFD2H',
                CH1_COUNT               LIT         'OFFD3H',
                CH2_ADDR                LIT         'OFFD4H',
                CH2_COUNT               LIT         'OFFD5H',
                CH3_ADDR                LIT         'OFFD6H',
                CH3_COUNT               LIT         'OFFD7H',

                        /* 8237 BIT ASSIGNMENTS */

                CH0_SEL                 LIT         '00H',
                CH1_SEL                 LIT         '01H',
                CH2_SEL                 LIT         '02H',
                CH3_SEL                 LIT         '03H',
                WRITE_XFER              LIT         '04H',
                READ_XFER               LIT         '08H',
                DEMAND_MODE             LIT         '00H',
                SINGLE_MODE             LIT         '40H',
                BLOCK_MODE              LIT         '80H',
                SET_MASK                LIT         '04H',

            $EJECT
                            /* 8259 PORTS */

               STATUS_POLL_59           LIT        'OFFE0H',
               ICW1_59                  LIT        'OFFE0H',
               OCW1_59                  LIT        'OFFE1H',
               OCW2_59                  LIT        'OFFE0H',
               OCW3_59                  LIT        'OFFE0H',
               ICW2_59                  LIT        'OFFE1H',
               ICW3_59                  LIT        'OFFE1H',
               ICW4_59                  LIT        'OFFE1H';

                        /* INTERRUPT SERVICE ROUTINE */

  3  1       OFF_RECV_DMA:   PROCEDURE   INTERRUPT 32;

  4  2          OUTPUT(SINGLE_MASK_37)=40H;
  5  2          WAIT=FALSE;
  6  2          END;
```

296166-4

Table 2. PL/M-86 Compiler RUPI/8088 Interface Example (Continued)

```
   7   1          DISABLE;

                          /* INITIALIZE 8237 */

   8   1          OUTPUT(MASTER_CLEAR_37)      =0;
   9   1          OUTPUT(COMMAND_37)           =040H;
  10   1          OUTPUT(ALL_MASK_37)          =0FH;
  11   1          OUTPUT(MODE_REG_37)          =(SINGLE_MODE OR WRITE_XFER OR CH0_SEL);
  12   1          OUTPUT(MODE_REG_37)          =(SINGLE_MODE OR READ_XFER OR CH1_SEL);
  13   1          OUTPUT(CLEAR_BYTE_PTR_37)    =0;
  14   1          OUTPUT(CH0_ADDR)             =00H;
  15   1          OUTPUT(CH0_ADDR)             =40H;
  16   1          OUTPUT(CH0_COUNT)            =64;
  17   1          OUTPUT(CH0_COUNT)            =00;
  18   1          OUTPUT(CH1_ADDR)             =40H;
  19   1          OUTPUT(CH1_ADDR)             =40H;
  20   1          OUTPUT(CH1_COUNT)            =64;
  21   1          OUTPUT(CH1_COUNT)            =00;

                          /* INITIALIZE 8259 */

  22   1          OUTPUT(ICW1_59)              =13H; /*SINGLE MODE, EDGE TRIGGERED
                                                       INPUT, 8086 INTERRUPT TYPE*/
  23   1          OUTPUT(ICW2_59)              =20H; /*INTERRUPT TYPE 32*/
  24   1          OUTPUT(ICW4_59)              =03H; /*AUTO-EOI*/
  25   1          OUTPUT(OCW1_59)              =0FEH; /*ENABLE INTERRUPT LEVEL 0*/

        $EJECT
  26   1          CALL SET$INTERRUPT  (32,OFF_RECV_DMA); /*LOAD INTERRUPT VECTOR LOCATION*/

  27   1          XMIT_BUFFER(0)=64; /*THE FIRST BYTE IN THE BLOCK OF DATA IS THE NUMBER
                                       OF BYTES TO BE TRANSFERED; NOT INCLUDING THE FIRST BYTE*/

  28   1          DO  I= 1 TO 64;   /* FILL UP THE XMIT_BUFFER WITH DATA */
  29   2          XMIT_BUFFER(I)=I;
  30   2          END;

  31   1          OUTPUT(ALL_MASK_37)=0FCH;    /*ENABLE CHANNEL 1 AND 2 */

  32   1          ENABLE;

  33   1          WAIT=TRUE;
  34   1          DO WHILE WAIT;
  35   2          END;                  /* A BLOCK OF DATA WILL BE TRANSFERRED TO THE RUPI.
                                           WHEN THE RUPI RECEIVES A BLOCK OF DATA IT WILL
                                           SEND IT TO THE 8088 MEMORY AND INTERRUPT THE 8088.
                                           THE INTERRUPT SERVICE ROUTINE WILL SHUT OFF THE DMA
                                           CONTROLLER AND SET 'WAIT' FALSE */

  36   1          DO WHILE 1;
  37   2          END;


  38   1      END;



MODULE INFORMATION:

     CODE AREA SIZE     = 00D7H    215D
     CONSTANT AREA SIZE = 0000H      0D
     VARIABLE AREA SIZE = 0082H    130D
     MAXIMUM STACK SIZE = 001EH     30D
     124 LINES READ
     0 PROGRAM WARNINGS
     0 PROGRAM ERRORS

END OF PL/M-86 COMPILATION
```

296166-5

A HIGH PERFORMANCE NETWORK USING THE 8044

2.0 INTRODUCTION

This section describes the design of an SDLC data link using the 8044 (RUPI) to implement a primary station and a secondary station. The design was implemented and tested. The following discussion assumes that the reader understands the 8044 and SDLC. This section is divided into two parts. First the data link design example is discussed. Second the software modules used to implement the data link are described. To help the reader understand the discussion of the software, flow charts and software listings are displayed in Appendix A and Appendix B, respectively.

APPLICATION DESCRIPTION

This particular data link design example uses a two wire half-duplex multidrop topology as shown in Figure 4. In an SDLC multidrop topology the primary station communicates with each secondary station. The secondary stations communicate only to the primary. Because of this hierarchial architecture, the logical topology for an SDLC multidrop is a star as shown in Figure 5. Although the physical topology of this data link is multidrop, the easiest way to understand the information flow is to think of the logical (star) topology. The term data link in this case refers to the logical communication pathways between the primary station and the secondary stations. The data links are shown in Figure 5 as two way arrows.

The application example uses dumb async terminals to interface to the SDLC network. Each secondary station has an async terminal connected to it. The secondary stations are in effect protocol converters which allows any async terminal to communicate with any other async terminal on the network. The secondary stations use an 8044 with a UART to convert SDLC to async. Figure 6 displays a block diagram of the data link. The primary station, controls the data link. In addition to data link control the primary provides a higher level layer which is a path control function or networking layer. The primary serves as a message exchange or switch. It receives information from one secondary station and retransmits it to another secondary station. Thus a virtual end to end connection is made between any two secondary stations on the network.

Three separate software modules were written for this network. The first module is a Secondary Station Driver (SSD) which provides an SDLC data link interface and a user interface. This module is a general purpose driver which requires application software to run it. The user interface to the driver provides four functions: OPEN, CLOSE, TRANSMIT, and SIU_RECV. Using these four functions properly will allow any application software to communicate over this SDLC data link without knowing the details of SDLC. The secondary station driver uses the 8044's AUTO mode.

The second module is an example of application software which is linked to the secondary station driver. This module drives the 8215A, buffers data, and interfaces with the secondary station driver's user interface.

The third module is a primary station, which is a standalone program (i.e., it is not linked to any other module). The primary station uses the 8044's NON-AUTO or FLEXIBLE mode. In addition to controlling the data link it acts as a message switch. Each time a secondary station transmits a frame, it places the destination address of the frame in the first byte of the information or I field. When the primary station receives a frame, it removes the first byte in the I field and retransmits the frame to the secondary station whose address matches this byte.

This network provides two complete layers of the OSI (Open Systems Interconnection) reference model: the physical layer and the data link layer. The physical layer implementation uses the RS-422 electrical interface. The mechanical medium consists of ribbon cable and connectors. The data link layer is defined by SDLC. SDLC's use of acknowledgements and frame numbering guarantees that messages will be received in the same order in which they were sent. It also guarantees message integrity over the data link. However this network will not guarantee secondary to secondary message delivery, since there are acknowledgements between secondary stations.

2.1 Hardware

The schematic of the hardware is given in Figure 7. The 8251A is used as an async communications controller, in support of the 8044. TxRDY and RxRDY on the 8251A are both tied to the two available external interrupts of the 8044 since the secondary station driver is totally interrupt driven. The 8044 buffers the data and some variables in a 2016 (2K x 8 static RAM). The 8254 programmable interval timer is employed as a programmable baud rate generator and system clock driver for the 8251A. The third output from the 8254 could be used as an external baud rate generator for the 8044. The 2732A shown in the diagram was not used

since the software for both the primary and secondary stations used far less than the 4K bytes provided on the 8744. For the async interface, the standard RS-232 mechanical and electrical interface was used. For the SDLC channel, a standard two wire three state RS-422 driver is used. A DIP switch connected to one of the available ports on the 8044 allows the baud rate, parity, and stop bits to be changed on the async interface. The primary station hardware does not use the USART, 8254, nor the RS-232 drivers.

2.2 SDLC Basic Repertoire

The SDLC commands and responses implemented in the data link include the SDLC Basic Repertoire as defined in the IBM SDLC General Information manual. Table 3 shows the commands and responses that the primary and the secondary station in this data link design recognize and send.

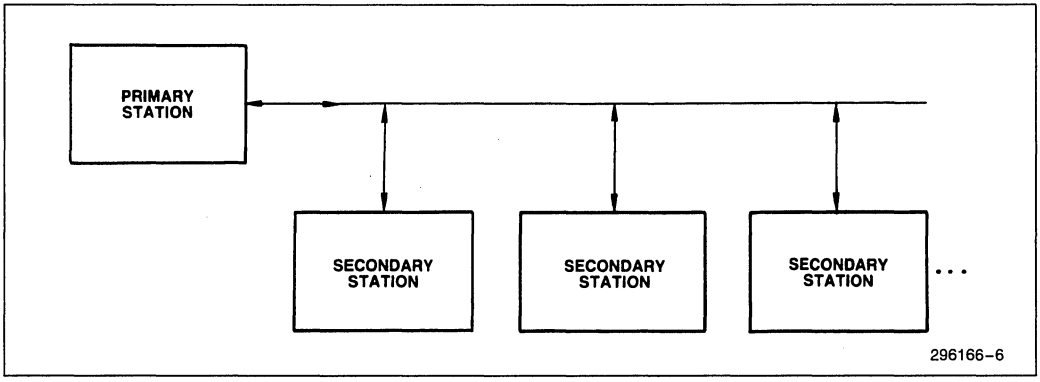


Figure 4. SDLC Multidrop Topology

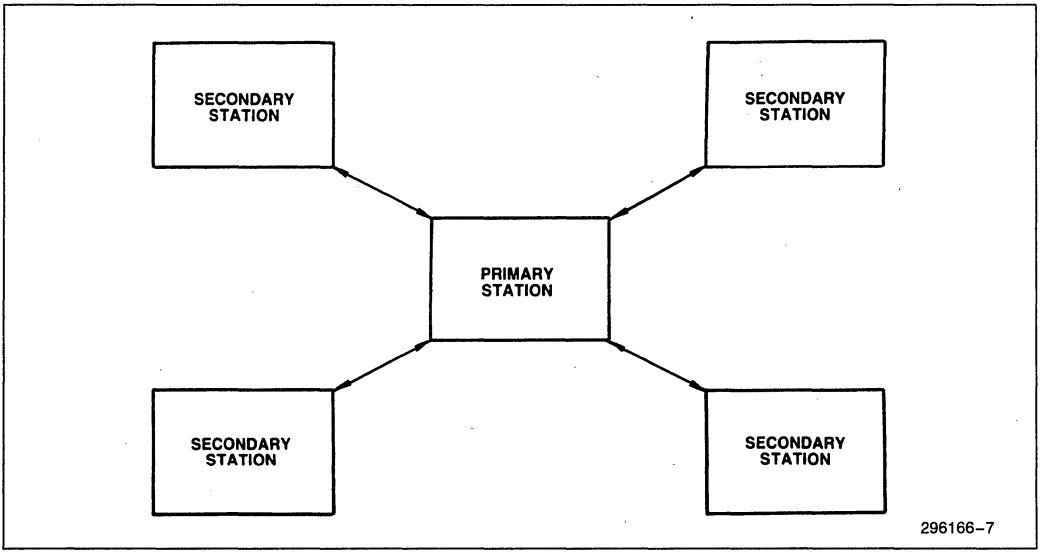


Figure 5. SDLC Logical Topology

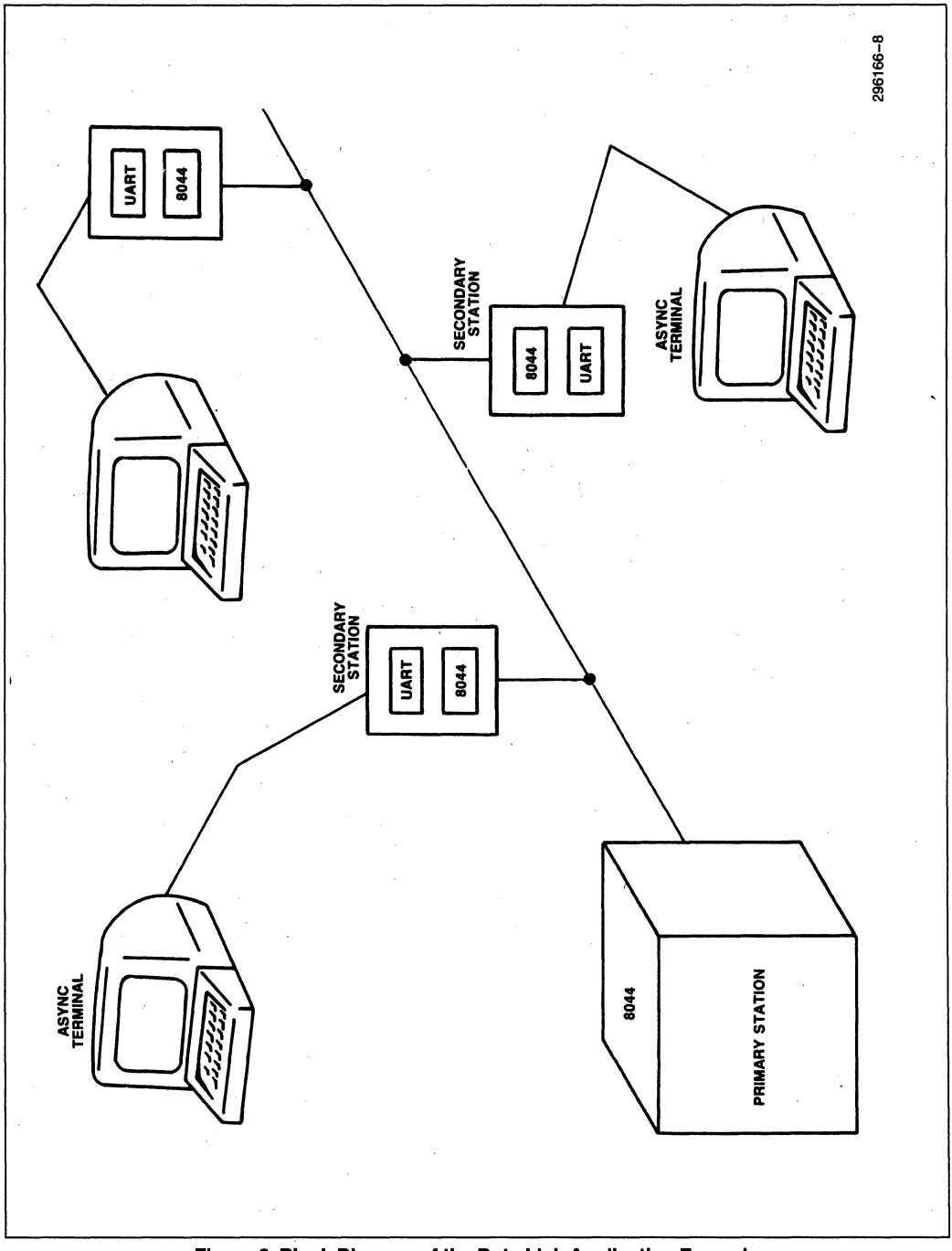


Figure 6. Block Diagram of the Data Link Application Example

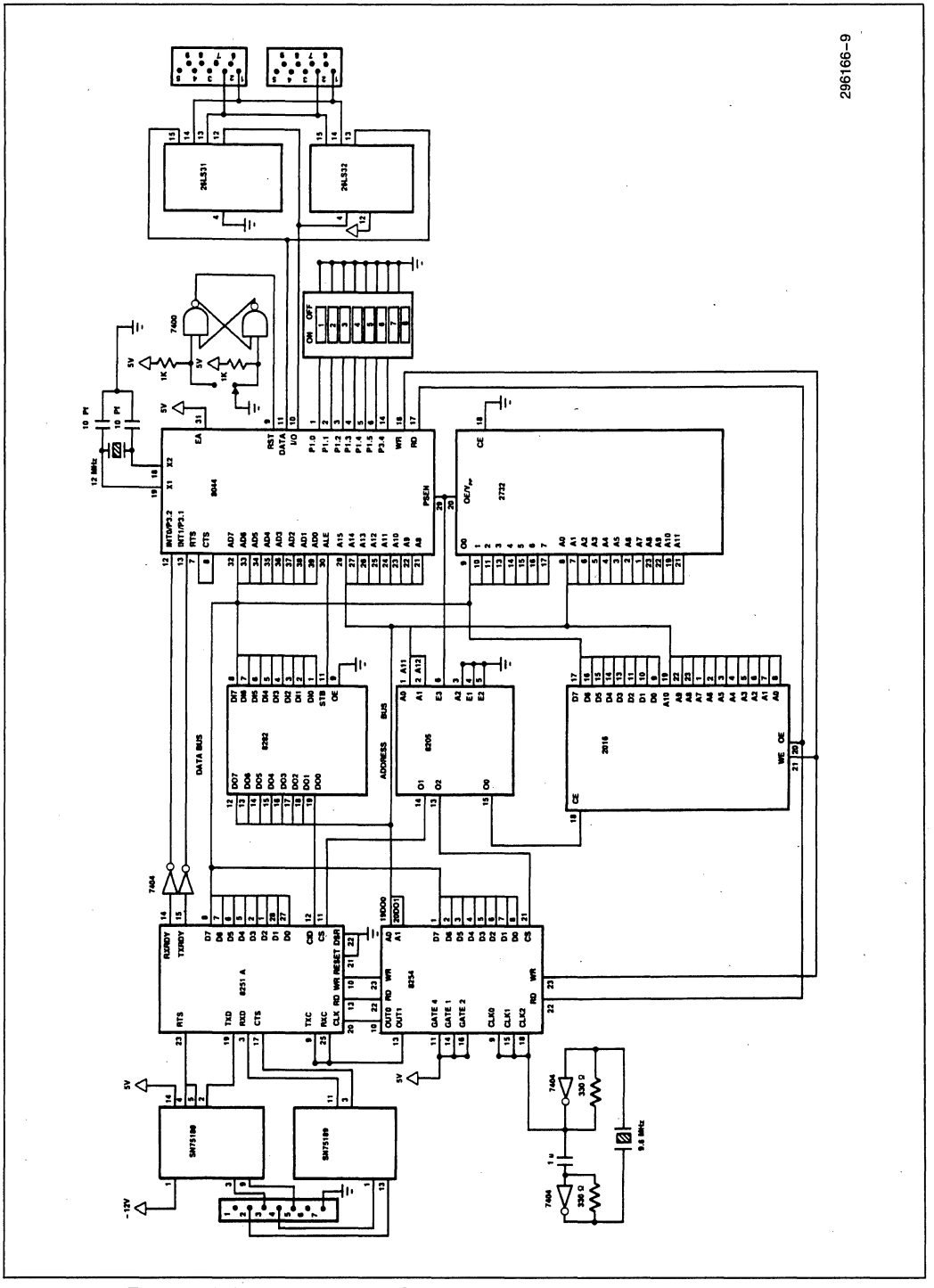


Figure 7. Schematic of Async/SDLC Secondary Station Protocol Converter

Table 3. Data Link Commands and Responses Implemented for This Design

Primary Station

	Responses Recognized	Commands Sent
Unnumbered	UA DM FRMR *RD	SNRM DISC
Supervisory	RR RNR	RR RNR
Information	I	I

Secondary Station

	Commands Recognized	Responses Sent
Unnumbered	SNRM DISC *TEST	UA DM FRMR *RD *TEST
Supervisory	RR RNR REJ	RR RNR
Information	I	I

*not included in the SDLC Basic Repertoire

The term command specifically means all frames which the primary station transmits and the secondary stations receive. Response refers to frames which the secondary stations transmit and the primary station receives.

NUMBER OF OUTSTANDING FRAMES

This particular data link design only allows one outstanding frame before it must receive an acknowledgement. Immediate acknowledgement allows the secondary station drivers to use the AUTO mode. In addition, one outstanding frame uses less memory for buffering, and the software becomes easier to manage.

2.3 Secondary Station Driver using AUTO Mode

The 8044 secondary station driver (SSD) was written as a general purpose SDLC driver. It was written to be linked to an application module. The application software implements the actual application in addition to interfacing to the SSD. The main application could be, a printer or plotter, a medical instrument, or a terminal. The SSD is independent of the main application, it just provides the SDLC communications. Existing 8051 applications could add high performance SDLC communications capability by linking the SSD to the existing software and providing additional software to be able to communicate with the SSD.

DATA LINK INTERFACE AND USER INTERFACE STATES

The SSD has two software interfaces: a data link interface and a user interface as shown in Figure 8. The data link interface is the part of the software which controls the SDLC communications. It handles link access, command recognition/response, acknowledgements, and error recovery. The user interface provides four functions: OPEN, CLOSE, TRANSMIT, and SIU_RECV. These are the only four functions which the application software has to interface in order to communicate using SDLC. These four functions are common to many I/O drivers like floppy and hard disks, keyboard/CRT, and async communication drivers.

The data link and the user interface each have their own states. Each interface can only be in one state at any time. The SSD uses the states of these two interfaces to help synchronize the application module to the data link.

There are three states which the secondary station data link interface can be in: Logical Disconnect State (L_D_S), Frame Reject State (FRMR_S), and the Information Transfer State (I_T_S). The Logical Disconnect State is when a station is physically connected to the channel but either the primary or secondary have not agreed to enter the Information Transfer State. Both the primary and the secondary stations synchronize to enter into the Information Transfer State. Only when the secondary station is in the I_T_S is it able to transfer data or information to the primary. The Frame Reject State (FRMR_S) indicates that the secondary station has lost software synchronization with the primary or encountered some kind of error condition. When the secondary station is in the FRMR_S, the primary station must reset the secondary to resynchronize.

The user interface has two states, open or closed. In the closed state, the user program does not want to communicate over the network. The communications channel is closed and not available for use. The secondary station tells the primary this by responding to all commands with DM. The primary continues to poll the secondary in case it wants to enter the I_T_S state. When the user program begins communication over the data link it goes into the open state. It does this by calling the OPEN procedure. When the user interface is in the open state it may transfer information to the primary.

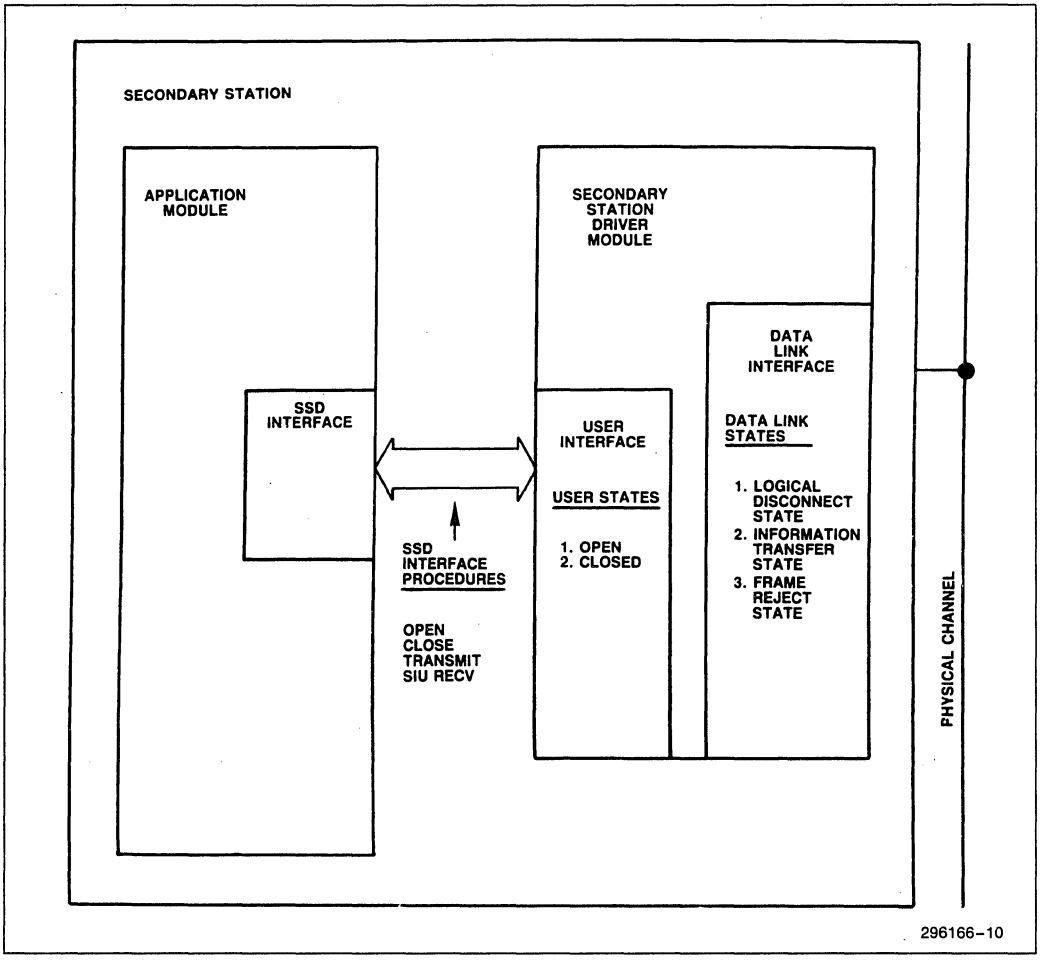


Figure 8. Secondary Station Software Modules

SECONDARY STATION COMMANDS, RESPONSES AND STATE TRANSITIONS

Table 4 shows the commands which the secondary station recognizes and the responses it generates. The first row in Table 4 displays commands the secondary station recognizes and each column shows the potential responses with respect to secondary station. For example, if the secondary is in the Logical Disconnect State it will only respond with DM, unless it receives a SNRM command and the user state is open. If this is the case, then the response will be UA and the secondary station will move into the I_T_S.

Figure 9 shows the state diagram of the secondary station. When power is first applied to the secondary station, it goes into the Logical Disconnect State. As mentioned above, the I_T_S is entered when the secondary station receives a SNRM command and the user state is open. The secondary responds with UA to let the primary know that it has accepted the SNRM and is entering the I_T_S. The I_T_S can go into either the L_D_S or the FRMR_S. The I_T_S goes into the L_D_S if the primary sends the secondary DISC. The secondary has to respond with UA, and then goes into the L_D_S. If the user interface changes from open to close state, then the secondary sends RD. This causes the primary to send a DISC.

The FRMR_S is entered when a secondary station is in the I_T_S and either one of the following conditions occurs.

— A command can not be recognized by the secondary station.
— There is a buffer overrun.
— The Nr that was received from the primary station is invalid.

The secondary station cannot leave the FRMR_S until it receives a SNRM or a DISC command.

SOFTWARE DESCRIPTION OF THE SSD

To aid in following the description of the software, the reader may either look at the flow charts which are given for each procedure, or read the PL/M-51 listing provided in Appendix A.

A block diagram of the software structure of the SSD is given in Figure 10. A complete module is identified by the dotted box, and a procedure is identified by the solid box. Therefore the SIU_RECV procedure is not included in the SSD module, it exists in the application software. Two or more procedures connected by a solid line means the procedure above calls the procedure below. Transmit, Power_on_D, Close, and Open are all called by the application software. Procedures without any solid lines connected above are interrupt procedures. The only interrupt procedure in the SSD module is the SIU_INT.

The entire SSD module is interrupt driven. Its design allows the application program to handle real time events or just dedicate more CPU time to the application program. The SIU_INT is the only interrupt procedure in the SSD. It is automatically entered when an SIU interrupt occurs. This particular interrupt can be the lowest priority interrupt in the system.

Table 4. Secondary Station Responses to Primary Station Commands

Data Link States	Primary Station-Commands					
	I	RR	RNR	SNRM	DISC	TEST
Information Transfer State	I RR RNR RD FRMR	I RR RNR RD FRMR	I RR RNR RD FRMR	 RD UA	 UA	 RD Test
Logical Disconnect State	DM	DM	DM	DM UA	DM	DM
Frame Reject State	FRMR	FRMR	FRMR	 UA	 UA	FRMR

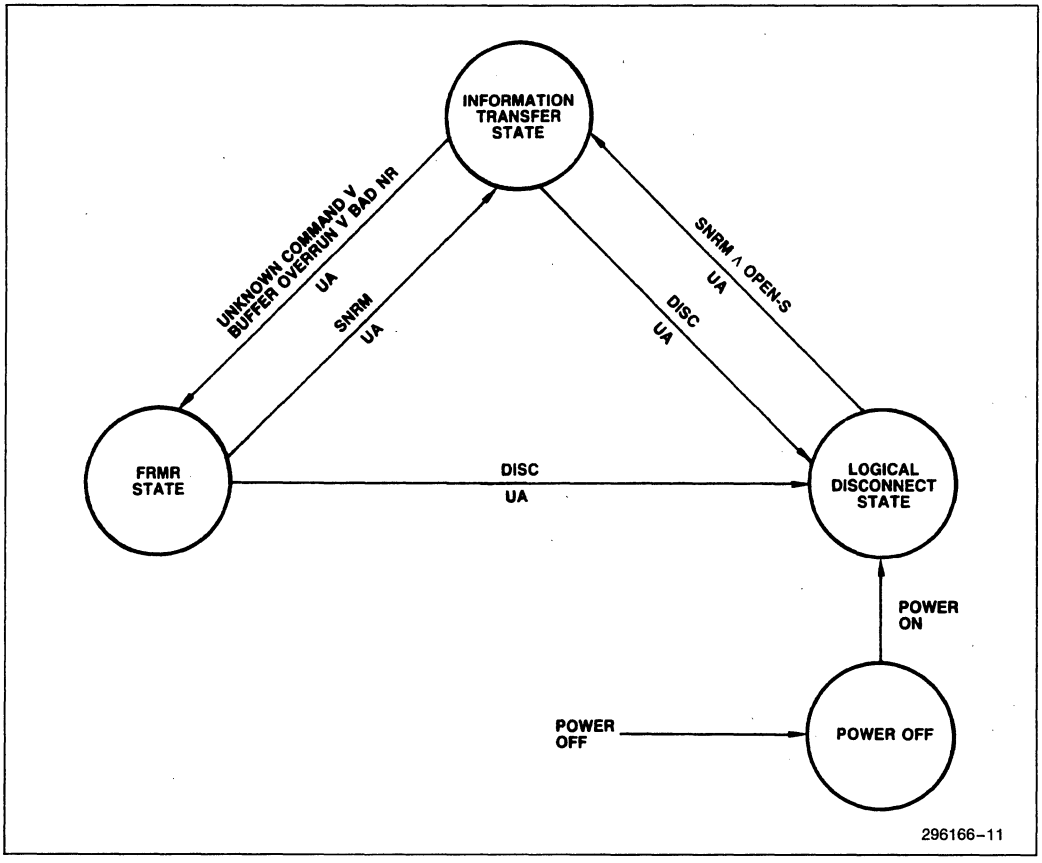


Figure 9. State Diagram of Secondary Station

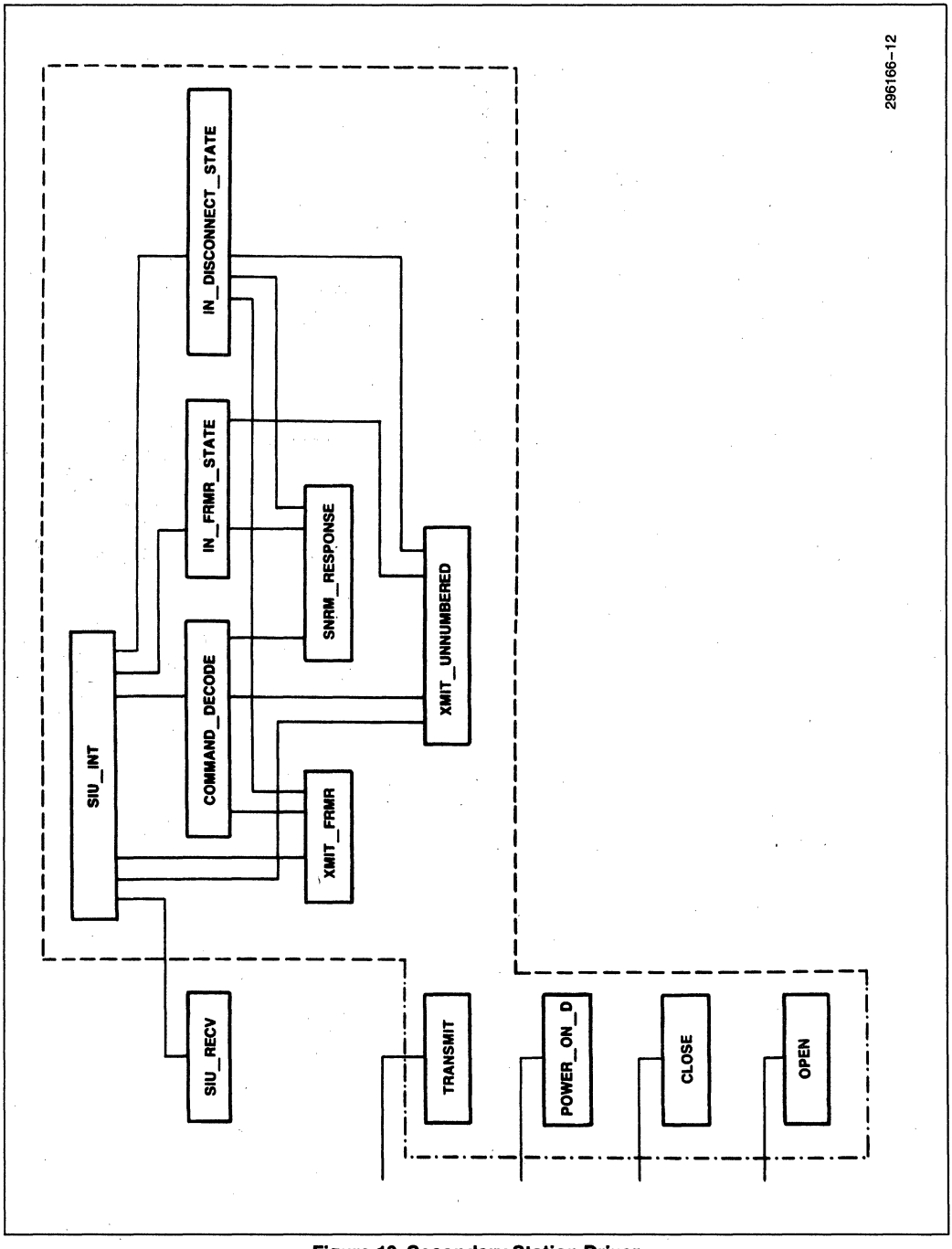


Figure 10. Secondary Station Driver

SSD INITIALIZATION

Upon reset the application software is entered first. The application software initializes its own variables then calls Power_On_D which is the SSD's initialization routine. The SSD's initialization sets up the transmit and receive data buffer pointers (TBS and RBS), the receive buffer length (RBL), and loads the State variables. The STATION_STATE begins in the L_D_S state, and the USER_STATE begins in the closed state. Finally Power_On_D initializes XMIT_BUFFER_EMPTY which is a bit flag. This flag serves as a semaphore between the SSD and the application software to indicate the status of the on chip transmit buffer. The SSD does not set the station address. It is the application software's responsibility to do this. After initialization, the SSD is read to respond to all of the primary station commands. Each time a frame is received with a matching station address and a good CRC, the SIU_INT procedure is entered.

SIU_INT PROCEDURE

The first thing the SIU_INT procedure clears is the serial interrupt_bit (SI) in the STS register. If the SIU_INT procedure returns with this bit set, another SI interrupt will occur.

The SIU_INT procedure is branches three independent cases. The first case is entered if the STATION_STATE is not in the I_T_S. If this is true, then the SIU is not in the AUTO mode, and the CPU will have to respond to the primary on its own. (Remember that the AUTO mode is entered when the STATION_STATE enters into I_T_S.) If the STATION_STATE is in the I_T_S, then either the SIU has just left the AUTO mode, or is still in the AUTO mode. This is the second and third case, respectively.

In the first case, if the STATION_STATE is not in the I_T_S, then it must be in either the L_D_S or the FRMR_S. In either case a separate procedure is called based on which state the station is in. The In_Disconnect_State procedure sends to the primary a DM response, unless it received a SNRM command and the USER_STATE equals open. In that case the SIU sends a UA and enters into the I_T_S. The In_FRMR_State procedure will send the primary the FRMR response unless it received either a DISC or an SNRM. If the primary's command was a DISC, then the secondary will send a UA and enter into the L_D_S. If the primary's command was a SNRM, then the secondary will send a UA, enter into the I_T_S, and clear NSNR register.

For the second case, if the STATION_STATE is in the I_T_S but the SIU left the AUTO mode, then the CPU must determine why the AUTO mode was exited, and generate a response to the primary. There are four reasons for the SIU to automatically leave the AUTO mode. The following is a list of these reasons, and the responses given by the SSD based on each reason.

1. The SIU has received a command field it does not recognize.

   Response: If the CPU recognizes the command, it generates the appropriate response. If neither the SIU nor the CPU recognize the command, then a FRMR response is sent.

2. The SIU has received a Sequence Error Sent (SES = 1 in NSNR register). Nr(P)≠Ns(S)+1, and Nr(P)≠Ns(S).

   Response: Send FRMR.

3. A buffer overrun has occurred. BOV = 1 in STS register.

   Response: Send FRMR.

4. An I frame with data was received while RPB = 1.

   Response: Go back into AUTO mode and send an AUTO mode response

In addition to the above reasons, there is one condition where the CPU forces the SIU out of the AUTO mode. This is discussed in the SSD's User Interface Procedures section in the CLOSED procedure description

Finally, case three is when the STATION_STATE is in the I_T_S and the AUTO mode. The CPU first looks at the TBF bit. If this bit is 0 then the interrupt may have been caused by a frame which was transmitted and acknowledged. Therefore the XMIT_BUFFER_EMPTY flag is set again, indicating that the application software can transmit another frame.

The other reason this section of code could be entered is if a valid I frame was received. When a good I frame is received the RBE bit equals 0. This means that the receiver is disabled. If the primary were to poll the 8044 while RBE = 0, it would time out since no response would be given. Time outs reduce network throughput. To improve network performance, the CPU first sets RBP, then sets RBE. Now when the primary polls the 8044 an immediate RNR response is given. At this point the SSD calls the application software procedure SIU_RECV and passes the length of the data as a parameter. The SIU_RECV procedure reads the data out of the receive buffer then returns to the SSD module. Now that the receive information has been transferred, RBP can be cleared.

COMMAND_DECODE PROCEDURE

The Command_Decode procedure is called from the SIU_INT procedure when the STATION_STATE = I_T_S and the SIU left the AUTO mode as a result of not being able to recognize the receive control byte. Commands which the SIU AUTO mode does not

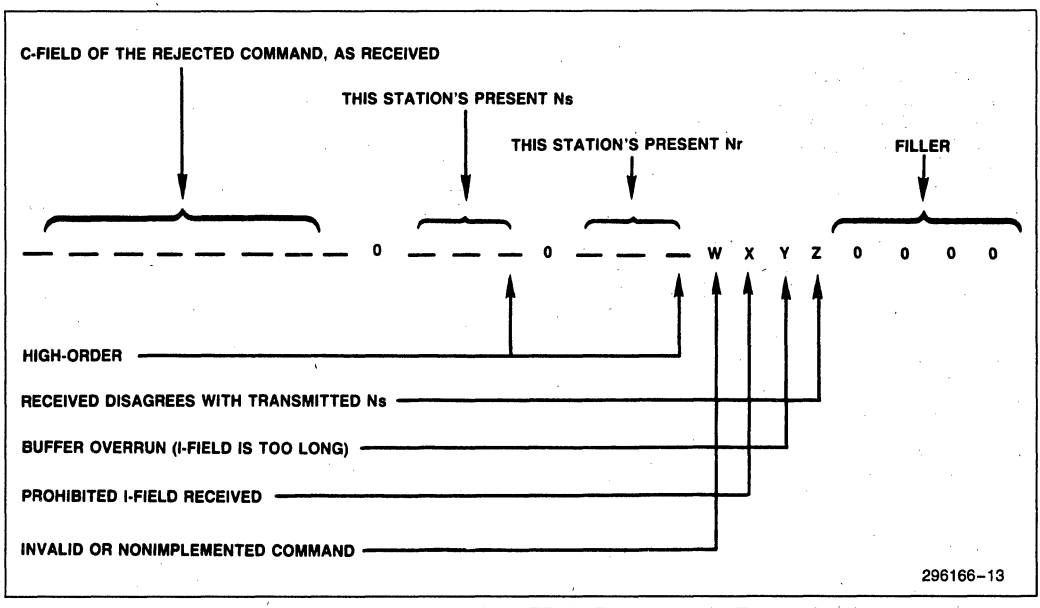


Figure 11. Information Field of the FRMR Response, as Transmitted

recognize are handled here. The commands recognized in this procedure are: SNRM, DISC, and TEST. Any other command received will generate a Frame Reject with the nonimplemented command bit set in the third data byte of the FRMR frame. Any additional unnumbered frame commands which the secondary station is going to implement, should be implemented in this procedure.

IF an SNRM is received the command_decode procedure calls the SNRM_Response procedure. The SNRM_Response procedure sets the STATION_STATE = I_T_S, clears the NSNR register and responds with a UA frame. If a DISC is received, the command_decode procedure sets the STATION_STATE = L_D_S, and responds with a UA frame. When a TEST frame is received, and there is no buffer overrun, the command_decode procedure responds with a TEST frame retransmitting the same data it received. However if a TEST frame is received and there is a buffer overrun, then a TEST frame will be sent without any data, instead of a FRMR with the buffer overrun bit set.

FRAME REJECT PROCEDURES

There are two procedures which handle the FRMR state: XMIT_FRMR and IN_FRMR_STATE. XMIT_FRMR is entered when the secondary station first goes into the FRMR state. The frame reject response frame contains the FRMR response in the command field plus three additional data bytes in the I field. Figure 11 displays the format for the three data bytes in the I field of a FRMR response. The XMIT_FRMR procedure sets up the Frame Reject response frame based on the parameter REASON which is passed to it. Each place in the SSD code that calls the XMIT_FRMR procedure, passes the REASON that this procedure was called, which in turn is communicated to the primary station. The XMIT_FRMR procedure uses three bytes of internal RAM which it initializes for the correct response. The TBS and TBL registers are then changed to point to the FRMR buffer so that when a response is sent these three bytes will be included in the I field.

The IN_FRMR_STATE procedure is called by the SIU_INT procedure when the STATION_STATE already is in the FRMR state and a response is required. The IN_FRMR_STATE procedure will only allow two commands to remove the secondary station from the FRMR state: SNRM and DISC. Any other command which is received while in the FRMR state will result in a FRMR response frame.

XMIT_UNNUMBERED PROCEDURE

This is a general purpose transmit procedure, used only in the FLEXIBLE mode, which sends unnumbered responses to the primary. It accepts the control byte as a parameter, and also expects the TBL register to be set before the procedure is called. This procedure waits until the frame has been transmitted before returning. If

this procedure returned before the transmit interrupt was generated, the SIU_INT routine would be entered. The SIU_INT routine would not be able to distinguish this condition.

SSD's User Interface Procedures—OPEN, CLOSE, TRANSMIT, SIU_RECV—are discussed in the following section.

The OPEN procedure is the simplest of all, it changes the USER_STATE to OPEN_S then returns. This lets the SSD know that the user wants to open the channel for communications. When the SSD receives a SNRM command, it checks the USER_STATE. If the USER_STATE is open, then the SSD will respond with a UA, and the STATION_STATE enters the I_T_S.

The CLOSE procedure is also simple, it changes the USER_STATE to CLOSED_S and sets the AM bit to 0. Note that when the CPU sets the AM bit to 0 it puts the SIU out of the AUTO mode. This event is asynchronous to the events on the network. As a result an I frame can be lost. This is what can happen.

1. AM is set to 0 by the CLOSE Procedure.
2. An I frame is received and an SI interrupt occurs.
3. The SIU_INT procedure enters case 2 (STATION_STATE = I_T_S, and AM = 0).
4. Case 2 detects that the USER_STATE = CLOSED_S, sends an RD response and ignores the fact that an I frame was received.

Therefore it is advised to never call the CLOSE procedure or take the SIU out of the AUTO mode when it is receiving I frames or an I frame will be lost.

For both the TRANSMIT and SIU_RECV procedures, it is the application software's job to put data into the transmit buffer, and take data out of the receive buffer. The SSD does not transfer data in or out of its transmit or receive buffers because it does not know what kind of buffering the application software is implementing. What the SSD does do is notify the application software when the transmit buffer is empty, XMIT_BUFFER_EMPTY = 1, and when the receive buffer is full.

One of the functions that the SSD performs to synchronize the application software to the SDLC data link. However some of the synchronization must also be done by the application software. Remember that the SSD does not want to hang up the application software waiting for some event to occur on the SDLC data link, therefore the SSD always returns to the application software as soon as possible.

For example, when the application software calls the OPEN procedure, the SSD returns immediately. The application software thinks that the SDLC channel is now open and it can transmit. This is not the case. For the channel to be open, the SSD must receive an SNRM from the primary and respond with a UA. However, the SSD does not want to hang up the application software waiting for an SNRM from the primary before returning from the OPEN procedure. When the TRANSMIT procedure is called, the SSD expects the STATION_STATE to be in the I_T_S. If it isn't, the SSD refuses to transmit the data. The TRANSMIT procedure first checks to see if the USER_STATE is open. If not, the USER_STATE_CLOSED parameter is passed back to the application module. The next thing TRANSMIT checks is the STATION_STATE. If this is not open, then TRANSMIT passes back LINK_DISCONNECTED. This means that the USER_STATE is open, but the SSD hasn't received an SNRM command from the primary yet. Therefore, the application software should wait awhile and try again. Based on network performance, one knows the maximum amount of time it will take for a station to be polled. If the application software waits this length of time and tries again but still gets a LINK_DISCONNECTED parameter passed back, higher level recovery must be implemented.

Before loading the transmit buffer and calling the TRANSMIT procedure, the application software must check to see that XMIT_BUFFER_EMPTY = 1. This flag tells the application software that it can write new data into the transmit buffer and call the TRANSMIT procedure. After the application software has verified that XMIT_BUFFER_EMPTY = 1, it fills the transmit buffer with the data and calls the TRANSMIT procedure passing the length of the buffer as a parameter. The TRANSMIT procedure checks for three reasons why it might not be able to transmit the frame. If any of these three reasons are true, the TRANSMIT procedure returns a parameter explaining why it couldn't send the frame. If the application software receives one of these responses, it must rectify the problem and try again. Assuming these three conditions are false, then the SSD clears XMIT_BUFFER_EMPTY, attempts to send the data and returns the parameter DATA_TRANSMITTED. XMIT_BUFFER_EMPTY will not be set to 1 again until the data has been transmitted and acknowledged.

The SIU_RECV procedure must be incorporated into the application software module. When a valid I frame is received by the SIU, it calls the SIU_RECV procedure and passes the length of the received data as a parameter. The SIU_RECV procedure must remove all of the data from the receive buffer before returning to the SIU_INT procedure.

LINKING UP TO THE SSD

Figure 12 shows the necessary parts to include in a PL/M-51 application program that will be linked to the SSD module. RL51 is used to link and locate the SSD and application modules. The command line used to do this is:

```
RL51 SSD.obj,filename.obj,PLM51.LIB TO
filename & RAMSIZE(192)

$registerbank(0)
user$mod: do;
$include (reg44.dcl)
declare
 lit                literally 'literally',
 buffer_length      lit           '60',
 siu_xmit_buffer
 (buffer_length)    byte    external idata,
 siu_recv_buffer
 (buffer_length)    byte        external,
 xmit_buffer_empty bit          external;

/* external procedures */

power_on_d: procedure  external;
end power_on_d;

close: procedure       external using 1;
end close;

open: procedure        external using 1;
end open;

transmit: procedure
(xmit_buffer_length) byte       external;
 declare xmit_buffer_length      byte;
end transmit;

/* local procedures */

siu_recv: procedure (length)     using 1;
public
 declare    length  byte,
        •
        •
        •
end siu_recv;
```

Figure 12. Applications Module Link Information

PL/M-51 AND REGISTER BANKS

The 8044 has four register banks. PL/M-51 assumes that an interrupt procedure never uses the same bank as the procedure it interrupts. The USING attribute of a procedure, or the $REGISTERBANK control, can be used to ensure that.

The SSD module uses the $REGISTERBANK(1) attribute. Some procedures are modified with the USING attribute based on the register bank level of the calling procedure.

2.4 Application Module; ASYNC to SDLC Protocol Converter

One of the purposes of this application module is to demonstrate how to interface software to the SSD. Another purpose is to implement and test a practical application. This application software performs I/O with an async terminal through a USART, buffers data, and also performs I/O with the SSD. In addition, it allows the user on the async terminal to: set the station address, set the destination address, and go online and offline. Setting the station address sets the byte in the STAD register. The destination address is the first byte in the I field. Going online or offline results in either calling the OPEN or CLOSE procedure respectively.

After the secondary station powers up, it enters the 'terminal mode', which accepts data from the terminal. However, before any data is sent, the user must configure the station. The station address and destination address must be set, and the station must be placed online. To configure the station the ESC character is entered at the terminal which puts the protocol converter into the 'configure mode'. Figure 13 shows the menu which appears on the terminal screen.

```
(/)8044 Secondary Station
 /

1 - Set the Station Address
2 - Set the Destination Address
3 - Go Online
4 - Go Offline
5 - Return to terminal mode
    Enter option __
```

Figure 13. Menu for the Protocol Converter

In the terminal mode data is buffered up in the secondary station. A Line Feed character 'LF' tells the secondary station to send an I frame. If more than 60 bytes are buffered in the secondary station when a 'LF' is received, the applications software packetizes the data into 60 bytes or less per frame. If a LF is entered when the station is offline, an error message comes on the screen which says 'Unable to Get Online'.

The secondary station also does error checking on the async interface for Parity, Framing Error, and Overrun Error. If one of these errors are detected, an error message is displayed on the terminal screen.

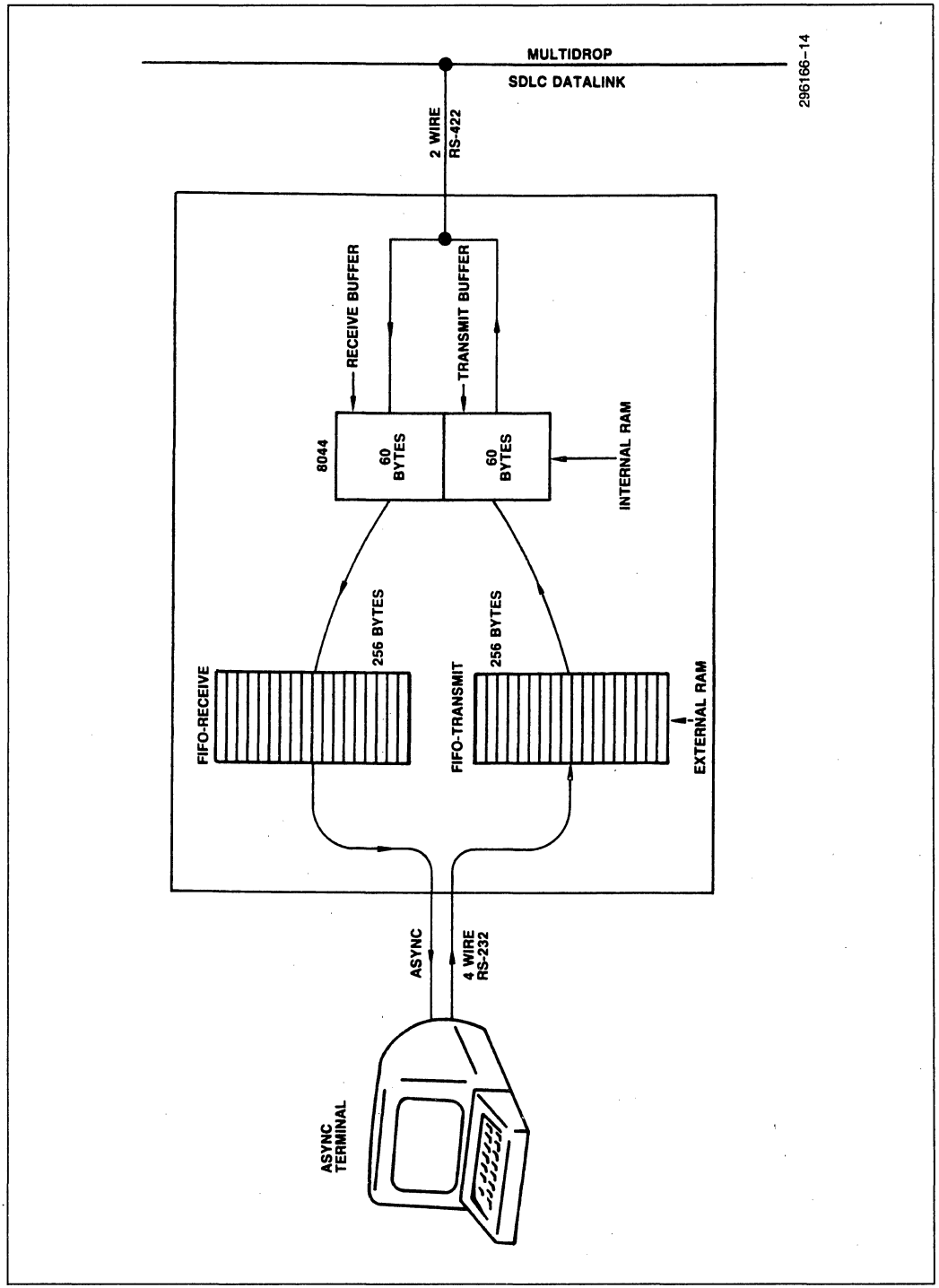


Figure 14. Block Diagram of Secondary Station Protocol Converter Illustrating Buffering

BUFFERING

There are two separate buffers in the application module: a transmit buffer and a receive buffer. The transmit buffer receives data from the USART, and sends data to the SSD. The receive buffer receives data from the SSD, and transmits data to the USART. Each buffer is a 256 byte software FIFO. If the transmit FIFO becomes full and no 'LF' character is received, the secondary station automatically begins sending the data. In addition, the application modules will shut off the terminal's transmitter using CTS until the FIFO has been partially emptied. A block diagram of the buffering for the protocol converter is given in Figure 14.

APPLICATION MODULE SOFTWARE

A block diagram of the application module software is given in Figure 15. There are three interrupt routines in this module: USART_RECV_INT, USART_XMIT_INT, and TIMER_0_INT. The first two are for servicing the USART. TIMER_0_INT is used if the TRANSMIT procedure in the SSD is called and does not return with the DATA_TRANSMITTED parameter. TIMER_0_INT employs Timer 0 to wait a finite amount of time before trying to transmit again. The highest priority interrupt is USART_RECV_INT. The main program and all the procedures it calls use register bank 0, USART_XMIT_INT and TIMER_0_INT and FIFO_R_OUT use bank 1, while USART_RECV_INT and all the procedures it calls use register bank 2.

POWER_ON PROCEDURE

The Power_On procedure initializes all of the chips in the system including the 8044. The 8044 is initialized to use the on-chip DPLL with NRZI coding, PreFrame Sync, and Timer 1 auto reload at a baud rate of 62.5 Kbps. The 8254 and the 8251A are initialized next based on the DIP switch values attached to port 1 on the 8044. Variables and pointers are initialized, then the SSD's Power-Up Procedure, Power_On_D, is called. Finally, the interrupt system is enabled and the main program is entered.

MAIN PROGRAM

The main program is a simple loop which waits for a frame transmit command. A frame transmit command is indicated when the variable SEND_DATA is greater than 0. The value of SEND_DATA equals the number of 'LF' characters in the transmit FIFO, hence it also indicates the number of frames pending transmission. Each time a frame is sent, SEND_DATA is decremented by one. Thus when SEND_DATA is greater than 0, the main program falls down into the next loop which polls the XMIT_BUFFER_EMPTY bit. When XMIT_BUFFER_EMPTY equals 1, the SIU_XMIT_BUFFER can be loaded. The first byte in the buffer is loaded with the destination address while the rest of the buffer is loaded with the data. Bytes are removed from the transmit FIFO and placed into the SIU_XMIT_BUFFER until one of three things happen: 1. a 'LF' character is read out of the FIFO, 2. the number of bytes loaded equals the size of the SIU_XMIT_BUFFER, or 3. the transmit FIFO is empty.

After the SIU_XMIT_BUFFER is filled, the SSD TRANSMIT procedure is called and the results from the procedure are checked. Any result other than DATA_TRANSMITTED will result in several retries within a finite amount of time. If all the retries fail, then the LINK_DISC procedure is called which sends a message to the terminal, 'Unable to Get Online'.

USART_RECV_INT PROCEDURE

When the 8251A receives a character, the RxRDY pin on the 8251A is activated, and this interrupt procedure is entered. The routine reads the USART status register to determine if there are any errors in the character received. If there are, the character is discarded and the ERROR procedure is called which prints the type of error on the screen. If there are no errors, the received character is checked to see if it's an ESC. If it is an ESC, the MENU procedure is called which allows the user to change the configuration. If neither one of these two conditions exists, the received character is inserted into the transmit FIFO. The received character may or may not be echoed back to the terminal based on the dip switch settings.

TRANSMIT FIFO

The transmit FIFO consists of two procedures: FIFO_T_IN and FIFO_T_OUT. FIFO_T_IN inserts a character into the FIFO, and FIFO_T_OUT removes a character from the FIFO. The FIFO itself is an array of 256 bytes called FIFO_T. There are two pointers used as indexes in the array to address the characters: IN_PTR_T and OUT_PTR_T. IN_PTR_T points to the location in the array which will store the next byte of data inserted. OUT_PTR_T points to the next byte of data removed from the array. Both IN_PTR_T and OUT_PTR_T are declared as bytes. The FIFO_T_IN procedure receives a character from the USART_RECV_INT procedure and stores it in the array location pointed to by IN_PTR_T, then IN_PTR_T is incremented. Similarly, when FIFO_T_OUT is called by the main program, to load the SIU_XMIT_BUFFER, the byte in the array

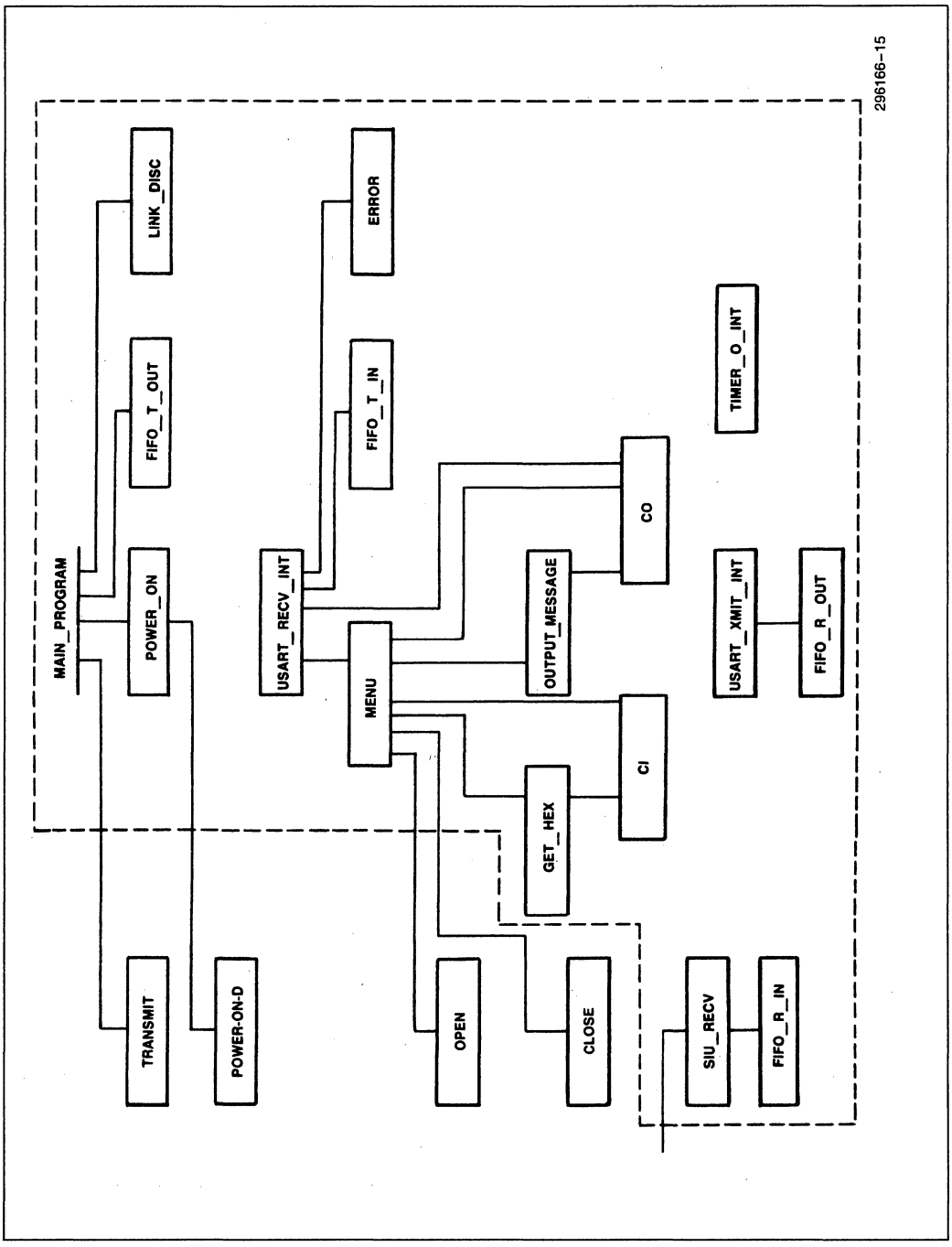


Figure 15. Block Diagram of User Software

pointed to by OUT_PTR_T is read, then OUT_PTR_T is incremented. Since IN_PTR_T and OUT_PTR_T are always incremented, they must be able to roll over when they hit the top of the 256 byte address space. This is done automatically by having both IN_PTR_T and OUT_PTR_T declared as bytes. Each character inserted into the transmit FIFO is tested to see if it's a LF. If it is a LF, the variable SEND_DATA is incremented, which lets the main program know that it is time to send an I frame. Similarly each character removed from the FIFO is tested. SEND_DATA is decremented for every LF character removed from the FIFO.

IN_PTR_T and OUT_PTR_T are also used to indicate how many bytes are in the FIFO, and whether it is full or empty. When a character is placed into the FIFO and IN_PTR_T is incremented, the FIFO is full if IN_PTR_T equals OUT_PTR_T. When a character is read from the FIFO and OUT_PTR_T is incremented, the FIFO is empty if OUT_PTR_T equals IN_PTR_T. If the FIFO is neither full nor empty, then it is in use. A byte called BUFFER_STATUS_T is used to indicate one of these three conditions. The application module uses the buffer status information to control the flow of data into and out of the FIFO. When the transmit FIFO is empty, the main program must stop loading bytes into the SIU_XMIT_BUFFER. Just before the FIFO is full, the async input must be shut off using CTS. Also, if the FIFO is full and SEND_DATA = 0, then SEND_DATA must be incremented to automatically send the data without an LF.

RECEIVE FIFO

The receive FIFO operates in a fashion similar to the transmit FIFO. Data is inserted into the receive FIFO from the SIU_RECV procedure. The SIU_RECV procedure is called by the SIU_INT procedure when a valid I frame is received. The SIU_RECV procedure merely polls the receive FIFO status to see if it's full before transferring each byte from the SIU_RECV_BUFFER into the receive FIFO. If the receive FIFO is full, the SIU_RECV procedure remains polling the FIFO status until it can insert the rest of the data. In the meantime, the SIU AUTO mode is responding to all polls from the primary with a RNR supervisory frame. The USART_XMIT_INT interrupt procedure removes data from the receive FIFO and transmits it to the terminal. The USART transmit interrupt remains enabled while the receive FIFO has data in it. When the receive FIFO becomes empty, the USART transmit interrupt is disabled.

2.5 Primary Station

The primary station is responsible for controlling the data link. It issues commands to the secondary stations and receives responses from them. The primary station controls link access, link level error recovery, and the flow of information. Secondaries can only transmit when polled by the primary.

Most primary stations are either micro/minicomputers, or front end processors to a mainframe computer. The example primary station in this design is standalone. It is possible for the 8044 to be used as an intelligent front end processor for a microprocessor, implementing the primary station functions. This latter type of design would extensively off-load link control functions for the microprocessor. The code listed in this paper can be used as the basis for this primary station design. Additional software is required to interface to the microprocessor. A hardware design example for interfacing the 8044 to a microprocessor can be found in the applications section of this handbook.

The primary station must know the addresses of all the stations which will be on the network. The software for this primary needs to know this before it is compiled, however a more flexible system would download these parameters.

From the listing of the software it can be seen that the variable NUMBER_OF_STATIONS is a literal declaration, which is 2 in this design example. There were three stations tested on this data link, two secondaries and one primary. Following the NUMBER_OF_STATIONS declaration is a table, loaded into the object code file at compile time, which lists the addresses of each secondary station on the network.

REMOTE STATION DATABASE

The primary station keeps a record of each secondary station on the network. This is called the Remote Station Database (RSD). The RSD in this software is an array of structures, which can be found in the listing and also in Figure 16. Each RSD stores the necessary information about that secondary station.

To add additional secondary stations to the network, one simply adjusts the NUMBER_OF_STATIONS declaration, and adds the additional addresses to the SECONDARY_ADDRESSES table. The number of RSDs is automatically allocated at compile time, and the primary automatically polls each station whose address is in the SECONDARY_ADDRESSES table.

Memory for the RSDs resides in external RAM. Based on memory requirements for each RSD, the maximum number of stations can be easily buffered in external RAM. (254 secondary stations is the maximum number SDLC will address on the data link; i.e. 8-bit address, FF H is the broadcast address, and 0 is the null address. Each RSD uses 70 bytes of RAM. 70 x 254 = 17,780.)

The station state, in the RSD structure, maintains the status of the secondary. If this byte indicates that the secondary is in the DISCONNECT_S, then the primary tries to put the station in the I_T_S by sending an SNRM. If the response is a UA then the station state changes into the I_T_S. Any other frame received results in the station state remaining in the DISCONNECT_S. When the RSD indicates that the station state is in the I_T_S, the primary will send either an I, RR, or RNR command, depending on the local and remote buffer status. When the station state equals GO_TO_DISC the primary will send a DISC command. If the response is a UA frame, the station state will change to DISCONNECT_S, else the station state will remain in GO_TO_DISC. The station state is set to GO_TO_DISC when one of the following responses occur:

1. A receive buffer overrun in the primary.
2. An I frame is received and Nr(P) $\neq$ Ns(S).
3. An I frame or a Supervisory frame is received and Ns(P) + 1 $\neq$ Nr(S) and Ns(P) $\neq$ Nr(S).
4. A FRMR response is received.
5. An RD response is received.
6. An unknown response is received.

The send count (Ns) and receive count (Nr) are also maintained in the RSD. Each time an I frame is sent by the primary and acknowledged by the secondary, Ns is incremented. Nr is incremented each time a valid I frame is received. BUFFER_STATUS indicates the status of the secondary station's buffer. If an RR response is received, BUFFER_STATUS is set to BUFFER_READY. If a RNR response is received, BUFFER_STATUS is set to BUFFER_NOT_READY.

BUFFERING

The buffering for the primary station is as follows: within each RSD is a 64 byte array buffer which is initially empty. When the primary receives an I frame, it looks for a match between the first byte of the I frame and the addresses of the secondaries on the network. If a match exists, the primary places the data in the RSD buffer of the destination station. The INFO_LENGTH in the RSD indicates how many bytes are in the buffer. If INFO_LENGTH equals 0, then the buffer is empty. The primary can buffer only one I frame per station. If a second I frame is received while the addressed secondary's RSD buffer is full, the primary cannot receive any more I frames. At this point the primary continues to poll the secondaries using RNR supervisory frame.

PRIMARY STATION SOFTWARE

A block diagram of the primary station software is shown in Figure 17. The primary station software consists of a main program, one interrupt routine, and several procedures. The POWER_ON procedure begins by initializing the SIU's DMA and enabling the receiver. Then each RSD is initialized. The DPLL and the timers are set, and finally the TIMER 0 interrupt is enabled.

The main program consists of an iterative do loop within a do forever loop. The iterative do loop polls each secondary station once through the do loop. The variable STATION_NUMBER is the counter for the iterative do statement which is also used as an index to the array of RSD structures. The primary station issues one command and receives one response from every secondary station each time through the loop. The first statement in the loop loads the secondary station address, indexed by STATION_NUMBER into the array of the RSD structures. Now when the primary sends a command, it will have the secondary's address in the address field of the frame. The automatic address recognition feature is used by the primary to recognize the response from the secondary.

Next, the main program determines the secondary station's state. Based on this state, the primary knows what command to send. If the station is in the DISCONNECT_S, the primary calls the SNRM_P procedure to try and put the secondary in the I_T_S. If the station state is in the GO_TO_DISC state, the DISC_P is called to try and put the secondary in the L_D_S. If the secondary is in neither one of the above two states, then it is in the I_T_S. When the secondary is in the I_T_S, the primary could send one of three commands: I, RR, or RNR. If the RSD's buffer has data in it, indicated by INFO_LENGTH being greater than zero, and the secondary's BUFFER_STATUS equals BUFFER_READY, then an I frame will be sent. Else if RPB = 0, an RR supervisory frame will be sent. If neither one of these cases is true, then an RNR will be sent. The last statement in the main program checks the RPB bit. If set to one, the BUFFER_TRANSFER procedure is called, which transfers the data from the SIU receive buffer to the appropriate RSD buffer.

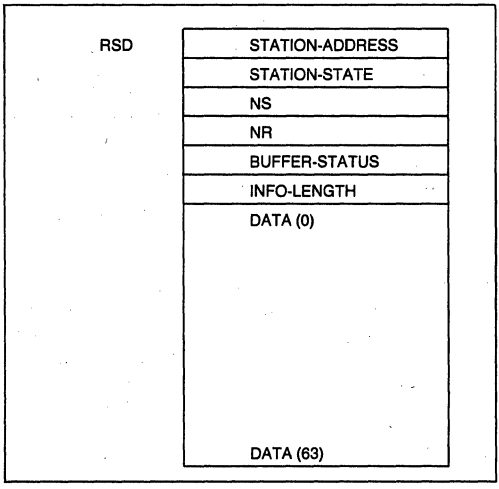


Figure 16. Remote Station Database Structure

RECEIVE TIME OUT

Each time a frame is transmitted, the primary sets a receive time out timer; Timer 0. If a response is not received within a certain time, the primary returns to the main program and continues polling the rest of the stations. The minimum length of time the primary should wait for a response can be calculated as the sum of the following parameters.

1. Propagation time to the secondary station
2. Clear-to-send at the secondary station's DCE
3. Appropriate time for secondary station processing
4. Propagation time from the secondary station
5. Maximum frame length time

The clear-to-send time and the propagation time are negligible for a local network at low bit rates. However, the turnaround time and the maximum frame length time are significant factors. Using the 8044 secondaries in the AUTO mode minimizes turnaround time. The maximum frame length time comes from the fact the 8044 does not generate an interrupt from a received frame until it has been completely received, and the CRC is verified as correct. This means that the time-out is bit rate dependent.

Ns AND Nr CHECK PROCEDURES

Each time an I frame or supervisory frame is received, the Nr field in the control byte must be checked. Since this data link only allows one outstanding frame, a valid Nr would satisfy either one of two equations; Ns(P) + 1 = Nr(S) the I frame previously sent by the primary is acknowledged, Ns(P) = Nr(S) the I frame previously sent is not acknowledged. If either one of these two cases is true, the CHECK_NR procedure returns a parameter of TRUE; otherwise a FALSE parameter is returned. If an acknowledgement is received, the Ns byte in the RSD structure is incremented, and the Information buffer may be cleared. Otherwise the information buffer remains full.

When an I frame is received, the Ns field has to be checked also. If Nr(P) = Ns(S), then the procedure returns TRUE, otherwise a FALSE is returned.

RECEIVE PROCEDURE

The receive procedure is called when a supervisory or information frame is sent, and a response is received before the time-out period. The RECEIVE procedure can be broken down into three parts. The first part is entered if an I frame is received. When an I frame is received, Ns, Nr and buffer overrun are checked. If there is a buffer overrun, or there is an error in either Ns or Nr, then the station state is set to GO_TO_DISC. Otherwise Nr in the RSD is incremented, the receive field length is saved, and the RPB bit is set. By incrementing the Nr field, the I frame just received is acknowledged the next time the primary polls the secondary with an I frame or a supervisory frame. Setting RBP protects the received data, and also tells the main program that there is data to transfer to one of the RSD buffers.

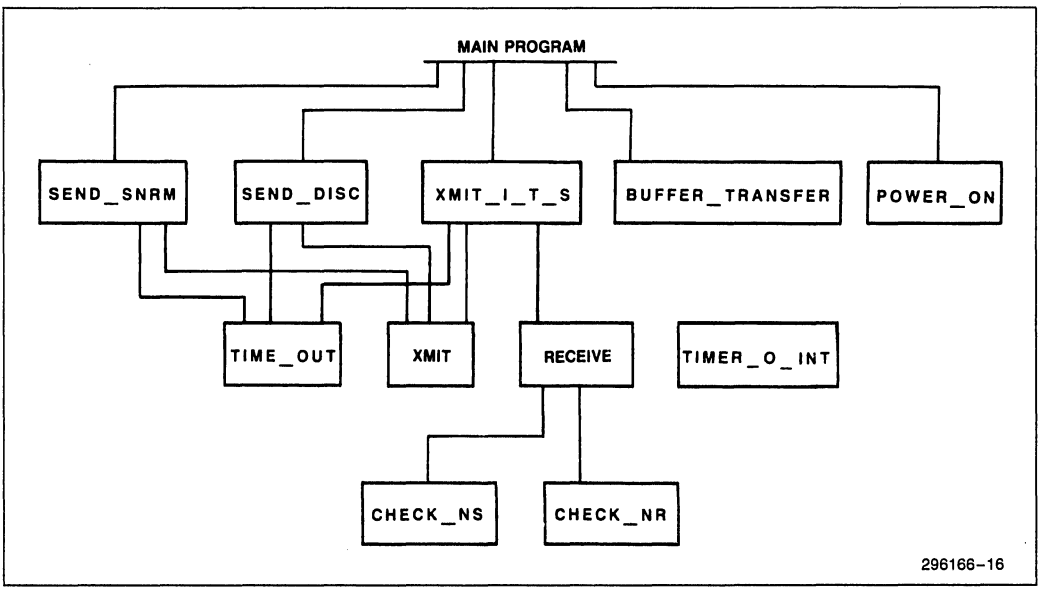


Figure 17. Block Diagram of Primary Station Software Structure

If a supervisory frame is received, the Nr field is checked. If a FALSE is returned, then the station state is set to GO_TO_DISC. If the supervisory frame received was an RNR, buffer status is set to not ready. If the response is not an I frame, nor a supervisory frame, then it must be an Unnumbered frame.

The only Unnumbered frames the primary recognizes are UA, DM, and FRMR. In any event, the station state is set to GO_TO_DISC. However, if the frame received is a FRMR, Nr in the second data byte of the I field is checked to see if the secondary acknowledged an I frame received before it went into the FRMR state. If this is not done and the secondary acknowledged an I frame which the primary did not recognize, the primary transmits the I frame when the secondary returns to the I_T_S. In this case, the secondary would receive duplicate I frames.

APPENDIX A
8044 SOFTWARE FLOWCHARTS

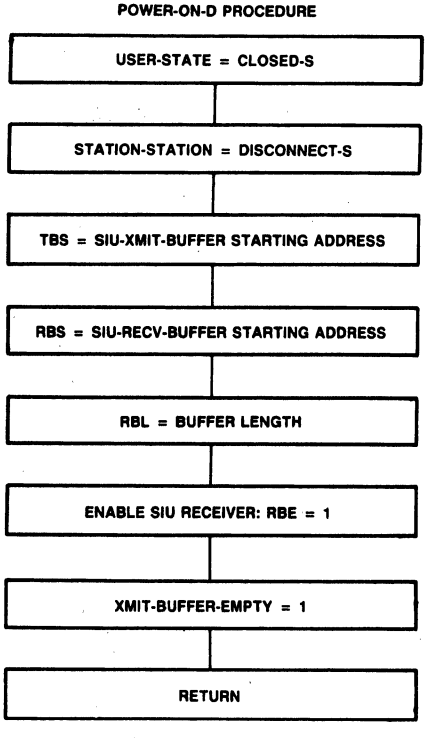
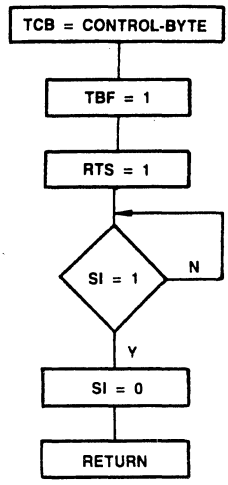


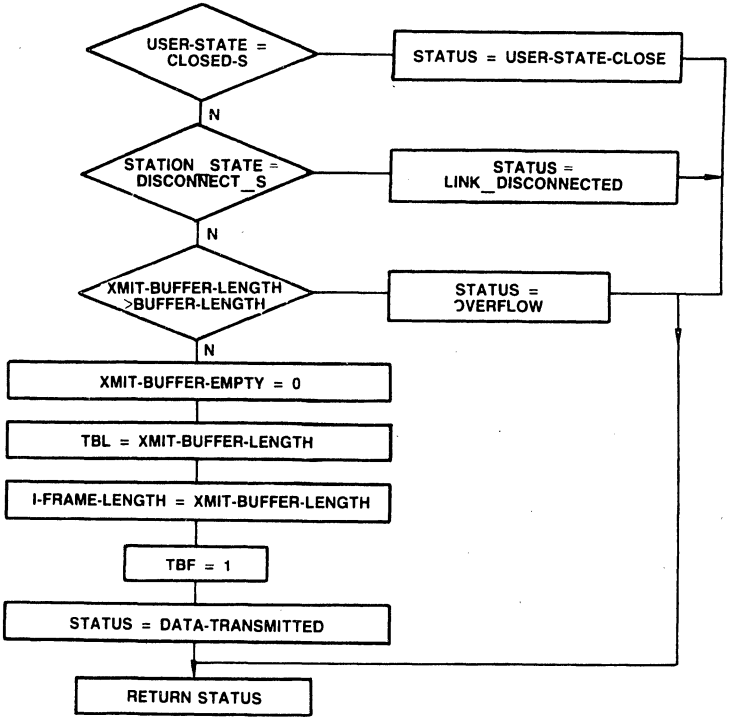
Figure 18. Secondary Station Driver Flow Chart

XMIT—UNNUMBERED PROCEDURE



296166-19

TRANSMIT PROCEDURE



296166-20

Figure 19. Secondary Station Driver Flow Chart

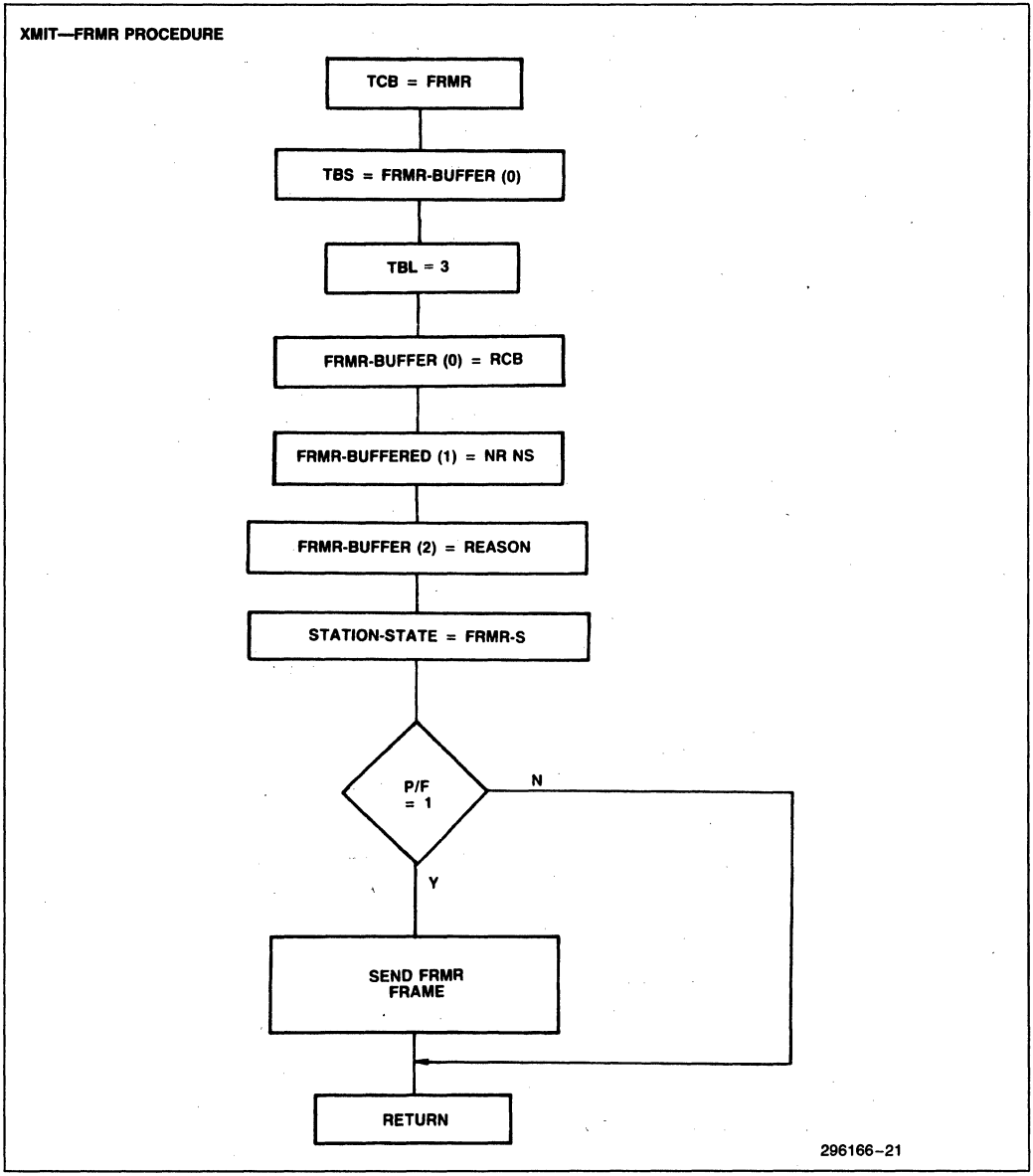
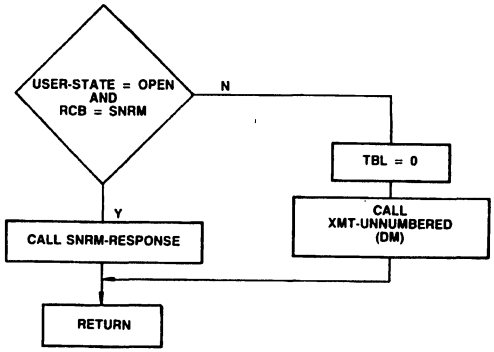


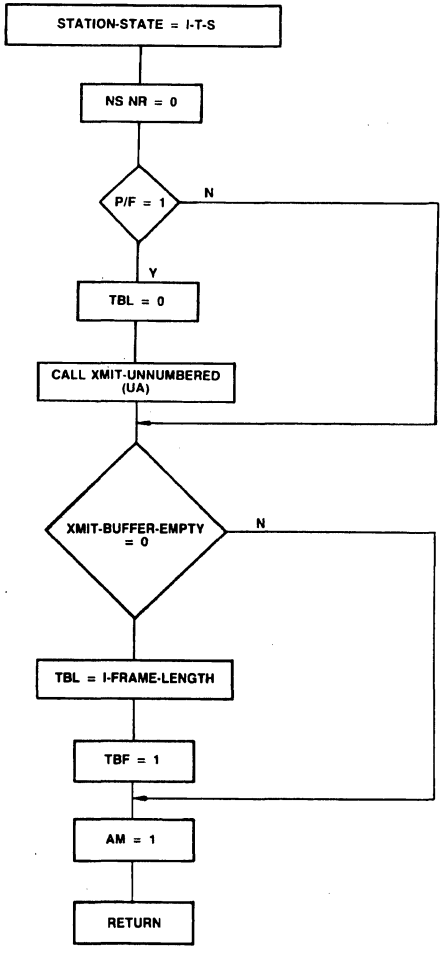
Figure 20. Secondary Station Driver Flow Chart

IN-DISCONNECT-STATE PROCEDURE



296166-22

SNRM-RESPONSE PROCEDURE



296166-23

Figure 21. Secondary Station Driver Flow Chart

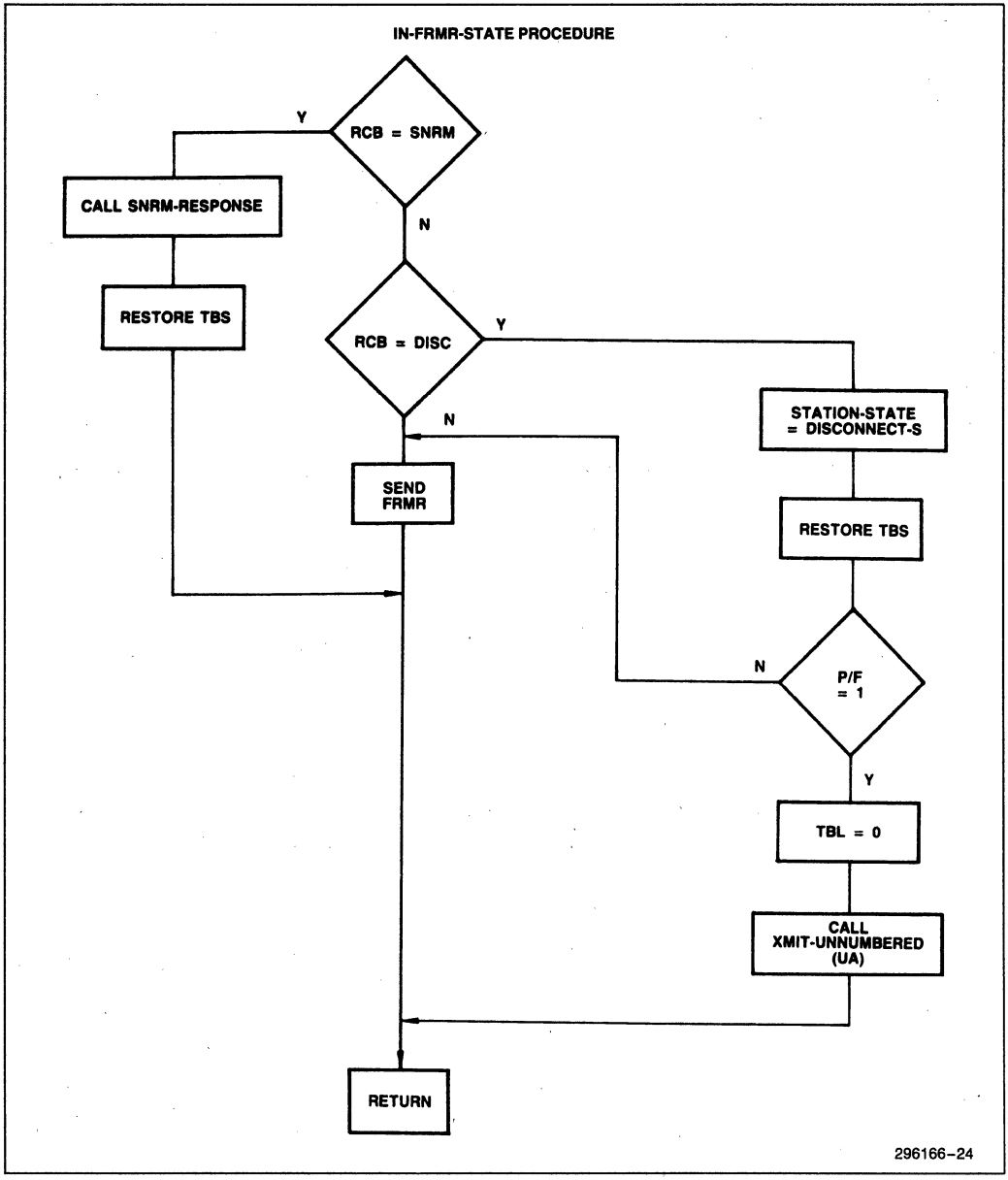
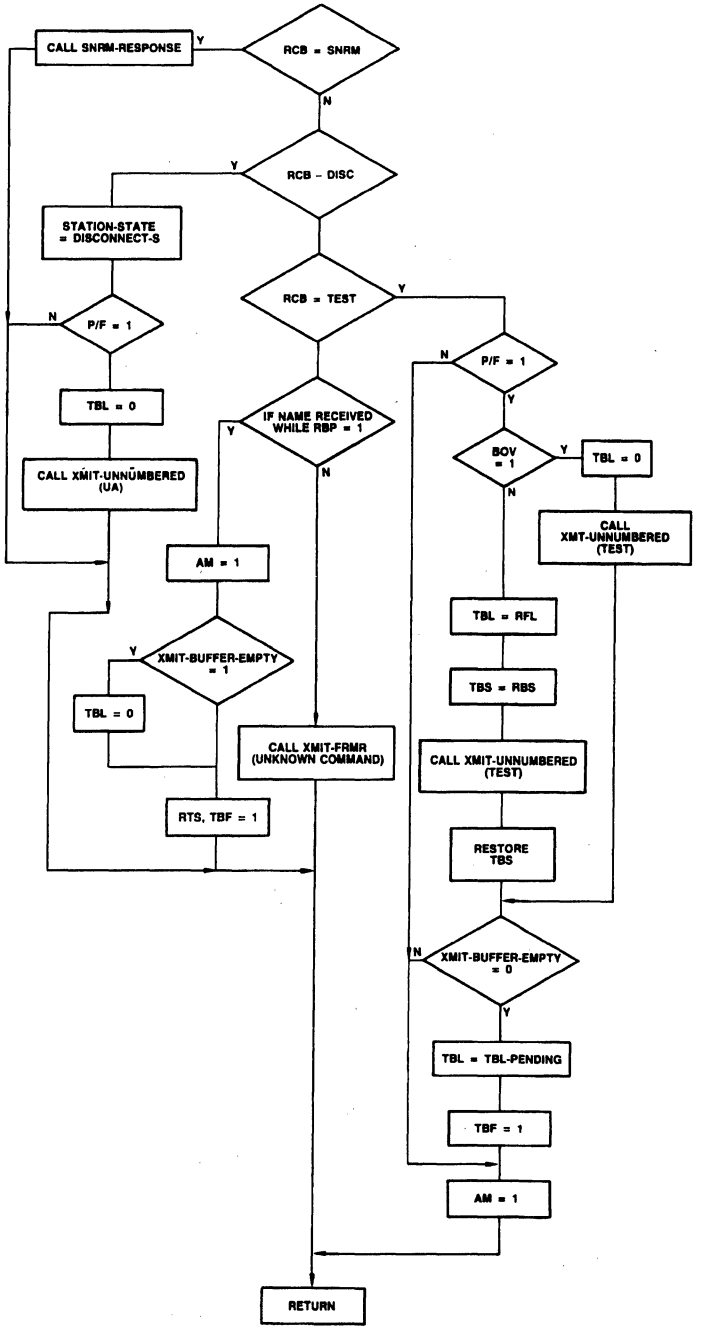


Figure 22. Secondary Station Driver Flow Chart

COMMAND DECODE PROCEDURE



296166-25

Figure 23. Secondary Station Driver Flow Chart

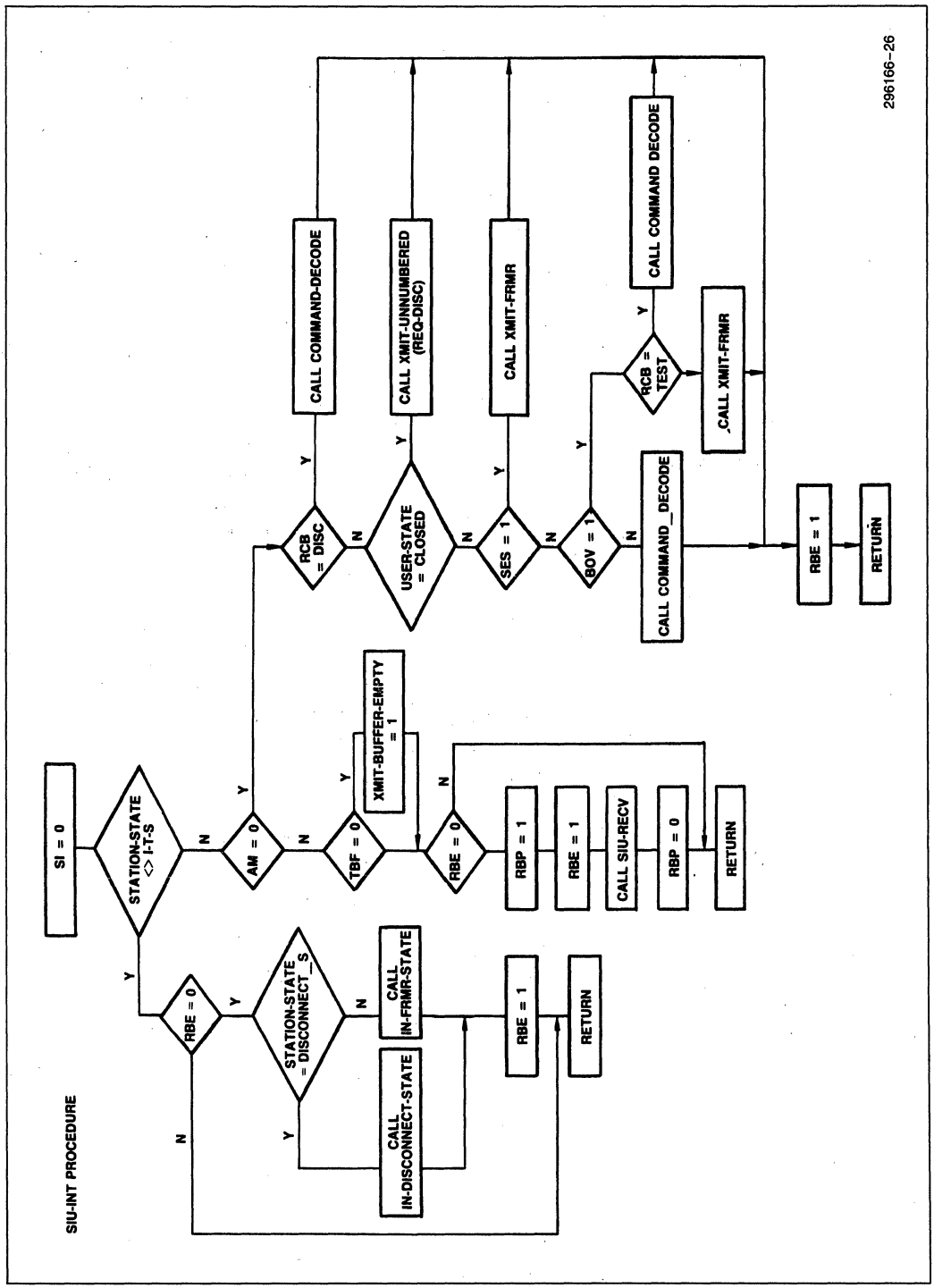


Figure 24. Secondary Station Driver Flow Chart

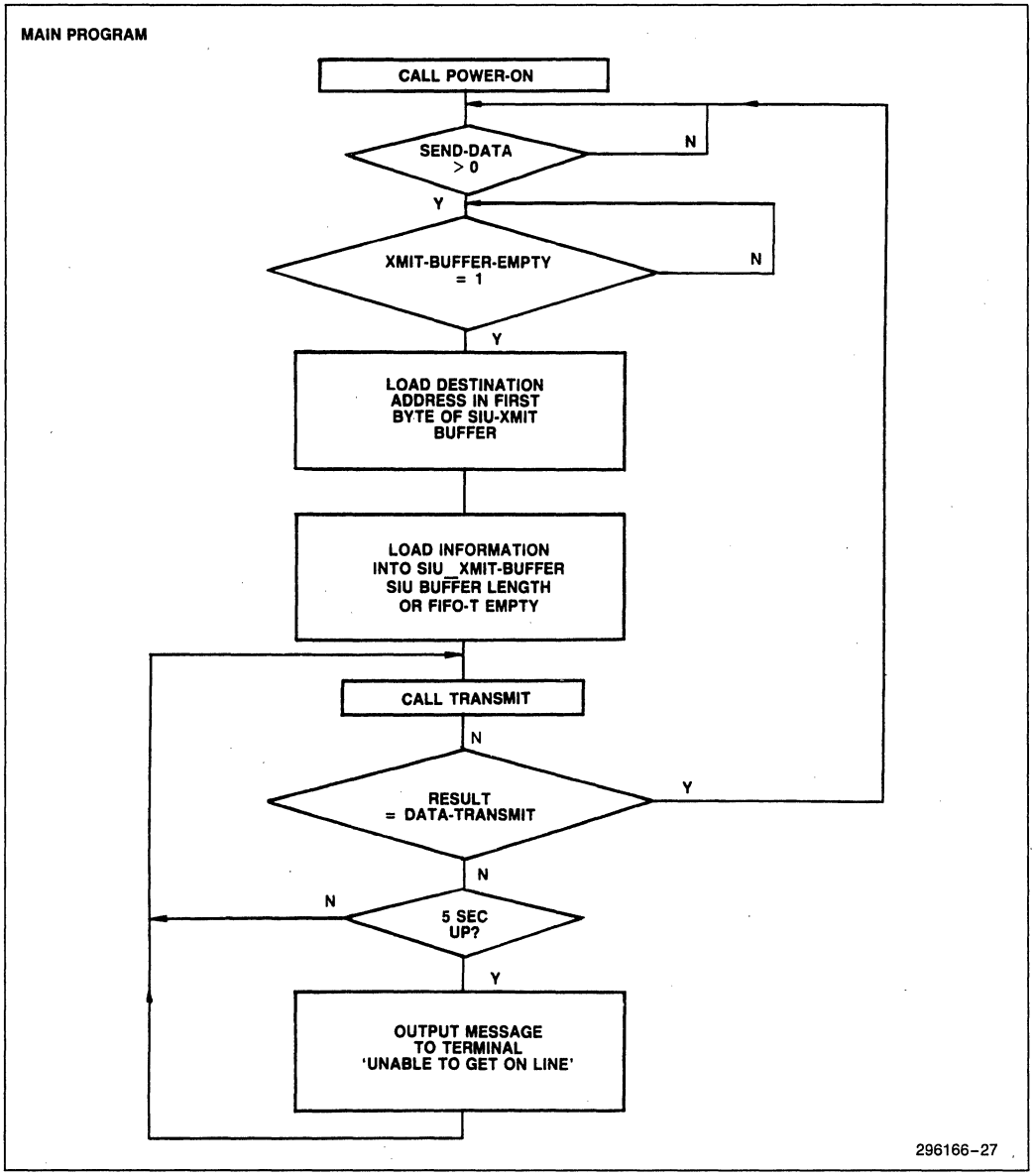


Figure 25. Application Module Flow Chart

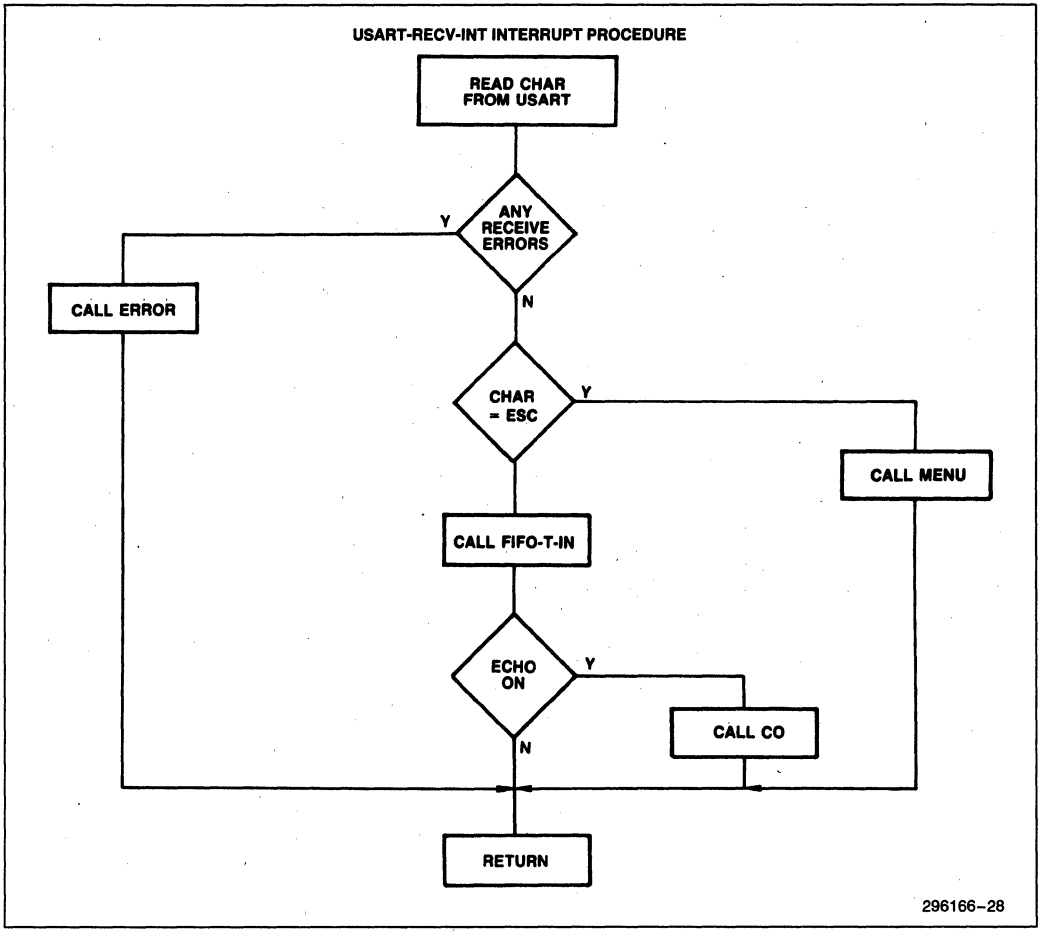


Figure 26. Application Module Flow Chart

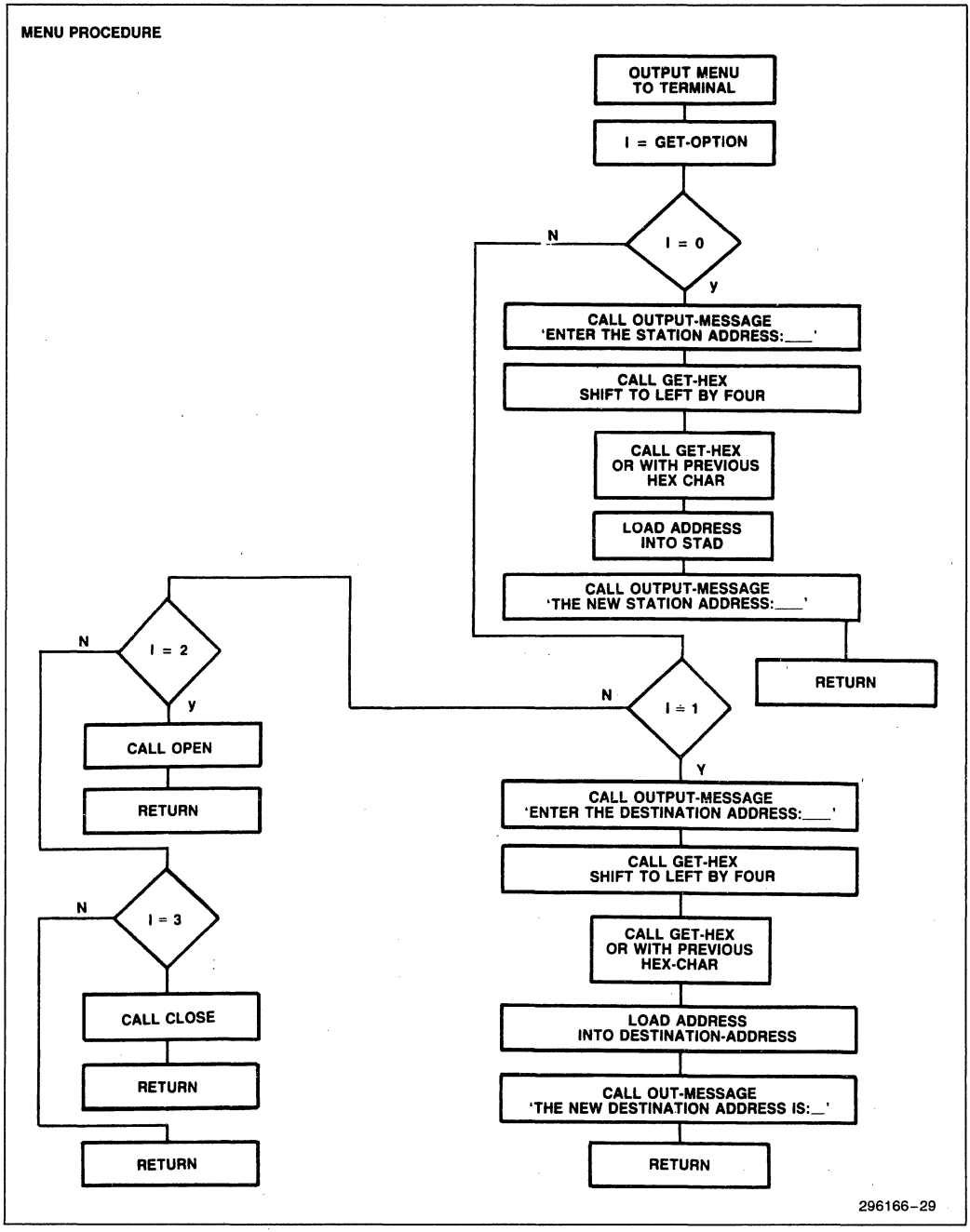


Figure 27. Application Module Flow Chart

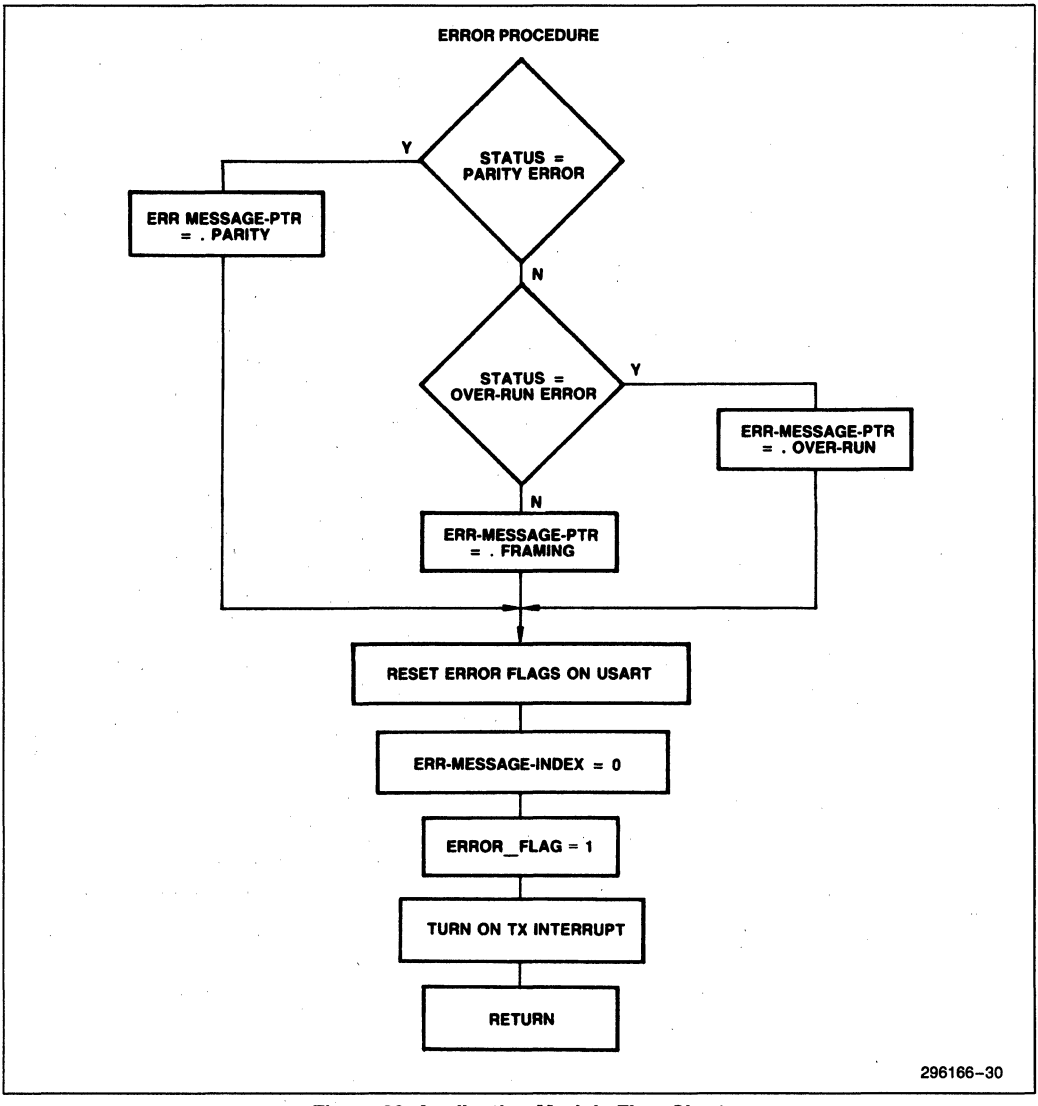


Figure 28. Application Module Flow Chart

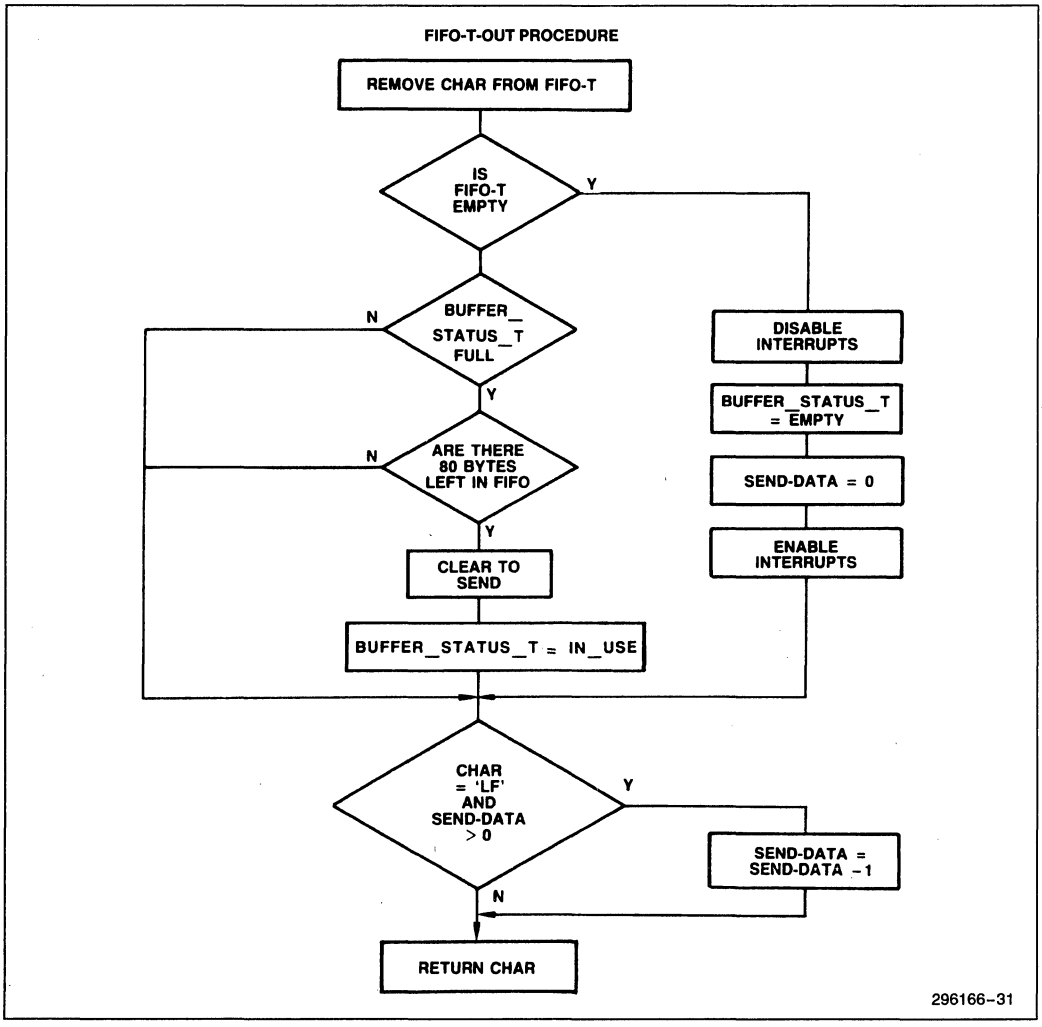


Figure 29. Application Module Flow Chart

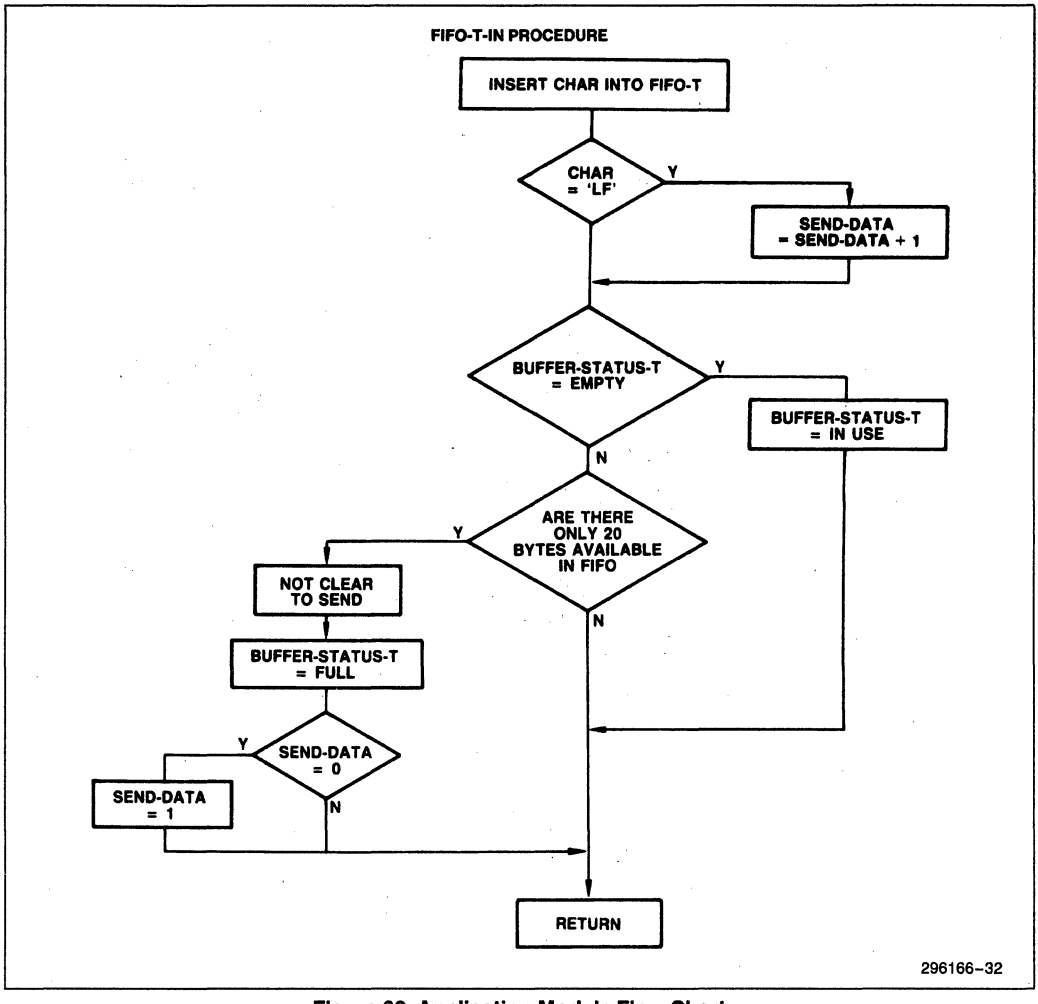


Figure 30. Application Module Flow Chart

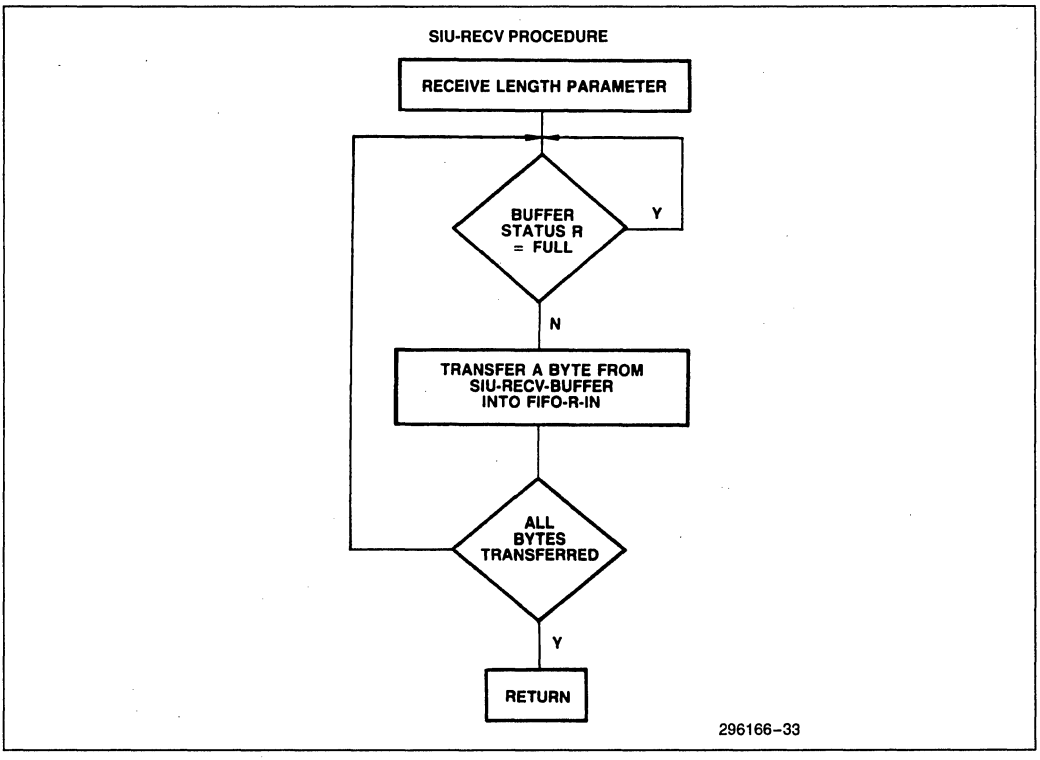


Figure 31. Application Module Flow Chart

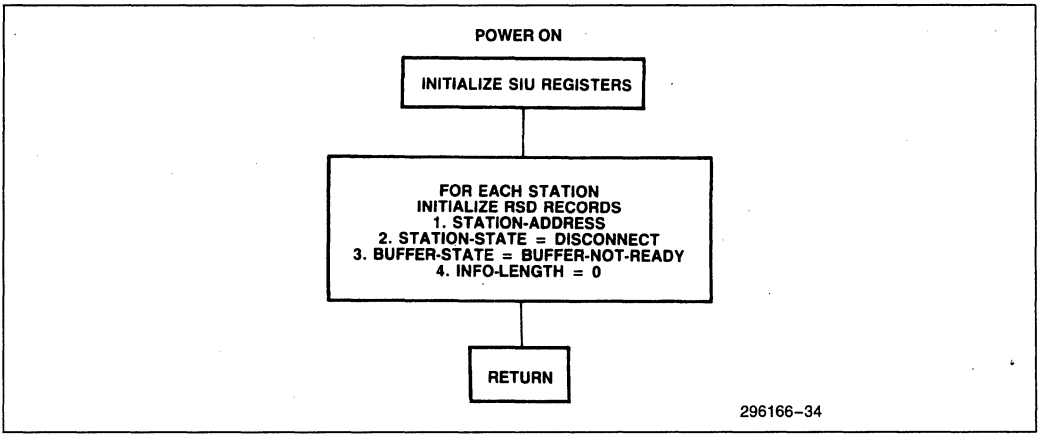


Figure 32. Primary Station Flow Charts

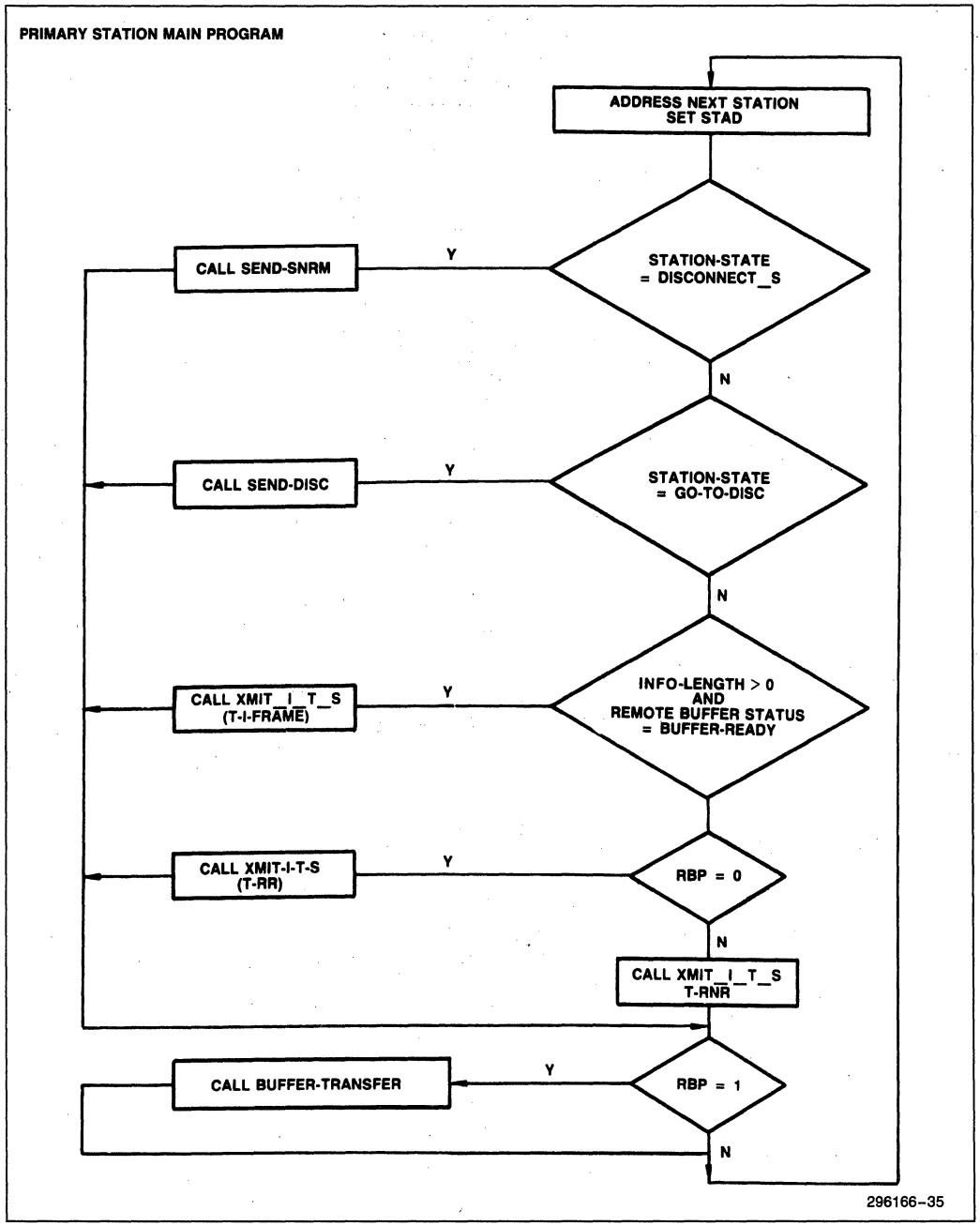


Figure 33. Primary Station Flow Charts

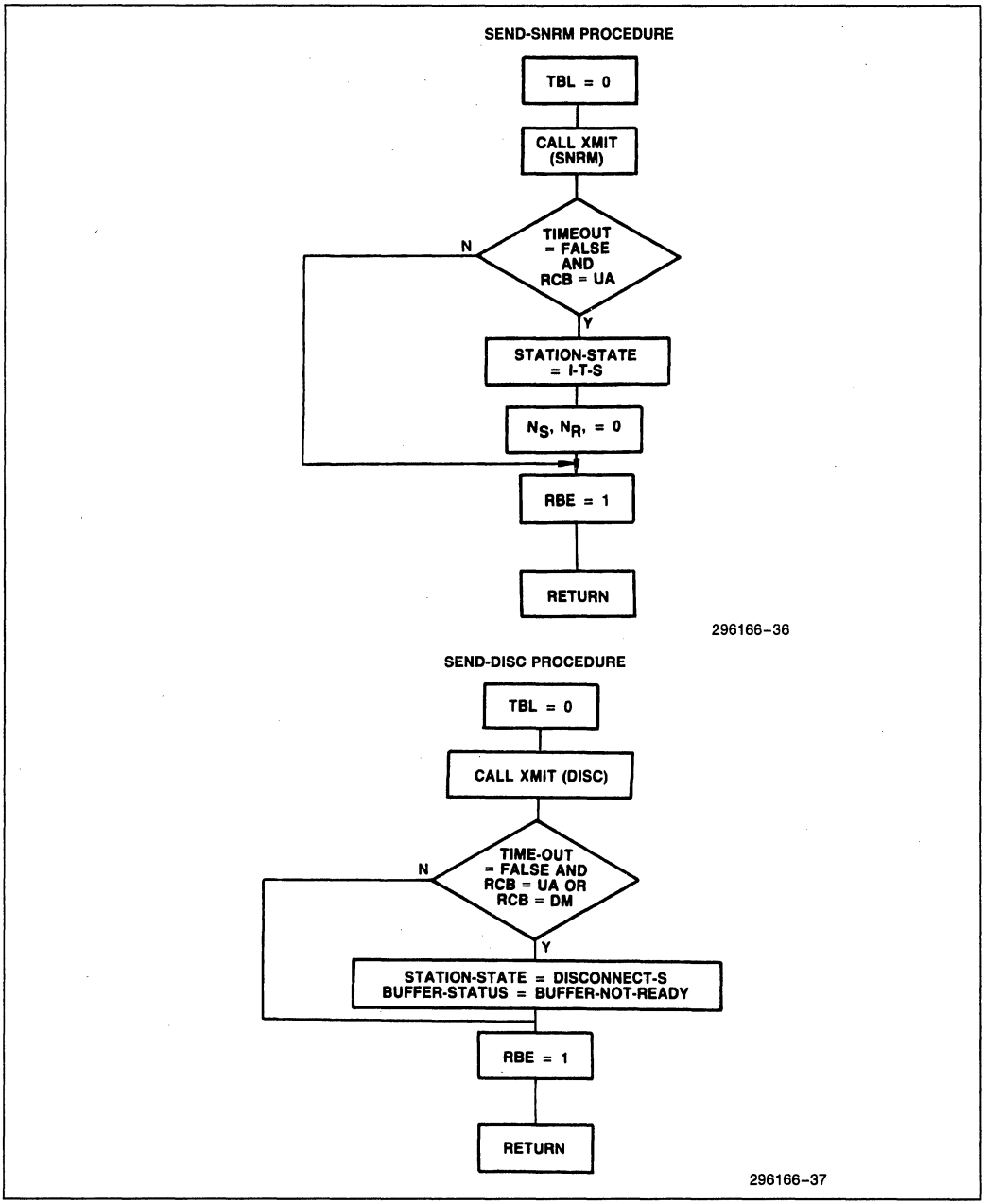


Figure 34. Primary Station Flow Charts

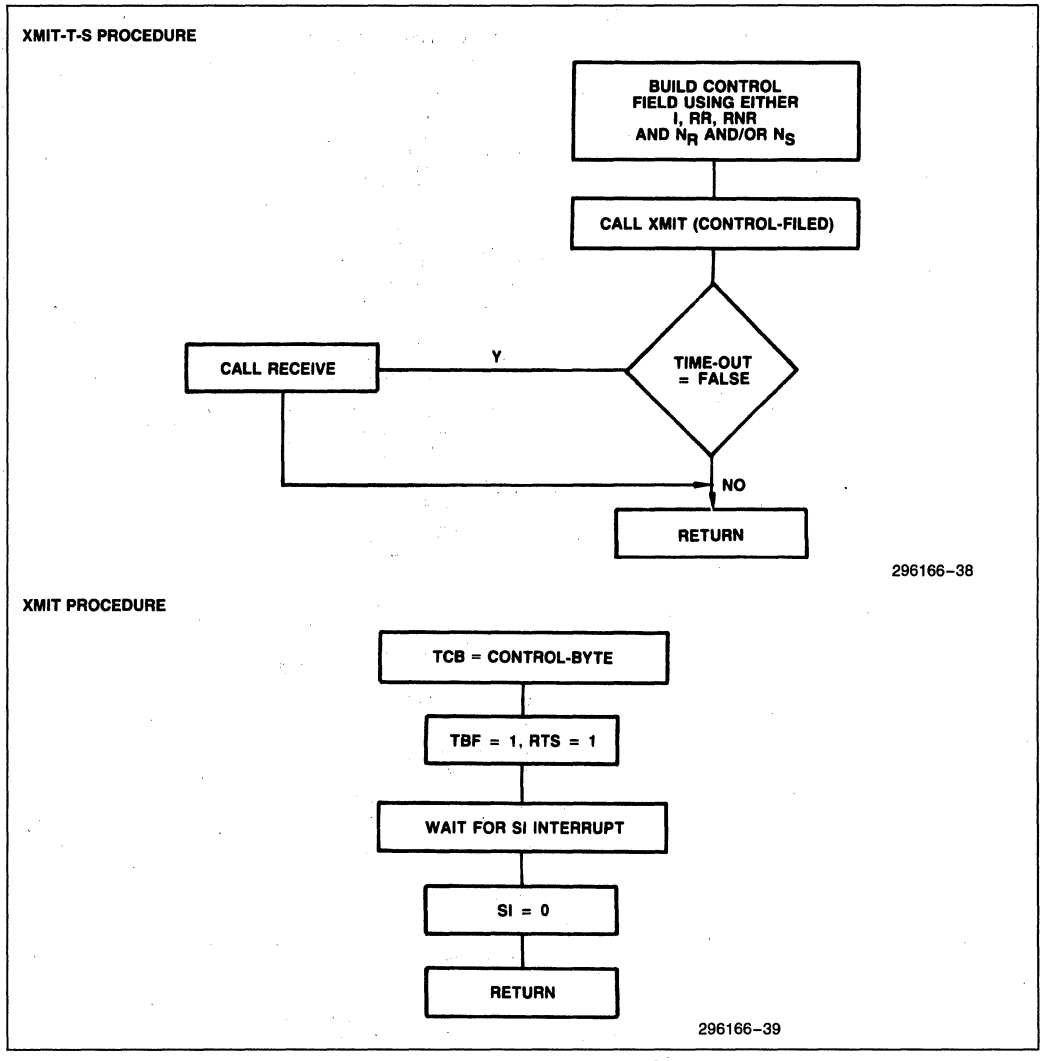


Figure 34. Primary Station Flow Charts

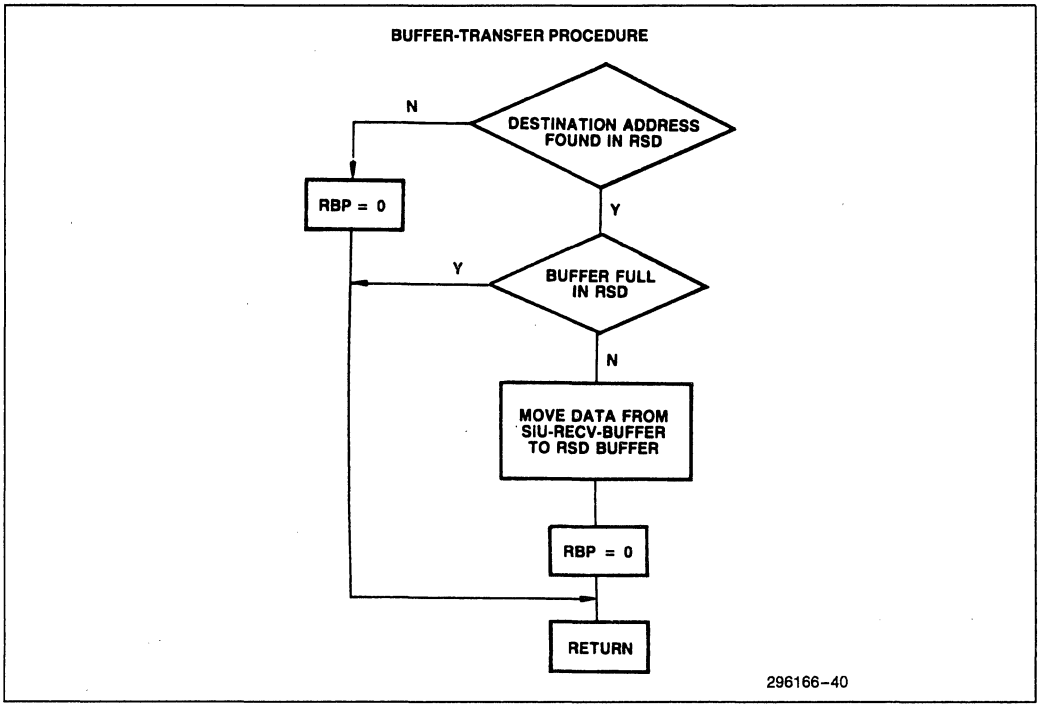


Figure 36. Primary Station Flow Charts

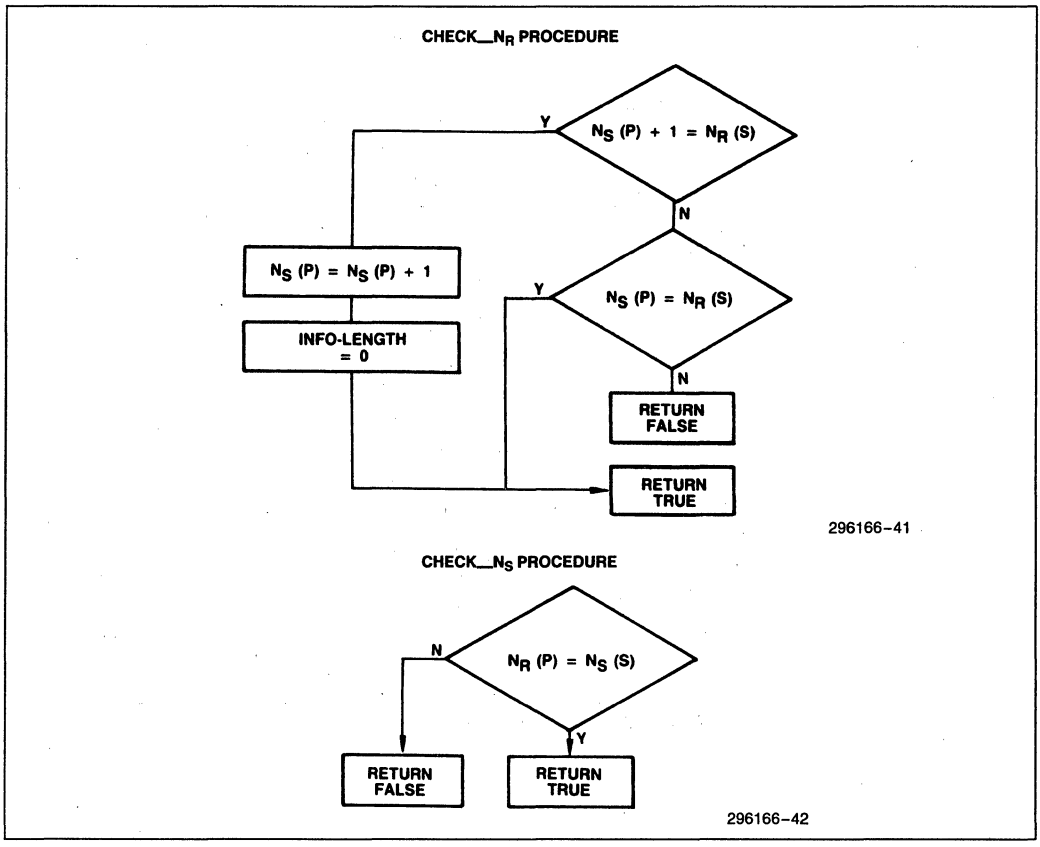


Figure 37. Primary Station Flow Charts

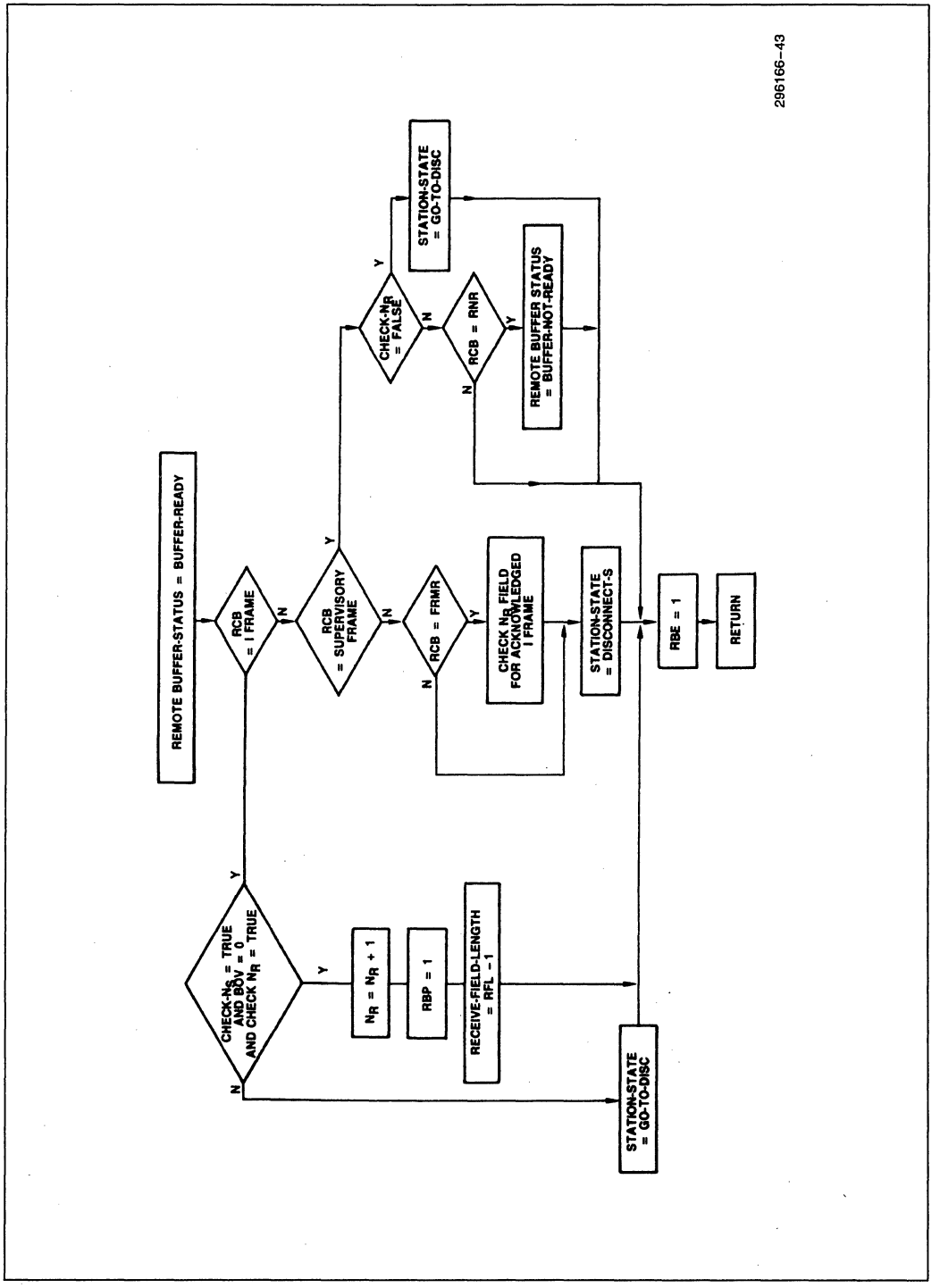


Figure 38. Primary Station Flow Charts

APPENDIX B
LISTINGS OF SOFTWARE MODULES

```
PL/M-51 COMPILER    RUPI-44 Secondary Station Driver                    20:24:47  09/20/83  PAGE   1

ISIS-II PL/M-51 V1.0
COMPILER INVOKED BY:   :F2:PLM51 :F2:APNOTE.SRC


                $TITLE      ('RUPI-44 Secondary Station Driver')
                $DEBUG
                $REGISTERBANK(1)
   1    1      MAIN$MOD:DO;
                $NOLIST

                /* To save paper the RUPI registers are not listed, but this is the statement
                   used to include them: $INCLUDE (:F2:REG44.DCL) */

   5    1      DECLARE LIT             LITERALLY   'LITERALLY',
                       TRUE            LIT         '0FFH',
                       FALSE           LIT         '00H',
                       FOREVER         LIT         'WHILE 1';

                /* SDLC commands and responses */

   6    1      DECLARE SNRM            LIT         '83H',
                       UA              LIT         '73H',
                       DISC            LIT         '43H',
                       DM              LIT         '1FH',
                       FRMR            LIT         '97H',
                       REQ_DISC        LIT         '53H',
                       UP              LIT         '33H',
                       TEST            LIT         '0E3H',


                              /* User states */

                       OPEN_S           LIT         '00H',
                       CLOSED_S         LIT         '01H',

                              /* Station states */

                       DISCONNECT_S    LIT         '00H', /* LOGICALLY DISCONNECTED STATE*/
                       FRMR_S          LIT         '01H', /* FRAME REJECT STATE */
                       I_T_S           LIT         '02H', /* INFORMATION TRANSFER STATE */


                       /* Status values returned from TRANSMIT procedure */

                       USER_STATE_CLOSED   LIT     '00H',
                       LINK_DISCONNECTED   LIT     '01H',
                       OVERFLOW            LIT     '02H',
                       DATA_TRANSMITTED    LIT     '03H',

                       /* Parameters passed to XMIT_FRMR */

                       UNASSIGNED_C        LIT     '00H',
                       NO_I_FIELD_ALLOWED  LIT     '01H',
                       BUFF_OVERRUN        LIT     '02H',
                       SEQ_ERR             LIT     '03H',
```

296166-44

```
PL/M-51 COMPILER    RUPI-44 Secondary Station Driver                    20:24:47  09/20/83   PAGE   2

                            /* Variables */

                        USER_STATE          BYTE    AUXILIARY,
                        STATION_STATE       BYTE    AUXILIARY,
                        I_FRAME_LENGTH      BYTE    AUXILIARY,

                            /* Buffers */

                        BUFFER_LENGTH                       LIT     '60',
                        SIU_XMIT_BUFFER(BUFFER_LENGTH)      BYTE    PUBLIC    IDATA,
                        SIU_RECV_BUFFER(BUFFER_LENGTH)      BYTE    PUBLIC,
                        FRMR_BUFFER(3)              BYTE,

                            /* Flags */

                        XMIT_BUFFER_EMPTY       BIT PUBLIC;

  7   2     SIU_RECV: PROCEDURE (LENGTH) EXTERNAL;
  8   2         DECLARE LENGTH  BYTE;
  9   1     END SIU_RECV;

 10   2     OPEN: PROCEDURE PUBLIC USING 2;
 11   2             USER_STATE=OPEN_S;
 12   1     END OPEN;

 13   2     CLOSE: PROCEDURE PUBLIC USING 2;
 14   2             AM=0;
 15   2             USER_STATE=CLOSED_S;
 16   1     END CLOSE;

 17   2     POWER_ON_D: PROCEDURE   PUBLIC USING 0;

 18   2             USER_STATE=CLOSED_S;
 19   2             STATION_STATE=DISCONNECT_S;
 20   2             TBS=.SIU_XMIT_BUFFER(0);
 21   2             RBS=.SIU_RECV_BUFFER(0);
 22   2             RBL=BUFFER_LENGTH;
 23   2             RBE=1;      /* Enable the SIU's receiver */
 24   2             XMIT_BUFFER_EMPTY=1;

 25   1     END POWER_ON_D;

 26   2     TRANSMIT: PROCEDURE (XMIT_BUFFER_LENGTH) BYTE   PUBLIC USING 0;

           /* User must check XMIT_BUFFER_EMPTY flag before calling this procedure */

 27   2             DECLARE XMIT_BUFFER_LENGTH  BYTE,
                         I                   BYTE    AUXILIARY,
                         STATUS              BYTE    AUXILIARY;

 28   2             IF USER_STATE=CLOSED_S
                        THEN STATUS=USER_STATE_CLOSED;
 30   2             ELSE IF STATION_STATE=DISCONNECT_S
                             THEN STATUS=LINK_DISCONNECTED;
 32   2             ELSE IF XMIT_BUFFER_LENGTH>BUFFER_LENGTH
                             THEN STATUS=OVERFLOW;
 34   3             ELSE DO;
```

296166-45

```
PL/M-51 COMPILER    RUPI-44 Secondary Station Driver                    20:24:47  09/20/83  PAGE   3

  35   3                    XMIT_BUFFER_EMPTY=0;
  36   3                    TBL=XMIT_BUFFER_LENGTH;
  37   3                    I_FRAME_LENGTH=XMIT_BUFFER_LENGTH; /* Store length in case station
                                                                 is reset by FRMR, SNRM etc. */
  38   3                    TBF=1;
  39   3                   STATUS=DATA_TRANSMITTED;
  40   3                  END;
  41   2              RETURN STATUS;
  42   1      END TRANSMIT;

  43   2      XMIT_UNNUMBERED: PROCEDURE (CONTROL_BYTE) ;

  44   2          DECLARE CONTROL_BYTE    BYTE;

  45   2              TCB=CONTROL_BYTE;
  46   2              TBF=1;
  47   2              RTS=1;
  48   3              DO WHILE NOT SI;
  49   3              END;
  50   2              SI=0;

  51   1      END XMIT_UNNUMBERED;

  52   2      SNRM_RESPONSE: PROCEDURE ;

  53   2          STATION_STATE=I_T_S;
  54   2          NSNR=0;
  55   2          IF (RCB AND 10H) <> 0 /* Respond if polled */
                     THEN DO;
  57   3                    TBL=0;
  58   3                    CALL XMIT_UNNUMBERED(UA);
  59   3                END;
  60   2          IF XMIT_BUFFER_EMPTY=0  /* If an I frame was left pending transmission
                                             then restore it */
                     THEN DO;
  62   3                    TBL=I_FRAME_LENGTH;
  63   3                    TBF=1;
  64   3                END;
  65   2          AM=1;

  66   1      END SNRM_RESPONSE;


  67   2      XMIT_FRMR: PROCEDURE (REASON)  ;

  68   2              DECLARE REASON  BYTE;


  69   2              TCB=FRMR;

  70   2              TBB=.FRMR_BUFFER(0);
  71   2              TBL=3;
  72   2              FRMR_BUFFER(0)=RCB;
                      /* Swap nibbles in NSNR */
  73   2              FRMR_BUFFER(1)=(SHL((NSNR AND 0EH),4) OR  SHR((NSNR AND 0E0H),4));
  74   3              DO CASE REASON;
  75   3                  FRMR_BUFFER(2)=01H;   /* UNASSIGNED_C */
```

296166-46

```
PL/M-51 COMPILER    RUPI-44 Secondary Station Driver                    20:24:47  09/20/83  PAGE   4

  76   3                 FRMR_BUFFER(2)=02H;   /* NO_I_FIELD_ALLOWED */
  77   3                 FRMR_BUFFER(2)=04H;   /* BUFF_OVERRUN */
  78   3                 FRMR_BUFFER(2)=08H;   /* SEQ_ERR */
  79   3              END;

  80   2             STATION_STATE=FRMR_S;

  81   2             IF (RCB AND 10H) <>0
                         THEN DO;
  83   3                           TBF=1;
  84   3                           RTS=1;
  85   4                           DO WHILE NOT SI;
  86   4                           END;
  87   3                           SI=0;
  88   3                       END;
  89   1      END XMIT_FRMR;


  90   2      IN_DISCONNECT_STATE: PROCEDURE ; /* Called from SIU_INT procedure */

  91   2              IF ((USER_STATE=OPEN_S) AND ((RCB AND OEFH)=SNRM))
                         THEN CALL SNRM_RESPONSE;

  93   2             ELSE IF (RCB AND 10H) <> 0
                           THEN DO;
  95   3                             TBL=0;
  96   3                             CALL XMIT_UNNUMBERED(DM);
  97   3                           END;
  98   1      END IN_DISCONNECT_STATE;


  99   2      IN_FRMR_STATE: PROCEDURE ; /* Called by SIU_INT when a frame has been received
                                           when in the FRMR state */

 100   2          IF (RCB AND OEFH)=SNRM
                     THEN DO;
 102   3                     CALL SNRM_RESPONSE;
 103   3                     TBS=.SIU_XMIT_BUFFER(0);  /* Restore transmit buffer start address */
 104   3                   END;


 105   2          ELSE IF (RCB AND OEFH)=DISC
                         THEN DO;
 107   3                     STATION_STATE=DISCONNECT_S;
 108   3                     TBS=.SIU_XMIT_BUFFER(0);   /* Restore transmit buffer start address */
 109   3                     IF (RCB AND 10H)<> 0
                                THEN DO;
 111   4                                 TBL=0;
 112   4                                 CALL XMIT_UNNUMBERED(UA);
 113   4                               END;
 114   3                   END;

 115   3          ELSE DO;       /* Receive control byte is something other than DISC or SNRM */
 116   3                 IF (RCB AND 10H) <> 0
                             THEN DO;
 118   4                               TBF=1;
 119   4                               RTS=1;
```

296166-47

```
PL/M-51 COMPILER    RUPI-44 Secondary Station Driver                    20:24:47  09/20/83  PAGE   5

 120   5                          DO WHILE NOT SI;
 121   5                          END;
 122   4                       END;
 123   3             END;

 124   1      END IN_FRMR_STATE;

 125   2      COMMAND_DECODE: PROCEDURE ;


 126   2          IF (RCB AND OEFH)=SNRM
                     THEN  CALL SNRM_RESPONSE;

 128   2          ELSE IF (RCB AND OEFH)=DISC
                         THEN  DO;
 130   3                          STATION_STATE=DISCONNECT_S;
 131   3                          IF (RCB AND 10H)<>0
                                     THEN DO;
 133   4                                   TBL=0;
 134   4                                   CALL XMIT_UNNUMBERED(UA);
 135   4                                END;
 136   3                       END;

 137   2          ELSE IF (RCB AND OEFH)=TEST
                         THEN DO;
 139   3                          IF (RCB AND 10H)>0  /* Respond if polled */
                                     THEN DO;    /* FOR BOV=1, SEND THE TEST RESPONSE WITHOUT AN I FIELD */
 141   4                                   IF (BOV=1)
                                             THEN DO;
 143   5                                            TBL=0;
 144   5                                            CALL XMIT_UNNUMBERED(TEST OR 10H);
 145   5                                        END;
 146   5                                   ELSE DO; /* If no BOV, send received I field back to primary */
 147   5                                        TBL=RFL;
 148   5                                        TBS=RBS;
 149   5                                        CALL XMIT_UNNUMBERED(TEST OR 10H);
 150   5                                        TBS=.SIU_XMIT_BUFFER(0);  /* Restore TBS */
 151   5                                    END;

                                             /* If an I frame was pending, set it up again */

 152   4                                   IF XMIT_BUFFER_EMPTY=0
                                             THEN DO;
 154   5                                            TBL=I_FRAME_LENGTH;
 155   5                                            TBF=1;
 156   5                                        END;
 157   4                                END;
 158   3                          AM=1;
 159   3                       END;

 160   2          ELSE IF (RCB AND 01H) = 0 /* Kicked out of the AUTO mode because
                                               an I frame was received while RPB = 1 */
                         THEN DO;
 162   3                          AM = 1;
 163   3                          IF XMIT_BUFFER_EMPTY = 1
                                     THEN TBL = 0;
 165   3                          TBF = 1;     /* Send an AUTO mode response */
```

296166-48

```
PL/M-51 COMPILER    RUPI-44 Secondary Station Driver                    20:24:47  09/20/83  PAGE   6

 166   3                        RTS = 1;
 167   3                      END;

 168   2          ELSE CALL XMIT_FRMR(UNASSIGNED_C);  /* Received an undefined or not implemented command */

 169   1      END COMMAND_DECODE;


 170   2      SIU_INT: PROCEDURE INTERRUPT 4;

 171   2            DECLARE     I   BYTE    AUXILIARY;

 172   2            SI=0;
 173   2            IF STATION_STATE<> I_T_S /* Must be in NON-AUTO mode */
                       THEN DO;
 175   3                        IF RBE=0 /* Received a frame?  Give response */
                              THEN DO;
 177   5                                DO CASE STATION_STATE;
 178   5                                    CALL IN_DISCONNECT_STATE;
 179   5                                    CALL IN_FRMR_STATE;
 180   5                                END;
 181   4                                RBE=1;
 182   4                              END;
 183   3                    RETURN;
 184   3                    END;

                     /* If the program reaches this point, STATION_STATE=I_T_S
                         which means the SIU either was, or still is in the AUTO MODE */


 185   2            IF AM=0
                       THEN DO;
 187   3                        IF (RCB AND OEFH)=DISC
                                  THEN CALL COMMAND_DECODE;
 189   3                        ELSE IF USER_STATE=CLOSED_S
                                  THEN DO;
 191   4                                    TBL=0;
 192   4                                    CALL XMIT_UNNUMBERED(REQ_DISC);
 193   4                                END;
 194   3                        ELSE IF SES=1
                                  THEN CALL XMIT_FRMR(SES_ERR);
 196   3                        ELSE IF BOV=1
                                  THEN DO; /* DON'T SEND FRMR IF A TEST WAS RECEIVED*/
 198   4                                    IF (RCB AND OEFH)=TEST
                                              THEN CALL COMMAND_DECODE;
 200   4                                    ELSE     CALL XMIT_FRMR(BUFF_OVERRUN);
 201   4                                END;
 202   3                        ELSE CALL COMMAND_DECODE;
 203   3                        RBE=1;
 204   3                     END;

 205   3            ELSE DO; /* MUST STILL BE IN AUTO MODE */
 206   3                    IF TBF=0
                              THEN XMIT_BUFFER_EMPTY=1;  /* TRANSMITTED A FRAME */
 208   3                    IF RBE=0
                              THEN DO;
```

296166-49

```
PL/M-51 COMPILER    RUPI-44 Secondary Station Driver                    20:24:47  09/20/83  PAGE   7

 210   4                                RBP=1; /* RNR STATE */
 211   4                                RBE=1; /* RE-ENABLE RECEIVER */
 212   4                                CALL SIU_RECV(RFL);
 213   4                                RBP=0; /* RR STATE */
 214   4                              END;
 215   3                    END;
 216   1      END SIU_INT;

 217   1      END MAIN&MOD;

Software and application note written by Charles Yager


WARNINGS:
    4 IS THE HIGHEST USED INTERRUPT


MODULE INFORMATION:                 (STATIC+OVERLAYABLE)
    CODE SIZE                       = 028FH       655D
    CONSTANT SIZE                   = 0000H         0D
    DIRECT VARIABLE SIZE            =   3FH+02H    63D+  2D
    INDIRECT VARIABLE SIZE          =   3CH+00H    60D+  0D
    BIT SIZE                        =   01H+00H     1D+  0D
    BIT-ADDRESSABLE SIZE            =   00H+00H     0D+  0D
    AUXILIARY VARIABLE SIZE         = 0006H         6D
    MAXIMUM STACK SIZE              = 0017H        23D
    REGISTER-BANK(S) USED:            0 1 2
    460 LINES READ
    0 PROGRAM ERROR(S)
END OF PL/M-51 COMPILATION
```

296166-50

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                18:50:53  09/19/83  PAGE   1

ISIS-II PL/M-51 V1.0
COMPILER INVOKED BY:   :f2:plm51 :f2:unote.src


                $TITLE      ('Application Module: Async/SDLC Protocol converter')
                $debug
                $registerbank(0)
  1   1         user$mod:do;
                $NOLIST
  5   1         DECLARE     LIT                 LITERALLY       'LITERALLY',
                            TRUE                LIT             'OFFH',
                            FALSE               LIT             '00H',
                            FOREVER             LIT             'WHILE 1',
                            ESC                 LIT             '1BH',
                            LF                  LIT             '0AH',
                            CR                  LIT             '0DH',
                            BS                  LIT             '08H',
                            BEL                 LIT             '07H',
                            EMPTY               LIT             '00H',
                            INUSE               LIT             '01H',
                            FULL                LIT             '02H',
                            USER_STATE_CLOSED   LIT             '00H',
                            LINK_DISCONNECTED   LIT             '01H',
                            OVERFLOW            LIT             '02H',
                            DATA_TRANSMITTED    LIT             '03H',

                                /* BUFFERS */

                            BUFFER_LENGTH       LIT             '60',
                            SIU_XMIT_BUFFER(BUFFER_LENGTH)      BYTE    EXTERNAL    IDATA,
                            SIU_RECV_BUFFER(BUFFER_LENGTH)      BYTE    EXTERNAL,
                            FIFO_T(256)                 BYTE    AUXILIARY,
                            IN_PTR_T                    BYTE    AUXILIARY,
                            OUT_PTR_T                   BYTE    AUXILIARY,
                            BUFFER_STATUS_T             BYTE    AUXILIARY,
                            FIFO_R(256)                 BYTE    AUXILIARY,
                            IN_PTR_R                    BYTE    AUXILIARY,
                            OUT_PTR_R                   BYTE    AUXILIARY,
                            BUFFER_STATUS_R             BYTE    AUXILIARY,

                                /* Variables and Parameters */

                            LENGTH                      BYTE    AUXILIARY,
                            CHAR                        BYTE    AUXILIARY,
                            I                           BYTE    AUXILIARY,
                            USART_CMD                   BYTE    AUXILIARY,
                            DESTINATION_ADDRESS         BYTE    AUXILIARY,
                            SEND_DATA                   BYTE    AUXILIARY,
                            RESULT                      BYTE    AUXILIARY,
                            ERR_MESSAGE_INDEX           BYTE    AUXILIARY,
                            ERR_MESSAGE_PTR             WORD    AUXILIARY,

                              /*   Messages Sent to the Terminal   */

                            PARITY(*) BYTE CONSTANT(LF,CR,'Parity Error Detected',LF,CR,00H),
                            FRAME(*) BYTE CONSTANT(LF,CR,'Framing Error Detected',LF,CR,00H),
```

296166-51

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                18:50:53  09/19/83  PAGE   2

                    OVER_RUN(*) BYTE CONSTANT(LF,CR,'Overrun Error Detected',LF,CR,0),
                    LINK(*) BYTE CONSTANT(LF,CR,'Unable to Get Online',LF,CR,00H),
                    DEST_ADDR(*) BYTE CONSTANT(CR,LF,LF,
                              'Enter the destination address: __', BS, BS, 0),

                    D_ADDR_ACK(*) BYTE CONSTANT(CR,LF,LF,
                              'The new destination address is ',0),

                    STAT_ADDR(*) BYTE CONSTANT(CR,LF,LF,
                              'Enter the station address: __',BS,BS,0),

                    S_ADDR_ACK(*) BYTE CONSTANT(CR,LF,LF,
                              'The new station address is ',0),

                    ADDR_ACK_FIN(*) BYTE CONSTANT('H',CR,LF,LF,0),



                    SIGN_ON(*)  BYTE CONSTANT(CR, LF,LF,
                    '(\/)  RUPI-44 Secondary Station', CR, LF,
                    ' \/', CR, LF, LF,
                    '1 - Set the Station Address',LF,CR,
                    '2 - Set the Destination Address',CR,LF,
                    '3 - Go Online',CR,LF,
                    '4 - Go Offline',CR,LF,
                    '5 - Return to terminal mode',CR,LF,LF,
                    '    Enter option: _', BS, 0),

                    FIN(*)  BYTE CONSTANT(CR,LF,LF,0),


                    /* Characters Received From the Terminal */

                    HEX_TABLE(17) BYTE CONSTANT('0123456789ABCDEF',BEL),
                    MENU_CHAR(6) BYTE CONSTANT('12345',BEL),

                      /* Flags and Bits */

                    XMIT_BUFFER_EMPTY       BIT     EXTERNAL, /* Semaphore  for RUPI SIOU Transmit Buffer */
                    STOP_BIT                BIT     AT(147) REG, /* Terminal parameters */
                    ECHO                    BIT     AT(0B4H) REG,
                    WAIT                    BIT,              /* Timeout flag */
                    ERROR_FLAG              BIT,              /* Error message Flag */

                      /* Periheral Addresses */

                    USART_STATUS            BYTE    AT(0801H) AUXILIARY,
                    USART_DATA              BYTE    AT(0800H) AUXILIARY,
                    TIMER_CONTROL           BYTE    AT(1003H) AUXILIARY,
                    TIMER_0                 BYTE    AT(1000H) AUXILIARY,
                    TIMER_1                 BYTE    AT(1001H) AUXILIARY,
                    TIMER_2                 BYTE    AT(1002H) AUXILIARY;


                       /* External Procedures */
```

296166-52

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                  18:50:53  09/19/83  PAGE   3


   6   2      POWER_ON_D: PROCEDURE   EXTERNAL;
   7   1      END POWER_ON_D;

   8   2      CLOSE: PROCEDURE EXTERNAL USING 2;
   9   1      END CLOSE;

  10   2      OPEN: PROCEDURE EXTERNAL USING 2;
  11   1      END OPEN;

  12   2      TRANSMIT:   PROCEDURE   (XMIT_BUFFER_LENGTH)    BYTE    EXTERNAL;
  13   2                  DECLARE XMIT_BUFFER_LENGTH  BYTE;
  14   1      END TRANSMIT;

                          /* Local Procedures */

  15   2      TIMER_0_INT: PROCEDURE INTERRUPT 1 USING 1;
  16   2          WAIT=0;
  17   1      END TIMER_0_INT;

  18   2      POWER_ON:   PROCEDURE USING 0;

  19   2              DECLARE TEMP    BYTE    AUXILIARY;

  20   2              SMD=54H;                /* Using DPLL, NRZI, PFS, TIMER 1, @ 62.5 Kbps */
  21   2              TMOD=21H;               /* Timer 0 16 bit, Timer 1 auto reload */
  22   2              TH1=OFFH;
  23   2              TCON=40H;

  24   2              TIMER_CONTROL=37H;    /* Initialize USART's system clock: 8254 */
  25   2              TIMER_0=04H;
  26   2              TIMER_0=00H;
  27   2              TIMER_CONTROL=77H;  /* Initialize TxC, RxC  */

              /*        Definition for dip switch tied to P1.0 to P1.6

                        Bit Rate    3       2       1
                       ------------------------------------
                          300       on      on      on
                          1200      on      on      off
                          2400      on      off     on
                          4800      on      off     off
                          9600      off     on      on
                          19200     off     on      off

                        Stop bit     4
                       ------------------
                          1         on
                          2         off

                        Parity      6       5
                       ------------------------
                        off         on      on
                        odd         on      off
                        off         off     on
                        even        off     off
```

296166-53

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                18:50:53  09/19/83  PAGE   4


                        Echo        7
                        ____________
                        on          on
                        off         off                 */

  28   2              TEMP=P1 AND 07H;  /* Read the dip switch to determine the bit rate */
  29   2              IF TEMP>5
                          THEN TEMP=0;
  31   3              DO CASE TEMP;
                /* 300 */
  32   4                  DO;
  33   4                      TIMER_1=83H;
  34   4                      TIMER_1=20H;
  35   4                  END;

  36   4       /* 1200 */  DO;
  37   4                      TIMER_1=20H;
  38   4                      TIMER_1=05H;
  39   4                  END;

  40   4       /* 2400 */  DO;
  41   4                      TIMER_1=60H;
  42   4                      TIMER_1=02H;
  43   4                  END;

  44   4       /* 4800 */  DO;
  45   4                      TIMER_1=30H;
  46   4                      TIMER_1=01H;
  47   4                  END;

  48   4       /* 9600 */  DO;
  49   4                      TIMER_1=65H;
  50   4                      TIMER_1=0;
  51   4                  END;

  52   4       /* 19200 */ DO;
  53   4                      TIMER_1=33H;
  54   4                      TIMER_1=0;
  55   4                  END;
  56   3              END;

  57   2              USART_STATUS=0;   /* Software power-on reset for 8251A */
  58   2              USART_STATUS=0;
  59   2              USART_STATUS=0;
  60   2              USART_STATUS=40H;

  61   2              TEMP=0AH;         /* Determine the parity and # of stop bits */
  62   2              TEMP=TEMP OR (P1 AND 30H);
  63   2              IF STOP_BIT=1
                          THEN TEMP=TEMP OR 0C0H;
  65   2              ELSE TEMP=TEMP OR 40H;

  66   2              USART_STATUS=TEMP;   /* USART Mode Word */
  67   2              USART_STATUS, USART_CMD=27H; /*USART Command Word RTS, RxE, DTR, TxEN=1*/

  68   2              STAD=0FFH;
```

296166-54

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                    18:50:53  09/19/83  PAGE   5


   69   2              SEND_DATA=0;  /* Intialize Flags */

   70   2             IN_PTR_T, OUT_PTR_T, IN_PTR_R, OUT_PTR_R = 0;/*Initialize FIFO PTRs*/

   71   2              BUFFER_STATUS_T, BUFFER_STATUS_R= EMPTY;

   72   2              CALL POWER_ON_D;

   73   2              IP=01H;            /* USART's RxRdy is the highest priority */
                                           /* Both external interrupts are level triggered*/
   74   2              IE=93H;            /* Enable USART RxRdy, SI, and Timer 0 interrupts*/

   75   2              ERROR_FLAG=0;

   76   1      END POWER_ON;

   77   2      FIFO_R_IN: PROCEDURE (CHAR) USING 1;
   78   2          DECLARE CHAR     BYTE;

   79   2          FIFO_R(IN_PTR_R)=CHAR;
   80   2          IN_PTR_R=IN_PTR_R+1;

   81   2          IF BUFFER_STATUS_R=EMPTY
                      THEN DO;
   83   3                       EA=0;
   84   3                       BUFFER_STATUS_R=INUSE;
   85   3                       EX1=1;       /* Enable USART's TxD interrupt */
   86   3                       EA=1;
   87   3                   END;
   88   2          ELSE IF ((BUFFER_STATUS_R=INUSE) AND (IN_PTR_R=OUT_PTR_R))
                       THEN BUFFER_STATUS_R=FULL;

   90   1      END FIFO_R_IN;

   91   2      FIFO_R_OUT: PROCEDURE BYTE USING 1;

   92   2          DECLARE CHAR     BYTE     AUXILIARY;

   93   2          CHAR=FIFO_R(OUT_PTR_R);
   94   2          OUT_PTR_R=OUT_PTR_R+1;
   95   2          IF OUT_PTR_R=IN_PTR_R
                      THEN DO;
   97   3                       EX1=0;  /* Shut off TxD interrupt */
   98   3                       BUFFER_STATUS_R=EMPTY;
   99   3                   END;
  100   2          ELSE IF ((BUFFER_STATUS_R=FULL) AND (OUT_PTR_R-20=IN_PTR_R))
                       THEN BUFFER_STATUS_R=INUSE;

  102   2          RETURN CHAR;

  103   1      END FIFO_R_OUT;


  104   2      USART_XMIT_INT: PROCEDURE INTERRUPT 2 USING 1;
```

296166-55

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                    18:50:53  09/19/83  PAGE   6

 105   2          DECLARE
                        MESSAGE BASED ERR_MESSAGE_PTR(1)    BYTE    CONSTANT;

 106   2          IF ERROR_FLAG
                    THEN DO;
 108   3                    IF MESSAGE(ERR_MESSAGE_INDEX)<>0  /* Then continue to send the message */
                                THEN DO;
 110   4                                 USART_DATA = MESSAGE(ERR_MESSAGE_INDEX);
 111   4                                 ERR_MESSAGE_INDEX=ERR_MESSAGE_INDEX+1;
 112   4                               END;

 113   4                           ELSE DO; /* If message is done reset ERROR_FLAG and shut off interrupt if FIFO is empty */
 114   4                                  ERROR_FLAG=0;
 115   4                                  IF BUFFER_STATUS_R = EMPTY
                                              THEN EX1=0;
 117   4                                END;
 118   3                         END;

 119   2          ELSE USART_DATA=FIFO_R_OUT;

 120   1      END USART_XMIT_INT;


 121   2      SIU_RECV: PROCEDURE (LENGTH) PUBLIC USING 1;

 122   2          DECLARE LENGTH  BYTE,
                         I       BYTE    AUXILIARY;

 123   3          DO I=0 TO LENGTH-1;
 124   4              DO WHILE BUFFER_STATUS_R=FULL; /* Check to see if fifo is full */
 125   4              END;
 126   3              CALL FIFO_R_IN(SIU_RECV_BUFFER(I));
 127   3          END;

 128   1      END SIU_RECV;


 129   2      FIFO_T_IN: PROCEDURE (CHAR) USING 2;

 130   2          DECLARE CHAR         BYTE;

 131   2          FIFO_T(IN_PTR_T)=CHAR;
 132   2          IN_PTR_T=IN_PTR_T+1;
 133   2          IF CHAR=LF
                    THEN SEND_DATA=SEND_DATA+1;

 135   2          IF BUFFER_STATUS_T=EMPTY
                    THEN BUFFER_STATUS_T=INUSE;
 137   2          ELSE IF ((BUFFER_STATUS_T=INUSE) AND (IN_PTR_T+20=OUT_PTR_T))
                         THEN DO; /* Stop reception using CTS */
 139   3                           USART_STATUS, USART_CMD=USART_CMD AND NOT(20H);
 140   3                           BUFFER_STATUS_T=FULL;
 141   3                           IF SEND_DATA=0
                                     THEN SEND_DATA=1;/*If the buffer is full and no LF
                                                        has been received then send data */
 143   3                      END;
 144   1      END FIFO_T_IN;
```

296166-56

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                    18:50:53  09/19/83  PAGE   7


  145   2     FIFO_T_OUT: PROCEDURE BYTE ;

  146   2         DECLARE CHAR    BYTE    AUXILIARY;

  147   2         CHAR=FIFO_T(OUT_PTR_T);
  148   2         OUT_PTR_T=OUT_PTR_T+1;
  149   2         IF OUT_PTR_T=IN_PTR_T   /* Then FIFO_T is empty */
                      THEN DO;
  151   3                   EA=0;
  152   3                   BUFFER_STATUS_T=EMPTY;
  153   3                   SEND_DATA=0;
  154   3                   EA=1;
  155   3                END;
  156   2         ELSE IF ((BUFFER_STATUS_T=FULL) AND (OUT_PTR_T-80=IN_PTR_T))
                      THEN DO;
  158   3                      USART_STATUS, USART_CMD=USART_CMD OR 20H;
  159   3                      BUFFER_STATUS_T=INUSE;
  160   3                   END;
  161   2         IF (CHAR=LF AND SEND_DATA>0) THEN SEND_DATA=SEND_DATA-1;
  163   2         RETURN CHAR;
  164   1     END FIFO_T_OUT;

  165   2     ERROR: PROCEDURE (STATUS) USING 2;

  166   2           DECLARE STATUS      BYTE;

  167   2           IF (STATUS AND 08H)<>0
                      THEN ERR_MESSAGE_PTR=.PARITY;
  169   2           ELSE IF (STATUS AND 10H)<>0
                      THEN ERR_MESSAGE_PTR=.OVER_RUN;
  171   2           ELSE IF (STATUS AND 20H)<>0
                      THEN ERR_MESSAGE_PTR=.FRAME;

  173   2           USART_STATUS=(USART_CMD OR 10H); /* Reset error flags on USART */

  174   2           ERR_MESSAGE_INDEX = 0;
  175   2           ERROR_FLAG=1;
  176   2           EX1=1;          /* Turn on Tx Interrupt */

  177   1     END ERROR;

  178   2     LINK_DISC: PROCEDURE ;

           /* This procedure sends the message 'Unable to Get Online' to the terminal */

  179   2           DECLARE MESSAGE_PTR WORD    AUXILIARY,
                       MESSAGE     BASED   MESSAGE_PTR(1)      BYTE    CONSTANT,
                       J           BYTE    AUXILIARY,
                       EX1_STORE   BIT;


  180   2           EX1_STORE=EX1;    /* Shut off async transmit interrupt */
  181   2           EX1=0;
  182   2           MESSAGE_PTR=.LINK;
  183   2           J=0;
  184   3           DO WHILE (MESSAGE(J)<>0);
```

296166-57

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                    18:50:53  09/19/83  PAGE   8

185   4             DO WHILE (USART_STATUS AND 01H)=0;  /* Wait for TxRDY on USART */
186   4             END;
187   3             USART_DATA=MESSAGE(J);
188   3             J=J+1;
189   3         END;
190   2         EX1=EX1_STORE;    /* Restore async transmit interrupt */
191   1     END LINK_DISC;


192   2     CO: PROCEDURE (CHAR) USING 2;
193   2             DECLARE CHAR    BYTE;

194   3             DO WHILE (USART_STATUS AND 01H) = 0;
195   3             END;
196   2             USART_DATA=CHAR;

197   1     END CO;

198   2     CI: PROCEDURE BYTE USING 2;

199   3         DO WHILE (USART_STATUS AND 02H) = 0;
200   3         END;
201   2         RETURN USART_DATA;

202   1     END CI;


203   2     GET_HEX: PROCEDURE BYTE USING 2;

204   2         DECLARE CHAR    BYTE    AUXILIARY,
                      I       BYTE    AUXILIARY;

205   2         L0: CHAR=CI;

206   3             DO I=0 TO 15;
207   3                IF CHAR=HEX_TABLE(I)
                           THEN GOTO L1;
209   3             END;

210   2         L1: CALL CO(HEX_TABLE(I));
211   2             IF I=16
                       THEN GOTO L0;

213   2             RETURN I;

214   1     END GET_HEX;

215   2     OUTPUT_MESSAGE: PROCEDURE (MESSAGE_PTR) USING 2;
216   2         DECLARE MESSAGE_PTR WORD,
                      MESSAGE     BASED   MESSAGE_PTR(1)  BYTE CONSTANT,
                      I           BYTE    AUXILIARY;

217   2         I=0;

218   3         DO WHILE MESSAGE(I) <> 0;
219   3             CALL CO(MESSAGE(I));
220   3             I=I+1;
```

296166-58

```
PL/M-51 COMPILER   Application Module: Async/SDLC Protocol converter                    18:50:53  09/19/83  PAGE   9

 221   3          END;

 222   1      END OUTPUT_MESSAGE;


 223   2      MENU: PROCEDURE USING 2;

 224   2          DECLARE I                BYTE    AUXILIARY,
                       CHAR               BYTE    AUXILIARY,
                       STATION_ADDRESS BYTE    AUXILIARY;


 225   2      START:
                   CALL OUTPUT_MESSAGE(.SIGN_ON);

 226   2          M0: CHAR=CI;     /* Read a character */

 227   3              DO I=0 TO 4;
 228   3                  IF CHAR=MENU_CHAR(I)
                             THEN GOTO M1;
 230   3              END;

 231   2          M1: CALL CO(MENU_CHAR(I));
 232   2              IF I=5
                         THEN GOTO M0;

 234   3          DO CASE I;

 235   4              DO;
 236   4                  CALL OUTPUT_MESSAGE(.STAT_ADDR);

 237   4                  STATION_ADDRESS=SHL(GET_HEX,4);

 238   4                  STATION_ADDRESS=(STATION_ADDRESS OR GET_HEX);

 239   4                  STAD=STATION_ADDRESS;

 240   4                  CALL OUTPUT_MESSAGE(.S_ADDR_ACK);

 241   4                  CALL CO(HEX_TABLE(SHR(STATION_ADDRESS,4)));
 242   4                  CALL CO(HEX_TABLE(0FH AND STATION_ADDRESS));

 243   4                  CALL OUTPUT_MESSAGE(.ADDR_ACK_FIN);
 244   4              END;

 245   4              DO;

 246   4                  CALL OUTPUT_MESSAGE(.DEST_ADDR);

 247   4                  DESTINATION_ADDRESS=SHL(GET_HEX,4);

 248   4                  DESTINATION_ADDRESS=(DESTINATION_ADDRESS OR GET_HEX );

 249   4                  CALL OUTPUT_MESSAGE(.D_ADDR_ACK);
```

296166-59

```
PL/M-51 COMPILER    Application Module: Async/BDLC Protocol converter                    18:50:53  09/19/83  PAGE  10

 250   4                  CALL CO(HEX_TABLE(SHR(DESTINATION_ADDRESS,4)));
 251   4                  CALL CO(HEX_TABLE(0FH AND DESTINATION_ADDRESS));

 252   4                  CALL OUTPUT_MESSAGE(.ADDR_ACK_FIN);
 253   4              END;

 254   4              DO;
 255   4                  CALL OUTPUT_MESSAGE(.FIN);
 256   4                  CALL OPEN;
 257   4              END;


 258   4              DO;
 259   4                  CALL OUTPUT_MESSAGE(.FIN);
 260   4                  CALL CLOSE;
 261   4              END;

 262   3              CALL OUTPUT_MESSAGE(.FIN);

 263   3          END; /* DO CASE */

 264   1      END MENU;


 265   2      USART_RECV_INT: PROCEDURE INTERRUPT 0 USING 2;

 266   2          DECLARE CHAR        BYTE    AUXILIARY,
                         STATUS      BYTE    AUXILIARY;


 267   2          CHAR=USART_DATA;
 268   2          STATUS=USART_STATUS AND 38H;
 269   2          IF STATUS<>0
                     THEN CALL ERROR(STATUS);
 271   2          ELSE IF CHAR=ESC
                         THEN CALL MENU;
 273   3          ELSE DO;
 274   3                  CALL FIFO_T_IN(CHAR);
 275   3                  IF ECHO=0
                             THEN CALL CO(CHAR);
 277   3              END;

 278   1      END USART_RECV_INT;


 279   1      BEGIN:
              CALL POWER_ON;

 280   2          DO FOREVER;
 281   2              IF SEND_DATA>0
                         THEN DO;
 283   4                          DO WHILE NOT(XMIT_BUFFER_EMPTY); /*Wait until SIU_XMIT_BUFFer
                                                                      is empty */
 284   4                          END;
 285   3                          LENGTH, CHAR =1;
 286   3                          SIU_XMIT_BUFFER(0)=DESTINATION_ADDRESS;
 287   4                          DO WHILE ((CHAR<>LF) AND (LENGTH<BUFFER_LENGTH) AND (BUFFER_STATUS_T<>EMPTY));
```

296166-60

```
PL/M-51 COMPILER    Application Module: Async/SDLC Protocol converter                18:50:53  09/19/83  PAGE  11

 288   4                              CHAR=FIFO_T_OUT;
 289   4                              SIU_XMIT_BUFFER(LENGTH)=CHAR;
 290   4                              LENGTH=LENGTH+1;
 291   4                          END;

              /* If the line entered at the terminal is greater than BUFFER_LENGTH char, send the
                 first BUFFER_LENGTH char, then send the rest; since the SIU buffer is only BUFFER_LENGTH bytes */
 292   3      L1:                 I=0; /* Use I to count the number of unsuccesful
                                        transmits */

 293   3      RETRY:          RESULT=TRANSMIT(LENGTH); /* Send the message */
 294   3                          IF RESULT<>DATA_TRANSMITTED
                                    THEN DO;
                             /* Wait 50 msec for link to connect then try again */
 296   4                              WAIT=1;
 297   4                              TH0=3CH;
 298   4                              TL0=0AFH;
 299   4                              TR0=1;
 300   5                              DO WHILE WAIT;
 301   5                              END;
 302   4                              TR0=0;
 303   4                              I=I+1;
 304   5                              IF I>100 THEN DO;  /* Wait 5 sec to get on line else
                                                            send error message to terminal
                                                            and try again  */
 306   5                                          CALL LINK_DISC;
 307   5                                          GOTO L1;
 308   5                                      END;
 309   4                              GOTO RETRY;
 310   4                          END;
 311   3                      END;

 312   2          END;
 313   1      END USER$MOD;



WARNING8:
    2 IS THE HIGHEST USED INTERRUPT


MODULE INFORMATION:                 (STATIC+OVERLAYABLE)
    CODE SIZE                       = 06B2H      1714D
    CONSTANT SIZE                   = 01CFH       463D
    DIRECT VARIABLE SIZE            =   00H+05H     0D+  5D
    INDIRECT VARIABLE SIZE          =   00H+00H     0D+  0D
    BIT SIZE                        =   02H+01H     2D+  1D
    BIT-ADDRESSABLE SIZE            =   00H+00H     0D+  0D
    AUXILIARY VARIABLE SIZE         = 021FH       543D
    MAXIMUM STACK SIZE              = 0028H        40D
    REGISTER-BANK(S) USED:            0 1 2
    713 LINES READ
    0 PROGRAM ERROR(S)
END OF PL/M-51 COMPILATION
```

296166-61

```
PL/M-51 COMPILER    RUPI-44 Primary Station                          20:47:13  09/26/83  PAGE   1

ISIS-II PL/M-51 V1.0
COMPILER INVOKED BY:   :F2:PLM51 :F2:PNOTE.SRC


                $TITLE      ('RUPI-44 Primary Station')
                $DEBUG
                $REGISTERBANK(0)
  1   1         MAIN$MOD:DO;

                /* To save paper the RUPI registers are not listed, but this is the statement
                   used to include them:  $INCLUDE (:F2:REG44.DCL) */

                $NOLIST

  5   1         DECLARE LIT             LITERALLY   'LITERALLY',
                        TRUE            LIT         '0FFH',
                        FALSE           LIT         '00H',
                        FOREVER         LIT         'WHILE 1';

                /* SDLC COMMANDS AND RESPONSES */

  6   1         DECLARE SNRM            LIT         '93H',
                        UA              LIT         '73H',
                        DISC            LIT         '53H',
                        DM              LIT         '1FH',
                        FRMR            LIT         '97H',
                        REQ_DISC        LIT         '53H',
                        UP              LIT         '33H',
                        TEST            LIT         '0F3H',
                        RR              LIT         '11H',
                        RNR             LIT         '15H',


                              /* REMOTE STATION BUFFER STATUS */

                        BUFFER_READY        LIT         '0',
                        BUFFER_NOT_READY    LIT         '1',

                              /* STATION STATES */
                        DISCONNECT_S    LIT         '00H', /* LOGICALLY DISCONNECTED STATE*/
                        GO_TO_DISC      LIT         '01H',
                        I_T_S           LIT         '02H', /* INFORMATION TRANSFER STATE */


                        /* PARAMETERS PASSED TO XMIT_I_T_S */
                        T_I_FRAME       LIT     '00H',
                        T_RR            LIT     '01H',
                        T_RNR           LIT     '02H',


                        /* SECONDARY STATION IDENTIFICATION */

                        NUMBER_OF_STATIONS  LIT     '2',
                        SECONDARY_ADDRESSES(NUMBER_OF_STATIONS)
                                        BYTE    CONSTANT(55H,43H),
```

296166-62

```
PL/M-51 COMPILER    RUPI-44 Primary Station                          20:47:13  09/26/83  PAGE   2


                        /* Remote Station Database */

                    RSD(NUMBER_OF_STATIONS) STRUCTURE
                    (STATION_ADDRESS     BYTE,
                     STATION_STATE       BYTE,
                     NS                  BYTE,
                     NR                  BYTE,
                     BUFFER_STATUS       BYTE,  /* The status of the secondary stations buffer */
                     INFO_LENGTH         BYTE,
                     DATA(64)            BYTE)   AUXILIARY,


                        /* VARIABLES */
                    STATION_NUMBER       BYTE    AUXILIARY,
                    RECV_FIELD_LENGTH    BYTE    AUXILIARY,
                    WAIT                 BIT,

                        /* BUFFERS */
                    SIU_XMIT_BUFFER(64)      BYTE    IDATA,
                    SIU_RECV_BUFFER(64)      BYTE;


  7   2     POWER_ON: PROCEDURE ;

  8   2         DECLARE I   BYTE    AUXILIARY;

  9   2             TBS=.SIU_XMIT_BUFFER(0);
 10   2             RBS=.SIU_RECV_BUFFER(0);
 11   2             RBL=64;      /* 64 Byte receive buffer */
 12   2             RBE=1;       /* Enable the SIU's receiver */

 13   3             DO I= 0 TO NUMBER_OF_STATIONS-1;

 14   3                 RSD(I).STATION_ADDRESS=SECONDARY_ADDRESSES(I);
 15   3                 RSD(I).STATION_STATE=DISCONNECT_S;
 16   3                 RSD(I).BUFFER_STATUS=BUFFER_NOT_READY;
 17   3                 RSD(I).INFO_LENGTH=0;

 18   3             END;

 19   2             SMD=54H; /* Using DPLL, NRZI, PFS, TIMER 1, @ 62.5 Kbps */
 20   2             TMOD=21H;
 21   2             TH1=0FFH;
 22   2             TCON=40H;  /* Use timer 0 for receive time out interrupt */
 23   2             IE=82H;

 24   1     END POWER_ON;

 25   2     XMIT: PROCEDURE (CONTROL_BYTE);

 26   2         DECLARE CONTROL_BYTE    BYTE;

 27   2             TCB=CONTROL_BYTE;
 28   2             TBF=1;
```

296166-63

```
PL/M-51 COMPILER    RUPI-44 Primary Station                          20:47:13  09/26/83  PAGE   3

  29   2              RTS=1;
  30   3              DO WHILE NOT SI;
  31   3              END;
  32   2              SI=0;

  33   1      END XMIT;


  34   2      TIMER_0_INT: PROCEDURE INTERRUPT 1  USING 1;
  35   2          WAIT=0;
  36   1      END TIMER_0_INT;


  37   2      TIME_OUT: PROCEDURE BYTE;          /* Time_out returns true if there wasn't
                                                    a frame received within 200 msec.
                                                    If there was a frame received within
                                                    200 msec then time_out returns false. */
  38   2          DECLARE     I   BYTE    AUXILIARY;

  39   3          DO I=0 TO 3;

  40   3              WAIT=1;
  41   3              TH0=3CH;
  42   3              TL0=0AFH;
  43   3              TR0=1;
  44   4                  DO WHILE WAIT;
  45   4                      IF SI=1
                                 THEN GOTO T_01;
  47   4                  END;
  48   3          END;

  49   2          RETURN TRUE;

  50   2      T_01:
                  SI=0;
  51   2              RETURN FALSE;

  52   1      END TIME_OUT;

  53   2      SEND_DISC: PROCEDURE;

  54   2                  TBL=0;
  55   2                  CALL XMIT(DISC);
  56   2                  IF TIME_OUT=FALSE
                              THEN IF RCB=UA OR RCB=DM
  57   3                                 THEN DO;
  59   3                                         RSD(STATION_NUMBER).BUFFER_STATUS=BUFFER_NOT_READY;
  60   3                                         RSD(STATION_NUMBER).STATION_STATE=DISCONNECT_S;
  61   3                                     END;
  62   2                  RBE=1;

  63   1      END SEND_DISC;


  64   2      SEND_SNRM: PROCEDURE;

  65   2          TBL=0;
```

296166-64

```
PL/M-51 COMPILER    RUPI-44 Primary Station                                    20:47:13  09/26/83  PAGE   4

  66   2          CALL XMIT(SNRM);
  67   2          IF (TIME_OUT=FALSE) AND (RCB=UA)
                        THEN DO;
  69   3                          RSD(STATION_NUMBER).STATION_STATE=I_T_S;
  70   3                          RSD(STATION_NUMBER).NS=0;
  71   3                          RSD(STATION_NUMBER).NR=0;
  72   3                      END;
  73   2          RBE=1;

  74   1      END SEND_SNRM;


  75   2      CHECK_NS: PROCEDURE BYTE;

               /* Check the Ns Field of the received frame.  If Nr(P)=Ns(S) return true */

  76   2          IF (RSD(STATION_NUMBER).NR=(SHR(RCB,1) AND 07H))
                   THEN RETURN TRUE;
  78   2              ELSE RETURN FALSE;

  79   1      END CHECK_NS;

  80   2      CHECK_NR: PROCEDURE BYTE;

              /* Check the Nr field of the received frame.  If Ns(P)+1=Nr(S) then the frame
                  has been acknowledged, else if Ns(P)=Nr(S) then the frame has not been
                  acknowledged, else reset the secondary */

  81   2          IF (((RSD(STATION_NUMBER).NS + 1) AND 07H) = SHR(RCB,5))
                   THEN DO;
  83   3                      RSD(STATION_NUMBER).NS=((RSD(STATION_NUMBER).NS+1) AND 07H);
  84   3                      RSD(STATION_NUMBER).INFO_LENGTH=0;
  85   3                  END;
  86   2          ELSE IF (RSD(STATION_NUMBER).NS <> SHR(RCB,5))
                       THEN RETURN FALSE;

  88   2          RETURN TRUE;

  89   1      END CHECK_NR;


  90   2      RECEIVE: PROCEDURE ;

  91   2              DECLARE     I   BYTE    AUXILIARY;

  92   2              RSD(STATION_NUMBER).BUFFER_STATUS=BUFFER_READY;

               /* If an RNR was received buffer_status will be changed in the supervisory
                  frame decode section futher down in this procedure, any other response
                  means the remote stations buffer is ready  */

  93   2              IF (RCB AND 01H)=0
                       THEN DO;  /* I Frame Received */
  95   3                          IF (CHECK_NS=TRUE AND BOV=0 AND CHECK_NR=TRUE)
                                   THEN DO;
  97   4                                  RSD(STATION_NUMBER).NR=((RSD(STATION_NUMBER).NR+1) AND 07H);
  98   4                                  RBP=1;
```

296166-65

```
PL/M-51 COMPILER    RUPI-44 Primary Station                                    20:47:13  09/26/83  PAGE   5

 99   4                                    RECV_FIELD_LENGTH=RFL-1;
100   4                                END;

101   3                          ELSE RSD(STATION_NUMBER).STATION_STATE=GO_TO_DISC;
102   3                       END;
103   2             ELSE IF (RCB AND 03H)=01H
                         THEN DO;  /* Supervisory frame received */
105   3                             IF CHECK_NR=FALSE
                                        THEN RSD(STATION_NUMBER).STATION_STATE=GO_TO_DISC;

107   3                             ELSE IF ((RCB AND 0FH)=05H) /* then RNR */
                                        THEN RSD(STATION_NUMBER).BUFFER_STATUS=BUFFER_NOT_READY;
109   3                           END;

110   3             ELSE DO;          /* Unnumbered frame or unknown frame received */
111   3                     IF RCB=FRMR
                              THEN DO;   /* If FRMR was received check Nr for an
                                            acknowledged I frame  */
113   4                                 RCB=SIU_RECV_BUFFER(1);
114   4                                 I=CHECK_NR;
115   4                               END;
116   3                     RSD(STATION_NUMBER).STATION_STATE=GO_TO_DISC;
117   3                 END;

118   2             RBE=1;

119   1      END RECEIVE;



120   2      XMIT_I_T_S: PROCEDURE (TEMP);

121   2          DECLARE      TEMP     BYTE;

122   2              IF TEMP=T_I_FRAME
                       THEN DO;    /* Transmit I frame */
                            /* Transfer the station buffer into internal ram */

124   4                        DO TEMP=0 TO RSD(STATION_NUMBER).INFO_LENGTH-1;
125   4                        SIU_XMIT_BUFFER(TEMP)=RSD(STATION_NUMBER).DATA(TEMP);
126   4                        END;

                               /* Build the I frame control field */

127   3                        TEMP=(SHL(RSD(STATION_NUMBER).NR,5) OR SHL(RSD(STATION_NUMBER).NS,1) OR 10H);
128   3                        TBL=RSD(STATION_NUMBER).INFO_LENGTH;
129   3                        CALL XMIT(TEMP);
130   3                        IF TIME_OUT=FALSE
                                   THEN CALL RECEIVE;
132   3                     END;

133   3              ELSE DO;    /* Transmit RR or RNR*/
134   3                     IF TEMP=T_RR
                                THEN TEMP=RR;
136   3                     ELSE TEMP=RNR;
```

296166-66

```
PL/M-51 COMPILER    RUPI-44 Primary Station                                  20:47:13  09/26/83  PAGE   6

 137   3                      TEMP=(SHL(RSD(STATION_NUMBER).NR,5) OR TEMP);
 138   3                      TBL=0;
 139   3                      CALL XMIT(TEMP);
 140   3                      IF TIME_OUT=FALSE
                                THEN CALL RECEIVE;
 142   3                  END;
 143   1      END XMIT_I_T_S;

 144   2      BUFFER_TRANSFER: PROCEDURE;

 145   2          DECLARE     I       BYTE    AUXILIARY,
                             J       BYTE    AUXILIARY;


 146   3          DO I=0 TO NUMBER_OF_STATIONS-1;
 147   3              IF RSD(I).STATION_ADDRESS=SIU_RECV_BUFFER(0)
                         THEN GOTO T1;
 149   3          END;

 150   2        T1:   IF I=NUMBER_OF_STATIONS  /* If the addressed station does not exits,
                                                  then discard the data */
                         THEN DO;
 152   3                          RBP=0;
 153   3                          RETURN;
 154   3                      END;
 155   2                  ELSE IF RSD(I).INFO_LENGTH=0
                             THEN DO;
 157   3                          RSD(I).INFO_LENGTH=RECV_FIELD_LENGTH;
 158   4                          DO J=1 TO RECV_FIELD_LENGTH;
 159   4                              RSD(I).DATA(J-1)=SIU_RECV_BUFFER(J);
 160   4                          END;
 161   3                          RBP=0;
 162   3                      END;

 163   1      END BUFFER_TRANSFER;


 164   1      BEGIN:
              CALL POWER_ON;

 165   2          DO FOREVER;

 166   3              DO STATION_NUMBER=0 TO NUMBER_OF_STATIONS-1;
 167   3                  STAD=RSD(STATION_NUMBER).STATION_ADDRESS;
 168   3                  IF RSD(STATION_NUMBER).STATION_STATE = DISCONNECT_S
                             THEN CALL SEND_SNRM;
 170   3                  ELSE IF RSD(STATION_NUMBER).STATION_STATE = GO_TO_DISC
                                 THEN CALL SEND_DISC;
 172   3                  ELSE IF ((RSD(STATION_NUMBER).INFO_LENGTH>0) AND
                                  (RSD(STATION_NUMBER).BUFFER_STATUS=BUFFER_READY))
                                 THEN CALL XMIT_I_T_S(T_I_FRAME);
 174   3                  ELSE IF RBP=0
                                 THEN CALL XMIT_I_T_S(T_RR);
 176   3                  ELSE CALL XMIT_I_T_S(T_RNR);

 177   3                  IF RBP=1
                             THEN CALL BUFFER_TRANSFER;
```

296166-67

```
PL/M-51 COMPILER    RUPI-44 Primary Station                                  20:47:13  09/26/83  PAGE   7

 179   3              END;

 180   2          END;

 181   1      END MAIN$MOD;



WARNINGS:
    1 IS THE HIGHEST USED INTERRUPT


MODULE INFORMATION:                 (STATIC+OVERLAYABLE)
    CODE SIZE                       = 053DH      1341D
    CONSTANT SIZE                   = 0002H         2D
    DIRECT VARIABLE SIZE            =   40H+02H    64D+  2D
    INDIRECT VARIABLE SIZE          =   40H+00H    64D+  0D
    BIT SIZE                        =   01H+00H     1D+  0D
    BIT-ADDRESSABLE SIZE            =   00H+00H     0D+  0D
    AUXILIARY VARIABLE SIZE         = 0093H       147D
    MAXIMUM STACK SIZE              = 0019H        25D
    REGISTER-BANK(S) USED:            0 1
    456 LINES READ
    0 PROGRAM ERROR(S)
END OF PL/M-51 COMPILATION
```

296166-68

8044 Data Sheet

8044AH/8344AH/8744H
HIGH PERFORMANCE 8-BIT MICROCONTROLLER WITH ON-CHIP SERIAL COMMUNICATION CONTROLLER

- **8044AH—Includes Factory Mask Programmable ROM**
- **8344AH—For Use with External Program Memory**
- **8744H—Includes User Programmable/Eraseable EPROM**

8051 MICROCONTROLLER CORE

- **Optimized for Real Time Control 12 MHz Clock, Priority Interrupts, 32 Programmable I/O Lines, Two 16-bit Timer/Counters**
- **Boolean Processor**
- **4K × 8 ROM, 192 × 8 RAM**
- **64K Accessible External Program Memory**
- **64K Accessible External Data Memory**
- **4 μs Multiply and Divide**

SERIAL INTERFACE UNIT (SIU)

- **Serial Communication Processor that Operates Concurrently to CPU**
- **2.4 Mbps Maximum Data Rate**
- **375 Kbps using On-Chip Phase Locked Loop**
- **Communication Software in Silicon:**
  — Complete Data Link Functions
  — Automatic Station Response
- **Operates as an SDLC Primary or Secondary Station**

The RUPI-44 family integrates a high performance 8-bit Microcontroller, the Intel 8051 Core, with an Intelligent/high performance HDLC/SDLC serial communication controller, called the Serial Interface Unit (SIU). See Figure 1. This dual architecture allows complex control and high speed data communication functions to be realized cost effectively.

Specifically, the 8044's Microcontroller features: 4K byte On-Chip program memory space; 32 I/O lines; two 16-bit timer/event counters; a 5-source; 2-level interrupt structure; a full duplex serial channel; a Boolean processor; and on-chip oscillator and clock circuitry. Standard TTL and most byte-oriented MCS-80 and MCS-85 peripherals can be used for I/O amd memory expansion.

The Serial Interface Unit (SIU) manages the interface to a high speed serial link. The SIU offloads the On-Chip 8051 Microcontroller of communication tasks, thereby freeing the CPU to concentrate on real time control tasks.

The RUPI-44 family consists of the 8044, 8744, and 8344. All three devices are identical except in respect of on-chip program memory. The 8044 contains 4K bytes of mask-programmable ROM. User programmable EPROM replaces ROM in the 8744. The 8344 addresses all program memory externally.

The RUPI-44 devices are fabricated with Intel's reliable +5 volt, silicon-gate HMOSII technology and packaged in a 40-pin DIP.

The 8744H is available in a hermetically sealed, ceramic, 40-lead dual in-line package which includes a window that allows for EPROM erasure when exposed to ultraviolet light (See Erasure Characteristics). During normal operation, ambient light may adversely affect the functionality of the chip. Therefore applications which expose the 8744H to ambient light may require an opaque label over the window.

8044's Dual Controller Architecture

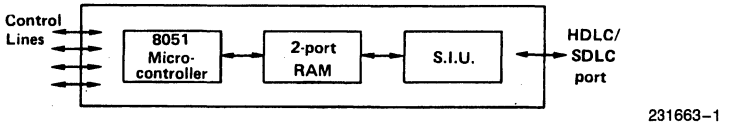


Figure 1. Dual Controller Architecture

October 1987
Order Number: 231663-004

Table 1. RUPI™-44 Family Pin Description

VSS

Circuit ground potential.

VCC

+5V power supply during operation and program verification.

PORT 0

Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads (six in 8744).

PORT 1

Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.

In non-loop mode two of the I/O lines serve alternate functions:

— $\overline{RTS}$ (P1.6). Request-to-Send output. A low indicates that the RUPI-44 is ready to transmit.

— $\overline{CTS}$ (P1.7) Clear-to-Send input. A low indicates that a receiving station is ready to receive.

PORT 2

Port 2 is an 8-bit quasi-bidirection I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.

PORT 3

Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and RD and WR pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS LTT loads.

In addition to I/O, some of the pins also serve alternate functions as follows:

— I/$\overline{O}$ RxD (P3.0). In point-to-point or multipoint configurations, this pin controls the direction of pin P3.1. Serves as Receive Data input in loop and diagnostic modes.

— DATA TxD (P3.1) In point-to-point or multipoint configurations, this pin functions as data input/output. In loop mode, it serves as transmit pin. A '0' written to this pin enables diagnostic mode.

— $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0.

— $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1.

— TO (P3.4). Input to counter 0.

— SCLK T1 (P3.5). In addition to I/O, this pin provides input to counter 1 or serves as SCLK (serial clock) input.

— $\overline{WR}$ (P3.6). The write control signal latches the data byte from Port 0 into the External Data Memory.

— $\overline{RD}$ (P3.7). The read control signal enables External Data Memory to Port 0.

RST

A high on this pin for two machine cycles while the oscillator is running resets the device. A small external pulldown resistor ($\approx$8.2KΩ) from RST to V_{SS} permits power-on reset when a capacitor ($\approx 10\mu f$) is also connected from this pin to V_{CC}.

ALE/$\overline{PROG}$

Provides Address Latch Enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access. It also receives the program pulse input for programming the EPROM version.

$\overline{PSEN}$

The Program Store Enable output is a control signal that enables the external Program Memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.

$\overline{EA}$/VPP

When held at a TTL high level, the RUPI-44 executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the RUPI-44 fetches all instructions from external Program Memory. The pin also receives the 21V EPROM programming supply voltage on the 8744.

Table 1. RUPI™-44 Family Pin Description (Continued)

XTAL 1	XTAL 2
Input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to VSS when external source is used on XTAL 2.	Output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.

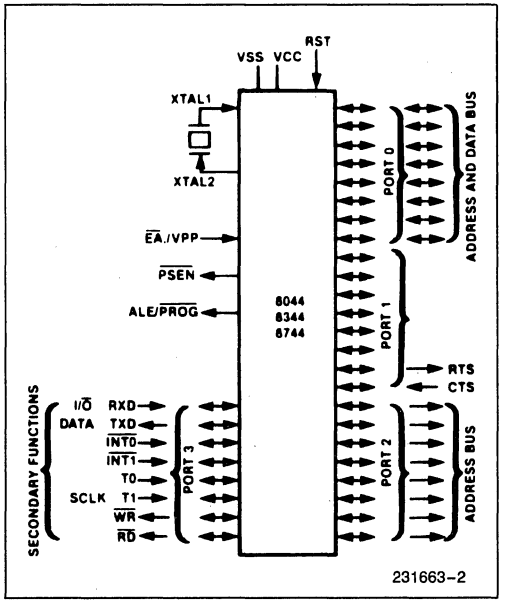


Figure 2. Logic Symbol

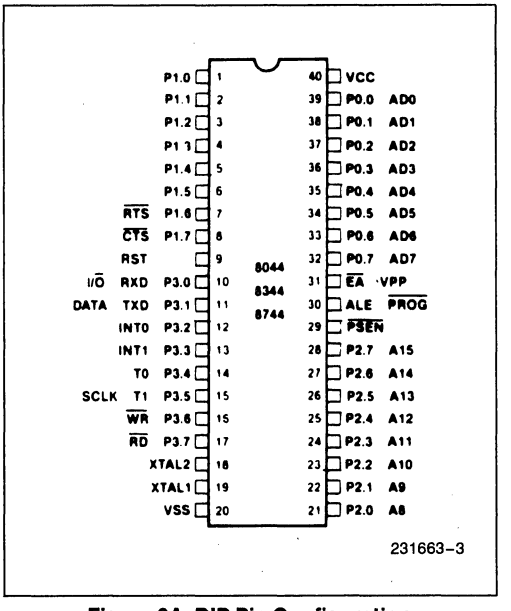


Figure 3A. DIP Pin Configuration

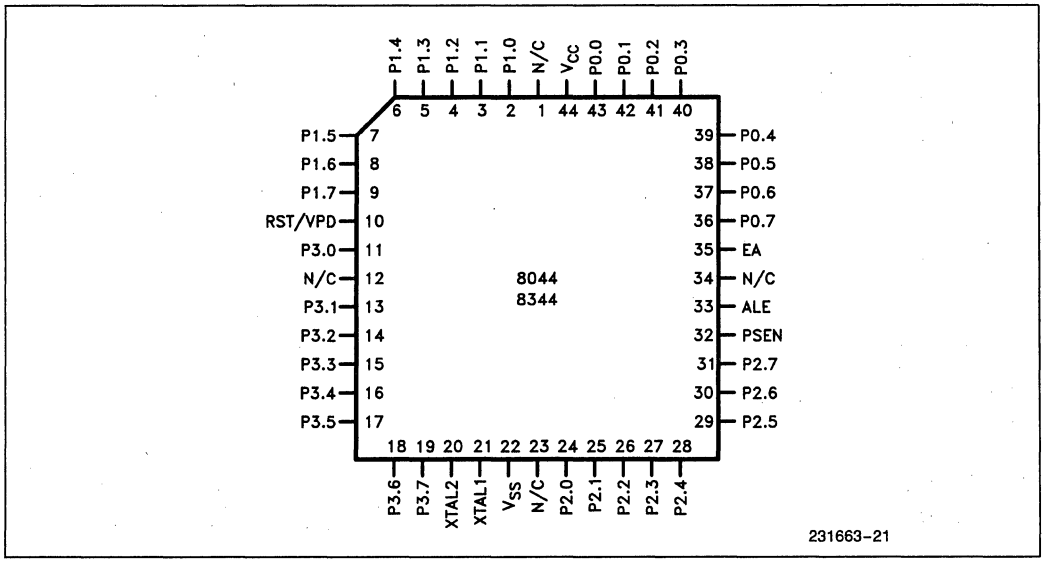


Figure 3B. PLCC Pin Configuration

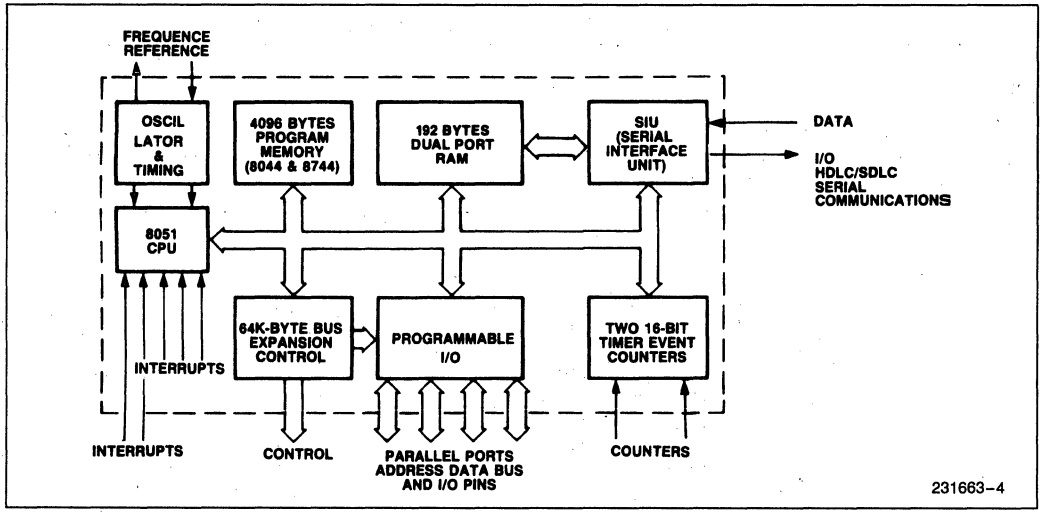


Figure 4. Block Diagram

FUNCTIONAL DESCRIPTION

General

The 8044 integrates the powerful 8051 microcontroller with an intelligent Serial Communication Controller to provide a single-chip solution which will efficiently implement a distributed processing or distributed control system. The microcontroller is a self-sufficient unit containing ROM, RAM, ALU, and its own peripherals. The 8044's architecture and instruction set are identical to the 8051's. The 8044 replaces the 8051's serial interface with an intelligent SDLC/HDLC Serial Interface Unit (SIU). 64 more bytes of RAM have been added to the 8051 RAM array. The SIU can communicate at bit rates up to 2.4 M bps. The SIU works concurrently with the Microcontroller so that there is no throughput loss in either unit. Since the SIU possesses its own intelligence, the CPU is off-loaded from many of the communications tasks, thus dedicating more of its computing power to controlling local peripherals or some external process.

The Microcontroller

The microcontroller is a stand-alone high-performance single-chip computer intended for use in sophisticated real-time application such as instrumentation, industrial control, and intelligent computer peripherals.

The major features of the microcontroller are:

- 8-bit CPU
- on-chip oscillator
- 4K bytes of ROM
- 192 bytes of RAM
- 32 I/O lines
- 64K address space for external Data Memory
- 64K address space for external Program Memory
- two fully programmable 16-bit timer/counters
- a five-source interrupt structure with two priority levels
- bit addressability for Boolean processing

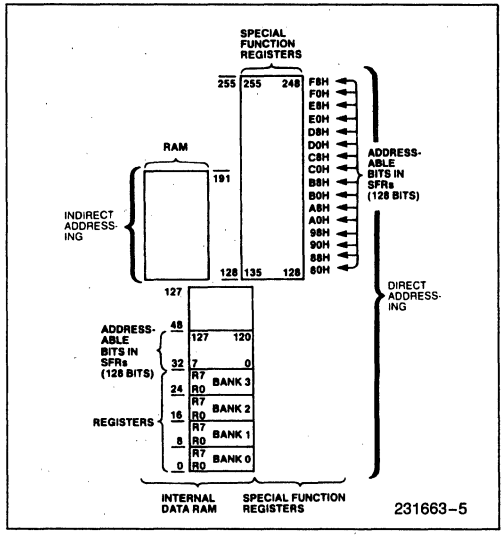


Figure 5. Internal Data Memory Address Space

- 1 μs instruction cycle time for 60% of the instructions 2 μs instruction cycle time for 40% of the instructions
- 4 μs cycle time for 8 by 8 bit unsigned Multiply/Divide

INTERNAL DATA MEMORY

Functionally the Internal Data Memory is the most flexible of the address spaces. The Internal Data Memory space is subdivided into a 256-byte Internal Data RAM address space and a 128-bit Special Function Register address space as shown in Figure 5.

The Internal Data RAM address space is 0 to 255. Four 8-Register Banks occupy locations 0 through 31. The stack can be located anywhere in the Internal Data RAM address space. In addition, 128 bit locations of the on-chip RAM are accessible through Direct Addressing. These bits reside in Internal Data RAM at byte locations 32 through 47. Currently locations 0 through 191 of the Internal Data RAM address space are filled with on-chip RAM.

Parallel I/O

The 8044 has 32 general-purpose I/O lines which are arranged into four groups of eight lines. Each group is called a port. Hence there are four ports; Port 0, Port 1, Port 2, and Port 3. Up to five lines from Port 3 are dedicated to supporting the serial channel when the SIU is invoked. Due to the nature of the serial port, two of Port 3's I/O lines (P3.0 and P3.1) do not have latched outputs. This is true whether or not the serial channel is used.

Port 0 and Port 2 also have an alternate dedicated function. When placed in the external access mode, Port 0 and Port 2 become the means by which the 8044 communicates with external program memory. Port 0 and Port 2 are also the means by which the 8044 communicates with external data memory. Peripherals can be memory mapped into the address space and controlled by the 8044.

Table 2. MCS®-51 Instruction Set Description

Mnemonic		Description	Byte	Cyc
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
INC	DPTR	Increment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

Table 2. MCS®-51 Instruction Set Description (Continued)

Mnemonic		Description	Byte	Cyc
LOGICAL OPERATIONS				
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@RI	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@RI	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1

Mnemonic		Description	Byte	Cyc
LOGICAL OPERATIONS (Continued)				
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@RI	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2

Table 2. MCS®-51 Instruction Set Description (Continued)

Mnemonic		Description	Byte	Cyc
DATA TRANSFER (Continued)				
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit) addr	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order Digit ind RAM w A	1	1
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry Flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry Flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2

Mnemonic		Description	Byte	Cyc
BOOLEAN VARIABLE MANIPULATION (Continued)				
ANL	C,/bit	AND complement of direct bit to Carry	2	2
ORL	C/bit	OR direct bit to Carry flag	2	2
ORL	C,/bit	OR complement of direct bit to Carry	2	2
MOV	C,/bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2
PROGRAM AND MACHINE CONTROL				
ACALL	addr11	Absolute Subroutine Call	2	2
LCALL	addr16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute Jump	2	2
LJMP	addr16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp, immed, to A & Jump if Not Equal	3	2

Table 2. MCS®-51 Instruction Set Description (Continued)

Mnemonic	Description	Byte	Cyc
PROGRAM AND MACHINE CONTROL (Continued)			
CJNE Rn,#data,rel	Comp, immed, to reg & Jump if Not Equal	3	2
CJNE @Ri,#data, rel	Comp, immed, to ind. & Jump if Not Equal	3	2
DJNZ Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ direct,rel	Decrement direct & Jump if Not Zero	3	2
NOP	No operation	1	1

Notes on data addressing modes:

Rn — Working register R0-R7

direct — 128 internal RAM locations, any I/O port, control or status register

@Ri — Indirect internal RAM location addressed by register R0 or R1

Notes on data addressing modes:

(Continued)

#data — 8-bit constant included in instruction

#data16 — 16-bit constant included as bytes 2 & 3 of instruction

bit — 128 software flags, any I/O pin, controll or status bit

Notes on program addressing modes:

addr16 — Destination address for LCALL & LJMP may be anywhere within the 64-K program memory address space

Addr11 — Destination address for ACALL & AJMP will be within the same 2-K page of program memory as the first byte of the following instruction

rel — SJMP and all conditional jumps include an 8-bit offset byte, Range is +127 −128 bytes relative to first byte of the following instruction

Timer/Counters

The 8044 contains two 16-bit counters which can be used for measuring time intervals, measuring pulse widths, counting events, generating precise periodic interrupt requests, and clocking the serial communications. Internally the Timers are clocked at 1/12 of the crystal frequency, which is the instruction cycle time. Externally the counters can run up to 500 KHz.

Interrupt System

External events and the real-time driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a sophisticated multiple-source, two priority level, nested interrupt system is provided. Interrupt response latency ranges from 3 μsec to 7 μsec when using a 12 MHz clock.

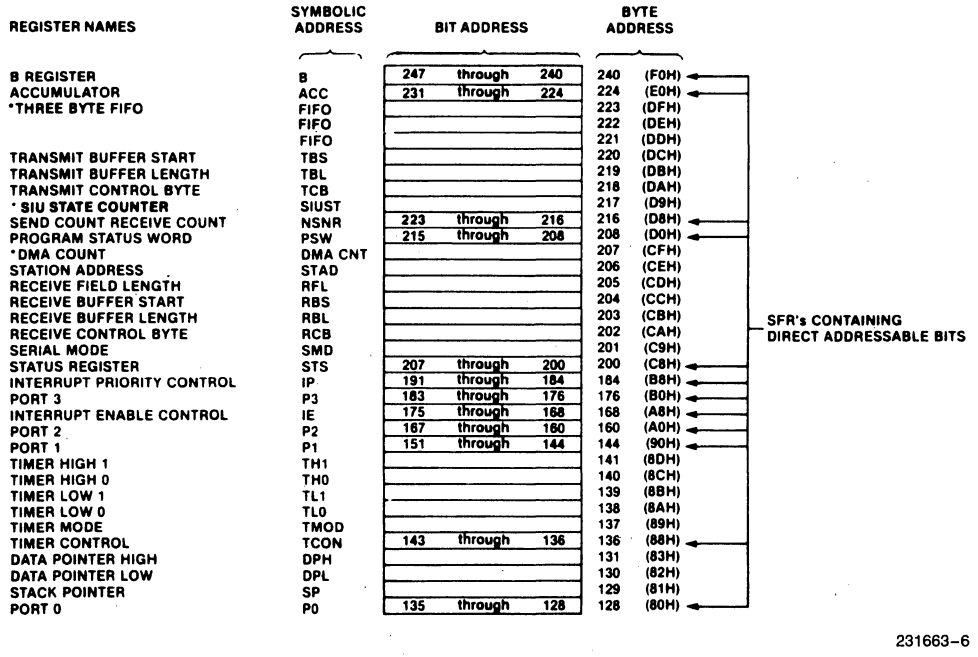
All five interrupt sources can be mapped into one of the two priority levels. Each interrupt source can be enabled or disabled individually or the entire interrupt system can be enabled or disabled. The five interrupt sources are: Serial Interface Unit, Timer 1, Timer 2, and two external interrupts. The external interrupts can be either level or edge triggered.

Serial Interface Unit (SIU)

The Serial Interface Unit is used for HDLC/SDLC communications. It handles Zero Bit Insertion/Deletion, Flags automatic access recognization, and a 16-bit cyclic redundancy check. In addition it implements in hardware a subset of the SDLC protocol certain applications it is advantageous to have the CPU control the reception or transmission of every single frame. For this reason the SIU has two modes of operation: "AUTO" and "FLEXIBLE" (or "NON-AUTO"). It is in the AUTO mode that the SIU responds to SDLC frames without CPU intervention; whereas, in the FLEXIBLE mode the reception or transmission of every single frame will be under CPU control.

There are three control registers and eight parameter registers that are used to operate the serial interface. These registers are shown in Figure 5 and Figure 6. The control register set the modes of operation and provide status information. The eight parameter registers buffer the station address, receive and transmit control bytes, and point to the on-chip transmit and receive buffers.

Data to be received or transmitted by the SIU must be buffered anywhere within the 192 bytes of on-chip RAM. Transmit and receive buffers are not allowed to "wrap around" in RAM; a "buffer end" is generated after address 191 is reached.



NOTE:
*ICE Support Hardware registers. Under normal operating conditions there is no need for the CPU to access these registers.

Figure 5. Mapping of Special Function Registers

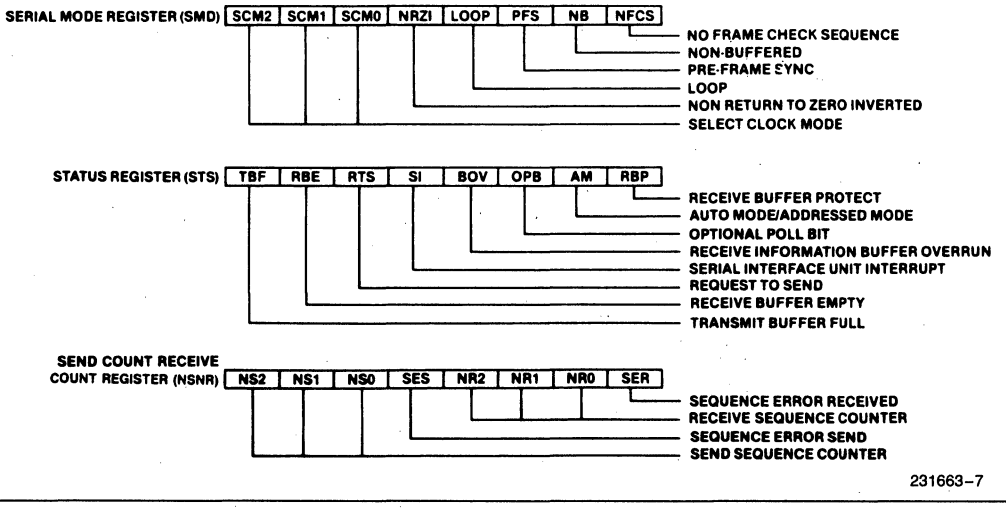


Figure 6. Serial Interface Unit Control Registers

With the addition of only a few bytes of code, the 8044's frame size is not limited to the size of its internal RAM (192 bytes), but rather by the size of external buffer with no degradation of the RUPI's features (e.g. NRZI, zero bit insertion/deletion, address recognition, cyclic redundancy check). There is a special function register called SIUST whose contents dictates the operation of the SIU. At low data rates, one section of the SIU (the Byte Processor) performs no function during known intervals. For a given data rate, these intervals (stand-by mode) are fixed. The above characteristics make it possible to program the CPU to move data to/from external RAM and to force the SIU to perform some desired hardware tasks while transmission or reception is taking place. With these modifications, external RAM can be utilized as a transmit and received buffer instead of the internal RAM.

AUTO Mode

In the AUTO mode the SIU implements in hardware a subset of the SDLC protocol such that it responds to many SDLC frames without CPU intervention. All AUTO mode responses to the primary station will comform to IBM's SDLC definition. The advantages of the AUTO mode are that less software is required to implement a secondary station, and the hardware generated response to polls is much faster than doing it in software. However, the Auto mode can not be used at a primary station.

To transmit in the AUTO mode the CPU must load the Transmit Information Buffer, Transmit Buffer Start register, Transmit Buffer Length register, and set the Transmit Buffer Full bit. The SIU automatically responds to a poll by transmitting an information frame with the P/F bit in the control field set. When the SIU receives a positive acknowledgement from the primary station, it automatically increments the Ns field in the NSNR register and interrupts the CPU. A negative acknowledgement would cause the SIU to retransmit the frame.

To receive in the AUTO mode, the CPU loads the Receive Buffer Start register, the Receive Buffer Length register, clears the Receive Buffer Protect bit, and sets the Receive Buffer Empty bit. If the SIU is polled in this state, and the TBF bit indicates that the Transmit Buffer is empty, an automatic RR response will be generated. When a valid information frame is received the SIU will automatically increment Nr in the NSNR register and interrupt the CPU.

While in the AUTO mode the SIU can recognize and respond to the following commands without CPU intervention: I (Information), RR (Receive Ready), RNR (Receive Not Ready), REJ (Reject), and UP (Unnumbered Poll). The SIU can generate the following responses without CPU intervention: I (Information), RR (Receive Ready), and RNR (Receive Not Ready).

When the Receive Buffer Empty bit (RBE) indicates that the Receive Buffer is empty, the receiver is enabled, and when the RBE bit indicates that the Receive Buffer is full, the receiver is disabled. Assuming that the Receiver Buffer is empty, the SIU will respond to a poll with an I frame if the Transmit Buffer is full. If the Transmit Buffer is empty, the SIU will respond to a poll with a RR command if the Receive Buffer Protect bit (RBP) is cleared, or an RNR command if RBP is set.

FLEXIBLE (or NON-AUTO) Mode

In the FLEXIBLE mode all communications are under control of the CPU. It is the CPU's task to encode and decode control fields, manage acknowledgements, and adhere to the requirements of the HDLC/SDLC protocols. The 8044 can be used as a primary or a secondary station in this mode.

To receive a frame in the FLEXIBLE mode, the CPU must load the Receive Buffer Start register, the Receive Buffer Length register, clear the Receive Buffer Protect bit, and set the Receive Buffer Empty bit. If a valid opening flag is received and the address field matches the byte in the Station Address register or the address field contains a broadcast address, the 8044 loads the control field in the receive control byte register, and loads the I field in the receive buffer. If there is no CRC error, the SIU interrupts the CPU, indicating a frame has just been received. If there is a CRC error, no interrupt occurs. The Receive Field Length register provides the number of bytes that were received in the information field.

To transmit a frame, the CPU must load the transmit information buffer, the Transmit Buffer Start register, the Transmit Buffer Length register, the Transmit Control Byte, and set the TBF and the RTS bit. The SIU, unsolicited by an HDLC/SDLC frame, will transmit the entire information frame, and interrupt the CPU, indicating the completion of transmission. For supervisory frames or unnumbered frames, the transmit buffer length would be 0.

CRC

The FCS register is initially set to all 1's prior to calculating the FCS field. The SIU will not interrupt the CPU if a CRC error occurs (in both AUTO and FLEXIBLE modes). The CRC error is cleared upon receiving of an opening flag.

Frame Format Options

In addition to the standard SDLC frame format, the 8044 will support the frames displayed in Figure 7. The standard SDLC frame is shown at the top of this figure. For the remaining frames the information field will incorporate the control or address bytes and the frame check sequences; therefore these fields will be stored in the Transmit and Receive buffers. For example, in the non-buffered mode the third byte is treated as the beginning of the information field. In the non-addressed mode, the information field begins after the opening flag. The mode bits to set the frame format options are found in the Serial Mode register and the Status register.

FRAME OPTION	NFCS	NB	AM[1]	FRAME FORMAT
Standard SDLC NON-AUTO Mode	0	0	0	F A C I FCS F
Standard SDLC AUTO Mode	0	0	1	F A C I FCS F
Non-Buffered Mode NON-AUTO Mode	0	1	1	F A I FCS F
Non-Addressed Mode NON-AUTO Mode	0	1	0	F I FCS F
No FCS Field NON-AUTO Mode	1	0	0	F A C I F
No FCS Field AUTO Mode	1	0	1	F A C I F
No FCS Field Non-Buffered Mode NON-AUTO Mode	1	1	1	F A I F
No FCS Field Non-Addressed Mode NON-AUTO Mode	1	1	0	F I F

Mode Bits:
AM — "AUTO" Mode/Addressed Mode
NB — Non-Buffered Mode
NFCS — No FCS Field Mode

Key to Abbreviations:
F = Flag (01111110)
A = Address Field
C = Control Field
I = Information Field
FCS= Frame Check Sequence

Note 1:
The AM bit function is controlled by the NB bit. When NB = 0, AM becomes AUTO mode select, when NB = 1, AM becomes Address mode select.

Figure 7. Frame Format Options

Extended Addressing

To realize an extended control field or an extended address field using the HDLC protocol, the FLEXIBLE mode must be used. For an extended control field, the SIU is programmed to be in the non-buffered mode. The extended control field will be the first and second bytes in the Receive and Transmit Buffers. For extended addressing the SIU is placed in the non-addressed mode. In this mode the CPU must implement the address recognition for received frames. The addressing field will be the initial bytes in the Transmit and Receive buffers followed by the control field.

The SIU can transmit and receive only frames which are multiples of 8 bits. For frames received with other than 8-bit multiples, a CRC error will cause the SIU to reject the frame.

SDLC Loop Networks

The SIU can be used in an SDLC loop as a secondary or primary station. When the SIU is placed in the Loop mode it receives the data on pin 10 and transmits the data one bit time delayed on pin 11. It can also recognize the Go ahead signal and change it into a flag when it is ready to transmit. As a secondary station the SIU can be used in the AUTO or FLEXIBLE modes. As a primary station the FLEXIBLE mode is used; however, additional hardware is required for generating the Go Ahead bit pattern. In the Loop mode the maximum data rate is 1 Mbps clocked or 375 Kpbs self-clocked.

SDLC Multidrop Networks

The SIU can be used in a SDLC non-loop configuration as a secondary or primary station. When the SIU is placed in the non-loop mode, data is received and transmitted on pin 11, and pin 10 drives a tri-state buffer. In non-loop mode, modem interface pins, RTS and CTS, become available.

Data Clocking Options

The 8044's serial port can operate in an externally clocked or self clocked system. A clocked system provides to the 8044 a clock synchronization to the data. A self-clocked system uses the 8044's on-chip Digital Phase Locked Loop (DPLL) to recover the clock from the data, and clock this data into the Serial Receive Shift Register.

In this mode, a clock synchronized with the data is externally fed into the 8044. This clock may be generated from an External Phase Locked Loop, or possibly supplied along with the data. The 8044 can transmit and receive data in this mode at rates up to 2.4 Mbps.

This self clocked mode allows data transfer without a common system data clock. An on-chip Digital Phase Locked Loop is employed to recover the data clock which is encoded in the data stream. The DPLL will converge to the nominal bit center within eight bit transitions, worst case. The DPLL requires a reference clock of either 16 times (16x) or 32 times (32x) the data rate. This reference clock may be externally applied or internally generated. When internally generated either the 8044's internal logic clock (crystal frequency divided by two) or the timer 1 overflow is used as the reference clock. Using the internal timer 1 clock the data rates can vary from 244 to 62.5 Kbps. Using the internal logic clock at a 16x sampling rate, receive data can either be 187.5 Kbps, or 375 Kbps. When the reference clock for the DPLL is externally applied the data rates can vary from 0 to 375 Kbps at a 16x sampling rate.

To aid in a Phase Locked Loop capture, the SIU has a NRZI (Non Return to Zero Inverted) data encoding and decoding option. Additionally the SIU has a preframe sync option that transmits two bytes of alternating 1's and 0's to ensure that the receive station DPLL will be synchronized with the data by the time it receives the opening flag.

Control and Status Registers

There are three SIU Control and Status Registers:

Serial Mode Register (SMD)

Status/Command Register (STS)

Send/Receive Count Register (NSNR)

The SMD, STS, and NSNR, registers are all cleared by system reset. This assures that the SIU will power up in an idle state (neither receiving nor transmitting).

These registers and their bit assignments are described below.

SMD: Serial Mode Register (byte-addressable)

Bit 7:	6	5	4	3	2	1	0
SCM2	SCM1	SCM0	NRZI	LOOP	PFS	NB	NFCS

The Serial Mode Register (Address C9H) selects the operational modes of the SIU. The 8044 CPU can both read and write SMD. The SIU can read SMD but cannot write to it. To prevent conflict between CPU and SIU access to SMD, the CPU should write SMD only when the Request To Send (RTS) and

Receive Buffer Empty (RBE) bits (in the STS register) are both false (0). Normally, SMD is accessed only during initialization.

The individual bits of the Serial Mode Register are as follows:

Bit#	Name	Description
SMD.0	NFCS	No FCS field in the SDLC frame.
SMD.1	NB	Non-Buffered mode. No control field in the SDLC frame.
SMD.2	PFS	Pre-Frame Sync mode. In this mode, the 8044 transmits two bytes before the first flag of a frame, for DPLL synchronization. If NRZI is enabled, 00H is sent; otherwise, 55H is sent. In either case, 16 preframe transitions are guaranteed.
SMD.3	LOOP	Loop configuration.
SMD.4	NRZI	NRZI coding option. If bit = 1, NRZI coding is used. If bit = 0, then it is straight binary (NRZ).
SMD.5	SCM0	Select Clock Mode—Bit 0
SMD.6	SCM1	Select Clock Mode—Bit 1
SMD.7	SCM2	Select Clock Mode—Bit 2

The SCM bits decode as follows:

SCM				
2	1	0	Clock Mode	Data Rate (Bits/sec)*
0	0	0	Externally clocked	0-2.4M**
0	0	1	Reserved	
0	1	0	Self clocked, timer overflow	244-62.5K
0	1	1	Reserved	
1	0	0	Self clocked, external 16x	0-375K
1	0	1	Self clocked, external 32x	0-187.5K
1	1	0	Self clocked, internal fixed	375K
1	1	1	Self clocked, internal fixed	187.5K

NOTES:
*Based on a 12 Mhz crystal frequency
**0-1 M bps in loop configuration

STS: Status/Command Register (bit-addressable)

Bit:	7	6	5	4	3	2	1	0
	TBF	RBE	RTS	SI	BOV	OPB	AM	RBP

The Status/Command Register (Address C8H) provides operational control of the SIU by the 8044 CPU, and enables the SIU to post status information for the CPU's access. The SIU can read STS, and can alter certain bits, as indicated below. The CPU can both read and write STS asynchronously. However, 2-cycle instructions that access STS during both cycles ('JBC/B, REL' and 'MOV/B, C.') should not be used, since the SIU may write to STS between the two CPU accesses.

The individual bits of the Status/Command Register are as follows:

Bit#	Name	Description
STS.0	RBP	Receive Buffer Protect. Inhibits writing of data into the receive buffer. In AUTO mode, RBP forces an RNR response instead of an RR.
STS.1	AM	AUTO Mode/Addressed Mode. Selects AUTO mode where AUTO mode is allowed. If NB is true, (= 1), the AM bit selects the addressed mode. AM may be cleared by the SIU.
STS.2	OPB	Optional Poll Bit. Determines whether the SIU will generate an AUTO response to an optional poll (UP with P = 0). OPM may be set or cleared by the SIU.
STS.3	BOV	Receive Buffer Overrun. BOV may be set or cleared by the SIU.
STS.4	SI	SIU Interrupt. This is one of the five interrupt sources to the CPU. The vector location = 23H. SI may be set by the SIU. It should be cleared by the CPU before returning from an interrupt routine.
STS.5	RTS	Request To Send. Indicates that the 8044 is ready to transmit or is transmitting. RTS may be read or written by the CPU. RTS may be read by the SIU, and in AUTO mode may be written by the SIU.
STS.6	RBE	Receive Buffer Empty. RBE can be thought of as Receive Enable. RBE is set to one by the CPU when it is ready to receive a frame, or has just read the buffer, and to zero by the SIU when a frame has been received.
STS.7	TBF	Transmit Buffer Full. Written by the CPU to indicate that it has filled the transmit buffer. TBF may be cleared by the SIU.

NSNR: Send/Receive Count Register (bit-addressable)

Bit:	7	6	5	4	3	2	1	0
	NS2	NS1	NS0	SES	NR2	NR1	NR0	SER

The Send/Receive Count Register (Address D8H) contains the transmit and receive sequence numbers, plus tally error indications. The SIU can both read and write NSNR. The 8044 CPU can both read and write NSNR asynchronously. However, 2-cycle instructions that access NSNR during both cycles ('JBC /B, REL,' and 'MOV /B,C') should not be used, since the SIU may write to NSMR between the two 8044 CPU accesses.

The individual bits of the Send/Receive Count Register are as follows:

Bit #	Name	Description
NSNR.0	SER	Receive Sequence Error: NS (P) ≠ NR (S)
NSNR.1	NR0	Receive Sequence Counter—Bit 0
NSNR.2	NR1	Receive Sequence Counter—Bit 1
NSNR.3	NR2	Receive Sequence Counter—Bit 2
NSNR.4	SES	Send Sequence Error: NR (P) ≠ NS (S) and NR (P) ≠ NS (S) + 1
NSNR.5	NS0	Send Sequence Counter—Bit 0
NSNR.6	NS1	Send Sequence Counter—Bit 1
NSNR.7	NS2	Send Sequence Counter—Bit 2

Parameter Registers

There are eight parameter registers that are used in connection with SIU operation. All eight registers may be read or written by the 8044 CPU. RFL and RCB are normally loaded by the SIU.

The eight parameter registers are as follows:

STAD: Station Address Register (byte-addressable)

The Station Address register (Address CEH) contains the station address. To prevent acess conflict, the CPU should access STAD only when the SIU is idle (RTS = 0 and RBE = 0). Normally, STAD is accessed only during initialization.

TBS: Transmit Buffer Start Address Register (byte-addressable)

The Transmit Buffer Start address register (Address DCH) points to the location in on-chip RAM for the beginning of the I-field of the frame to be transmitted. The CPU should access TBS only when the SIU is not transmitting a frame (when TBF = 0).

TBL: Transmit Buffer Length Register (byte = addressable)

The Transmit Buffer Length register (Address DBH) contains the length (in bytes) of the I-field to be transmitted. A blank I-field (TBL = 0) is valid. The CPU should access TBL only when the SIU is not transmitting a frame (when TBF = 0).

NOTE:
The transmit and receive buffers are not allowed to "wrap around" in the on-chip RAM. A "buffer end" is automatically generated if address 191 (BFH) is reached.

TCB: Transmit Control Byte Register (byte-addressable)

The Transmit Control Byte register (Address DAH) contains the byte which is to be placed in the control field of the transmitted frame, during NON-AUTO mode transmission. The CPU should access TCB only when the SIU is not transmitting a frame (when TBF = 0). The N_S and N_R counters are not used in the NON-AUTO mode.

RBS: Receive Buffer Start Address Register (byte-addressable)

The Receive Buffer Start address register (Address CCH) points to the location in on-chip RAM where the beginning of the I-field of the frame being received is to be stored. The CPU should write RBS only when the SIU is not receiving a frame (when RBE = 0).

RBL: Receive Buffer Length Register (byte-addressable)

The Receive Buffer Length register (Address CBH) contains the length (in bytes) of the area in on-chip RAM allocated for the received I-field. RBL=0 is valid. The CPU should write RBL only when RBE=0.

RFL: Receive Field Length Register (byte-addressable)

The Receive Field Length register (Address CDH) contains the length (in bytes) of the received I-field that has just been loaded into on-chip RAM. RFL is loaded by the SIU. RFL = 0 is valid. RFL should be accessed by the CPU only when RBE = 0.

RCB: Receive Control Byte Register (byte-addressable)

The Received Control Byte register (Address CAH) contains the control field of the frame that has just been received. RCB is loaded by the SIU. The CPU can only read RCB, and should only access RCB when RBE = 0.

ICE Support

The 8044 In-Circuit Emulator (ICE-44) allows the user to exercise the 8044 application system and monitor the execution of instructions in real time.

The emulator operates with Intel's Intellec™ development system. The development system interfaces with the user's 8044 system through an in-cable buffer box. The cable terminates in a 8044 pin-compatible plug, which fits into the 8044 socket in the user's system. With the emulator plug in place, the user can excercise his system in real time while collecting up to 255 instruction cycles of real-time data. In addition, he can single-step the program.

Static RAM is available (in the in-cable buffer box) to emulate the 8044 internal and external program memory and external data memory. The designer can display and alter the contents of the replacement memory in the buffer box, the internal data memory, and the internal 8044 registers, including the SFR's.

SIUST: SIU State Counter (byte-addressable)

The SIU State Counter (Address D9H) reflects the state of the internal logic which is under SIU control. Therefore, care must be taken not to write into this register. This register provides a useful means for debugging 8044 receiver problem.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature.....−65°C to −150°C
Voltage on $\overline{EA}$, VPP Pin to VSS...−0.5V to −21.5V
Voltage on Any Other Pin to VSS....−0.5V to −7V
Power Dissipation 2W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, VCC = 5V ± 10%, VSS = 0V

Symbol	Parameter		Min	Max	Unit	Test Conditions
VIL	Input Low Voltage (Except $\overline{EA}$ Pin of 8744H)		−0.5	0.8	V	
VIL1	Input Low Voltage to $\overline{EA}$ Pin of 8744H		0	0.8	V	
VIH	Input High Voltage (Except XTAL2, RST)		2.0	VCC + 0.5	V	
VIH1	Input High Voltage to XTAL2, RST		2.5	VCC + 0.5	V	XTAL1 = VSS
VOL	Output Low Voltage (Ports 1, 2, 3)*			0.45	V	IOL = 1.6mA
VOL1	Output Low Voltage (Port 0,ALE,$\overline{PSEN}$)*					
		8744H		0.60 0.45	V V	IOL = 3.2 mA IOL = 2.4 mA
		8044AH/8344AH		0.45	V	IOL = 3.2 mA
VOH	Output High Voltage (Ports 1, 2, 3)		2.4		V	IOH = −80 μA
VOH1	Output High Voltage (Port 0 in External Bus Mode, ALE, $\overline{PSEN}$)		2.4		V	IOH = −400 μA
IIL	Logical 0 Input Current (Ports 1, 2, 3)			−500	μA	Vin = 0.45V
IIL1	Logical 0 Input Current to $\overline{EA}$ Pin of 8744H only			−15	mA	
IIL2	Logical 0 Input Current (XTAL2)			−3.6	mA	Vin = 0.45V
ILI	Input Leakage Current (Port 0) 8744H 8044AH/8344AH			 ±100 ±10	 μA μA	 0.45 < Vin < VCC 0.45 < Vin < VCC
IIH	Logical 1 Input Current to $\overline{EA}$ Pin of 8744H			500	μA	
IIH1	Input Current to RST to Activate Reset			500	μA	Vin < (VCC − 1.5V)
ICC	Power Supply Current: 8744H 8044AH/8344AH			 285 170	 mA mA	All Outputs Disconnected: $\overline{EA}$ = VCC
CIO	Pin Capacitance			10	pF	Test Freq. = 1MHz(1)

***NOTES:**
1. Sampled not 100% tested. T_A = 25°C.
2. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLs of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pin when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.

A.C. CHARACTERISTICS

T_A = 0°C to +70°C, VCC = 5V ± 10%, VSS = 0V, Load Capacitance for Port 0, ALE, and PSEN = 100 pF, Load Capacitance for All Other Outputs = 80 pF

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Osc		Variable Clock 1/TCLCL = 3.5 MHz to 12 MHz		Unit
		Min	Max	Min	Max	
TLHLL	ALE Pulse Width	127		2TCLCL-40		ns
TAVLL	Address Valid to ALE Low	43		TCLCL-40		ns
TLLAX[1]	Address Hold After ALE Low	48		TCLCL-35		ns
TLLIV	ALE Low to Valid Instr in 8744H 8044AH/8344AH		 183 233		 4TCLCL-150 4TCLCL-100	ns
TLLPL	ALE Low to $\overline{PSEN}$ Low	58		TCLCL-25		ns
TPLPH	$\overline{PSEN}$ Pulse Width 8744H 8044AH/8344AH	 190 215		 3TCLCL-60 3TCLCL-35		 ns ns
TPLIV	$\overline{PSEN}$ Low to Valid Instr in 8744H 8044AH/8344AH		 100 125		 3TCLCL-150 3TCLCL-125	 ns ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		0		ns
TPXIZ[2]	Input Instr Float After $\overline{PSEN}$		63		TCLCL-20	ns
TPXAV[2]	$\overline{PSEN}$ to Address Valid	75		TCLCL-8		ns
TAVIV	Address to Valid Instr in 8744H 8044AH/8344AH		 267 302		 5TCLCL-150 5TCLCL-115	 ns ns
TAZPL	Address Float to $\overline{PSEN}$	−25		−25		ns

NOTES:
1. TLLAX for access to program memory is different from TLLAX for data memory.
2. Interfacing RUPI-44 devices with float times up to 75ns is permissible. This limited bus contention will not cause any damage to Port 0 drivers.

EXTERNAL DATA MEMORY CHARACTERISTICS

Symbol	Parameter	12 MHz Osc		Variable Clock 1/TCLCL = 3.5 MHz to 12 MHz		Unit
		Min	Max	Min	Max	
TRLRH	$\overline{RD}$ Pulse Width	400		6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		6TCLCL-100		ns
TLLAX	Address Hold after ALE	48		TCLCL-35		ns
TRLDV	$\overline{RD}$ Low to Valid Data in		252		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		0		ns
TRHDZ	Data Float After $\overline{RD}$		97		2TCLCL-70	ns
TLLDV	ALE Low to Valid Data In		517		8TCLCL-150	ns
TAVDV	Address to Valid Data In		585		9TCLCL-165	ns
TLLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	3TCLCL-50	3TLCLCL+50	ns
TAVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition 8744H 8044AH/8344AH	 13 23		 TCLCL-70 TCLCL-60		 ns ns
TQVWH	Data Setup Before $\overline{WR}$ High	433		7TCLCL-150		ns
TWHQX	Data Held After $\overline{WR}$	33		TCLCL-50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		25		25	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High 8744H 8044AH/8344AH	 33 43	 133 123	 TCLCL-50 TCLCL-40	 TCLCL+50 TCLCL+50	 ns ns

NOTE:
1. TLLAX for access to program memory is different from TLLAX for access data memory.

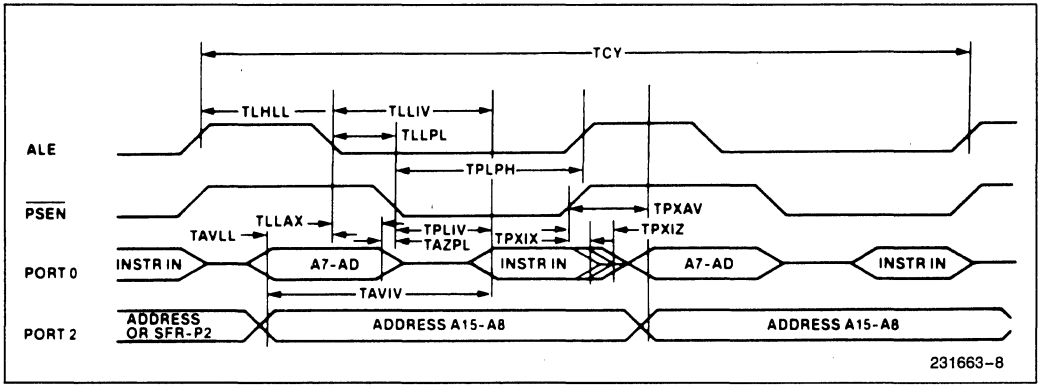
Serial Interface Characteristics

Symbol	Parameter	Min	Max	Unit
TDCY	Data Clock	420		ns
TDCL	Data Clock Low	180		ns
TDCH	Data Clock High	100		ns
tTD	Transmit Data Delay		140	ns
tDSS	Data Setup Time	40		ns
tDHS	Data Hold Time	40		ns

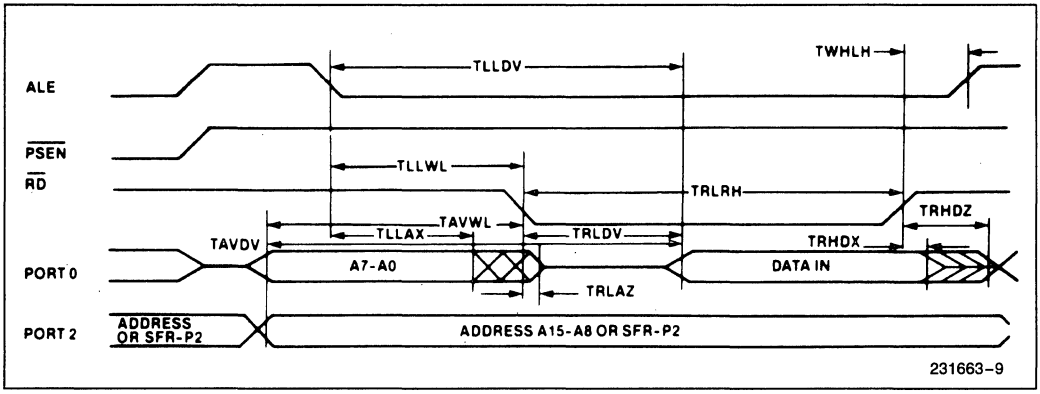
WAVEFORMS

Memory Access

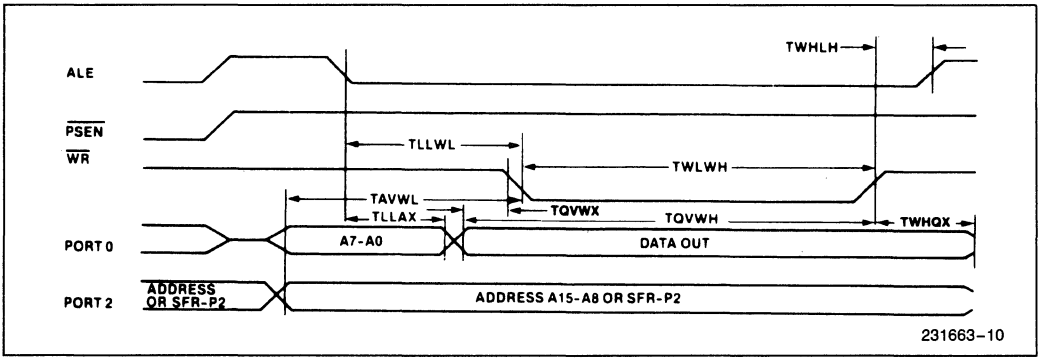
PROGRAM MEMORY READ CYCLE



DATA MEMORY READ CYCLE

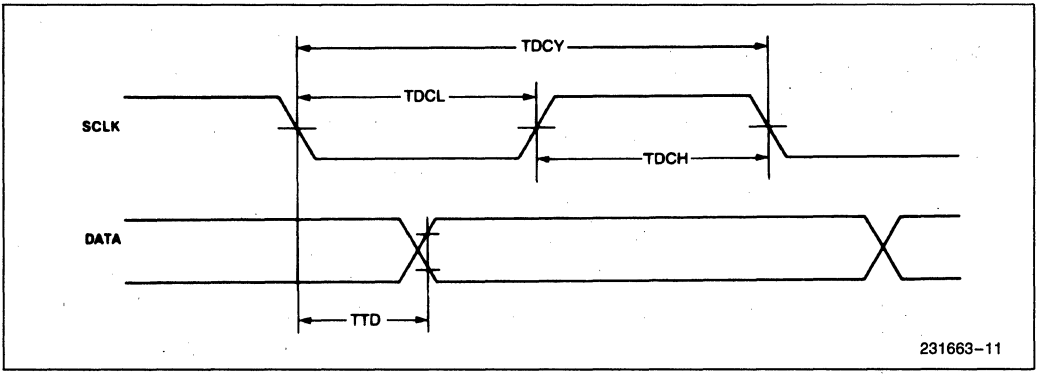


DATA MEMORY WRITE CYCLE

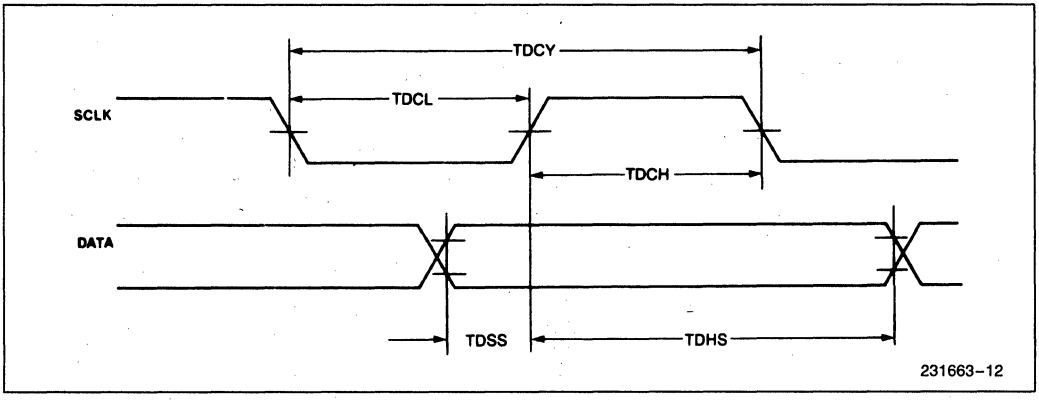


SERIAL I/O WAVEFORMS

SYNCHRONOUS DATA TRANSMISSION



SYNCHRONOUS DATA RECEPTION



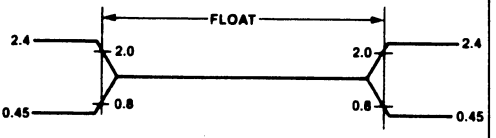
AC TESTING INPUT, OUTPUT, FLOAT WAVEFORMS

INPUT/OUTPUT

231663-13

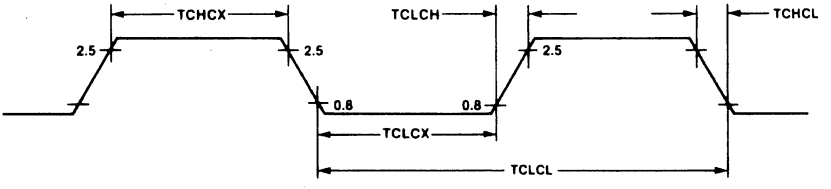
AC testing inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0" Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

FLOAT



231663-14

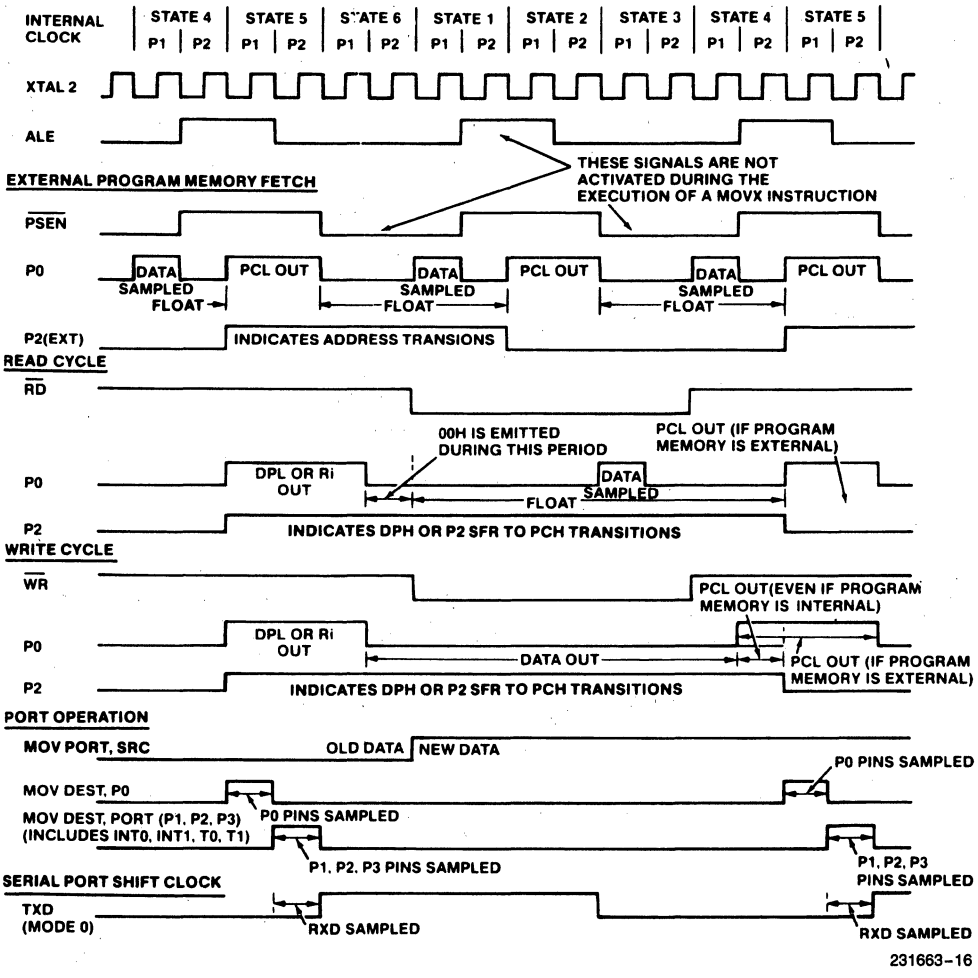
EXTERNAL CLOCK DRIVE XTAL2



231663-15

Symbol	Parameter	Variable Clock Freq = 3.5 MHz to 12 MHz		Unit
		Min	Max	
TCLCL	Oscillator Period	83.3	285.7	ns
TCHCX	High Time	20	TCLCL-TCLCX	ns
TCLCX	Low Time	20	TCLCL-TCHCX	ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns

CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component to component. Typically though, (T_A = 25°C, fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

8744H EPROM CHARACTERISTICS

Erasure Characteristics

Erasure of the 8744H Program Memory begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 Ångstroms. Since sunlight and fluorescent lighting have wavelengths in this range, constant exposure to these light sources over an extended period of time (about 1 week in sunlight, or 3 years in room-level fluorescent lighting) could cause unintentional erasure. If an application subjects the 8744H to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 Ångstroms) to an integrated dose of at least 15 W-sec/cm^2 rating for 20 to 30 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.

Programming the EPROM

To be programmed, the 8744H must be running with a 4 to 6 MHz oscillator. (The reason the oscillator needs to be running is that the internal bus is being used to transfer address and program data to appropriate registers.) The address of an EPROM location to be programmed is applied to Port 1 and pins P2.0-P2.3 of Port 2, while the data byte is applied to Port 0. Pins P2.4-P2.6 and $\overline{PSEN}$ should be held low, and P2.7 and RST high. (These are all TTL levels except RST, which requires 2.5V for high.) $\overline{EA}$/VPP is held normally high, and is pulsed to +21V. While EA/VPP is at 21V, the ALE/$\overline{PROG}$ pin, which is normally being held high, is pulsed low for 50 msec. Then EA/VPP is returned to high. This is illustrated in Figure 8. Detailed timing specifications are provided in the EPROM Programming and Verification Characteristics section of this data sheet.

Program Memory Security

The program memory security feature is developed around a "security bit" in the 8744H EPROM array. Once this "hidden bit" is programmed, electrical access to the contents of the entire program memory array becomes impossible. Activation of this feature is accomplished by programming the 8744H as described in "Programming the EPROM" with the exception that P2.6 is held at a TTL high rather than a TTL low. In addition, Port 1 and P2.0-P2.3 may be in any state. Figure 9 illustrates the security bit programming configuration. Deactivating the security feature, which again allows programmability of the EPROM, is accomplished by exposing the EPROM to ultraviolet light. This exposure, as described in "Erasure Characteristics," erases the entire EPROM array. Therefore, attempted retrieval of "protected code" results in its destruction.

Program Verification

Program Memory may be read only when the "security feature" has not been activated. Refer to Figure 10 for Program Verification setup. To read the Program Memory, the following procedure can be used. The unit must be running with a 4 to 6 MHz oscillator. The address of a Program Memory location to be read is applied to Port 1 and pins P2.0-P2.3 of Port 2. Pins P2.4-P2.6 and $\overline{PSEN}$ are held at TTL low, while the ALE/$\overline{PROG}$, RST, and $\overline{EA}$/VPP pins are held at TTL high. (These are all TTL levels except RST, which requires 2.5V for high.) Port 0 will be the data output lines. P2.7 can be used as a read strobe. While P2.7 is held high, the Port 0 pins float. When P2.7 is strobed low, the contents of the addressed location will appear at Port 0. External pullups (e.g., 10K) are required on Port 0 during program verification.

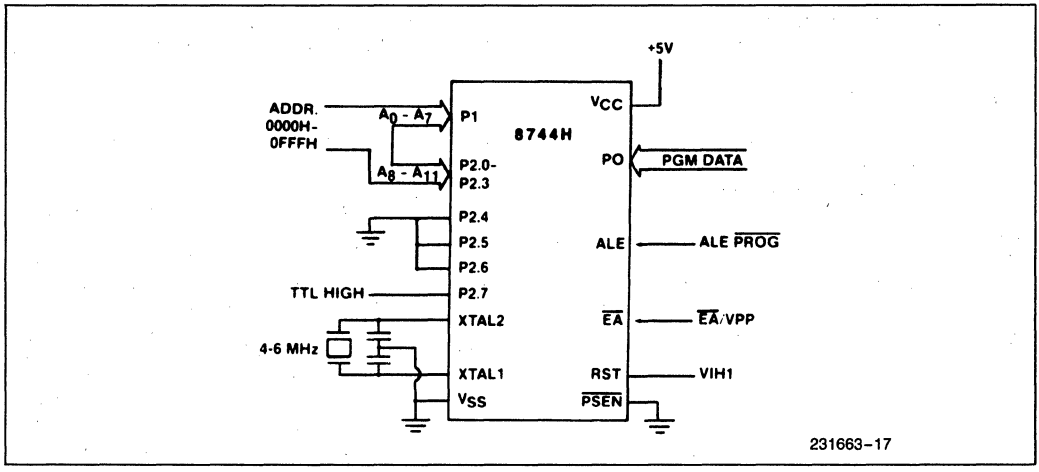


Figure 8. Programming Configuration

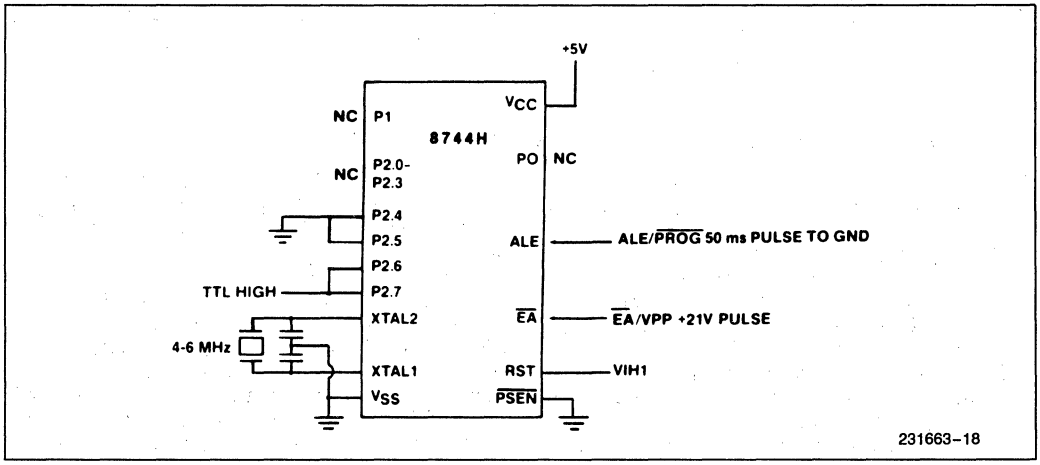


Figure 9. Security Bit Programming Configuration

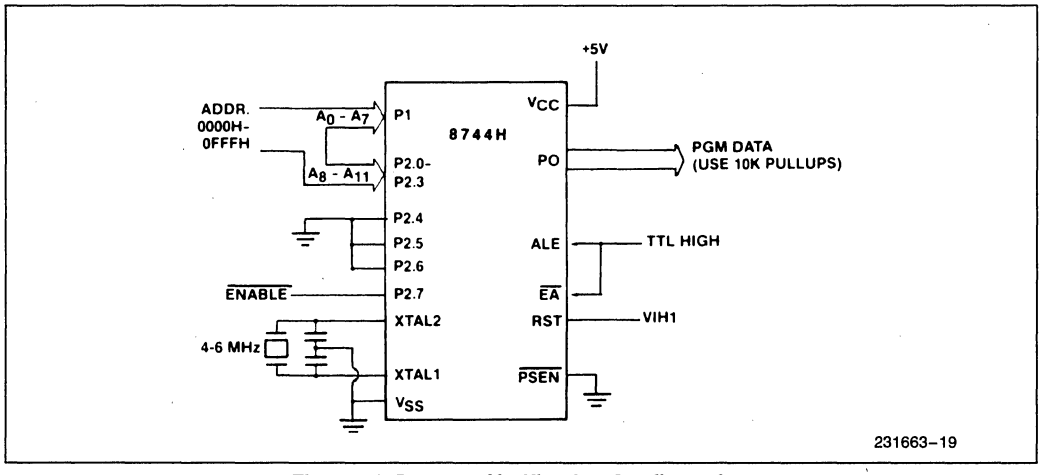


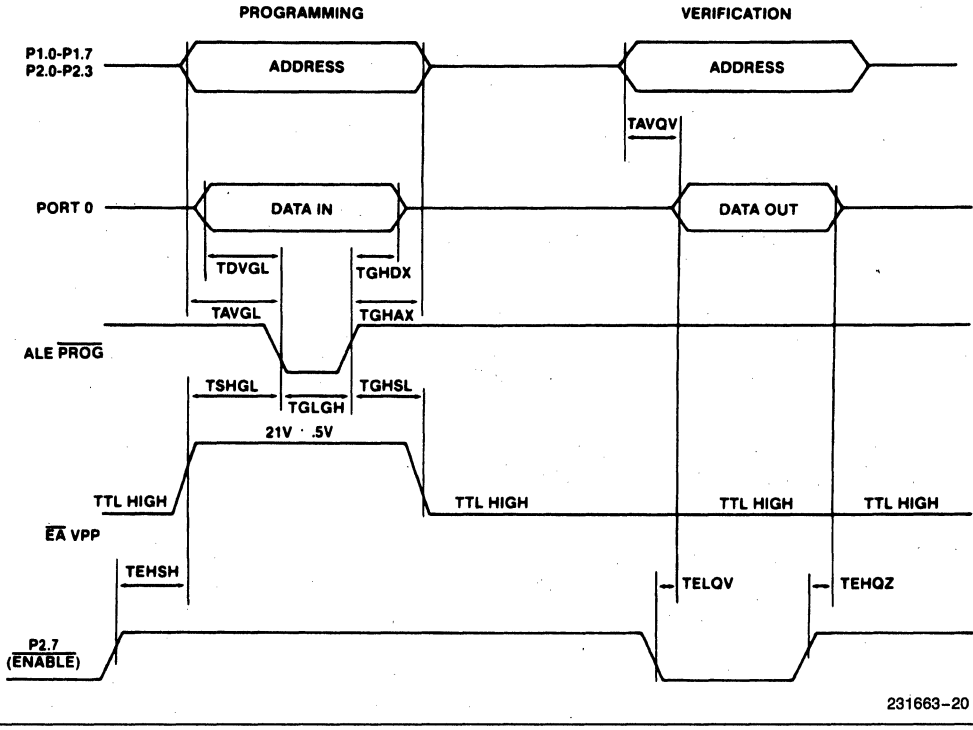
Figure 10. Program Verification Configuration

EPROM PROGRAMMING, SECURITY BIT PROGRAMMING AND VERIFICATION CHARACTERISTICS

TA = 21°C to 27°C, V_{CC} = 4.5V to 5.5V, V_{SS} = 0V

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	20.5	21.5	V
IPP	Programming Current		30	mA
1/TCLCL	Oscillator Frequency	4	6	MHz
TAVGL	Address Setup to $\overline{PROG}$	48TCLCL		
TGHAX	Address Hold after $\overline{PROG}$	48TCLCL		
TDVGL	Data Setup to $\overline{PROG}$	48TCLCL		
TGHDX	Data Hold after $\overline{PROG}$	48TCLCL		
TEHSH	$\overline{ENABLE}$ High to Vpp	48TCLCL		
TSHGL	V_{PP} Setup to $\overline{PROG}$	10		μsec
TGHSL	V_{PP} Hold after $\overline{PROG}$	10		μsec
TGLGH	$\overline{PROG}$ Width	45	55	msec
TAVQV	Address to Data Valid		48TCLCL	
TELQV	$\overline{ENABLE}$ to Data Valid		48TCLCL	
TEHQZ	Data Float after $\overline{ENABLE}$	0	48TCLCL	

EPROM PROGRAMMING, SECURITY BIT PROGRAMMING AND VERIFICATION WAVEFORMS



RUPI™ Development Support Tool

23

ICE-5100/452 In-Circuit Emulator

IN-CIRCUIT EMULATOR FOR THE UPI™-452 FAMILY OF PROGRAMMABLE I/O PROCESSORS

The ICE-5100/452 In-Circuit Emulator is a complete hardware/software debug environment for developing embedded control applications based on the Intel UPI™-452 family of I/O peripherals. With high-performance full-speed emulation, symbolic debugging, and flexible memory mapping, the ICE-5100/452 emulator expedites all stages of development: hardware development, software development, system integration, and system test; shortening your project's time to market.

FEATURES

- Full speed to the speed of the component.
- 64KB of emulation mapped memory.
- 254 frames of execution trace.
- Symbolic debug.
- Serial link to an IBM PC XT, AT, 100% compatible.
- Four address breakpoints with in-range, out-of-range, and page breaks.
- On-line disassembler and single line assembler.
- Full emulation and debug support for the FIFO Buffer.
- Source code display.
- ASM-51 and PL/M-51 language support.
- Pop-up help.
- DOS shell escape.
- On-line tutorial.
- Built-in CRT based editor.
- System self-test diagnostics.
- Worldwide service and support.

intel

September, 1988

Order Number: 280817-001

ONE TOOL FOR ENTIRE DEVELOPMENT CYCLE

The ICE-5100/452 emulator speeds target system development by allowing hardware and software design to proceed simultaneously. You can develop software even before prototype hardware is finished. And because the ICE-5100/452 emulator precisely matches the component's electrical and timing characteristics, it's a valuable tool for hardware development and debug. Thus, the ICE-5100/452 emulator can debug a prototype or production system at any stage in its development, without introducing extraneous hardware or software test tools.

HIGH-SPEED, REAL-TIME EMULATION

The ICE-5100/452 emulator provides full-speed, real-time emulation up to the speed of the component. Because the emulator is fully transparent to the target system, you have complete control over hardware and software debug and system integration.

64KB of zero wait-state emulation memory is available to replace target system code memory, allowing software debug to begin even before prototype hardware is finished.

FLEXIBLE BREAKPOINTING FOR QUICK PROBLEM ISOLATION

The ICE-5100/452 emulator supports three different types of break specifications: specific address breaks on up to 64,000 possible addresses; range breaks, both within and outside a user-defined range; and page breaks, up to 256 pages on 256-byte boundaries. 254 frames of execution trace memory provide ample debug information, with each frame divided into 16 bits of program execution address and 8 bits of external event information. A maximum of four tracepoints allows qualified trace for a variety of debug conditions.

SYMBOLIC DEBUGGING FOR FAST DEVELOPMENT

Design team productivity is enhanced by the use of symbolic debug references to program line, high-level statements, and module and variable names. The terms used to develop programs are the same used for system debugging.

PATCH CODE WITHOUT RECOMPILING

Code-patching is easy with the ICE-5100/452 emulator's single-line assembler. Machine code can be disassembled to mnemonics for significantly easier debugging and project development.

EASY TO LEARN AND USE

The ICE-5100/452 is accompanied by a full tutorial that explains all system functions and provides many examples. Additional features such as on-line help, a built-in CRT-based editor, and DOS shell escape make the emulator fast and easy to use for both novice and experienced users. You can develop your own test suites or save frequently-used debug routines as debug procedures (PROCs) that can be invoked with a single command.

WORLDWIDE SERVICE AND SUPPORT

The ICE-5100/452 emulator is supported by Intel's worldwide service and support organization. In addition to an extended warranty, you can choose from hotline support, on-site system engineering assistance, and a variety of hands-on training workshops.

ELECTRICAL CONSIDERATIONS

The emulation processor's user-pin timings and loadings are identical to the 452 component except as follows:

- Up to 25 pF of additional pin capacitance is contributed by the processor module and target adaptor assemblies.

PROCESSOR MODULE DIMENSIONS

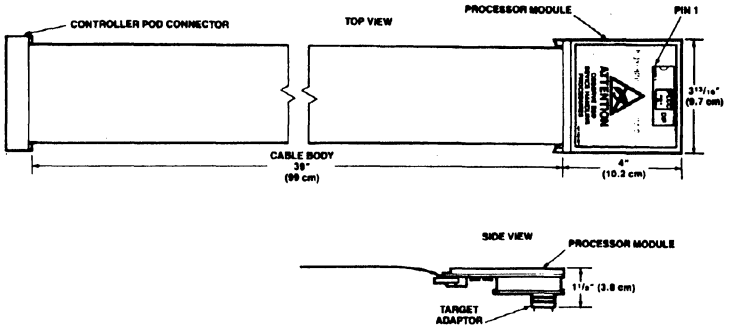


Figure 1: Processor Module Dimensions

SPECIFICATIONS

Host Requirements:

IBM PC-XT, AT or compatible
PC-DOS 3.0 or later
512K RAM
One floppy drive and hard disk

Physical Characteristics:

The ICE-5100/452 emulator consists of the following components:

Unit	Width		Height		Length	
	Inch	Cm	Inch	Cm	Inch	Cm
Controller Pod	8.25	21.0	1.5	3.8	13.5	34.3
User Cable					39.0	99.0
Processor Module*	3.8	9.7	1.5	3.8	4.0	10.2
Power Supply	7.6	18.1	4.0	10.2	11.0	28.0
Serial Cable					144.0	360.0

*with supplied target adapter.

Electrical Characteristics:

Power supply
100-120V or 220-240V selectable
50-60 Hz
2 amps (AC max) @ 120V
1 amp (AC max) @ 240V

Environmental Characteristics:

Operating temperature: +10°C to +40°C (50°F to 104°F)
Operating humidity: Maximum of 85% relative humidity, non-condensing

ORDERING INFORMATION

Order Code	Description
pl452KITAD	Kit contains ICE-5100/452 user probe assembly, power supply and cables, serial cables, target adapter, crystal power accessory, emulator controller pod, emulator software, DOS host communication cables, ASM-51 and AEDIT text editor (requires software license).
pl452KITD	Kit contains the same components as pl452KITAD, excluding ASM-51 and the AEDIT text editor (requires software license).
pC452KITD	Conversion kit for ICE-5100/451, ICE-5100/252, or ICE-5100/044 running PC-DOS 3.0 or later, to provide emulation support for 80C452 components (requires software license).
TA452E	Target adapter for 68-pin PLCC package support.
D86ASM51	ASM/RL 51 package for PC-DOS (requires software license).
D86PLM51	PL/M/RL 51 package for PC-DOS (requires software license).
D86EDINL	AEDIT text editor for PC-DOS.

MCS is a registered trademark and ICE is a trademark of Intel Corporation.

IBM and PC/AT are registered trademarks and PC/XT a trademark of International Business Machines Corporation.

For direct information on Intel's Development Tools, or for the number of your nearest sales office or distributor, call 800-874-6835 (U.S.). For information or literature on additional Intel products, call 800-548-4725 (U.S. and Canada).

MCS®-80/85 Data Sheets

24

8080A/8080A-1/8080A-2
8-BIT N-CHANNEL MICROPROCESSOR

- **TTL Drive Capability**
- **2 μs (−1:1.3 μs, −2:1.5 μs) Instruction Cycle**
- **Powerful Problem Solving Instruction Set**
- **6 General Purpose Registers and an Accumulator**
- **16-Bit Program Counter for Directly Addressing up to 64K Bytes of Memory**
- **16-Bit Stack Pointer and Stack Manipulation Instructions for Rapid Switching of the Program Environment**
- **Decimal, Binary, and Double Precision Arithmetic**
- **Ability to Provide Priority Vectored Interrupts**
- **512 Directly Addressed I/O Ports**
- **Available in EXPRESS — Standard Temperature Range**
- **Available in 40-Lead Cerdip and Plastic Packages**
  (See Packaging Spec. Order #231369)

The Intel® 8080A is a complete 8-bit parallel central processing unit (CPU). It is fabricated on a single LSI chip using Intel's n-channel silicon gate MOS process. This offers the user a high performance solution to control and processing applications.

The 8080A contains 6 8-bit general purpose working registers and an accumulator. The 6 general purpose registers may be addressed individually or in pairs providing both single and double precision operators. Arithmetic and logical instructions set or reset 4 testable flags. A fifth flag provides decimal arithmetic operation.

The 8080A has an external stack feature wherein any portion of memory may be used as a last in/first out stack to store/retrieve the contents of the accumulator, flags, program counter, and all of the 6 general purpose registers. The 16-bit stack pointer controls the addressing of this external stack. This stack gives the 8080A the ability to easily handle multiple level priority interrupts by rapidly storing and restoring processor status. It also provides almost unlimited subroutine nesting.

This microprocessor has been designed to simplify systems design. Separate 16-line address and 8-line bidirectional data busses are used to facilitate easy interface to memory and I/O. Signals to control the interface to memory and I/O are provided directly by the 8080A. Ultimate control of the address and data busses resides with the HOLD signal. It provides the ability to suspend processor operation and force the address and data busses into a high impedance state. This permits OR-tying these busses with other controlling devices for (DMA) direct memory access or multi-processor operation.

NOTE:
The 8080A is functionally and electrically compatible with the Intel 8080.

November 1986
Order Number: 231453-001

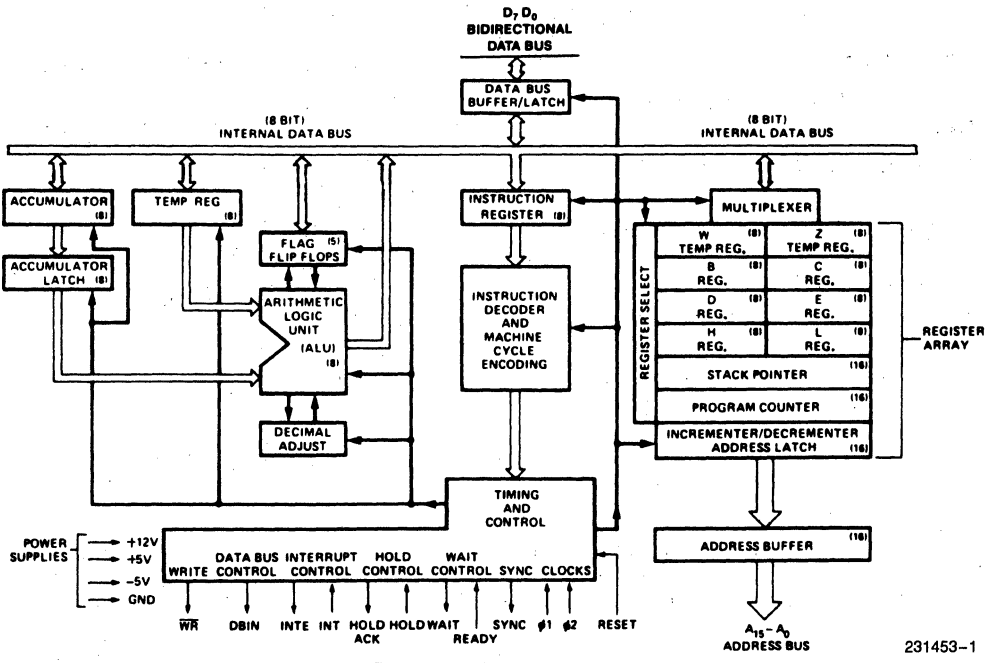


Figure 1. Block Diagram

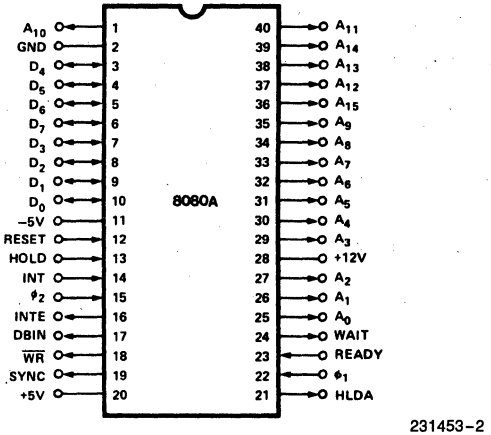


Figure 2. Pin Configuration

Table 1. Pin Description

Symbol	Type	Name and Function
$A_{15}-A_0$	O	**ADDRESS BUS:** The address bus provides the address to memory (up to 64K 8-bit words) or denotes the I/O device number for up to 256 input and 256 output devices. A_0 is the least significant address bit.
D_7-D_0	I/O	**DATA BUS:** The data bus provides bi-directional communication between the CPU, memory, and I/O devices for instructions and data transfers. Also, during the first clock cycle of each machine cycle, the 8080A outputs a status word on the data bus that describes the current machine cycle. D_0 is the least significant bit.
SYNC	O	**SYNCHRONIZING SIGNAL:** The SYNC pin provides a signal to indicate the beginning of each machine cycle.
DBIN	O	**DATA BUS IN:** The DBIN signal indicates to external circuits that the data bus is in the input mode. This signal should be used to enable the gating of data onto the 8080A data bus from memory or I/O.
READY	I	**READY:** The READY signal indicates to the 8080A that valid memory or input data is available on the 8080A data bus. This signal is used to synchronize the CPU with slower memory or I/O devices. If after sending an address out the 8080A does not receive a READY input, the 8080A will enter a WAIT state for as long as the READY line is low. READY can also be used to single step the CPU.
WAIT	O	**WAIT:** The WAIT signal acknowledges that the CPU is in a WAIT state.
$\overline{WR}$	O	**WRITE:** The $\overline{WR}$ signal is used for memory WRITE or I/O output control. The data on the data bus is stable while the $\overline{WR}$ signal is active low ($\overline{WR} = 0$).
HOLD	I	**HOLD:** The HOLD signal requests the CPU to enter the HOLD state. The HOLD state allows an external device to gain control of the 8080A address and data bus as soon as the 8080A has completed its use of these busses for the current machine cycle. It is recognized under the following conditions: • the CPU is in the HALT state. • the CPU is in the T2 or TW state and the READY signal is active. As a result of entering the HOLD state the CPU ADDRESS BUS ($A_{15}-A_0$) and DATA BUS (D_7-D_0) will be in their high impedance state. The CPU acknowledges its state with the HOLD ACKNOWLEDGE (HLDA) pin.
HLDA	O	**HOLD ACKNOWLEDGE:** The HLDA signal appears in response to the HOLD signal and indicates that the data and address bus will go to the high impedance state. The HLDA signal begins at: • T3 for READ memory or input. • The Clock Period following T3 for WRITE memory or OUTPUT operation. In either case, the HLDA signal appears after the rising edge of ϕ_2.
INTE	O	**INTERRUPT ENABLE:** Indicates the content of the internal interrupt enable flip/flop. This flip/flop may be set or reset by the Enable and Disable Interrupt instructions and inhibits interrupts from being accepted by the CPU when it is reset. It is automatically reset (disabling further interrupts) at time T1 of the instruction fetch cycle (M1) when an interrupt is accepted and is also reset by the RESET signal.
INT	I	**INTERRUPT REQUEST:** The CPU recognizes an interrupt request on this line at the end of the current instruction or while halted. If the CPU is in the HOLD state or if the Interrupt Enable flip/flop is reset it will not honor the request.
RESET[1]	I	**RESET:** While the RESET signal is activated, the content of the program counter is cleared. After RESET, the program will start at location 0 in memory. The INTE and HLDA flip/flops are also reset. Note that the flags, accumulator, stack pointer, and registers are not cleared.
V_{SS}		**GROUND:** Reference.
V_{DD}		**POWER:** +12 ±5% V.
V_{CC}		**POWER:** +5 ±5% V.
V_{BB}		**POWER:** −5 ±5% V.
ϕ_1, ϕ_2		**CLOCK PHASES:** 2 externally supplied clock phases. (non TTL compatible)

NOTE:
1. The RESET signal must be active for a minimum of 3 clock cycles.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....0°C to +70°C
Storage Temperature .....−65°C to +150°C
All Input or Output Voltages with Respect to V_{BB} .....−0.3V to +20V
V_{CC}, V_{DD} and V_{SS} with Respect to V_{BB} .....−0.3V to +20V
Power Dissipation.....1.5W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

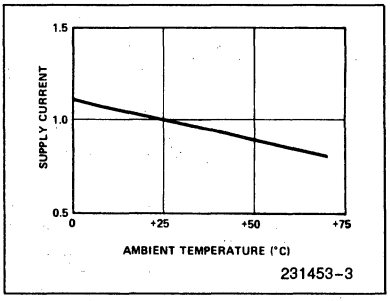
T_A = 0°C to 70°C, V_{DD} = +12V ±5%, V_{CC} = +5V ±5%, V_{BB} = −5V ±5%, V_{SS} = 0V; unless otherwise noted

Symbol	Parameter	Min	Typ	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	V_{SS} − 1		V_{SS} + 0.8	V	
V_{IHC}	Clock Input High Voltage	9.0		V_{DD} + 1	V	
V_{IL}	Input Low Voltage	V_{SS} − 1		V_{SS} + 0.8	V	
V_{IH}	Input High Voltage	3.3		V_{CC} + 1	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 1.9 mA on All Outputs, I_{OH} = −150 μA.
V_{OH}	Output High Voltage	3.7			V	
$I_{DD\ (AV)}$	Avg. Power Supply Current (V_{DD})		40	70	mA	Operation T_{CY} = 0.48 μs
$I_{CC\ (AV)}$	Avg. Power Supply Current (V_{CC})		60	80	mA	
$I_{BB\ (AV)}$	Avg. Power Supply Current (V_{BB})		0.01	1	mA	
I_{IL}	Input Leakage			±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{CL}	Clock Leakage			±10	μA	$V_{SS} \le V_{CLOCK} \le V_{DD}$
I_{DL}	Data Bus Leakage in Input Mode			−100 −2.0	μA mA	$V_{SS} \le V_{IN} \le V_{SS}$ + 0.8V V_{SS} + 0.8V $\le V_{IN} \le V_{CC}$
I_{FL}	Address and Data Bus Leakage During HOLD			+10 −100	μA	$V_{ADDR/DATA} = V_{CC}$ $V_{ADDR/DATA} = V_{SS}$ + 0.45V

CAPACITANCE

T_A = 25°C, V_{CC} = V_{DD} = V_{SS} = 0V, V_{BB} = −5V

Symbol	Parameter	Typ	Max	Unit	Test Condition
C_ϕ	Clock Capacitance	17	25	pF	f_c = 1 MHz
C_{IN}	Input Capacitance	6	10	pF	Unmeasured Pins
C_{OUT}	Output Capacitance	10	20	pF	Returned to V_{SS}

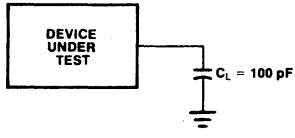


Typical Supply Current vs Temperature, Normalized
ΔI Supply/ΔT_A = −0.45%/°C

A.C. CHARACTERISTICS (8080A) T_A = 0°C to 70°C, V_{DD} = +12V ±5%, V_{CC} = +5V ±5%, V_{BB} = −5V ±5%, V_{SS} = 0V; unless otherwise noted

Symbol	Parameter	Min	Max	−1 Min	−1 Max	−2 Min	−2 Max	Unit	Test Condition
t_{CY}(3)	Clock Period	0.48	2.0	0.32	2.0	0.38	2.0	μs	
t_r, t_f	Clock Rise and Fall Time	0	50	0	25	0	50	ns	
$t_{\phi 1}$	ϕ_1 Pulse Width	60		50		60		ns	
$t_{\phi 2}$	ϕ_2 Pulse Width	220		145		175		ns	
t_{D1}	Delay ϕ_1 to ϕ_2	0		0		0		ns	
t_{D2}	Delay ϕ_1 to ϕ_2	70		60		70		ns	
t_{D3}	Delay ϕ_1 to ϕ_2 Leading Edges	80		60		70	ns		
t_{DA}	Address Output Delay From ϕ_2		200		150		175	ns	C_L = 100 pF
t_{DD}	Data Output Delay From ϕ_2		200		180		200	ns	
t_{DC}	Signal Output Delay From ϕ_1 or ϕ_2 (SYNC, WR, WAIT, HLDA)		120		110		120	ns	C_L = 50 pF
t_{DF}	DBIN Delay From ϕ_2	25	140	25	130	25	140	ns	
t_{DI}(1)	Delay for Input Bus to Enter Input Mode		t_{DF}		t_{DF}		t_{DF}	ns	
t_{DS1}	Data Setup Time During ϕ_1 and DBIN	30		10		20		ns	
t_{DS2}	Data Setup Time to ϕ_2 During DBIN	150		120		130		ns	
t_{DH}(1)	Data Hold Time From ϕ_2 and DBIN	(1)		(1)		(1)		ns	
t_{IE}	INTE Output Delay From ϕ_2		200		200		200	ns	C_L = 50 pF
t_{RS}	READY Setup Time During ϕ_2	120		90		90		ns	
t_{HS}	HOLD Setup Time During ϕ_2	140		120		120		ns	
t_{IS}	INT Setup Time During ϕ_2	120		100		100		ns	
t_H	Hold Time From ϕ_2 (READY, INT, HOLD)	0		0		0		ns	
t_{FD}	Delay to Float During Hold (Address and Data Bus)		120		120		120	ns	
t_{AW}	Address Stable Prior to WR	(5)		(5)		(5)		ns	
t_{DW}	Output Data Stable Prior to WR	(6)		(6)		(6)		ns	
t_{WD}	Output Data Stable From WR	(7)		(7)		(7)		ns	
t_{WA}	Address Stable From WR	(7)		(7)		(7)		ns	
t_{HF}	HLDA to Float Delay	(8)		(8)		(8)		ns	
t_{WF}	WR to Float Delay	(9)		(9)		(9)		ns	
t_{AH}	Address Hold Time After DBIN During HLDA	−20		−20		−20		ns	

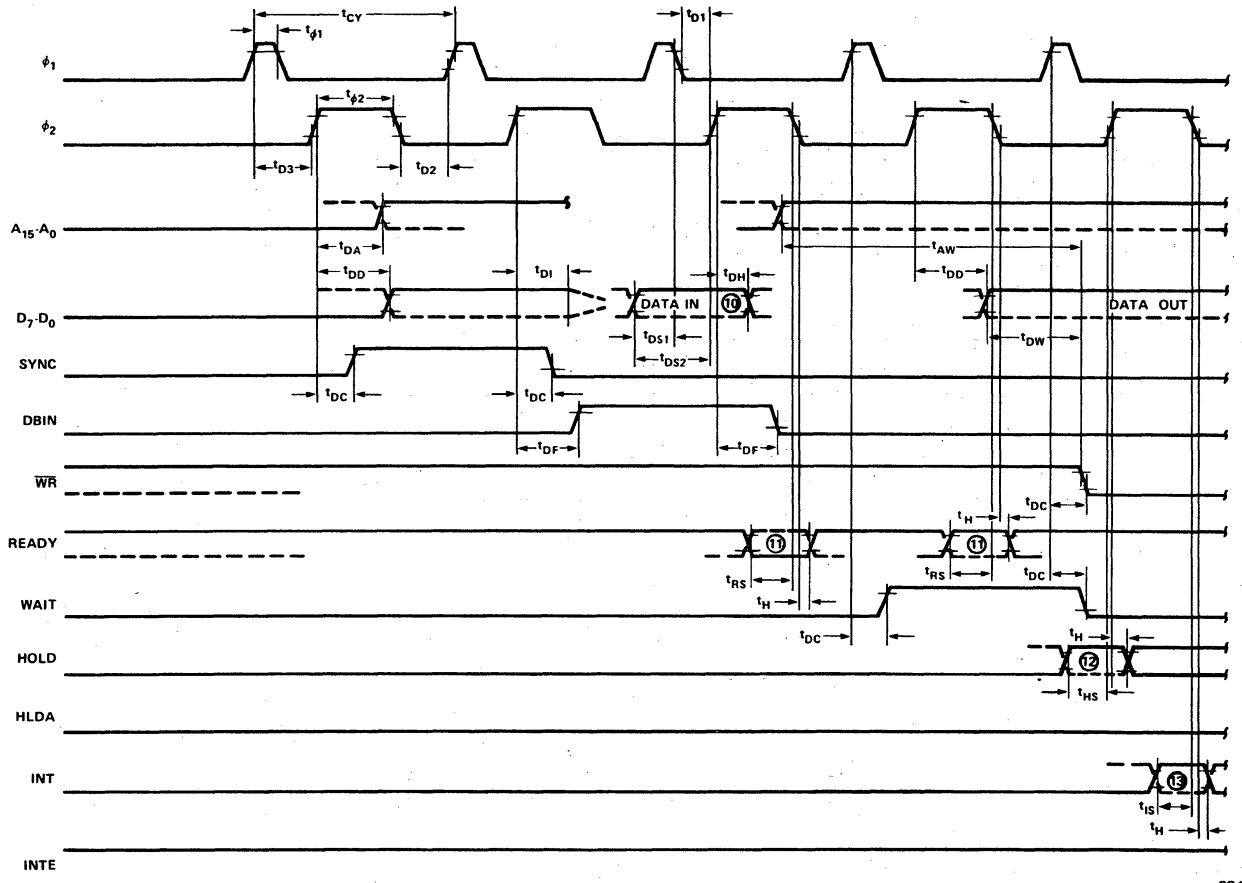
A.C. TESTING LOAD CIRCUIT



231453-4

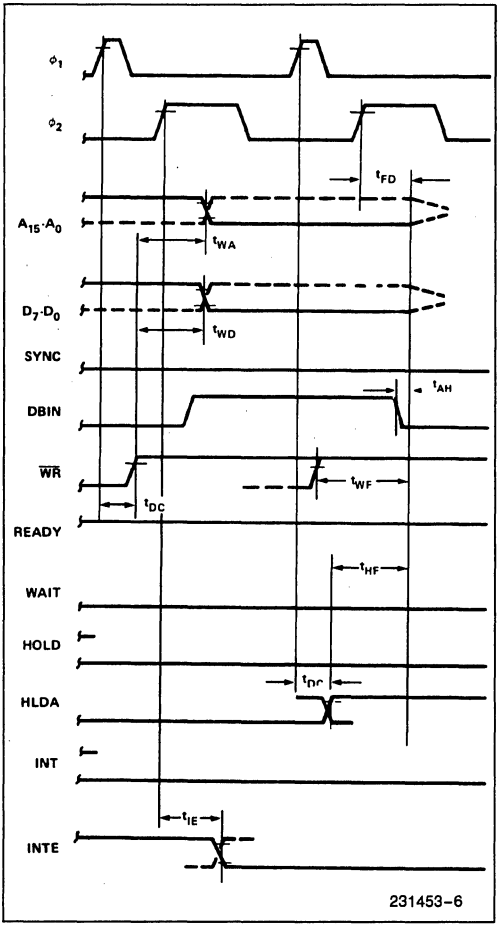
C_L = 100 pF
C_L Includes Jig Capacitance

WAVEFORMS



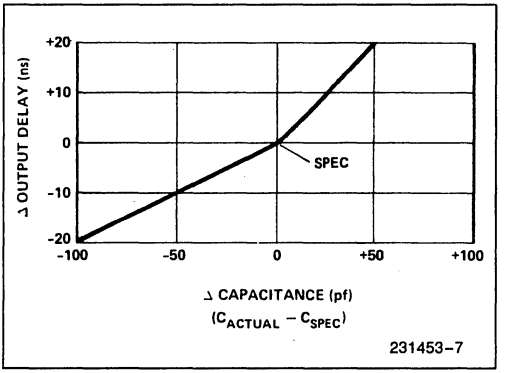
NOTE:
Timing measurements are made at the following reference voltages: CLOCK "1" = 8.0V, "0" = 1.0V; INPUTS "1" = 3.3V, "0" = 0.8V; OUTPUTS "1" = 2.0V, "0" = 0.8V

WAVEFORMS (Continued)



NOTES:
(Parenthesis gives −1, −2 specifications, respectively.)
1. Data input should be enabled with DBIN status. No bus conflict can then occur and data hold time is assured.
t_{DH} = 50 ns or t_{DF}, whichever is less.
2. $t_{CY} = t_{D3} + t_{r\phi2} + t_{\phi2} + t_{f\phi2} + t_{D2} + t_{r\phi1} \geq 480$ ns (−1:320 ns, − 2:380 ns).

Typical Δ Output Delay vs Δ Capacitance



3. The following are relevant when interfacing the 8080A to devices having V_{IH} = 3.3V:
   a) Maximum output rise time from 0.8V to 3.3V = 100 ns @ C_L = SPEC.
   b) Output delay when measured to 3.0V = SPEC +60 ns @ C_L = SPEC.
   c) If C_L = SPEC, add 0.6 ns/pF if $C_L > C_{SPEC}$, subtract 0.3 ns/pF (from modified delay) if $C_L < C_{SPEC}$.
4. $t_{AW} = 2\,t_{CY} - t_{D3} - t_{r\phi2} - 140$ ns (−1:110 ns, − 2:130 ns).
5. $t_{DW} = t_{CY} - t_{D3} - t_{r\phi2} - 170$ ns (−1:150 ns, − 2:170 ns).
6. If not HLDA, $t_{WD} = t_{WA} = t_{D3} + t_{r\phi2} + 10$ ns. If HLDA, $t_{WD} = t_{WA} = t_{WF}$.
7. $t_{HF} = t_{D3} + t_{r\phi2} - 50$ ns.
8. $t_{WF} = t_{D3} + t_{r\phi2} - 10$ ns.
9. Data in must be stable for this period during DBIN T_3. Both t_{DS1} and t_{DS2} must be satisfied.
10. Ready signal must be stable for this period during T_2 or T_W. (Must be externally synchronized.)
11. Hold signal must be stable for this period during T_2 or T_W when entering hold mode, and during T_3, T_4, T_5 and T_{WH} when in hold mode. (External synchronization is not required.)
12. Interrupt signal must be stable during this period of the last clock cycle of any instruction in order to be recognized on the following instruction. (External synchronization is not required.)
13. This timing diagram shows timing relationships only; it does not represent any specific machine cycle.

INSTRUCTION SET

The accumulator group instructions include arithmetic and logical operators with direct, indirect, and immediate addressing modes.

Move, load, and store instruction groups provide the ability to move either 8 or 16 bits of data between memory, the six working registers and the accumulator using direct, indirect, and immediate addressing modes.

The ability to branch to different portions of the program is provided with jump, jump conditional, and computed jumps. Also the ability to call to and return from subroutines is provided both conditionally and unconditionally. The RESTART (or single byte call instruction) is useful for interrupt vector operation.

Double precision operators such as stack manipulation and double add instructions extend both the arithmetic and interrupt handling capability of the 8080A. The ability to increment and decrement memory, the six general registers and the accumulator is provided as well as extended increment and decrement instructions to operate on the register pairs and stack pointer. Further capability is provided by the ability to rotate the accumulator left or right through or around the carry bit.

Input and output may be accomplished using memory addresses as I/O ports or the directly addressed I/O provided for in the 8080A instruction set.

The following special instruction group completes the 8080A instruction set: the NOP instruction, HALT to stop processor execution and the DAA instructions provide decimal arithmetic capability. STC allows the carry flag to be directly set, and the CMC instruction allows it to be complemented. CMA complements the contents of the accumulator and XCHG exchanges the contents of two 16-bit register pairs directly.

Data and Instruction Formats

Data in the 8080A is stored in the form of 8-bit binary integers. All data transfers to they system data bus will be in the same format.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

DATA WORD

The program instructions may be one, two, or three bytes in length. Multiple byte instructions must be stored in successive words in program memory. The instruction formats then depend on the particular operation executed.

		TYPICAL INSTRUCTIONS
One Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	Register to register, memory reference, arithmetic or logical, rotate, return, push, pop, enable or disable Interrupt instructions
Two Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	Immediate mode or I/O instructions
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OPERAND	
Three Byte Instructions		
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	OP CODE	Jump, call or direct load and store instructions
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	LOW ADDRESS OR OPERAND 1	
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	HIGH ADDRESS OR OPERAND 2	

For the 8080A a logic "1" is defined as a high level and a logic "0" is defined as a low level.

Table 2. Instruction Set Summary

Mnemonic*	Instruction Code (1) D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description	Clock Cycles (2)
MOVE, LOAD, AND STORE			
MOVr1,r2	0 1 D D D S S S	Move register to register	5
MOV M,r	0 1 1 1 0 S S S	Move register to memory	7
MOV r,M	0 1 D D D 1 1 0	Move memory to register	7
MVI r	0 0 D D D 1 1 0	Move immediate register	7
MVI M	0 0 1 1 0 1 1 0	Move immediate memory	10
LXI B	0 0 0 0 0 0 0 1	Load immediate register Pair B & C	10
LXI D	0 0 0 1 0 0 0 1	Load immediate register Pair D & E	10
LXI H	0 0 1 0 0 0 0 1	Load immediate register Pair H & L	10
STAX B	0 0 0 0 0 0 1 0	Store A indirect	7
STAX D	0 0 0 1 0 0 1 0	Store A indirect	7
LDAX B	0 0 0 0 1 0 1 0	Load A indirect	7
LDAX D	0 0 0 1 1 0 1 0	Load A indirect	7
STA	0 0 1 1 0 0 1 0	Store A direct	13
LDA	0 0 1 1 1 0 1 0	Load A direct	13
SHLD	0 0 1 0 0 0 1 0	Store H & L direct	16
LHLD	0 0 1 0 1 0 1 0	Load H & L direct	16
XCHG	1 1 1 0 1 0 1 1	Exchange D & E, H & L Registers	4
STACK OPS			
PUSH B	1 1 0 0 0 1 0 1	Push register Pair B & C on stack	11
PUSH D	1 1 0 1 0 1 0 1	Push register Pair D & E on stack	11
PUSH H	1 1 1 0 0 1 0 1	Push register Pair H & L on stack	11
PUSH PSW	1 1 1 1 0 1 0 1	Push A and Flags on stack	11
POP B	1 1 0 0 0 0 0 1	Pop register Pair B & C off stack	10
POP D	1 1 0 1 0 0 0 1	Pop register Pair D & E off stack	10
POP H	1 1 1 0 0 0 0 1	Pop register Pair H & L off stack	10
POP PSW	1 1 1 1 0 0 0 1	Pop A and Flags off stack	10
XTHL	1 1 1 0 0 0 1 1	Exchange top of stack, H & L	18
SPHL	1 1 1 1 1 0 0 1	H & L to stack pointer	5
LXI SP	0 0 1 1 0 0 0 1	Load immediate stack pointer	10
INX SP	0 0 1 1 0 0 1 1	Increment stack pointer	5
DCX SP	0 0 1 1 1 0 1 1	Decrement stack pointer	5
JUMP			
JMP	1 1 0 0 0 0 1 1	Jump unconditional	10
JC	1 1 0 1 1 0 1 0	Jump on carry	10
JNC	1 1 0 1 0 0 1 0	Jump on no carry	10
JZ	1 1 0 0 1 0 1 0	Jump on zero	10
JNZ	1 1 0 0 0 0 1 0	Jump on no zero	10
JP	1 1 1 1 0 0 1 0	Jump on positive	10
JM	1 1 1 1 1 0 1 0	Jump on minus	10
JPE	1 1 1 0 1 0 1 0	Jump on parity even	10
JPO	1 1 1 0 0 0 1 0	Jump on parity odd	10
PCHL	1 1 1 0 1 0 0 1	H & L to program counter	5
CALL			
CALL	1 1 0 0 1 1 0 1	Call unconditional	17
CC	1 1 0 1 1 1 0 0	Call on carry	11/17
CNC	1 1 0 1 0 1 0 0	Call on no carry	11/17
CZ	1 1 0 0 1 1 0 0	Call on zero	11/17
CNZ	1 1 0 0 0 1 0 0	Call on no zero	11/17
CP	1 1 1 1 0 1 0 0	Call on positive	11/17
CM	1 1 1 1 1 1 0 0	Call on minus	11/17
CPE	1 1 1 0 1 1 0 0	Call on parity even	11/17
CPO	1 1 1 0 0 1 0 0	Call on parity odd	11/17
RETURN			
RET	1 1 0 0 1 0 0 1	Return	10
RC	1 1 0 1 1 0 0 0	Return on carry	5/11
RNC	1 1 0 1 0 0 0 0	Return on no carry	5/11
RZ	1 1 0 0 1 0 0 0	Return on zero	5/11
RNZ	1 1 0 0 0 0 0 0	Return on no zero	5/11
RP	1 1 1 1 0 0 0 0	Return on positive	5/11
RM	1 1 1 1 1 0 0 0	Return on minus	5/11
RPE	1 1 1 0 0 0 0 0	Return on parity even	5/11
RPO	1 1 1 0 0 0 0 0	Return on parity odd	5/11
RESTART			
RST	1 1 A A A 1 1 1	Restart	11
INCREMENT AND DECREMENT			
INR r	0 0 D D D 1 0 0	Increment register	5
DCR r	0 0 D D D 1 0 1	Decrement register	5
INR M	0 0 1 1 0 1 0 0	Increment memory	10
DCR M	0 0 1 1 0 1 0 1	Decrement memory	10
INX B	0 0 0 0 0 0 1 1	Increment B & C registers	5
INX D	0 0 0 1 0 0 1 1	Increment D & E registers	5
INX H	0 0 1 0 0 0 1 1	Increment H & L registers	5
DCX B	0 0 0 0 1 0 1 1	Decrement B & C	5
DCX D	0 0 0 1 1 0 1 1	Decrement D & E	5
DCX H	0 0 1 0 1 0 1 1	Decrement H & L	5
ADD			
ADD r	1 0 0 0 0 S S S	Add register to A	4
ADC r	1 0 0 0 1 S S S	Add register to A with carry	4
ADD M	1 0 0 0 0 1 1 0	Add memory to A	7
ADC M	1 0 0 0 1 1 1 0	Add memory to A with carry	7
ADI	1 1 0 0 0 1 1 0	Add immediate to A	7
ACI	1 1 0 0 1 1 1 0	Add immediate to A with carry	7
DAD B	0 0 0 0 1 0 0 1	Add B & C to H & L	10
DAD D	0 0 0 1 1 0 0 1	Add D & E to H & L	10
DAD H	0 0 1 0 1 0 0 1	Add H & L to H & L	10
DAD SP	0 0 1 1 1 0 0 1	Add stack pointer to H & L	10

Table 2. Instruction Set Summary (Continued)

Mnemonic*	Instruction Code (1) D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description	Clock Cycles (2)
SUBTRACT			
SUB r	1 0 0 1 0 S S S	Subtract register from A	4
SBB r	1 0 0 1 1 S S S	Subtract register from A with borrow	4
SUB M	1 0 0 1 0 1 1 0	Subtract memory from A	7
SBB M	1 0 0 1 1 1 1 0	Subtract memory from A with borrow	7
SUI	1 1 0 1 0 1 1 0	Subtract immediate from A	7
SBI	1 1 0 1 1 1 1 0	Subtract immediate from A with borrow	7
LOGICAL			
ANA r	1 0 1 0 0 S S S	And register with A	4
XRA r	1 0 1 0 1 S S S	Exclusive or register with A	4
ORA r	1 0 1 1 0 S S S	Or register with A	4
CMP r	1 0 1 1 1 S S S	Compare register with A	4
ANA M	1 0 1 0 0 1 1 0	And memory with A	7
XRA M	1 0 1 0 1 1 1 0	Exclusive Or memory with A	7
ORA M	1 0 1 1 0 1 1 0	Or memory with A	7
CMP M	1 0 1 1 1 1 1 0	Compare memory with A	7
ANI	1 1 1 0 0 1 1 0	And immediate with A	7
XRI	1 1 1 0 1 1 1 0	Exclusive Or immediate with A	7
ORI	1 1 1 1 0 1 1 0	Or immediate with A	7
CPI	1 1 1 1 1 1 1 0	Compare immediate with A	7

Mnemonic*	Instruction Code (1) D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description	Clock Cycles (2)
ROTATE			
RLC	0 0 0 0 0 1 1 1	Rotate A left	4
RRC	0 0 0 0 1 1 1 1	Rotate A right	4
RAL	0 0 0 1 0 1 1 1	Rotate A left through carry	4
RAR	0 0 0 1 1 1 1 1	Rotate A right through carry	4
SPECIALS			
CMA	0 0 1 0 1 1 1 1	Complement A	4
STC	0 0 1 1 0 1 1 1	Set carry	4
CMC	0 0 1 1 1 1 1 1	Complement carry	4
DAA	0 0 1 0 0 1 1 1	Decimal adjust A	4
INPUT/OUTPUT			
IN	1 1 0 1 1 0 1 1	Input	10
OUT	1 1 0 1 0 0 1 1	Output	10
CONTROL			
EI	1 1 1 1 1 0 1 1	Enable Interrupts	4
DI	1 1 1 1 0 0 1 1	Disable Interrupt	4
NOP	0 0 0 0 0 0 0 0	No-operation	4
HLT	0 1 1 1 0 1 1 0	Halt	7

NOTES:
1. DDD or SSS: B = 000, C = 001, D = 010, E = 011, H = 100, L = 101, Memory = 110, A = 111.
2. Two possible cycle times (6/12) indicate instruction cycles dependent on condition flags.

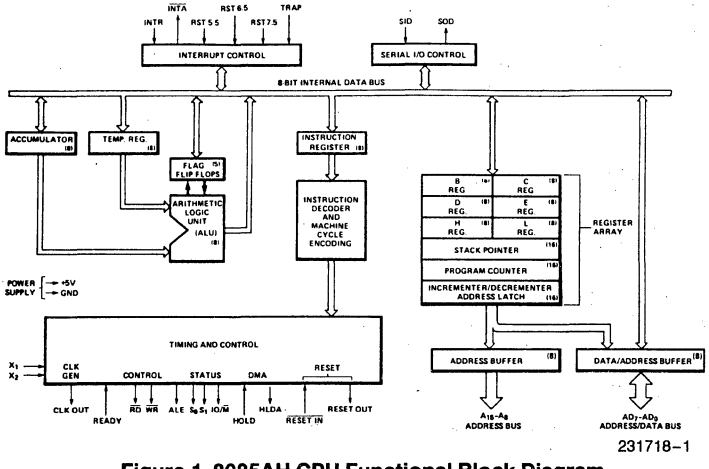
8085AH/8085AH-2/8085AH-1
8-BIT HMOS MICROPROCESSORS

- **Single +5V Power Supply with 10% Voltage Margins**
- **3 MHz, 5 MHz and 6 MHz Selections Available**
- **20% Lower Power Consumption than 8085A for 3 MHz and 5 MHz**
- **1.3 μs Instruction Cycle (8085AH); 0.8 μs (8085AH-2); 0.67 μs (8085AH-1)**
- **100% Software Compatible with 8080A**
- **On-Chip Clock Generator (with External Crystal, LC or RC Network)**
- **On-Chip System Controller; Advanced Cycle Status Information Available for Large System Control**
- **Four Vectored Interrupt Inputs (One Is Non-Maskable) Plus an 8080A-Compatible Interrupt**
- **Serial In/Serial Out Port**
- **Decimal, Binary and Double Precision Arithmetic**
- **Direct Addressing Capability to 64K Bytes of Memory**
- **Available in 40-Lead Cerdip and Plastic Packages**
  (See Packaging Spec., Order #231369)

The Intel 8085AH is a complete 8-bit parallel Central Processing Unit (CPU) implemented in N-channel, depletion load, silicon gate technology (HMOS). Its instruction set is 100% software compatible with the 8080A microprocessor, and it is designed to improve the present 8080A's performance by higher system speed. Its high level of system integration allows a minimum system of three IC's [8085AH (CPU), 8156H (RAM/IO) and 8755A (EPROM/IO)] while maintaining total system expandability. The 8085AH-2 and 8085AH-1 are faster versions of the 8085AH.

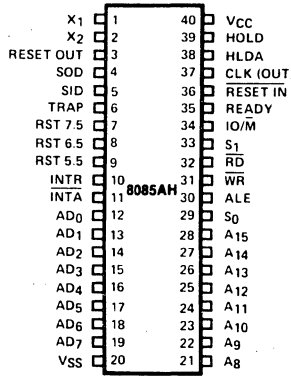
The 8085AH incorporates all of the features that the 8224 (clock generator) and 8228 (system controller) provided for the 8080A, thereby offering a higher level of system integration.

The 8085AH uses a multiplexed data bus. The address is split between the 8-bit address bus and the 8-bit data bus. The on-chip address latches of 8155H/8156H/8755A memory products allow a direct interface with the 8085AH.



231718-1

Figure 1. 8085AH CPU Functional Block Diagram



231718-2

Figure 2. 8085AH Pin Configuration

September 1987
Order Number: 231718-001

Table 1. Pin Description

Symbol	Type	Name and Function
A_8–A_{15}	O	**ADDRESS BUS:** The most significant 8 bits of memory address or the 8 bits of the I/O address, 3-stated during Hold and Halt modes and during RESET.
AD_{0-7}	I/O	**MULTIPLEXED ADDRESS/DATA BUS:** Lower 8 bits of the memory address (or I/O address) appear on the bus during the first clock cycle (T state) of a machine cycle. It then becomes the data bus during the second and third clock cycles.
ALE	O	**ADDRESS LATCH ENABLE:** It occurs during the first clock state of a machine cycle and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE is set to guarantee setup and hold times for the address information. The falling edge of ALE can also be used to strobe the status information. ALE is never 3-stated.
S_0, S_1 and IO/$\overline{M}$	O	**MACHINE CYCLE STATUS:** **IO/$\overline{M}$ S_1 S_0 Status** 0 0 1 Memory write 0 1 0 Memory read 1 0 1 I/O write 1 1 0 I/O read 0 1 1 Opcode fetch 1 1 1 Interrupt Acknowledge * 0 0 Halt * X X Hold * X X Reset * = 3-state (high impedance) X = unspecified S_1 can be used as an advanced R/$\overline{W}$ status. IO/$\overline{M}$, S_0 and S_1 become valid at the beginning of a machine cycle and remain stable throughout the cycle. The falling edge of ALE may be used to latch the state of these lines.
$\overline{RD}$	O	**READ CONTROL:** A low level on $\overline{RD}$ indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer, 3-stated during Hold and Halt modes and during RESET.
$\overline{WR}$	O	**WROTE CONTROL:** A low level on $\overline{WR}$ indicates the data on the Data Bus is to be written into the selected memory or I/O location. Data is set up at the trailing edge of $\overline{WR}$. 3-stated during Hold and Halt modes and during RESET.
READY	I	**READY:** If READY is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If READY is low, the CPU will wait an integral number of clock cycles for READY to go high before completing the read or write cycle. READY must conform to specified setup and hold times.
HOLD	I	**HOLD:** Indicates that another master is requesting the use of the address and data buses. The CPU, upon receiving the hold request, will relinquish the use of the bus as soon as the completion of the current bus transfer. Internal processing can continue. The processor can regain the bus only after the HOLD is removed. When the HOLD is acknowledged, the Address, Data $\overline{RD}$, $\overline{WR}$, and IO/$\overline{M}$ lines are 3-stated.
HLDA	O	**HOLD ACKNOWLEDGE:** Indicates that the CPU has received the HOLD request and that it will relinquish the bus in the next clock cycle. HILDA goes low after the Hold request is removed. The CPU takes the bus one half clock cycle after HLDA goes low.
INTR	I	**INTERRUPT REQUEST:** Is used as a general purpose interrupt. It is sampled only during the next to the last clock cycle of an instruction and during Hold and Halt states. If it is active, the Program Counter (PC) will be inhibited from incrementing and an $\overline{INTA}$ will be issued. During this cycle a RESTART or CALL instruction can be inserted to jump to the interrupt service routine. The INTR is enabled and disabled by software. It is disabled by Reset and immediately after an interrupt is accepted.

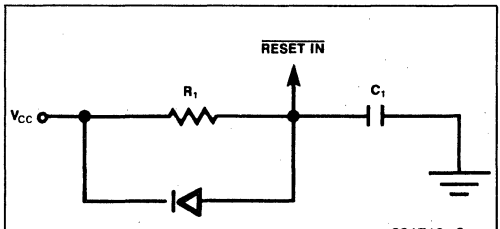
Table 1. Pin Description (Continued)

Symbol	Type	Name and Function
$\overline{INTA}$	O	**INTERRUPT ACKNOWLEDGE:** Is used instead of (and has the same timing as) $\overline{RD}$ during the Instruction cycle after an INTR is accepted. It can be used to activate an 8259A Interrupt chip or some other interrupt port.
RST 5.5 RST 6.5 RST 7.5	I	**RESTART INTERRUPTS:** These three inputs have the same timing as INTR except they cause an internal RESTART to be automatically inserted. The priority of these interrupt is ordered as shown in Table 2. These interrupts have a higher priority than INTR. In addition, they may be individually masked out using the SIM instruction.
TRAP	I	**TRAP:** Trap interrupt is a non-maskable RESTART interrupt. It is recognized at the same time as INTR or RST 5.5–7.5. It is unaffected by any mask or Interrupt Enable. It has the highest priority of any interrupt. (See Table 2.)
$\overline{RESET\ IN}$	I	**RESET IN:** Sets the Program Counter to zero and resets the Interrupt Enable and HLDA flip-flops. The data and address buses and the control lines are 3-stated during RESET and because of the asynchronous nature of RESET, the processor's internal registers and flags may be altered by RESET with unpredictable results. $\overline{RESET\ IN}$ is a Schmitt-triggered input, allowing connection to an R-C network for power-on RESET delay (see Figure 3). Upon power-up, $\overline{RESET\ IN}$ must remain low for at least 10 ms after minimum V_{CC} has been reached. For proper reset operation after the power-up duration, $\overline{RESET\ IN}$ should be kept low a minimum of three clock periods. The CPU is held in the reset condition as long as $\overline{RESET\ IN}$ is applied.
RESET OUT	O	**RESET OUT:** Reset Out indicates CPU is being reset. Can be used as a system reset. The signal is synchronized to the processor clock and lasts an integral number of clock periods.
X_1, X_2	I	**X_1 and X_2:** Are connected to a crystal, LC, or RC network to drive the internal clock generator. X_1 can also be an external clock input from a logic gate. The input frequency is divided by 2 to give the processor's internal operating frequency.
CLK	O	**CLOCK:** Clock output for use as a system clock. The period of CLK is twice the X_1, X_2 input period.
SID	I	**SERIAL INPUT DATA LINE:** The data on this line is loaded into accumulator bit 7 whenever a RIM instruction is executed.
SOD	O	**SERIAL OUTPUT DATA LINE:** The output SOD is set or reset as specified by the SIM instruction.
V_{CC}		**POWER:** +5 volt supply.
V_{SS}		**GROUND:** Reference.

Table 2. Interrupt Priority, Restart Address and Sensitivity

Name	Priority	Address Branched to(1) When Interrupt Occurs	Type Trigger
TRAP	1	24H	Rising Edge AND High Level until Sampled
RST 7.5	2	3CH	Rising Edge (Latched)
RST 6.5	3	34H	High Level until Sampled
RST 5.5	4	2CH	High Level until Sampled
INTR	5	(Note 2)	High Level until Sampled

NOTES:
1. The processor pushes the PC on the stack before branching to the indicated address.
2. The address branched to depends on the instruction provided to the CPU when the interrupt is acknowledged.



Typical Power-On Reset RC Values*
R_1 = 75 KΩ
C_1 = 1 μF
*Values May Have to Vary Due to Applied Power Supply Ramp Up Time.

Figure 3. Power-On Reset Circuit

FUNCTIONAL DESCRIPTION

The 8085AH is a complete 8-bit parallel central processor. It is designed with N-channel, depletion load, silicon gate technology (HMOS), and requires a single +5V supply. Its basic clock speed is 3 MHz (8085AH), 5 MHz (8085AH-2), or 6 MHz (8085-AH-1), thus improving on the present 8080A's performance with higher system speed. Also it is designed to fit into a minimum system of three IC's: The CPU (8085AH), a RAM/IO (8156H), and an EPROM/IO chip (8755A).

The 8085AH has twelve addressable 8-bit registers. Four of them can function only as two 16-bit register pairs. Six others can be used interchangeably as 8-bit registers or as 16-bit register pairs. The 8085AH register set is as follows:

Mnemonic	Register	Contents
ACC or A	Accumulator	8 Bits
PC	Program Counter	16-Bit Address
BC, DE, HL	General-Purpose Registers; data pointer (HL)	8-Bits x 6 or 16 Bits x 3
SP	Stack Pointer	16-Bit Address
Flags or F	Flag Register	5 Flags (8-Bit Space)

The 8085AH uses a multiplexed Data Bus. The address is split between the higher 8-bit Address Bus and the lower 8-bit Address/Data Bus. During the first T state (clock cycle) of a machine cycle the low order address is sent out on the Address/Data bus. These lower 8 bits may be latched externally by the Address Latch Enable signal (ALE). During the rest of the machine cycle the data bus is used for memory or I/O data.

The 8085AH provides $\overline{RD}$, $\overline{WR}$, S_0, S_1, and IO/$\overline{M}$ signals for bus control. An Interrupt Acknowledge signal ($\overline{INTA}$) is also provided. HOLD and all Interrupts are synchronized with the processor's internal clock. The 8085AH also provides Serial Input Data (SID) and Serial Output Data (SOD) lines for simple serial interface.

In addition to these features, the 8085AH has three maskable, vector interrupt pins, one nonmaskable TRAP interrupt, and a bus vectored interrupt, INTR.

INTERRUPT AND SERIAL I/O

The 8085AH has 5 interrupt inputs: INTR, RST 5.5, RST 6.5, RST 7.5, and TRAP. INTR is identical in function to the 8080A INT. Each of the three RESTART inputs, 5.5, 6.5, and 7.5, has a programmable mask. TRAP is also a RESTART interrupt but it is nonmaskable.

The three maskable interrupt cause the internal execution of RESTART (saving the program counter in the stack and branching to the RESTART address) if the interrupts are enabled and if the interrupt mask is not set. The nonmaskable TRAP causes the internal execution of a RESTART vector independent of the state of the interrupt enable or masks. (See Table 2.)

There are two different types of inputs in the restart interrupts. RST 5.5 and RST 6.5 are *high level-sensitive* like INTR (and INT on the 8080) and are recognized with the same timing as INTR. RST 7.5 is *rising edge-sensitive.*

For RST 7.5, only a pulse is required to set an internal flip-flop which generates the internal interrupt request (a normally high level signal with a low going pulse is recommended for highest system noise immunity). The RST 7.5 request flip-flop remains set until the request is serviced. Then it is reset automatically. This flip-flop may also be reset by using the SIM instruction or by issuing a $\overline{RESET\ IN}$ to the 8085AH. The RST 7.5 internal flip-flop will be set by a pulse on the RST 7.5 pin even when the RST 7.5 interrupt is masked out.

The status of the three RST interrupt masks can only be affected by the SIM instruction and $\overline{RESET\ IN}$. (See SIM, Chapter 5 of the 8080/8085 User's Manual.)

The interrupts are arranged in a fixed priority that determines which interrupt is to be recognized if more than one is pending as follows: TRAP—highest priority, RST 7.5, RST 6.5, RST 5.5, INTR—lowest priority. This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. RST 5.5 can interrupt an RST 7.5 routine if the interrupts are re-enabled before the end of the RST 7.5 routine.

The TRAP interrupt is useful for catastrophic events such as power failure or bus error. The TRAP input is recognized just as any other interrupt but has the

highest priority. It is not affected by any flag or mask. The TRAP input is both *edge and level sensitive.* The TRAP input must go high and remain high until it is acknowledged. It will not be recognized again until it goes low, then high again. This avoids any false triggering due to noise or logic glitches. Figure 4 illustrates the TRAP interrupt request circuitry within the 8085AH. Note that the servicing of any interrupt (TRAP, RST 7.5, RST 6.5, RST 5.5, INTR) disables all future interrupts (except TRAPs) until an EI instruction is executed.

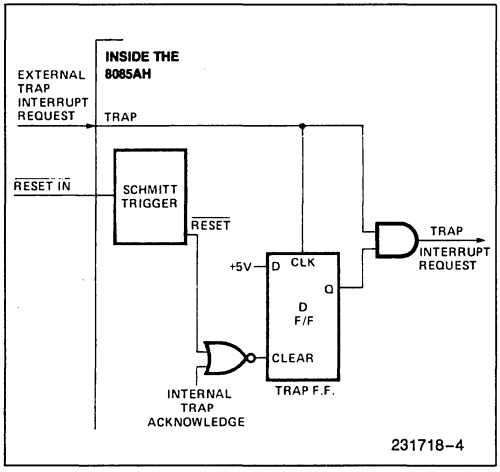


Figure 4. TRAP and RESET In Circuit

The TRAP interrupt is special in that it disables interrupts, but preserves the previous interrupt enable status. Performing the first RIM instruction following a TRAP interrupt allows you to determine whether interrupts were enabled or disabled prior to the TRAP. All subsequent RIM instructions provide current interrupt enable status. Performing a RIM instruction following INTR, or RST 5.5–7.5 will provide current Interrupt Enable status, revealing that interrupts are disabled. See the description of the RIM instruction in the 8080/8085 Family User's Manual.

The serial I/O system is also controlled by the RIM and SIM instruction. SID is read by RIM, and SIM sets the SOD data.

DRIVING THE X_1 AND X_2 INPUTS

You may drive the clock inputs of the 8085AH, 8085AH-2, or 8085AH-1 with a crystal, an LC tuned circuit, an RC network, or an external clock source. The crystal frequency must be at least 1 MHz, and must be twice the desired internal clock frequency; hence, the 8085AH is operated with a 6 MHz crystal (for 3 MHz clock), the 8085AH-2 operated with a 10 MHz crystal (for 5 MHz clock), and the 8085AH-1 can be operated with a 12 MHz crystal (for 6 MHz clock). If a crystal is used, it must have the following characteristics:

Parallel resonance at twice the clock frequency desired
C_L (load capacitance) $\leq$ 30 pF
C_S (Shunt capacitance) $\leq$ 7 pF
R_S (equivalent shunt resistance) $\leq 75\Omega$
Drive level: 10 mW
Frequency tolerance: $\pm 0.005\%$ (suggested)

Note the use of the 20 pF capacitor between X_2 and ground. This capacitor is required with crystal frequencies below 4 MHz to assure oscillator startup at the correct frequency. A parallel-resonant LC citcuit may be used as the frequency-determining network for the 8085AH, providing that its frequency tolerance of approximately $\pm 10\%$ is acceptable. The components are chosen from the formula:

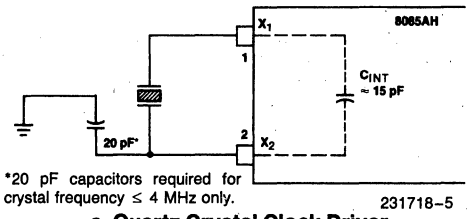
$$f = \frac{1}{2\pi\sqrt{L(C_{ext} + C_{int})}}$$

To minimize variations in frequency, it is recommended that you choose a value for C_{ext} that is at least twice that of C_{int}, or 30 pF. The use of an LC circuit is not recommended for frequencies higher than approximately 5 MHz.

An RC circuit may be used as the frequency-determining network for the 8085AH if maintaining a precise clock frequency is of no importance. Variations in the on-chip timing generation can cause a wide variation in frequency when using the RC mode. Its advantage is its low component cost. The driving frequency generated by the circuit shown is approximately 3 MHz. It is not recommended that frequencies greatly higher or lower than this be attempted.

Figure 5 shows the recommended clock driver circuits. Note in d and e that pullup resistors are required to assure that the high level voltage of the input is at least 4V and maximum low level voltage of 0.8V.

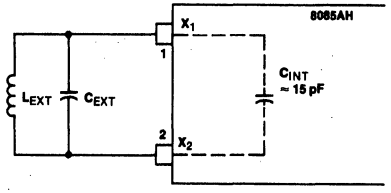
For driving frequencies up to and including 6 MHz you may supply the driving signal to X_1 and leave X_2 open-circuited (Figure 5d). If the driving frequency is from 6 MHz to 12 MHz, stability of the clock generator will be improved by driving both X_1 and X_2 with a push-pull source (Figure 5e). To prevent self-oscillation of the 8085AH, be sure that X_2 is not coupled back to X_1 through the driving circuit.



*20 pF capacitors required for crystal frequency ≤ 4 MHz only.

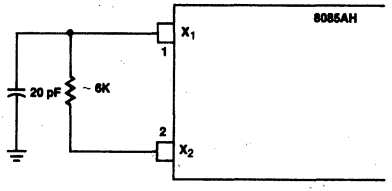
231718-5

a. Quartz Crystal Clock Driver



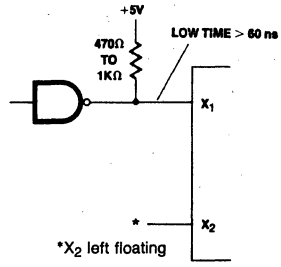
231718-6

b. LC Tuned Circuit Clock Driver



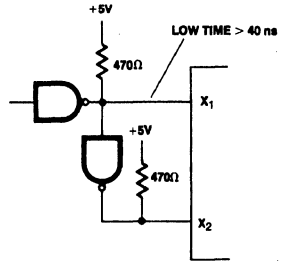
231718-7

c. RC Circuit Clock Driver



231718-8

d. 1-6 MHz Input Frequency Clock Driver Circuit



231718-9

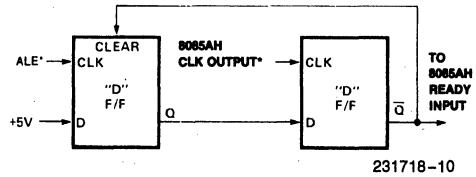
e. 1-12 MHz Input Frequency External Clock Driver Circuit

Figure 5. Clock Driver Circuits

GENERATING AN 8085AH WAIT STATE

If your system requirements are such that slow memories or peripheral devices are being used, the circuit shown in Figure 6 may be used to insert one WAIT state in each 8085AH machine cycle.

The D flip-flops should be chosen so that
- CLK is rising edge-triggered
- CLEAR is low-level active.



231718-10

*ALE and CLK (OUT) should be buffered if CLK input of latch exceeds 8085AH IOL or IOH.

Figure 6. Generation of a Wait State for 8085AH CPU

As in the 8080, the READY line is used to extend the read and write pulse lengths so that the 8085AH can be used with slow memory. HOLD causes the CPU to relinquish the bus when it is through with it by floating the Address and Data Buses.

SYSTEM INTERFACE

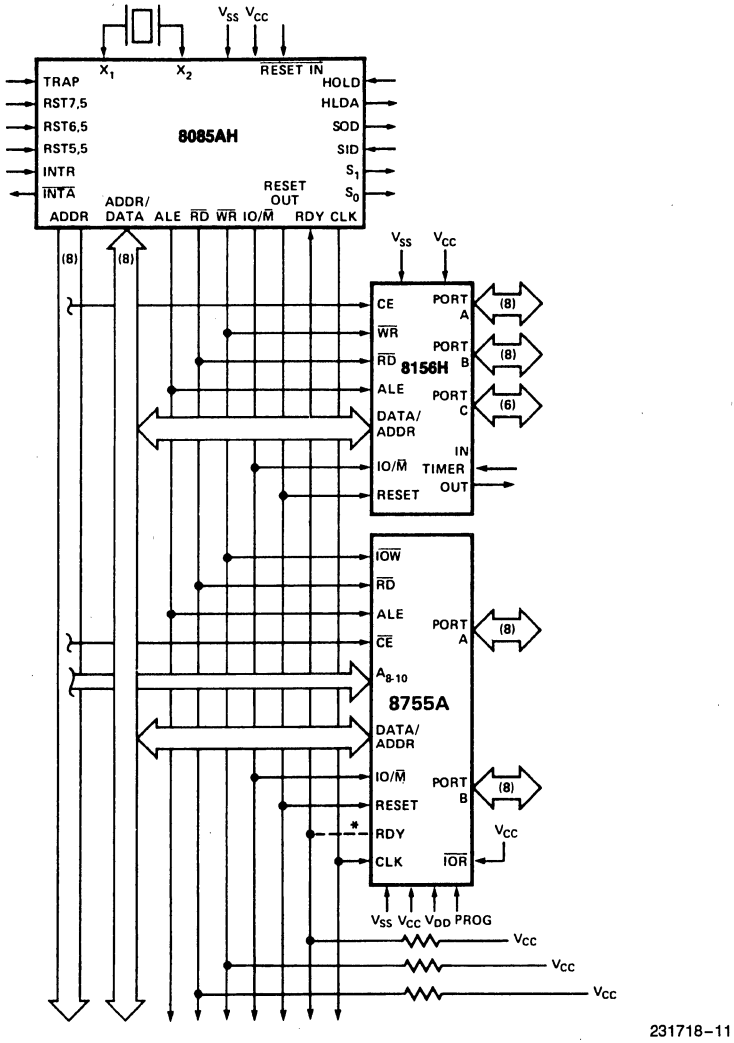
The 8085AH family includes memory components, which are directly compatible to the 8085AH CPU. For example, a system consisting of the three chips, 8085AH, 8156H and 8755A will have the following features:

- 2K Bytes EPROM
- 256 Bytes RAM
- 1 Timer/Counter
- 4 8-bit I/O Ports
- 1 6-bit I/O Port
- 4 Interrupt Levels
- Serial In/Serial Out Ports

This minimum system, using the standard I/O technique is as shown in Figure 7.

In addition to the standard I/O, the memory mapped I/O offers an efficient I/O addressing technique. With this technique, an area of memory address space is assigned for I/O address, thereby, using the memory address for I/O manipulation. Figure 8 shows the system configuration of Memory Mapped I/O using 8085AH.

The 8085AH CPU can also interface with the standard memory that does *not* have the multiplexed address/data bus. It will require a simple 8-bit latch as shown in Figure 9.



***NOTE:**
Optional Connection

Figure 7. 8085AH Minimum System (Standard I/O Technique)

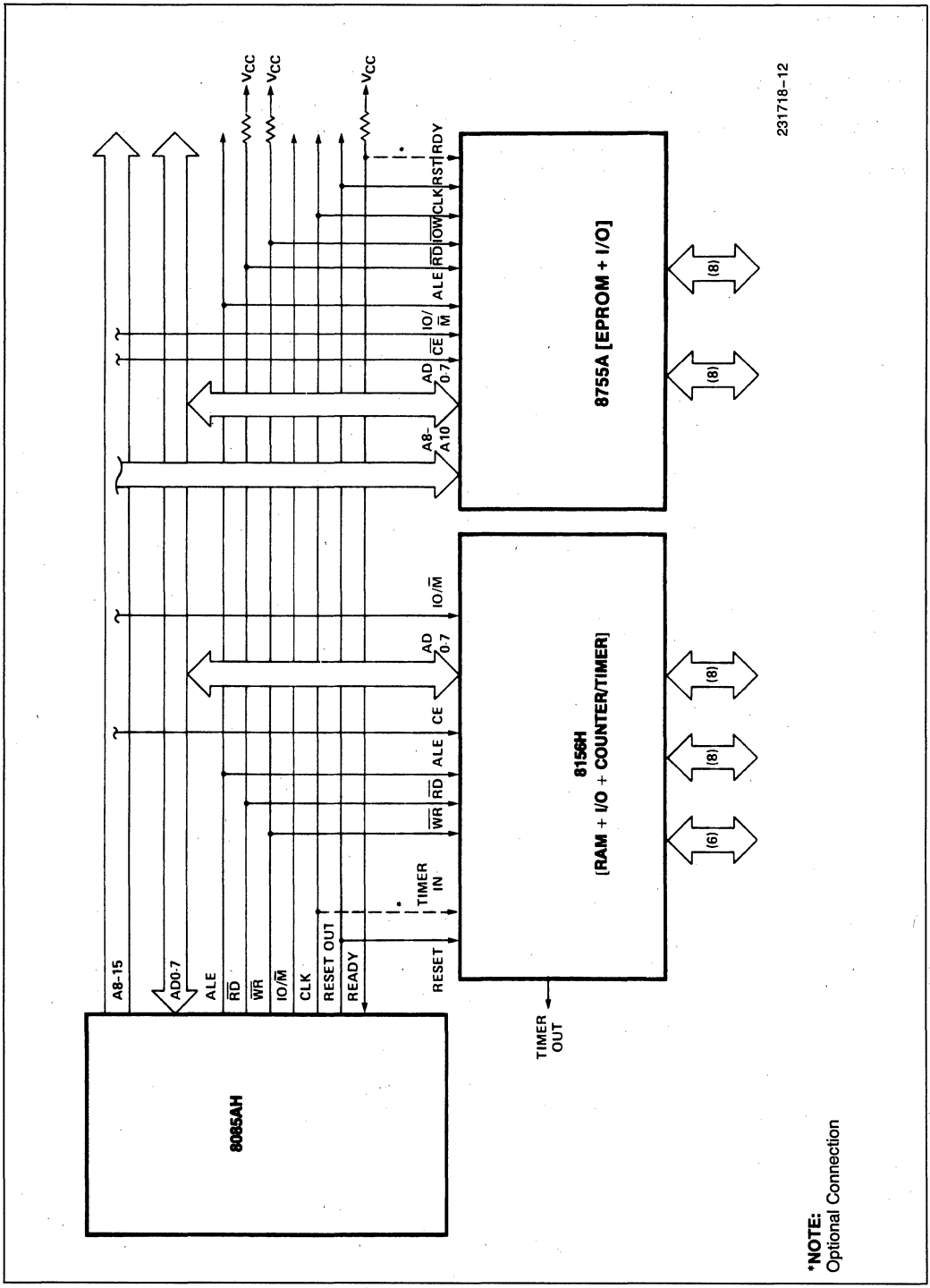


Figure 8. 8085 Minimum System (Memory Mapped I/O)

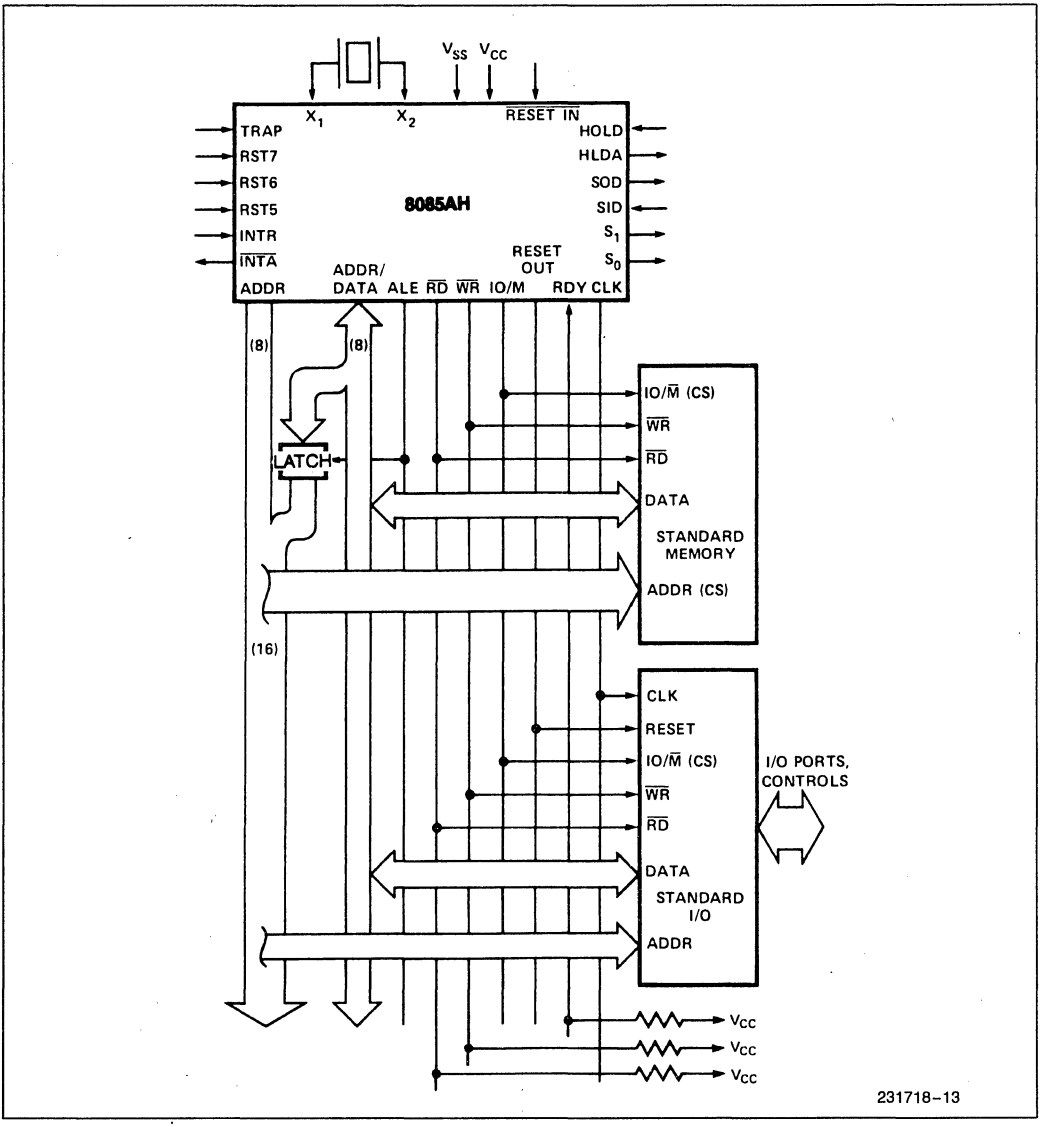


Figure 9. 8085 System (Using Standard Memories)

BASIC SYSTEM TIMING

The 8085AH has a multiplexed Data Bus. ALE is used as a strobe to sample the lower 8-bits of address on the Data Bus. Figure 10 shows an instruction fetch, memory read and I/O write cycle (as would occur during processing of the OUT instruction). Note that during the I/O write and read cycle that the I/O port address is copied on both the upper and lower half of the address.

There are seven possible types of machine cycles. Which of these seven takes place is defined by the status of the three status lines ($IO/\overline{M}$, S_1, S_0) and the three control signals ($\overline{RD}$, $\overline{WR}$, and $\overline{INTA}$). (See Table 3.) The status lines can be used as advanced controls (for device selection, for example), since they become active at the T_1 state, at the outset of each machine cycle. Control lines $\overline{RD}$ and $\overline{WR}$ become active later, at the time when the transfer of data is to take place, so are used as command lines.

A machine cycle normally consists of three T states, with the exception of OPCODE FETCH, which normally has either four or six T states (unless $\overline{WAIT}$ or HOLD states are forced by the receipt of $\overline{READY}$ or HOLD inputs). Any T state must be one of ten possible states, shown in Table 4.

Table 3. 8085AH Machine Cycle Chart

Machine Cycle			Status			Control		
			IO/$\overline{M}$	S1	S0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
OPCODE FETCH	(OF)		0	1	1	0	1	1
MEMORY READ	(MR)		0	1	0	0	1	1
MEMORY WRITE	(MW)		0	0	1	1	0	1
I/O READ	(IOR)		1	1	0	0	1	1
I/O WRITE	(IOW)		1	0	1	1	0	1
ACKNOWLEDGE OF INTR	(INA)		1	1	1	1	1	0
BUS IDLE	(BI):	DAD	0	1	0	1	1	1
		ACK.OF RST,TRAP	1	1	1	1	1	1
		HALT	TS	0	0	TS	TS	1

Table 4. 8085AH Machine State Chart

Machine State	Status & Buses				Control		
	S1,S0	IO/$\overline{M}$	A_8–A_{15}	AD_0–AD_7	$\overline{RD}$, $\overline{WR}$	$\overline{INTA}$	ALE
T_1	X	X	X	X	1	1	1*
T_2	X	X	X	X	X	X	0
T_{WAIT}	X	X	X	X	X	X	0
T_3	X	X	X	X	X	X	0
T_4	1	0†	X	TS	1	1	0
T_5	1	0†	X	TS	1	1	0
T_6	1	0†	X	TS	1	1	0
T_{RESET}	X	TS	TS	TS	TS	1	0
T_{HALT}	0	TS	TS	TS	TS	1	0
T_{HOLD}	X	TS	TS	TS	TS	1	0

0 = Logic "0" TS = High Impedance
1 = Logic "1" X = Unspecified
*ALE not generated during 2nd and 3rd machine cycles of DAD instruction.
†$IO/\overline{M}$ = 1 during T_4–T_6 of INA machine cycle.

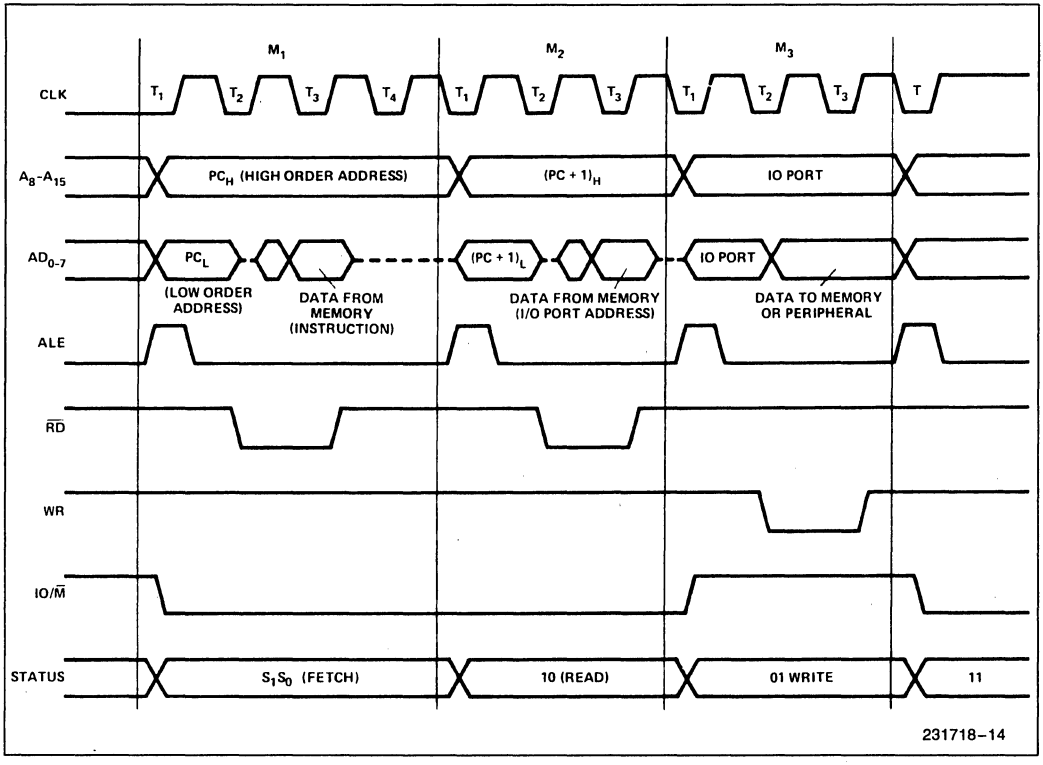


Figure 10. 8085AH Basic System Timing

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin
with Respect to Ground.....−0.5V to +7V
Power Dissipation.....1.5W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

8085AH, 8085AH-2: T_A = 0°C to 70°C, V_{CC} = 5V ±10%, V_{SS} = 0V; unless otherwise specified*
8085AH-1: T_A = 0°C to 70°C, V_{CC} = 5V ±5%, V_{SS} = 0V; unless otherwise specified*

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} +0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{CC}	Power Supply Current		135	mA	8085AH, 8085AH-2
			200	mA	8085AH-1
I_{IL}	Input Leakage		±10	μA	$0 \le V_{IN} \le V_{CC}$
I_{LO}	Output Leakage		±10	μA	$0.45V \le V_{OUT} \le V_{CC}$
V_{ILR}	Input Low Level, RESET	−0.5	+0.8	V	
V_{IHR}	Input High Level, RESET	2.4	V_{CC} + 0.5	V	
V_{HY}	Hysteresis, RESET	0.15		V	

A.C. CHARACTERISTICS

8085AH, 8085AH-2: T_A = 0°C to 70°C, V_{CC} = 5V ±10%, V_{SS} = 0V*
8085AH-1: T_A = 0°C to 70°C, V_{CC} = 5V ±5%, V_{SS} = 0V

Symbol	Parameter	8085AH (2)		8085AH-2 (2)		8085AH-1 (2)		Units
		Min	Max	Min	Max	Min	Max	
t_{CYC}	CLK Cycle Period	320	2000	200	2000	167	2000	ns
t_1	CLK Low Time (Standard CLK Loading)	80		40		20		ns
t_2	CLK High Time (Standard CLK Loading)	120		70		50		ns
t_r, t_f	CLK Rise and Fall Time		30		30		30	ns
t_{XKR}	X_1 Rising to CLK Rising	20	120	20	100	20	100	ns
t_{XKF}	X_1 Rising to CLK Falling	20	150	20	110	20	110	ns
t_{AC}	A_{8-15} Valid to Leading Edge of Control (1)	270		115		70		ns
t_{ACL}	A_{0-7} Valid to Leading Edge of Control	240		115		60		ns
t_{AD}	A_{0-15} Valid to Valid Data In		575		350		225	ns
t_{AFR}	Address Float after Leading Edge of $\overline{READ}$ ($\overline{INTA}$)		0		0		0	ns
t_{AL}	A_{8-15} Valid before Trailing Edge of ALE (1)	115		50		25		ns

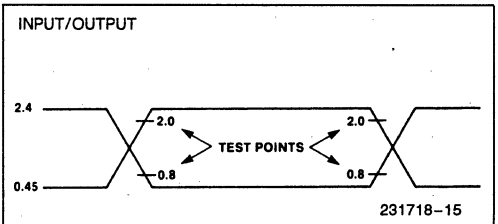
***NOTE:**
For Extended Temperature EXPRESS use M8085AH Electricals Parameters.

A.C. CHARACTERISTICS (Continued)

Symbol	Parameter	8085AH (2)		8085AH-2 (2)		8085AH-1 (2)		Units
		Min	Max	Min	Max	Min	Max	
t_{ALL}	A_{0-7} Valid before Trailing Edge of ALE	90		50		25		ns
t_{ARY}	READY Valid from Address Valid		220		100		40	ns
t_{CA}	Address (A_{8-15}) Valid after Control	120		60		30		ns
t_{CC}	Width of Control Low ($\overline{RD}$, $\overline{WR}$, $\overline{INTA}$) Edge of ALE	400		230		150		ns
t_{CL}	Trailing Edge of Control to Leading Edge of ALE	50		25		0		ns
t_{DW}	Data Valid to Trialing Edge of $\overline{WRITE}$	420		230		140		ns
t_{HABE}	HLDA to Bus Enable		210		150		150	ns
t_{HABF}	Bus Float after HLDA		210		150		150	ns
t_{HACK}	HLDA Valid to Trailing Edge of CLK	110		40		0		ns
t_{HDH}	HOLD Hold Time	0		0		0		ns
t_{HDS}	HOLD Setup Time to Trailing Edge of CLK	170		120		120		ns
t_{INH}	INTR Hold Time	0		0		0		ns
t_{INS}	INTR, RST, and TRAP Setup Time to Falling Edge of CLK	160		150		150		ns
t_{LA}	Address Hold Time after ALE	100		50		20		ns
t_{LC}	Trailing Edge of ALE to Leading Edge of Control	130		60		25		ns
t_{LCK}	ALE Low During CLK High	100		50		15		ns
t_{LDR}	ALE to Valid Data during Read		460		270		175	ns
t_{LDW}	ALE to Valid Data during Write		200		140		110	ns
t_{LL}	ALE Width	140		80		50		ns
t_{LRY}	ALE to READY Stable		110		30		10	ns
t_{RAE}	Trailing Edge of $\overline{READ}$ to Re-Enabling of Address	150		90		50		ns
t_{RD}	$\overline{READ}$ (or $\overline{INTA}$) to Valid Data		300		150		75	ns
t_{RV}	Control Trailing Edge to Leading Edge of Next Control	400		220		160		ns
t_{RDH}	Data Hold Time after $\overline{READ}$ $\overline{INTA}$	0		0		0		ns
t_{RYH}	READY Hold Time	0		0		5		ns
t_{RYS}	READY Setup Time to Leading Edge of CLK	110		100		100		ns
t_{WD}	Data Valid after Trailing Edge of $\overline{WRITE}$	100		60		30		ns
t_{WDL}	LEADING Edge of $\overline{WRITE}$ to Data Valid		40		20		30	ns

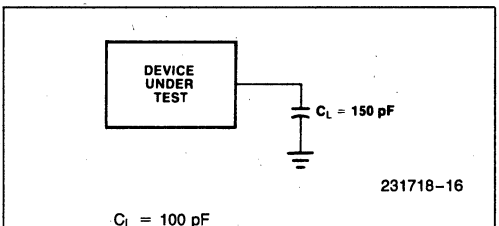
NOTES:
1. A_8–A_{15} address Specs apply IO/$\overline{M}$, S_0, and S_1 except A_8–A_{15} are undefined during T_4–T_6 of OF cycle whereas IO/$\overline{M}$, S_0, and S_1 are stable.
2. *Test Conditions:* t_{CYC} = 320 ns (8085AH)/200 ns (8085AH-2);/167 ns (8085AH-1); C_L = 150 pF.
3. For all output timing where C ≠ 150 pF use the following correction factors:
   25 pF ≤ C_L < 150 pF: −0.10 ns/pF
   150 pF < C_L ≤ 300 pF: +0.30 ns/pF
4. Output timings are measured with purely capacitive load.
5. To calculate timing specifications at other values of t_{CYC} use Table 5.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L Includes Jig Capacitance

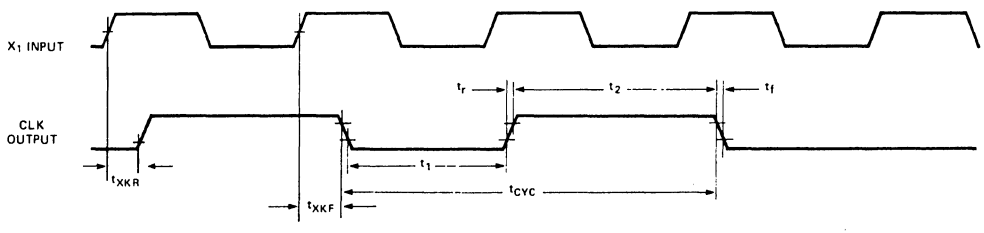
Table 5. Bus Timing Specification as a T_{CYC} Dependent

Symbol	8085AH	8085AH-2	8085AH-1	
t_{AL}	(1/2)T − 45	(1/2)T − 50	(1/2)T − 58	Minimum
t_{LA}	(1/2)T − 60	(1/2)T − 50	(1/2)T − 63	Minimum
t_{LL}	(1/2)T − 20	(1/2)T − 20	(1/2)T − 33	Minimum
t_{LCK}	(1/2)T − 60	(1/2)T − 50	(1/2)T − 68	Minimum
t_{LC}	(1/2)T − 30	(1/2)T − 40	(1/2)T − 58	Minimum
t_{AD}	(5/2 + N)T − 225	(5/2 + N)T − 150	(5/2 + N)T − 192	Maximum
t_{RD}	(3/2 + N)T − 180	(3/2 + N)T − 150	(3/2 + N)T − 175	Maximum
t_{RAE}	(1/2)T − 10	(1/2)T − 10	(1/2)T − 33	Minimum
t_{CA}	(1/2)T − 40	(1/2)T − 40	(1/2)T − 53	Minimum
t_{DW}	(3/2 + N)T − 60	(3/2 + N)T − 70	(3/2 + N)T − 110	Minimum
t_{WD}	(1/2)T − 60	(1/2)T − 40	(1/2)T − 53	Minimum
t_{CC}	(3/2 + N)T − 80	(3/2 + N)T − 70	(3/2 + N)T − 100	Minimum
t_{CL}	(1/2)T − 110	(1/2)T − 75	(1/2)T − 83	Minimum
t_{ARY}	(3/2)T − 260	(3/2)T − 200	(3/2)T − 210	Maximum
t_{HACK}	(1/2)T − 50	(1/2)T − 60	(1/2)T − 83	Minimum
t_{HABF}	(1/2)T + 50	(1/2)T + 50	(1/2)T + 67	Maximum
t_{HABE}	(1/2)T + 50	(1/2)T + 50	(1/2)T + 67	Maximum
t_{AC}	(2/2)T − 50	(2/2)T − 85	(2/2)T − 97	Minimum
t_1	(1/2)T − 80	(1/2)T − 60	(1/2)T − 63	Minimum
t_2	(1/2)T − 40	(1/2)T − 30	(1/2)T − 33	Minimum
t_{RV}	(3/2)T − 80	(3/2)T − 80	(3/2)T − 90	Minimum
t_{LDR}	(4/2 + N)T − 180	(4/2)T − 130	(4/2)T − 159	Maximum

NOTE:
N is equal to the total WAIT states. T = t_{CYC}.

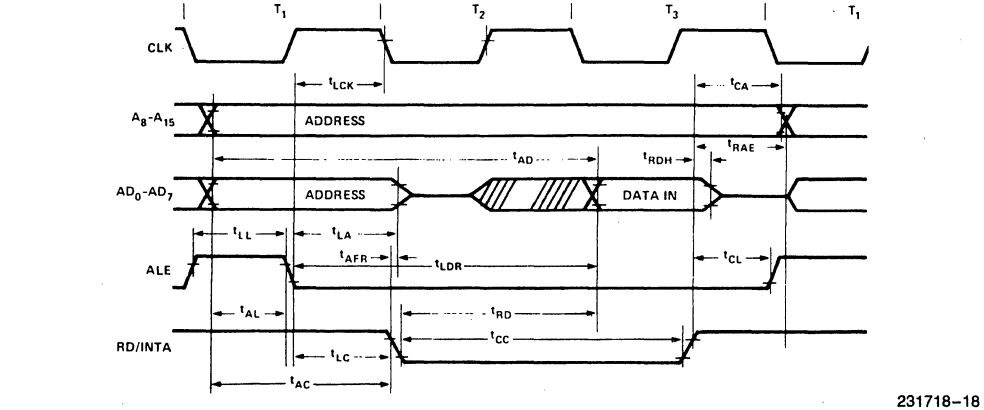
WAVEFORMS

CLOCK



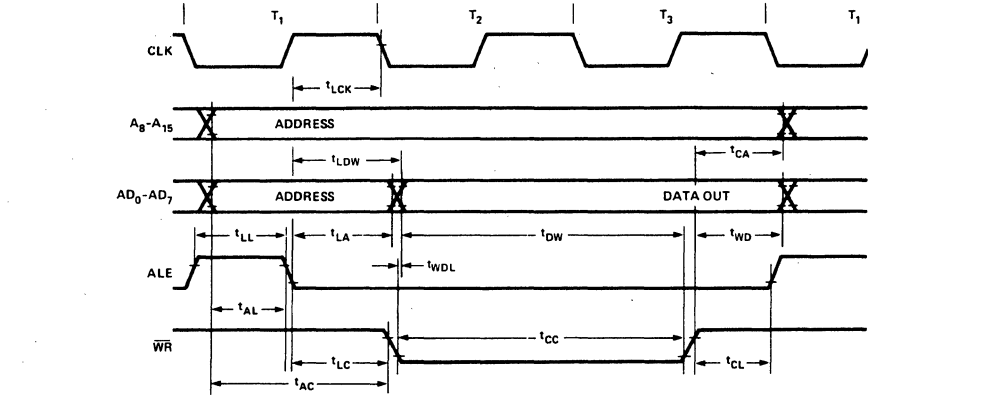
231718-17

READ



231718-18

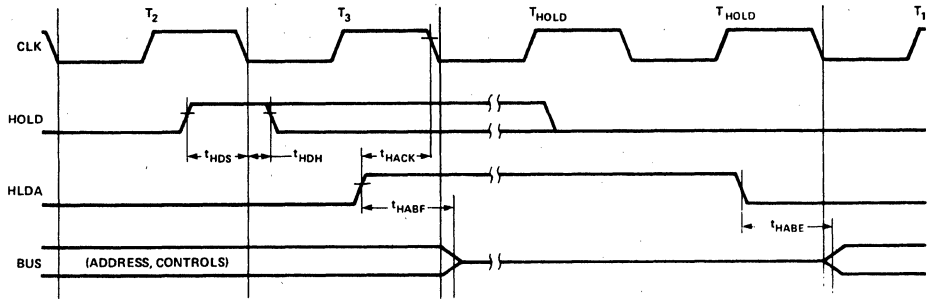
WRITE



231718-19

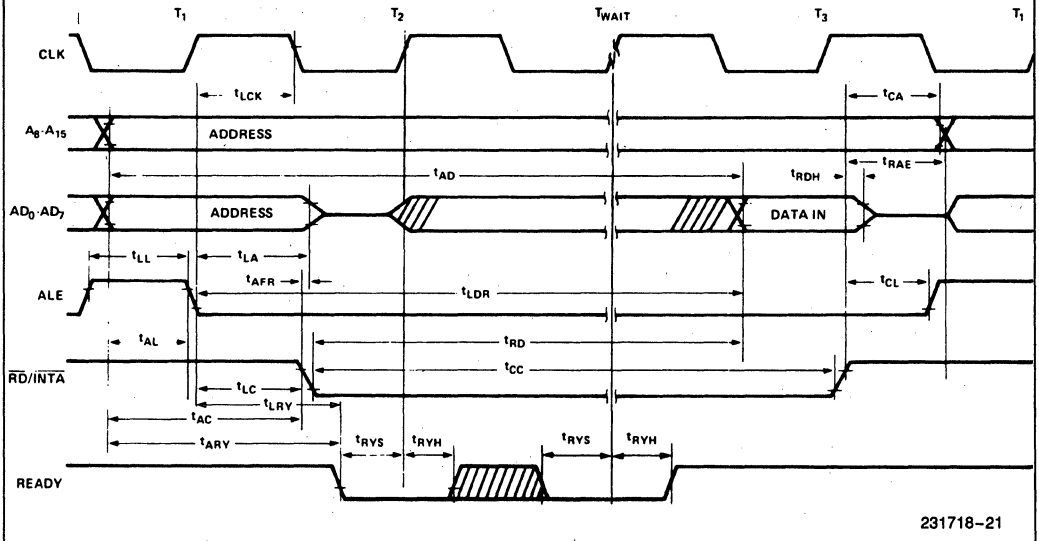
WAVEFORMS (Continued)

HOLD



231718-20

READ OPERATION WITH WAIT CYCLE (TYPICAL)—SAME READY TIMING APPLIES TO WRITE

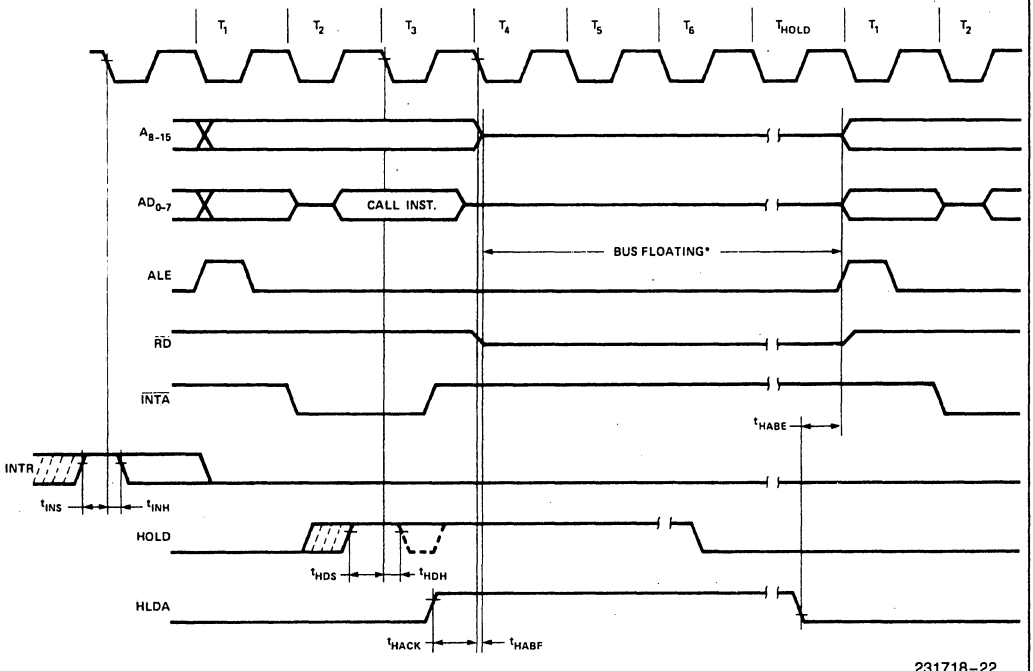


231718-21

NOTE:
1. Ready must remain stable during setup and hold times.

WAVEFORMS (Continued)

INTERRUPT AND HOLD



***NOTE:**
IO/$\overline{M}$ is also floating during this time.

Table 6. Instruction Set Summary

Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description
MOVE, LOAD AND STORE		
MOVr1 r2	0 1 D D D S S S	Move register to register
MOV M.r	0 1 1 1 0 S S S	Move register to memory
MOV r.M	0 1 D D D 1 1 0	Move memory to register
MVI r	0 0 D D D 1 1 0	Move immediate register
MVI M	0 0 1 1 0 1 1 0	Move immediate memory
LXI B	0 0 0 0 0 0 0 1	Load immediate register Pair B & C
LXI D	0 0 0 1 0 0 0 1	Load immediate register Pair D & E
LXI H	0 0 1 0 0 0 0 1	Load immediate register Pair H & L
STAX B	0 0 0 0 0 0 1 0	Store A indirect
STAX D	0 0 0 1 0 0 1 0	Store A indirect
LDAX B	0 0 1 0 1 0 1 0	Load A indirect
LDAX D	0 0 0 1 1 0 1 0	Load A indirect
STA	0 0 1 1 0 0 1 0	Store A direct
LDA	0 0 1 1 1 0 1 0	Load A direct
SHLD	0 0 1 0 0 0 1 0	Store H & L direct
LHLD	0 0 1 0 1 0 1 0	Load H & L direct
XCHG	1 1 1 0 1 0 1 1	Exchange D & E, H & L Registers
STACK OPS		
PUSH B	1 1 0 0 0 1 0 1	Push register Pair B & C on stack
PUSH D	1 1 0 1 0 1 0 1	Push register Pair D & E on stack
PUSH H	1 1 1 0 0 1 0 1	Push register Pair H & L on stack
PUSH PSW	1 1 1 1 0 1 0 1	Push A and Flags on stack
POP B	1 1 0 0 0 0 0 1	Pop register Pair B & C off stack
POP D	1 1 0 1 0 0 0 1	Pop register Pair D & E off stack
POP H	1 1 1 0 0 0 0 1	Pop register Pair H & L off stack

Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description
STACK OPS (Continued)		
POP PSW	1 1 1 1 0 0 0 1	Pop A and Flags off stack
XTHL	1 1 1 0 0 0 1 1	Exchange top of stack, H & L
SPHL	1 1 1 1 1 0 0 1	H & L to stack pointer
LXI SP	0 0 1 1 0 0 0 1	Load immediate stack pointer
INX SP	0 0 1 1 0 0 1 1	Increment stack pointer
DCX SP	0 0 1 1 1 0 1 1	Decrement stack pointer
JUMP		
JMP	1 1 0 0 0 0 1 1	Jump unconditional
JC	1 1 0 1 1 0 1 0	Jump on carry
JNC	1 1 0 1 0 0 1 0	Jump on no carry
JZ	1 1 0 0 1 0 1 0	Jump on zero
JNZ	1 1 0 0 0 0 1 0	Jump on no zero
JP	1 1 1 1 0 0 1 0	Jump on positive
JM	1 1 1 1 1 0 1 0	Jump on minus
JPE	1 1 1 0 1 0 1 0	Jump on parity even
JPO	1 1 1 0 0 0 1 0	Jump on parity odd
PCHL	1 1 1 0 1 0 0 1	H & L to program counter
CALL		
CALL	1 1 0 0 1 1 0 1	Call unconditional
CC	1 1 0 1 1 1 0 0	Call on carry
CNC	1 1 0 1 0 1 0 0	Call on no carry
CZ	1 1 0 0 1 1 0 0	Call on zero
CNZ	1 1 0 0 0 1 0 0	Call on no zero
CP	1 1 1 1 0 1 0 0	Call on positive
CM	1 1 1 1 1 1 0 0	Call on minus
CPE	1 1 1 0 1 1 0 0	Call on parity even
CPO	1 1 1 0 0 1 0 0	Call on parity odd
RETURN		
RET	1 1 0 0 1 0 0 1	Return
RC	1 1 0 1 1 0 0 0	Return on carry
RNC	1 1 0 1 0 0 0 0	Return on no carry
RZ	1 1 0 0 1 0 0 0	Return on zero

Table 6. Instruction Set Summary (Continued)

Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description
RETURN (Continued)		
RNZ	1 1 0 0 0 0 0 0	Return on no zero
RP	1 1 1 1 0 0 0 0	Return on positive
RM	1 1 1 1 1 0 0 0	Return on minus
RPE	1 1 1 0 1 0 0 0	Return on parity even
RPO	1 1 1 0 0 0 0 0	Return on parity odd
RESTART		
RST	1 1 A A A 1 1 1	Restart
INPUT/OUTPUT		
IN	1 1 0 1 1 0 1 1	Input
OUT	1 1 0 1 0 0 1 1	Output
INCREMENT AND DECREMENT		
INR r	0 0 D D D 1 0 0	Increment register
DCR r	0 0 D D D 1 0 1	Decrement register
INR M	0 0 1 1 0 1 0 0	Increment memory
DCR M	0 0 1 1 0 1 0 1	Decrement memory
INX B	0 0 0 0 0 0 1 1	Increment B & C registers
INX D	0 0 0 1 0 0 1 1	Increment D & E registers
INX H	0 0 1 0 0 0 1 1	Increment H & L registers
DCX B	0 0 0 0 1 0 1 1	Decrement B & C
DCX D	0 0 0 1 1 0 1 1	Decrement D & E
DCX H	0 0 1 0 1 0 1 1	Decrement H & L
ADD		
ADD r	1 0 0 0 0 S S S	Add register to A
ADC r	1 0 0 0 1 S S S	Add register to A with carry
ADD M	1 0 C 0 0 1 1 0	Add memory to A
ADC M	1 0 0 0 1 1 1 0	Add memory to A with carry
ADI	1 1 0 0 0 1 1 0	Add immediate to A
ACI	1 1 0 0 1 1 1 0	Add immediate to A with carry
DAD B	0 0 0 0 1 0 0 1	Add B & C to H & L

Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description
ADD (Continued)		
DAD D	0 0 0 1 1 0 0 1	Add D & E to H & L
DAD H	0 0 1 0 1 0 0 1	Add H & L to H & L
DAD SP	0 0 1 1 1 0 0 1	Add stack pointer to H & L
SUBTRACT		
SUB r	1 0 0 1 0 S S S	Subtract register from A
SBB r	1 0 0 1 1 S S S	Subtract register from A with borrow
SUB M	1 0 0 1 0 1 1 0	Subtract memory from A
SBB M	1 0 0 1 1 1 1 0	Subtract memory from A with borrow
SUI	1 1 0 1 0 1 1 0	Subtract immediate from A
SBI	1 1 0 1 1 1 1 0	Subtract immediate from A with borrow
LOGICAL		
ANA r	1 0 1 0 0 S S S	And register with A
XRA r	1 0 1 0 1 S S S	Exclusive OR register with A
ORA r	1 0 1 1 0 S S S	OR register with A
CMP r	1 0 1 1 1 S S S	Compare register with A
ANA M	1 0 1 0 0 1 1 0	And memory with A
XRA M	1 0 1 0 1 1 1 0	Exclusive OR memory with A
ORA M	1 0 1 1 0 1 1 0	OR memory with A
CMP M	1 0 1 1 1 1 1 0	Compare memory with A
ANI	1 1 1 0 0 1 1 0	And immediate with A
XRI	1 1 1 0 1 1 1 0	Exclusive OR immediate with A
ORI	1 1 1 1 0 1 1 0	OR immediate with A
CPI	1 1 1 1 1 1 1 0	Compare immediate with A

Table 6. Instruction Set Summary (Continued)

Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description
ROTATE		
RLC	0 0 0 0 0 1 1 1	Rotate A left
RRC	0 0 0 0 1 1 1 1	Rotate A right
RAL	0 0 0 1 0 1 1 1	Rotate A left through carry
RAR	0 0 0 1 1 1 1 1	Rotate A right through carry
SPECIALS		
CMA	0 0 1 0 1 1 1 1	Complement A
STC	0 0 1 1 0 1 1 1	Set carry
CMC	0 0 1 1 1 1 1 1	Complement carry
DAA	0 0 1 0 0 1 1 1	Decimal adjust A

Mnemonic	Instruction Code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Operations Description
CONTROL		
EI	1 1 1 1 1 0 1 1	Enable Interrupts
DI	1 1 1 1 0 0 1 1	Disable Interrupt
NOP	0 0 0 0 0 0 0 0	No-operation
HLT	0 1 1 1 0 1 1 0	Halt
NEW 8085AH INSTRUCTIONS		
RIM	0 0 1 0 0 0 0 0	Read Interrupt Mask
SIM	0 0 1 1 0 0 0 0	Set Interrupt Mask

NOTES:
1. DDS or SSS: B 000, C 001, D 010, E011, H 100, L101, Memory 110, A 111.
2. Two possible cycle times (6/12) indicate instruction cycles dependent on condition flags.

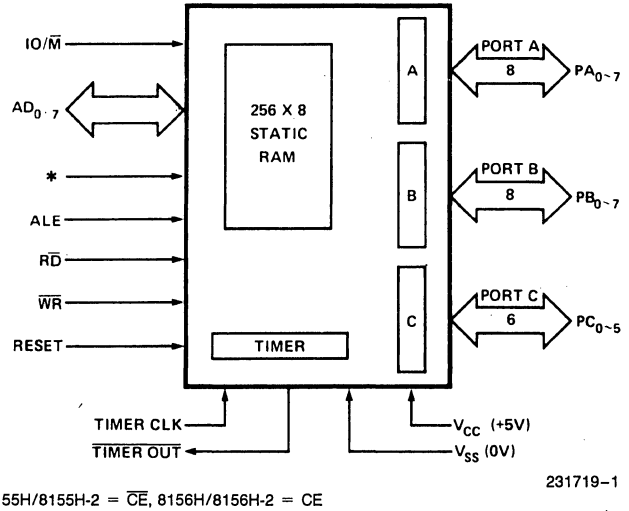
8155H/8156H/8155H-2/8156H-2
2048-BIT STATIC HMOS RAM WITH I/O PORTS AND TIMER

- **Single +5V Power Supply with 10% Voltage Margins**
- **30% Lower Power Consumption than the 8155 and 8156**
- **256 Word x 8 Bits**
- **Completely Static Operation**
- **Internal Address Latch**
- **2 Programmable 8-Bit I/O Ports**
- **1 Programmable 6-Bit I/O Port**
- **Programmable 14-Bit Binary Counter/ Timer**
- **Compatible with 8085AH and 8088 CPU**
- **Multiplexed Address and Data Bus**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

The Intel® 8155H and 8156H are RAM and I/O chips implemented in N-Channel, depletion load, silicon gate technology (HMOS), to be used in the 8085AH and 8088 microprocessor systems. The RAM portion is designed with 2048 static cells organized as 256 x 8. They have a maximum access time of 400 ns to permit use with no wait states in 8085AH CPU. The 8155H-2 and 8156H-2 have maximum access times of 330 ns for use with the 8085H-2 and the 5 MHz 8088 CPU.

The I/O portion consists of three general purpose I/O ports. One of the three ports can be programmed to be status pins, thus allowing the other two ports to operate in handshake mode.

A 14-bit programmable counter/timer is also included on chip to provide either a square wave or terminal count pulse for the CPU system depending on timer mode.



*8155H/8155H-2 = $\overline{CE}$, 8156H/8156H-2 = CE

Figure 1. Block Diagram

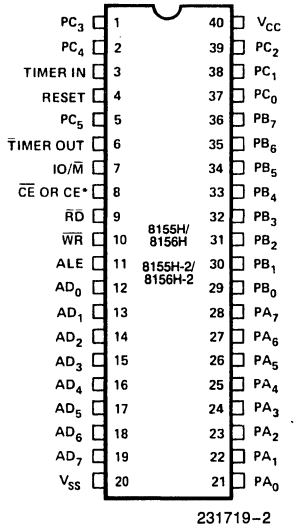


Figure 2. Pin Configuration

December 1986
Order Number: 231719-001

Table 1. Pin Description

Symbol	Type	Name and Function
RESET	I	**RESET:** Pulse provided by the 8085AH to initialize the system (connect to 8085AH RESET OUT). Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be two 8085AH clock cycle times.
AD_{0-7}	I/O	**ADDRESS/DATA:** 3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latch inside the 8155H/56H on the falling edge of ALE. The address can be either for the memory section or the I/O section depending on the IO/$\overline{M}$ input. The 8-bit data is either written into the chip or read from the chip, depending on the $\overline{WR}$ or $\overline{RD}$ input signal.
CE or $\overline{CE}$	I	**CHIP ENABLE:** On the 8155H, this pin is $\overline{CE}$ and is ACTIVE LOW. On the 8156H, this pin is CE and is ACTIVE HIGH.
$\overline{RD}$	I	**READ CONTROL:** Input low on this line with the Chip Enable active enables and AD_{0-7} buffers. If IO/$\overline{M}$ pin is low, the RAM content will be read out to the AD bus. Otherwise the content of the selected I/O port or command/status registers will be read to the AD bus.
$\overline{WR}$	I	**WRITE CONTROL:** Input low on this line with the Chip Enable active causes the data on the Address/Data bus to be written to the RAM or I/O ports and command/status register, depending on IO/$\overline{M}$.
ALE	I	**ADDRESS LATCH ENABLE:** This control signal latches both the address on the AD_{0-7} lines and the state of the Chip Enable and IO/$\overline{M}$ into the chip at the falling edge of ALE.
IO/$\overline{M}$	I	**I/O MEMORY:** Selects memory if low and I/O and command/status registers if high.
PA_{0-7} (8)	I/O	**PORT A:** These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
PB_{0-7} (8)	I/O	**PORT B:** These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
PC_{0-5} (6)	I/O	**PORT C:** These 6 pins can function as either input port, output port, or as control signals for PA and PB. Programming is done through the command register. When PC_{0-5} are used as control signals, they will provide the following: PC_0—A INTR (Port A Interrupt) PC_1—ABF (Port A Buffer Full) PC_2—$\overline{A\ STB}$ (Port A Strobe) PC_3—B INTR (Port B Interrupt) PC_4—B BF (Port B Buffer Full) PC_5—$\overline{B\ STB}$ (Port B Strobe)
TIMER IN	I	**TIMER INPUT:** Input to the timer-counter.
$\overline{TIMER\ OUT}$	O	**TIMER OUTPUT:** This output can be either a square wave or a pulse, depending on the timer mode.
V_{CC}		**VOLTAGE:** +5V supply.
V_{SS}		**GROUND:** Ground reference.

FUNCTIONAL DESCRIPTION

The 8155H/8156H contains the following:

- 2K Bit Static RAM organized as 256 x 8
- Two 8-bit I/O ports (PA & PB) and one 6-bit I/O port (PC)
- 14-bit timer-counter

The IO/$\overline{M}$ (IO/Memory Select) pin selects either the five registers (Command, Status, PA_{0-7}, PB_{0-7}, PC_{0-5}) or the memory (RAM) portion.

The 8-bit address on the Address/Data lines, Chip Enable input CE or $\overline{CE}$, and IO/$\overline{M}$ are all latched on-chip at the falling edge of ALE.

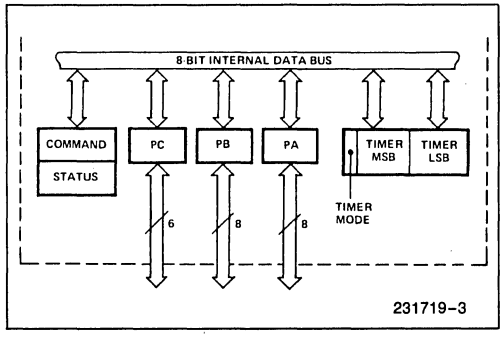
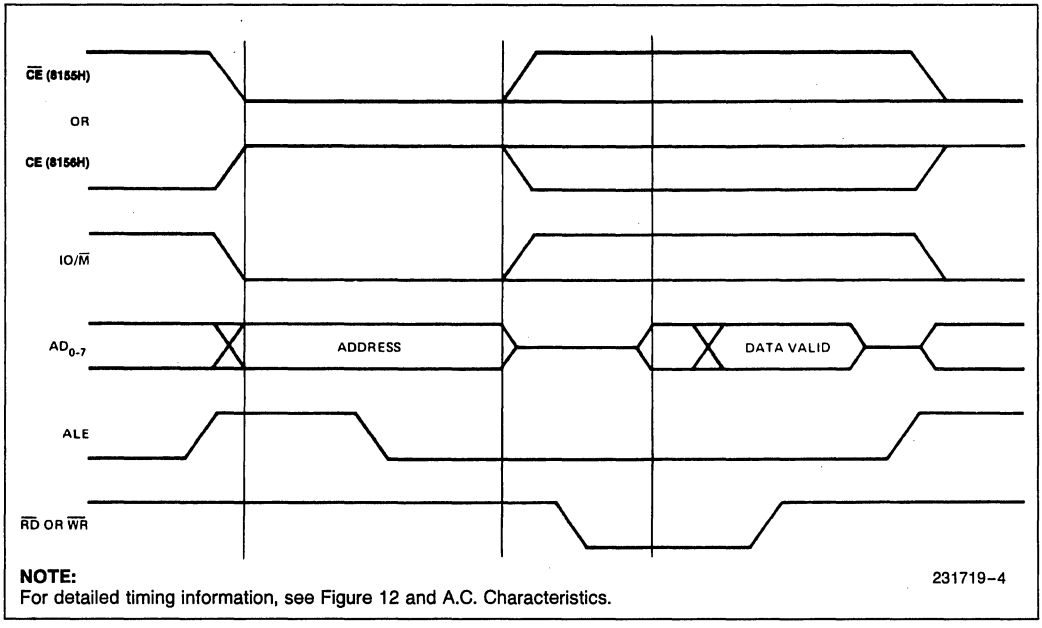


Figure 3. 8155H/8156H Internal Registers



NOTE:
For detailed timing information, see Figure 12 and A.C. Characteristics.

Figure 4. 8155H/8156H On-Board Memory Read/Write Cycle

PROGRAMMING OF THE COMMAND REGISTER

The command register consists of eight latches. Four bits (0–3) define the mode of the ports, two bits (4–5) enable or disable the interrupt from port C when it acts as control port, and the last two bits (6–7) are for the timer.

The command register contents can be altered at any time by using the I/O address XXXXX000 during a WRITE operation with the Chip Enable active and IO/$\overline{M}$ = 1. The meaning of each bit of the command byte is defined in Figure 5. The contents of the command register may never be read.

READING THE STATUS REGISTER

The status register consists of seven latches, one for each bit; six (0–5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the Status Register (Address XXXXX000). Status word format is shown in Figure 6. Note that you may never write to the status register since the command register shares the same I/O address and the command register is selected when a write to that address is issued.

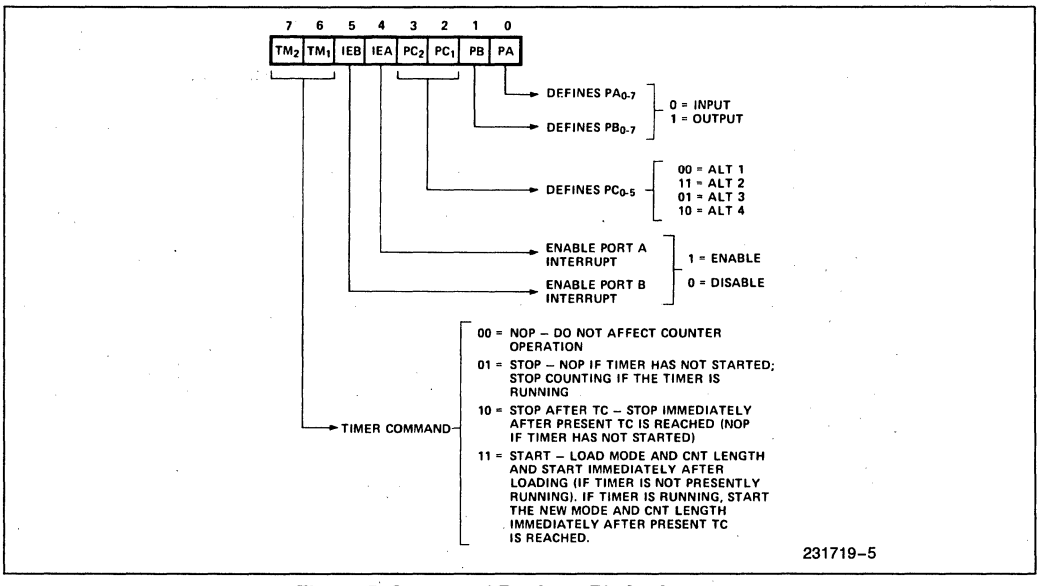


Figure 5. Command Register Bit Assignment

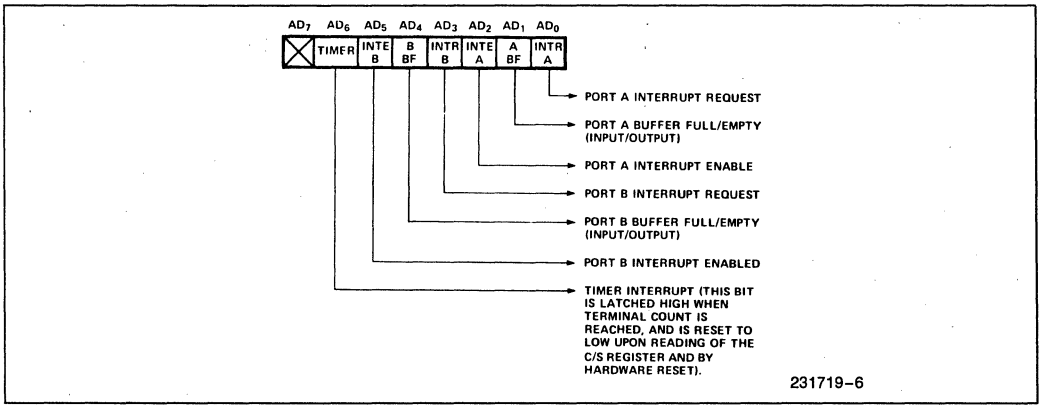


Figure 6. Status Register Bit Assignment

INPUT/OUTPUT SECTION

The I/O section of the 8155H/8156H consists of five registers: (see Figure 7.)

- **Command/Status Register (C/S)**—Both registers are assigned the address XXXXX000. The C/S address serves the dual purpose.
  When the C/S registers are selected during WRITE operation, a command is written into the command register. The contents of this register are *not* accessible through the pins.
  When the C/S (XXXXX000) is selected during a READ operation, the status information of the I/O ports and the timer becomes available on the AD_{0-7} lines.
- **PA Register**—This register can be programmed to be either input or output ports depending on the status of the contents of the C/S Register. Also depending on the command, this port can operate in either the basic mode or the strobed mode (see timing diagram). The I/O pins assigned in relation to this register are PA_{0-7}. The address of this register is XXXXX001.
- **PB Register**—This register functions the same as PA Register. The I/O pins assigned are PB_{0-7}. The address of this register is XXXXX010.
- **PC Register**—This register has the address XXXXX011 and contains only 6 bits. The 6 bits can be programmed to be either input ports, output ports or as control signals for PA and PB by properly programming the AD_2 and AD_3 bits of the C/S register.
  When PC_{0-5} is used as a control port, 3 bits are assigned for Port A and 3 for Port B. The first bit is an interrupt that the 8155H sends out. The second is an output signal indicating whether the buffer is full or empty, and the third is an input pin to accept a strobe for the strobed input mode. (See Table 2.)

When the 'C' port is programmed to either ALT3 or ALT4, the control signals for PA and PB are initialized as follows:

Control	Input Mode	Output Mode
BF	Low	Low
INTR	Low	High
STB	Input Control	Input Control

I/O Address†								Selection
A7	A6	A5	A4	A3	A2	A1	A0	
X	X	X	X	X	0	0	0	Interval Command/Status Register
X	X	X	X	X	0	0	1	General Purpose I/O Port A
X	X	X	X	X	0	1	0	General Purpose I/O Port B
X	X	X	X	X	0	1	1	Port C—General Purpose I/O or Control
X	X	X	X	X	1	0	0	Low-Order 8 bits of Timer Count
X	X	X	X	X	1	0	1	High 6 bits of Timer Count and 2 bits of Timer Mode

X: Don't Care.
†: I/O Address must be qualified by CE = 1 (8156H) or $\overline{CE}$ = 0 (8155H) and IO/$\overline{M}$ = 1 in order to select the appropriate register.

Figure 7. I/O Port and Timer Addressing Scheme

Figure 8 shows how I/O PORTS A and B are structured within the 8155H and 8156H:

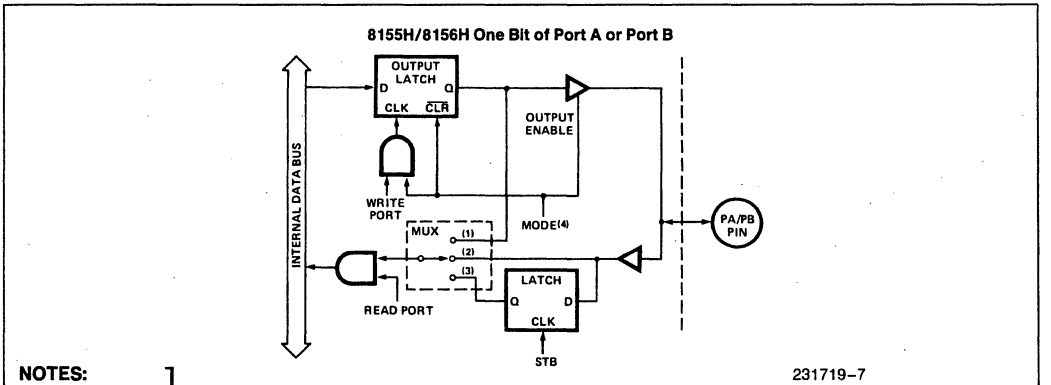


Figure 8. 8155H/8156H Port Functions

Table 2. Port Control Assignment

Pin	ALT 1	ALT 2	ALT 3	ALT 4
PC0	Input Port	Output Port	A INTR (Port A Interrupt)	A INTR (Port A Interrupt)
PC1	Input Port	Output Port	A BF (Port A Buffer Full)	A BF (Port A Buffer Full)
PC2	Input Port	Output Port	A $\overline{STB}$ (Port A Strobe)	A $\overline{STB}$ (Port A Strobe)
PC3	Input Port	Output Port	Output Port	B INTR (Port B Interrupt)
PC4	Input Port	Output Port	Output Port	B BF (Port B Buffer Full)
PC5	Input Port	Output Port	Output Port	B $\overline{STB}$ (Port B Strobe)

Note in the diagram that when the I/O ports are programmed to be output ports, the contents of the output ports can still be read by a READ operation when appropriately addressed.

The outputs of the 8155H/8156H are "glitch-free" meaning that you can write a "1" to a bit position that was previously "1" and the level at the output pin will not change.

Note also that the output latch is cleared when the port enters the input mode. The output latch cannot be loaded by writing to the port if the port is in the input mode. The result is that each time a port mode is changed from input to output, the output pin will go low. When the 8155H/56H is RESET, the output latches are all cleared and all 3 ports enter the input mode.

When in the ALT 1 or ALT 2 modes, the bits of PORT C are structured like the diagram above in the simple input or output mode, respectively.

Reading from an input port with nothing connected to the pins will provide unpredictable results.

Figure 9 shows how the 8155H/8156H I/O ports might be configured in a typical MCS®-85 system.

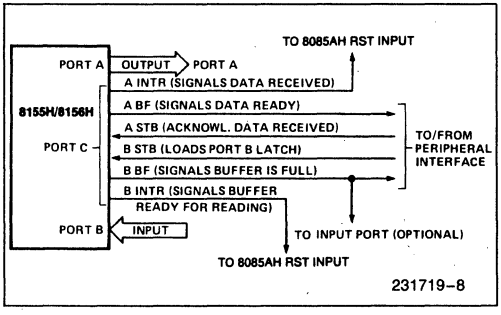


Figure 9. Example: Command Register = 00111001

TIMER SECTION

The time is a 14-bit down-counter that counts the TIMER IN pulses and provides either a square wave or pulse when terminal count (TC) is reached.

The timer has the I/O address XXXXX100 for the low order byte of the register and the I/O address XXXXX101 for the high order byte of the register. (See Figure 7.)

To program the timer, the COUNT LENGTH REG is loaded first, one byte at a time, by selecting the timer addresses. Bits 0–13 of the high order count register will specify the length of the next count and bits 14–15 of the high order register will specify the timer output mode (see Figure 10). The value loaded into the count length register can have any value from 2H through 3FFFH in Bits 0–13.

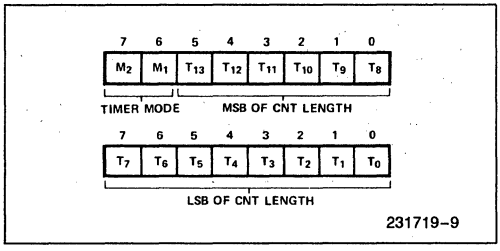


Figure 10. Timer Format

There are four modes to choose from: M2 and M1 define the timer mode, as shown in Figure 11.

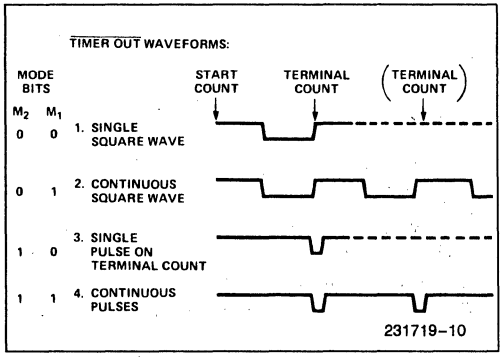


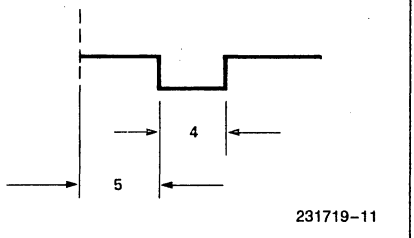
Figure 11. Timer Modes

Bits 6–7 (TM_2 and TM_1) of command register contents are used to start and stop the counter. There are four commands to choose from:

TM_2	TM_1	
0	0	NOP—Do not affect counter operation.
0	1	STOP—NOP if timer has not started; stop counting if the timer is running.
1	0	STOP AFTER TC—Stop immediately after present TC is reached (NOP if timer has not started)
1	1	START—Load mode and CNT length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and CNT length immediately after present TC is reached.

Note that while the counter is counting, you may load a new count and mode into the count length registers. Before the new count and mode will be used by the counter, you **must** issue a START command to the counter. This applies even though you may only want to change the count and use the previous mode.

In case of an odd-numbered count, the first half-cycle of the squarewave output, which is high, is one count longer than the second (low) half-cycle, as shown in Figure 12.



NOTE:
5 and 4 refer to the number of clocks in that time period.

Figure 12. Asymmetrical Square-Wave Output Resulting from Count of 9

The counter in the 8155H is not initialized to any particular mode or count when hardware RESET occurs, but RESET does *stop* the counting. Therefore, counting cannot begin following RESET until a START command is issued via the C/S register.

Please note that the timer circuit on the 8155H/8156H chip is designed to be a square-wave timer, not an event counter. To achieve this, it counts down by twos twice in completing one cycle. Thus, its registers do not contain values directly representing the number of TIMER IN pulses received. You cannot load an initial value of 1 into the count register and cause the timer to operate, as its terminal count value is 10 (binary) or 2 (decimal). (For the detection of single pulses, it is suggested that one of the hardware interrupt pins on the 8085AH be used.) After the timer has started counting down, the values residing in the count registers can be used to calculate the actual number of TIMER IN pulses required to complete the timer cycle if desired. To obtain the remaining count, perform the following operations in order:

1. Stop the count
2. Read in the 16-bit value from the count length registers
3. Reset the upper two mode bits
4. Reset the carry and rotate right one position all 16 bits through carry
5. If carry is set, add ½ of the full original count (½ full count—1 if full count is odd).

NOTE:
If you started with an odd count and you read the count length register before the third count pulse occurs, you will not be able to discern whether one or two counts has occurred. Regardless of this, the 8155H/56H always counts out the right number of pulses in generating the $\overline{\text{TIMER OUT}}$ waveforms.

8085AH MINIMUM SYSTEM CONFIGURATION

Figure 13a shows a minimum system using three chips, containing:

- 256 Bytes RAM
- 2K Bytes EPROM
- 38 I/O Pins
- 1 Interval Timer
- 4 Interrupt Levels

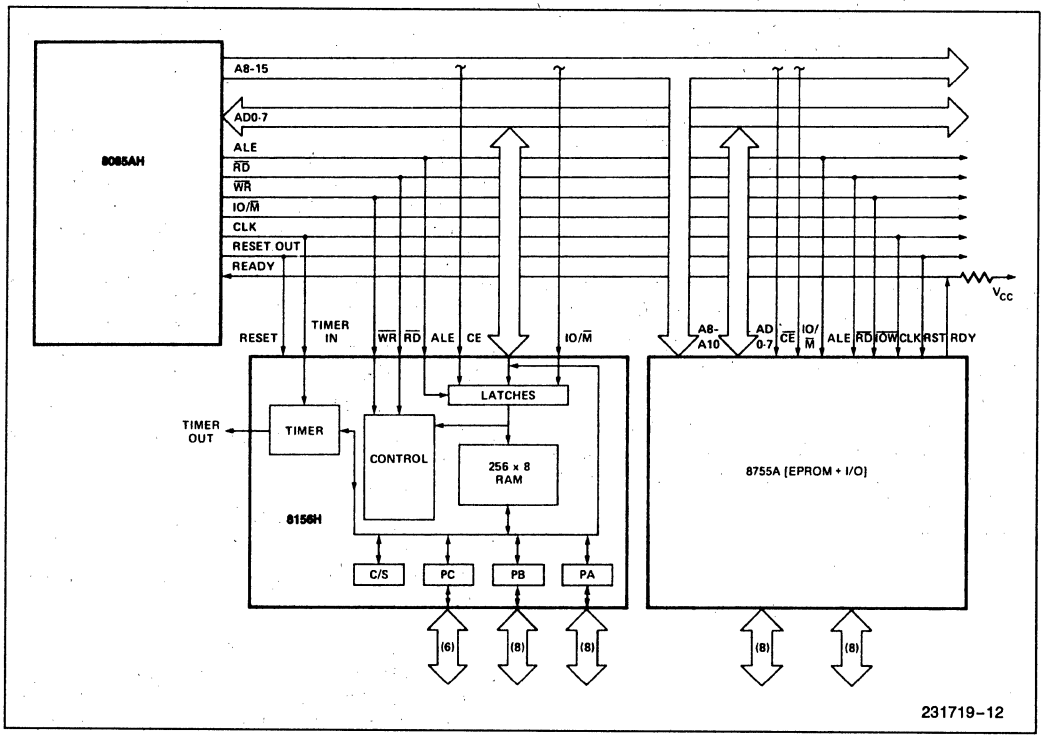


Figure 13a. 8085AH Minimum System Configuration (Memory Mapped I/O)

8088 FIVE CHIP SYSTEM

Figure 13b shows a five chip system containing:

- 1.25K Bytes RAM
- 2K Bytes EPROM
- 38 I/O Pins
- 1 Interval Timer
- 2 Interrupt Levels

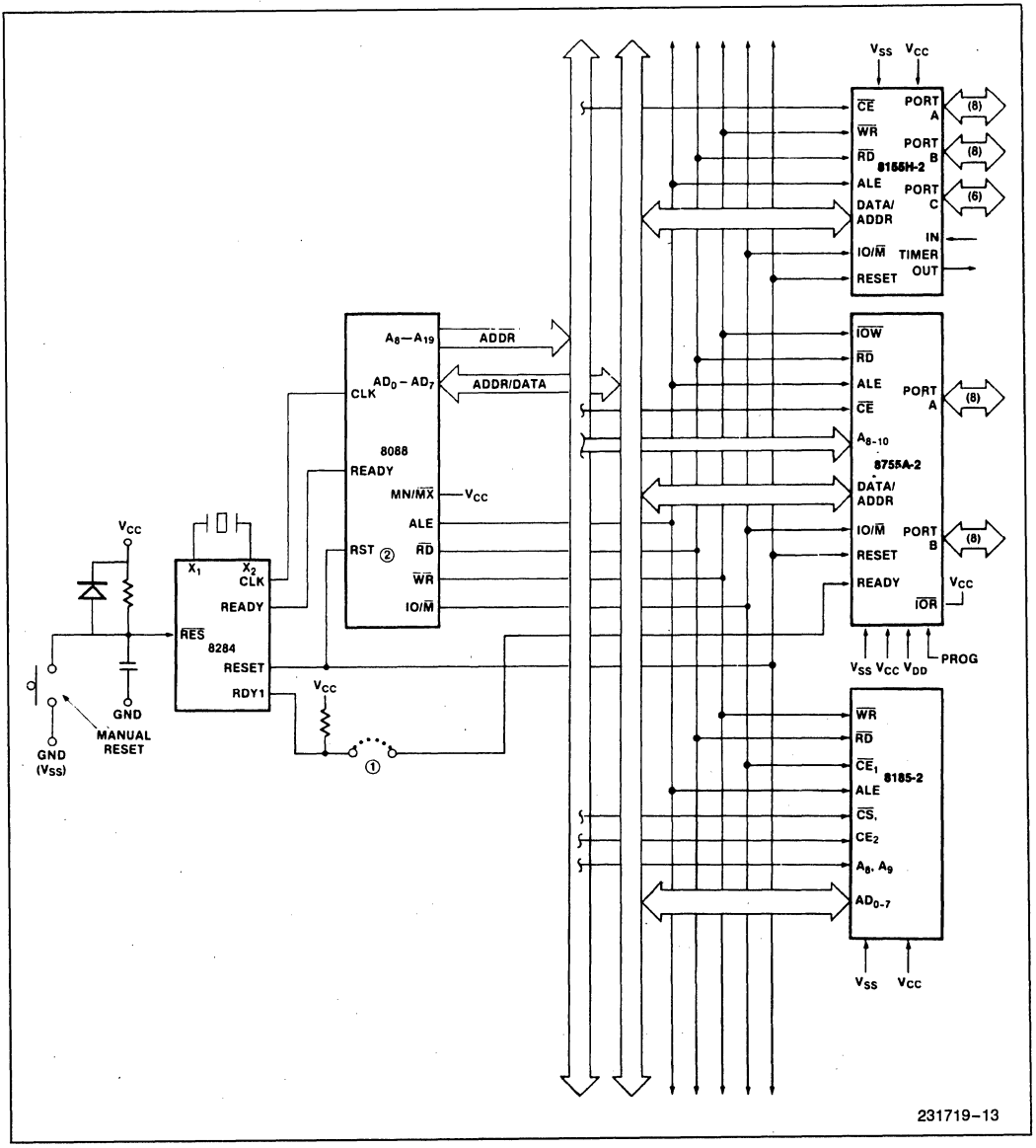


Figure 13b. 8088 Five Chip System Configuration

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... 0°C to +70°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5V to +7V
Power Dissipation 1.5W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{IL}	Input Leakage		±10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current		125	mA	
I_{IL} (CE)	Chip Enable Leakage 8155H 8156H		 +100 −100	 μA μA	0V ≤ V_{IN} ≤ V_{CC}

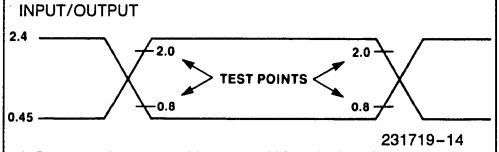
A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	8155H/8156H		8155H-2/8156H-2		Units
		Min	Max	Min	Max	
t_{AL}	Address to Latch Setup Time	50		30		ns
t_{LA}	Address Hold Time after Latch	80		30		ns
t_{LC}	Latch to READ/WRITE Control	100		40		ns
t_{RD}	Valid Data Out Delay from READ Control		170		140	ns
t_{LD}	Latch to Data Out Valid		350		270	ns
t_{AD}	Address Stable to Data Out Valid		400		330	ns
t_{LL}	Latch Enable Width	100		70		ns
t_{RDF}	Data Bus Float after READ	0	100	0	80	ns
t_{CL}	READ/WRITE Control to Latch Enable	20		10		ns
t_{CLL}	WRITE Control to Latch Enable for C/S Register	125		125		ns
t_{CC}	READ/WRITE Control Width	250		200		ns
t_{DW}	Data In to WRITE Setup Time	150		100		ns
t_{WD}	Data In Hold Time after WRITE	25		25		ns
t_{RV}	Recovery Time between Controls	300		200		ns
t_{WP}	WRITE to Port Output		400		300	ns

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10% (Continued)

Symbol	Parameter	8155H/8156H		8155H-2/8156H-2		Units
		Min	Max	Min	Max	
t_{PR}	Port Input Setup Time	70		50		ns
t_{RP}	Port Input Hold Time	50		10		ns
t_{SBF}	Strobe to Buffer Full		400		300	ns
t_{SS}	Strobe Width	200		150		ns
t_{RBE}	READ to Buffer Empty		400		300	ns
t_{SI}	Strobe to INTR On		400		300	ns
t_{RDI}	READ to INTR Off		400		300	ns
t_{PSS}	Port Setup Time to Strobe	50		0		ns
t_{PHS}	Port Hold Time After Strobe	120		100		ns
t_{SBE}	Strobe to Buffer Empty		400		300	ns
t_{WBF}	WRITE to Buffer Full		400		300	ns
t_{WI}	WRITE to INTR Off		400		300	ns
t_{TL}	TIMER-IN to $\overline{\text{TIMER-OUT}}$ Low		400		300	ns
t_{TH}	TIMER-IN to $\overline{\text{TIMER-OUT}}$ High		400		300	ns
t_{RDE}	Data Bus Enable from READ Control	10		10		ns
t_1	TIMER-IN Low Time	80		40		ns
t_2	TIMER-IN High Time	120		70		ns
t_{WT}	WRITE to TIMER-IN (for writes which start counting)	360		200		ns

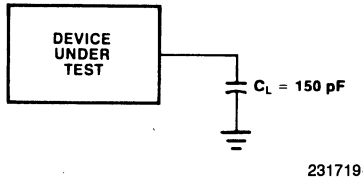
A.C. TESTING INPUT, OUTPUT WAVEFORM



231719-14

A.C. testing: inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

A.C. TESTING LOAD CIRCUIT

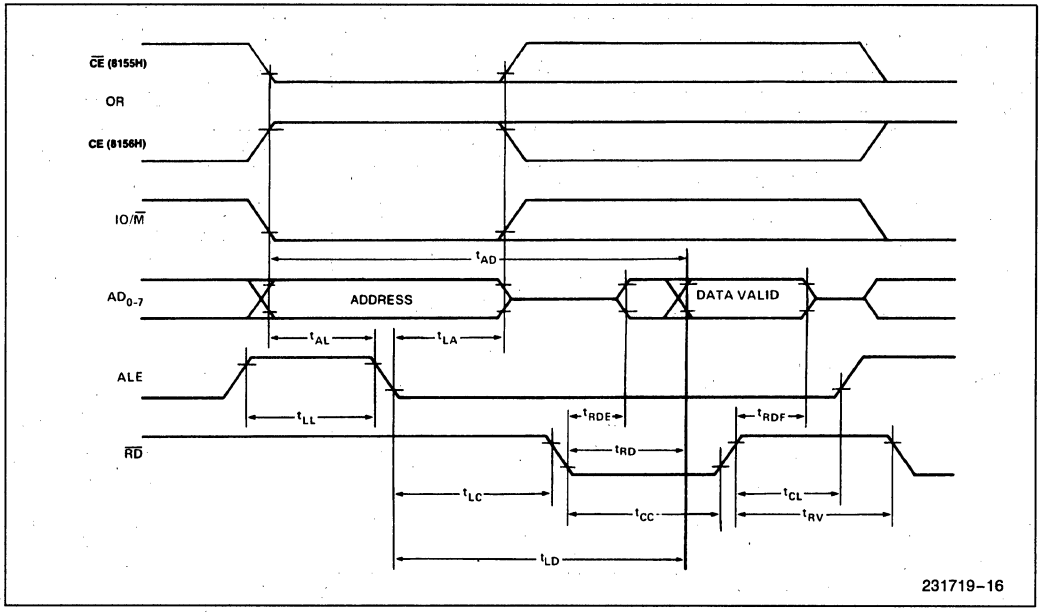


231719-15

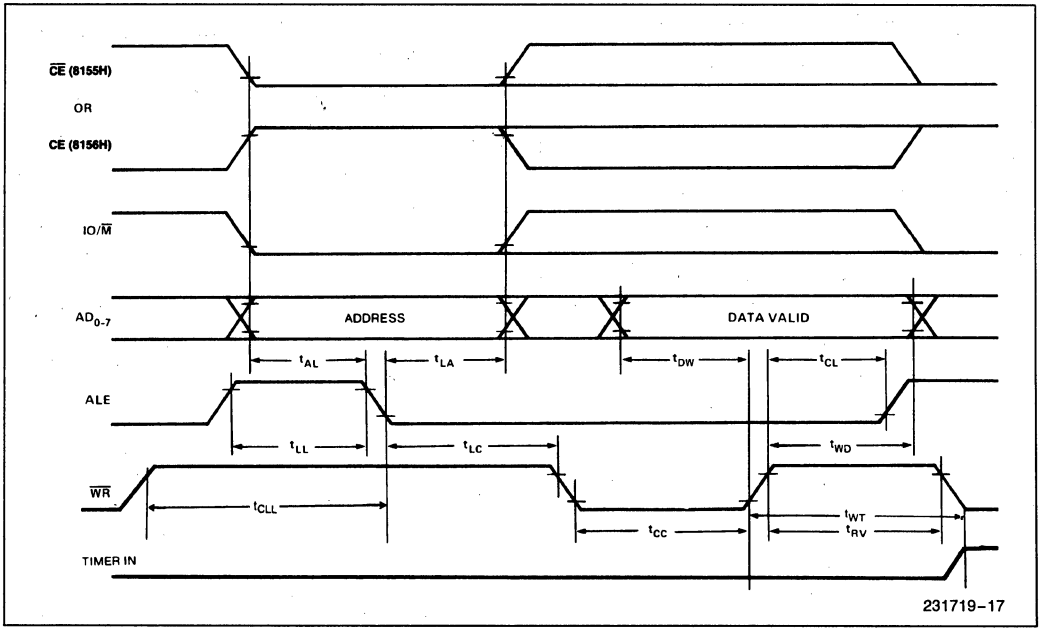
C_L = 150 pF
C_L Includes Jig Capacitance

WAVEFORMS

READ

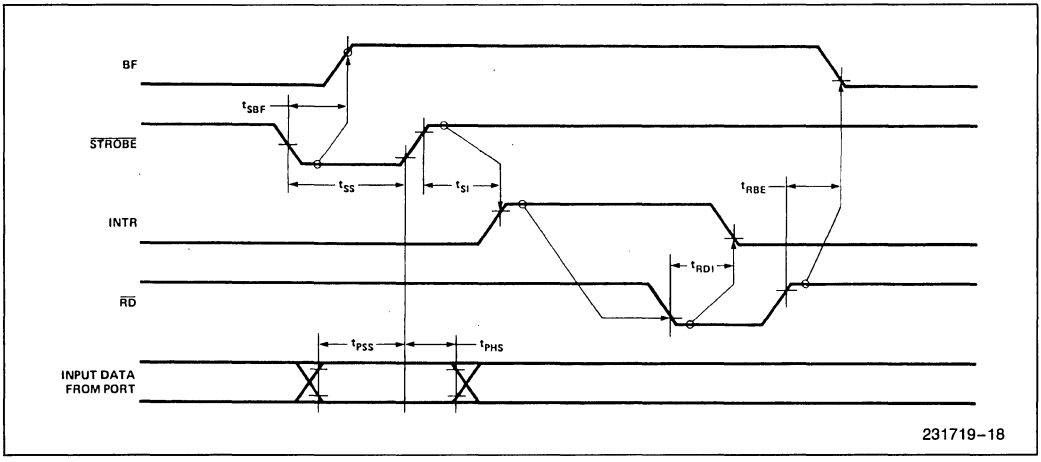


WRITE

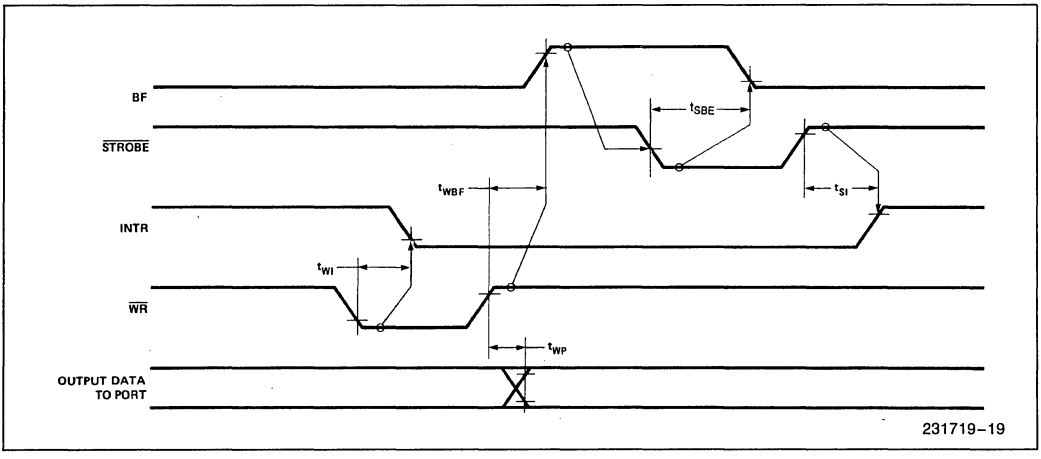


WAVEFORMS (Continued)

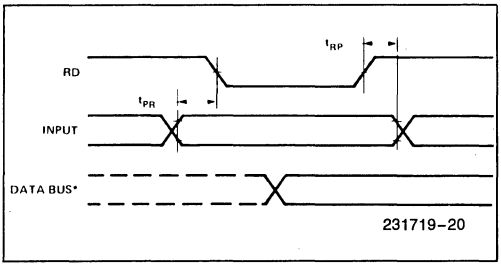
STROBED INPUT



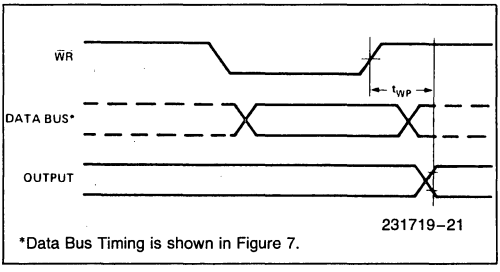
STROBED OUTPUT



BASIC INPUT



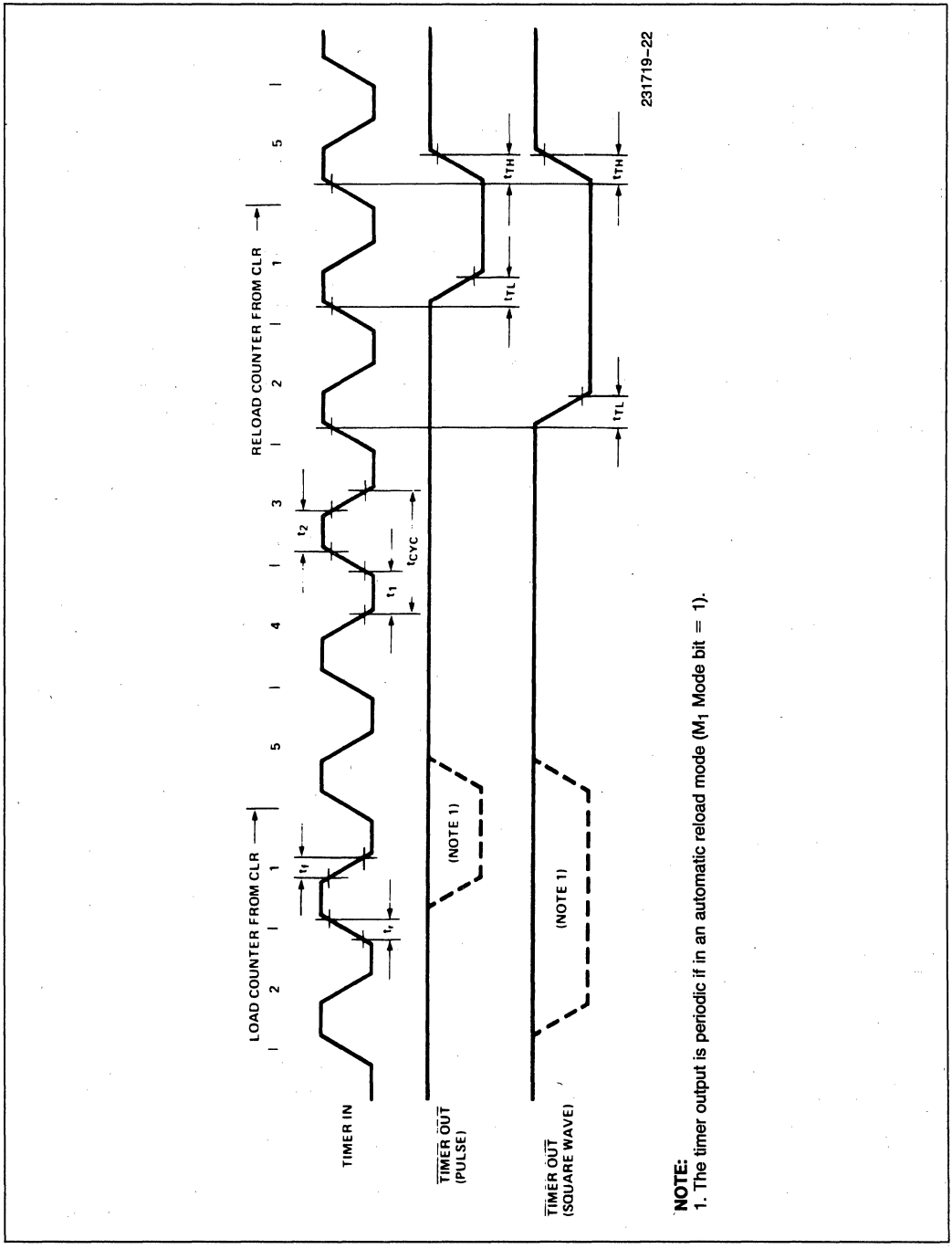
BASIC OUTPUT



*Data Bus Timing is shown in Figure 7.

WAVEFORMS (Continued)

TIMER OUTPUT COUNTDOWN FROM 5 TO 1



NOTE:
1. The timer output is periodic if in an automatic reload mode (M_1 Mode bit = 1).

8185/8185-2
1024 x 8-BIT STATIC RAM FOR MCS®-85

- **Multiplexed Address and Data Bus**
- **Directly Compatible with 8085AH and 8088 Microprocessors**
- **Low Operating Power Dissipation**
- **Low Standby Power Dissipation**
- **Single +5V Supply**
- **High Density 18-Pin Package**

The Intel 8185 is an 8192-bit static random access memory (RAM) organized as 1024 words by 8-bits using N-channel Silicon-Gate MOS technology. The multiplexed address and data bus allows the 8185 to interface directly to the 8085AH and 8088 microprocessors to provide a maximum level of system integration.

The low standby power dissipation minimizes system power requirements when the 8185 is disabled.

The 8185-2 is a high-speed selected version of the 8185 that is compatible with the 5 MHz 8085AH-2 and the 5 MHz 8088.

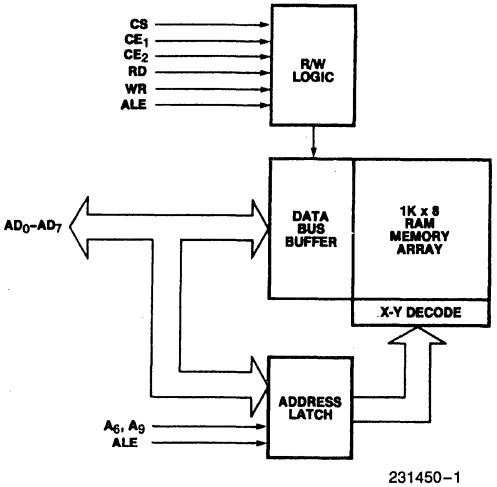


Figure 1. Block Diagram

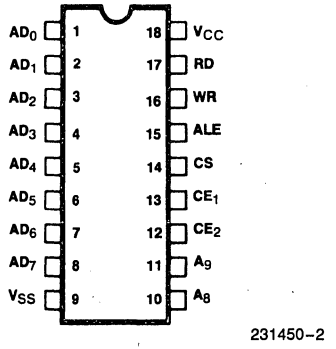


Figure 2. Pin Configuration

Pin Names

AD_0–AD_7	Address/Data Lines
A_8, A_9	Address Lines
CS	Chip Select
CE_1	Chip Enable (IO/M)
CE_2	Chip Enable
ALE	Address Latch Enable
WR	Write Enable

May 1987
Order Number: 231450-002

FUNCTIONAL DESCRIPTION

The 8185 has been designed to provide for direct interface to the multiplexed bus structure and bus timing of the 8085A microprocessor.

At the beginning of an 8185 memory access cycle, the 8-bit address on AD_{0-7}, A_8 and A_9, and the status of $\overline{CE}_1$ and CE_2 are all latched internally in the 8185 by the falling edge of ALE. If the latched status of both $\overline{CE}_1$ and CE_2 are active, the 8185 powers itself up, but no action occurs until the $\overline{CS}$ line goes low and the appropriate $\overline{RD}$ or $\overline{WR}$ control signal input is activated.

The $\overline{CS}$ input is not latched by the 8185 in order to allow the maximum amount of time for address decoding in selecting the 8185 chip. Maximum power consumption savings will occur, however, only when $\overline{CE}_1$ and CE_2 are activated selectively to power down the 8185 when it is not in use. A possible connection would be to wire the 8085A's IO/$\overline{M}$ line to the 8185's $\overline{CE}_1$ input, thereby keeping the 8185 powered down during I/O and interrupt cycles.

Table 1. Truth Table for Power Down and Function Enable

$\overline{CE}_1$	CE_2	$\overline{CS}$	(CS*)(2)	8185 Status
1	X	X	0	Power Down and Function Disable(1)
X	0	X	0	Power Down and Function Disable(1)
0	1	1	0	Powered Up and Function Disable(1)
0	1	0	1	Powered Up and Enabled

NOTES:
X = Don't Care.
1: Function Disable implies Data Bus in high impedance state and not writing.
2: CS* = ($\overline{CE}_1$ = 0) × (CE_2 = 1) × ($\overline{CS}$ = 0).
CS* = 1 signifies all chip enables and chip select active.

Table 2. Truth Table for Control and Data Bus Pin Status

(CS*)	$\overline{RD}$	$\overline{WR}$	AD_{0-7} During Data Portion of Cycle	8185 Function
0	X	X	Hi-Impedance	No Function
1	0	1	Data from Memory	Read
1	1	0	Data to Memory	Write
1	1	1	Hi-Impedance	Reading, but not Driving Data Bus

NOTE:
X = Don't Care.

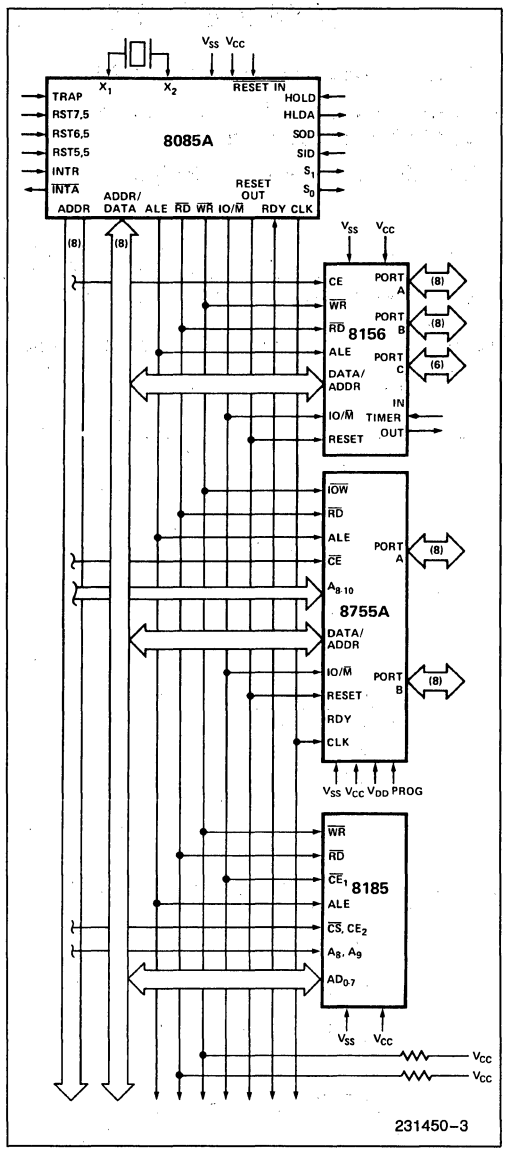


Figure 3. 8185 in an MCS®-85 System

4 Chips:
2K Bytes EPROM
1.25K Bytes RAM
38 I/O Lines
1 Counter/Timer
2 Serial I/O Lines
5 Interrupt Inputs

iAPX 88 FIVE CHIP SYSTEM:

- 1.25K Bytes RAM
- 2K Bytes EPROM
- 38 I/O Pins
- 1 Internal Timer
- 2 Interrupt Levels

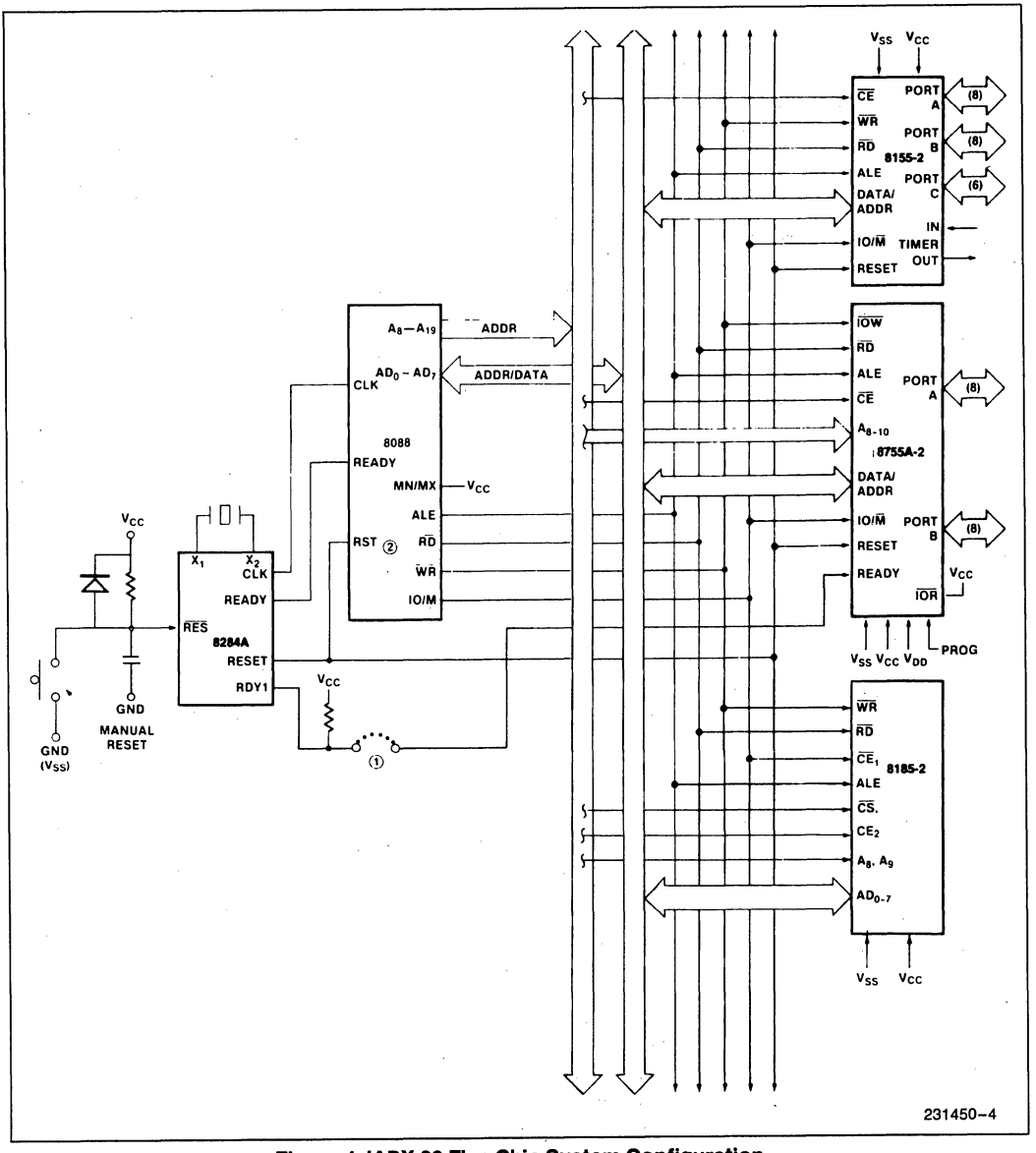


Figure 4. iAPX 88 Five Chip System Configuration

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin
with Respect to Ground.....−0.5V to +7V
Power Dissipation.....1.5W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

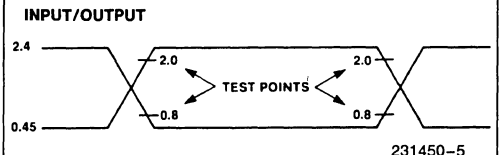
D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4			I_{OH} = −400 μA
I_{IL}	Input Leakage		±10	μA	0V ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current Powered Up		100	mA	
	Powered Down		35	mA	

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±10%

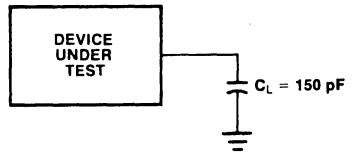
Symbol	Parameter	8185		8185-2		
		Min	Max	Min	Max	Units
t_{AL}	Address to Latch Set Up Time	50		30		ns
t_{LA}	Address Hold Time After Latch	80		30		ns
t_{LC}	Latch to READ/WRITE Control	100		40		ns
t_{RD}	Valid Data Out Delay from READ Control		170		140	ns
t_{LD}	ALE to Data Out Valid		300		200	ns
t_{LL}	Latch Enable Width	100		70		ns
t_{RDF}	Data Bus Float After READ	0	100	0	80	ns
t_{CL}	READ/WRITE Control to Latch Enable	20		10		ns
t_{CC}	READ/WRITE Control Width	250		200		ns
t_{DW}	Data In to WRITE Set Up Time	150		150		ns
t_{WD}	Data In Hold Time After WRITE	20		20		ns
t_{SC}	Chip Select Set Up to Control Line	10		10		ns
t_{CS}	Chip Select Hold Time After Control	10		10		ns
t_{ALCE}	Chip Enable Set Up to ALE Falling	30		10		ns
t_{LACE}	Chip Enable Hold Time After ALE	50		30		ns

A.C. TESTING INPUT, OUTPUT WAVEFORM



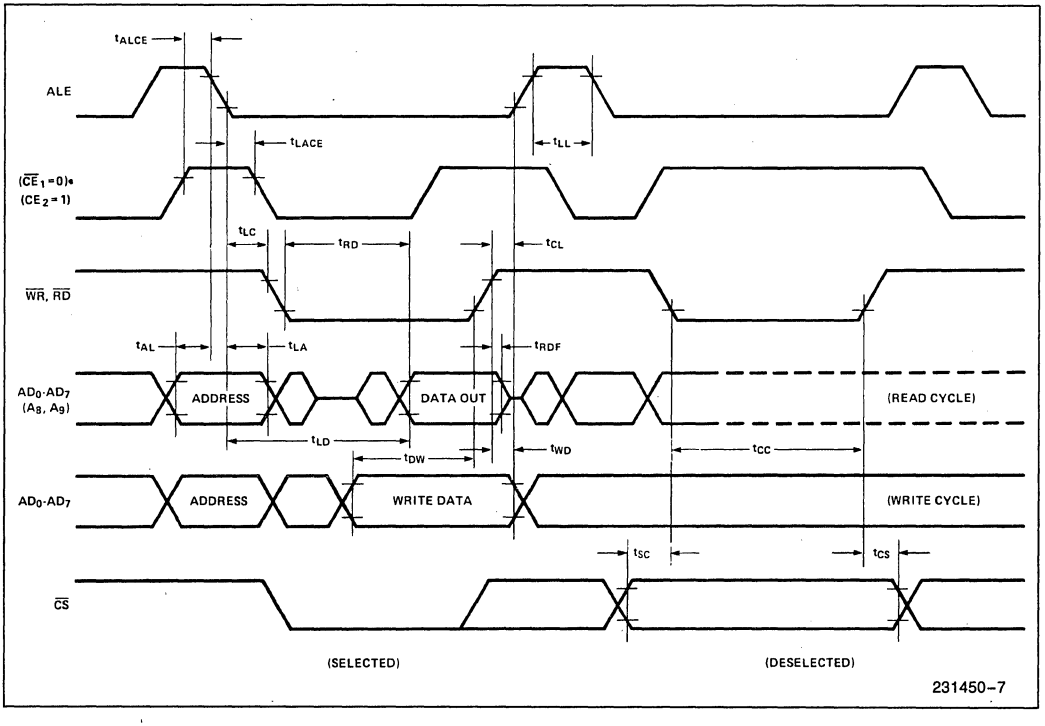
A.C. Testing: Inputs Are Driven at 2,.4V for a Logic "1" and 0.45V for a Logic "0." Timing Measurements Are Made at 2.0V for a Logic "1" and 0.8V for a Logic "0."

A.C. TESTING LOAD CIRCUIT



C_L = 150 pF
C_L Includes Jig Capacitance

WAVEFORM



8224
CLOCK GENERATOR AND DRIVER FOR 8080A CPU

- **Single Chip Clock Generator/Driver for 8080A CPU**
- **Power-Up Reset for CPU**
- **Ready Synchronizing Flip-Flop**
- **Advanced Status Strobe**
- **Oscillator Output for External System Timing**
- **Crystal Controlled for Stable System Operation**
- **Reduces System Package Count**
- **Available in EXPRESS — Standard Temperature Range**
- **Available in 16-Lead Cerdip Package**
  (See Packaging Spec, Order #231369)

The Intel 8224 is a single chip clock generator/driver for the 8080A CPU. It is controlled by a crystal, selected by the designer to meet a variety of system speed requirements.

Also included are circuits to provide power-up reset, advance status strobe, and synchronization of ready.

The 8224 provides the designer with a significant reduction of packages used to generate clocks and timing for 8080A.

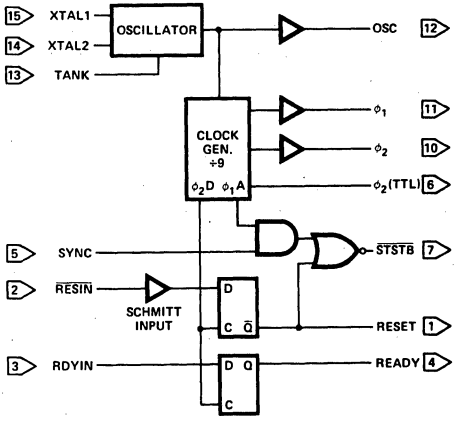
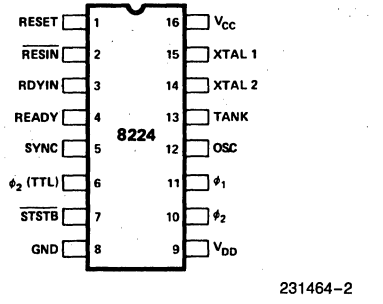


Figure 1. Block Diagram



$\overline{\text{RESIN}}$	Reset Input
RESET	Reset Output
RDYIN	Ready Input
READY	Ready Output
SYNC	Sync Input
$\overline{\text{STSTB}}$	Status STB (Active Low)
ϕ_1	8080 Clocks
ϕ_2	

XTAL 1	Connections for Crystal
XTAL 2	
TANK	Used with Overtone XTAL
OSC	Oscillator Output
ϕ_2 (TTL)	ϕ_2 CLK (TTL Level)
V_{CC}	+5V
V_{DD}	+12V
GND	0V

Figure 2. Pin Configuration

December 1986
Order Number: 231464-001

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... 0°C to +70°C
Storage Temperature ..... −65°C to +150°C
Supply Voltage, V_{CC} ..... −0.5V to +7V
Supply Voltage, V_{DD} ..... −0.5V to +13.5V
Input Voltage ..... −1.5V to +7V
Output Current ..... 100 mA

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = +5.0V ±5%, V_{DD} = +12V ±5%

Symbol	Parameter	Limits			Units	Test Conditions
		Min	Typ	Max		
I_F	Input Current Loading			−0.25	mA	V_F = 0.45V
I_R	Input Leakage Current			10	μA	V_R = 5.25V
V_C	Input Forward Clamp Voltage			1.0	V	I_C = −5 mA
V_{IL}	Input "Low" Voltage			0.8	V	V_{CC} = 5.0V
V_{IH}	Input "High" Voltage	2.6			V	Reset Input
		2.0			V	All Other Inputs
V_{IH}−V_{IL}	RESIN Input Hysteresis	0.25			V	V_{CC} = 5.0V
V_{OL}	Output "Low" Voltage			0.45	V	(ϕ_1, ϕ_2), Ready, Reset, $\overline{STSTB}$ I_{OL} = 2.5 mA
				0.45	V	All Other Outputs I_{OL} = 15 mA
V_{OH}	Output "High" Voltage ϕ_1, ϕ_2	9.4			V	I_{OH} = −100 μA
	READY, RESET	3.6			V	I_{OH} = −100 μA
	All Other Outputs	2.4			V	I_{OH} = −1 mA
I_{CC}	Power Supply Current			115	mA	
I_{DD}	Power Supply Current			12	mA	

NOTE:
1. For crystal frequencies of 18 MHz connect 510Ω resistors between the X1 input and ground as well as the X2 input and ground to prevent oscillation at harmonic frequencies.

Crystal Requirements

Tolerance: 0.005% at 0°C–70°C
Resonance: Series (Fundamental)*
Load Capacitance: 20 pF–35 pF
Equivalent Resistance: 75Ω–20Ω
Power Dissipation (Min): 4 mW

***NOTE:**
With tank circuit use 3rd overtone mode.

A.C. CHARACTERISTICS

Symbol	Parameter	Limits			Units	Test Conditions
		Min	Typ	Max		
$t_{\phi 1}$	ϕ_1 Pulse Width	$\frac{2tcy}{9} - 20$ ns			ns	C_L = 20 pF to 50 pF
$t_{\phi 2}$	ϕ_2 Pulse Width	$\frac{5tcy}{9} - 35$ ns				
t_{D1}	ϕ_1 to ϕ_2 Delay	0				
t_{D2}	ϕ_2 to ϕ_1 Delay	$\frac{2tcy}{9} - 14$ ns				
t_{D3}	ϕ_1 to ϕ_2 Delay	$\frac{2tcy}{9}$		$\frac{2tcy}{9} + 20$ ns		
t_R	ϕ_1 and ϕ_2 Rise Time			20		
t_F	ϕ_1 and ϕ_2 Fall Time			20		
$t_{D\phi 2}$	ϕ_2 to ϕ_2 (TTL) Delay	−5		+15	ns	ϕ_2 TTL, CL = 30 R_1 = 300Ω R_2 = 600Ω
t_{DSS}	ϕ_2 to $\overline{STSTB}$ Delay	$\frac{6tcy}{9} - 30$ ns		$\frac{6tcy}{9}$	ns	$\overline{STSTB}$, C_L = 15 pF R_1 = 2K R_2 = 4K
t_{PW}	$\overline{STSTB}$ Pulse Width	$\frac{tcy}{9} - 15$ ns			ns	
t_{DRS}	RDYIN Setup Time to Status Strobe	$50\text{ ns} - \frac{4tcy}{9}$				
t_{DRH}	RDYIN Hold Time after $\overline{STSTB}$	$\frac{4tcy}{9}$				
t_{DR}	RDYIN or RESIN to ϕ_2 Delay	$\frac{4tcy}{9} - 25$ ns			ns	Ready & Reset C_L = 10 pF R_1 = 2K R_2 = 4K
t_{CLK}	CLK Period		$\frac{tcy}{9}$		ns	
f_{max}	Maximum Oscillating Frequency			27	MHz	
C_{in}	Input Capacitance			8	pF	V_{CC} = +5.0V V_{DD} = +12V V_{BIAS} = 2.5V f = 1 MHz

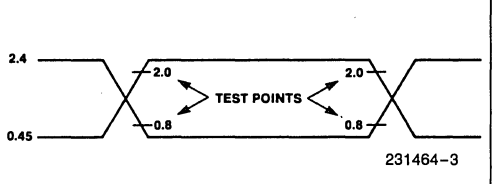
NOTE:
These formulas are based on the internal workings of the part and intended for customer convenience. Actual testing of the part is done at t_{cy} = 488.28 ns.

A.C. CHARACTERISTICS (Continued)

For t_{CY} = 488.28 ns; T_A = 0°C to 70°C, V_{CC} = +5V ±5%, V_{DD} = +12V ±5%

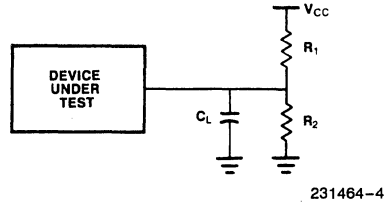
Symbol	Parameter	Limits			Units	Test Conditions
		Min	Typ	Max		
$t_{\phi 1}$	ϕ_1 Pulse Width	89			ns	t_{CY} = 488.28 ns
$t_{\phi 2}$	ϕ_2 Pulse Width	236			ns	
t_{D1}	Delay ϕ_1 to ϕ_2	0			ns	
t_{D2}	Delay ϕ_2 to ϕ_1	95			ns	ϕ_1 & ϕ_2 Loaded to C_L = 20 pF to 50 pF
t_{D3}	Delay ϕ_1 to ϕ_2 Leading Edges	109		129	ns	
t_r	Output Rise Time			20	ns	
t_f	Output Fall Time			20	ns	
t_{DSS}	ϕ_2 to $\overline{STSTB}$ Delay	296		326	ns	
$t_{D\phi 2}$	ϕ_2 to ϕ_2 (TTL) Delay	−5		+15	ns	
t_{PW}	Status Strobe Pulse Width	40			ns	Ready & Reset Loaded to 2 mA/10 pF All measurements referenced to 1.5V unless specified otherwise.
t_{DRS}	RDYIN Setup Time to $\overline{STSTB}$	−167			ns	
t_{DRH}	RDYIN Hold Time after $\overline{STSTB}$	217			ns	
t_{DR}	READY or RESET to ϕ_2 Delay	192			ns	
f_{MAX}	Oscillator Frequency			18.432	MHz	

A.C. TESTING, INPUT, OUTPUT WAVEFORM



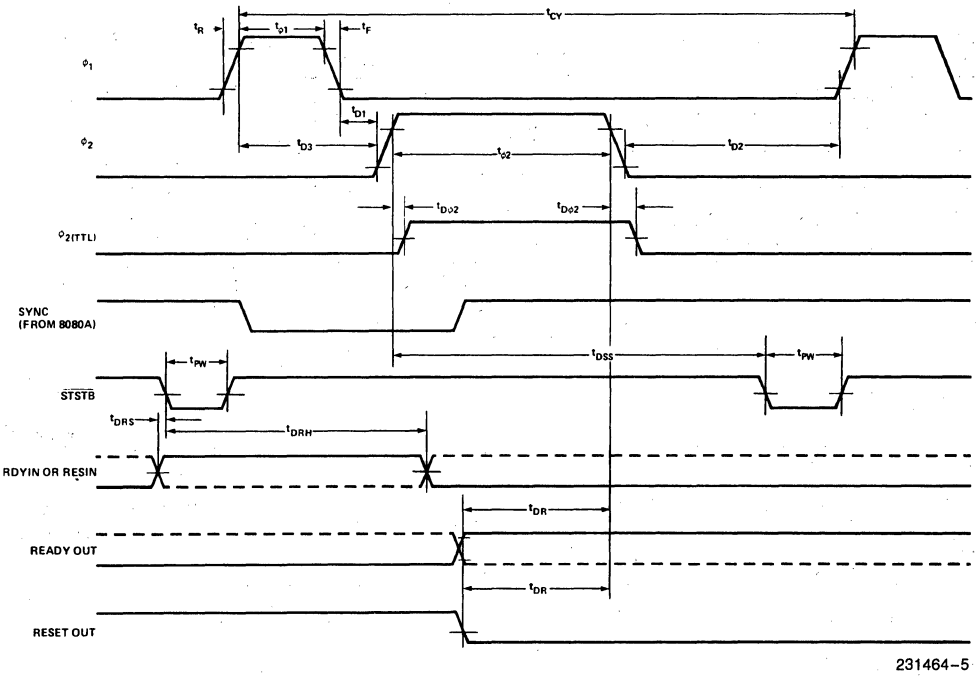
A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0" (unless otherwise noted).

A.C. TESTING LOAD CIRCUIT



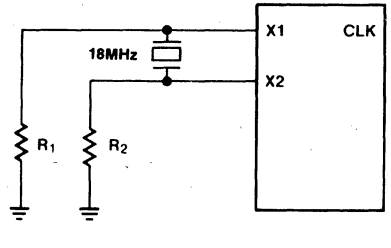
C_L Includes Jig Capacitance

WAVEFORMS



VOLTAGE MEASUREMENT POINTS: ϕ_1, ϕ_2 Logic "0" = 1.0V, Logic "1" = 8.0V. All other signals measured at 1.5V.

CLOCK HIGH AND LOW TIME (USING X1, X2)



8228
SYSTEM CONTROLLER AND BUS DRIVER FOR 8080A CPU

- **Single Chip System Control for MCS®-80 Systems**
- **Built-In Bidirectional Bus Driver for Data Bus Isolation**
- **Allows the Use of Multiple Byte Instructions (e.g. CALL) for Interrupt Acknowledge**
- **Reduces System Package Count**
- **User Selected Single Level Interrupt Vector (RST 7)**
- **Available in EXPRESS — Standard Temperature Range**
- **Available in 28-Lead Cerdip and Plastic Packages**
  (See Packaging Spec, Order #231369)

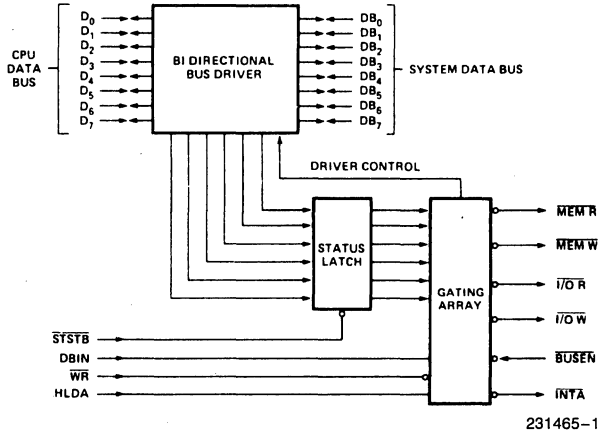
The Intel® 8228 is a single chip system controller and bus driver for MCS®-80. It generates all signals required to directly interface MCS-80 family RAM, ROM, and I/O components.

A bidirectional bus driver is included to provide high system TTL fan-out. It also provides isolation of the 8080 data bus from memory and I/O. This allows for the optimization of control signals, enabling the systems designer to use slower memory and I/O. The isolation of the bus driver also provides for enhanced system noise immunity.

A user selected single level interrupt vector (RST 7) is provided to simplify real time, interrupt driven, small system requirements. The 8228 also generates the correct control signals to allow the use of multiple byte instructions (e.g., CALL) in response to an interrupt acknowledge by the 8080A. This feature permits large, interrupt driven systems to have an unlimited number of interrupt levels.

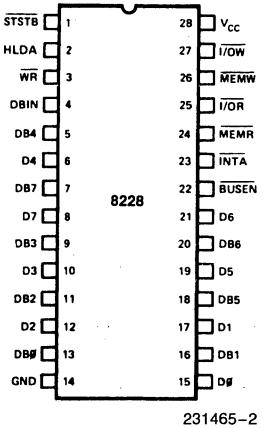
The 8228 is designed to support a wide variety of system bus structures and also reduce system package count for cost effective, reliable design of MCS-80 systems.

NOTE:
The specifications for the 3228 are identical with those for the 8228.



231465-1

Figure 1. Block Diagram



231465-2

D7-D0	Data Bus (8080 Side)
DB7-DB0	Data Bus (System Side)
I/OR	I/O Read
I/OW	I/O Write
MEMR	Memory Read
MEMW	Memory Write
DBIN	DBIN (from 8080)

INTA	Interrupt Acknowledge
HLDA	HLDA (from 8080)
WR	WR (from 8080)
BUSEN	Bus Enable Input
STSTB	Status Strobe (from 8224)
VCC	+5V
GND	0 Volts

Figure 2. Pin Configuration

September 1987
Order Number: 231465-002

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... 0°C to +70°C
Storage Temperature ..... −65°C to +150°C
Supply Voltage, V_{CC} ..... −0.5V to +7V
Input Voltage ..... −1.5 to +7V
Output Current ..... 100 mA

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5V ±5%

Symbol	Parameter		Limits			Unit	Test Conditions
			Min	Typ(1)	Max		
V_C	Input Clamp Voltage, All Input			0.75	−1.0	V	V_{CC} = 4.75V; I_C = −5 mA
I_F	Input Load Current	$\overline{STSTB}$			500	μA	V_{CC} = 5.25V
		D_2 & D_6			750	μA	V_F = 0.45V
		D_0, D_1, D_4, D_5 & D_7			250	μA	
		All Other Inputs			250	μA	
I_R	Input Leakage Current	$\overline{STSTB}$			100	μA	V_{CC} = 5.25V V_R = 5.25V
		DB_0–DB_7			20	μA	
		All Other Inputs			100	μA	
V_{TH}	Input Threshold Voltage, All Inputs		0.8		2.0	V	V_{CC} = 5V
I_{CC}	Power Supply Current			140	190	mA	V_{CC} = 5.25V
V_{OL}	Output Low Voltage	D_0–D_7			0.45	V	V_{CC} = 4.75V; I_{OL} = 2 mA
		All Other Outputs			0.45	V	I_{OL} = 10 mA
V_{OH}	Output High Voltage	D_0–D_7	3.6	3.8		V	V_{CC} = 4.75V; I_{OH} = −10μA
		All Other Outputs	2.4			V	I_{OH} = −1 mA
I_{OS}	Short Circuit Current, All Outputs		15		90	mA	V_{CC} = 5V
$I_{O(off)}$	Off State Output Current All Control Outputs				100	μA	V_{CC} = 5.25V; V_O = 5.25V
					−100	μA	V_O = 0.45V
I_{INT}	INTA Current				5	mA	(See INTA Test Circuit)

NOTE:
1. Typical values are for T_A = 25°C and nominal supply voltages.

CAPACITANCE V_{BIAS} = 2.5V, V_{CC} = 5.0V, T_A = 25°C, f = 1 MHz

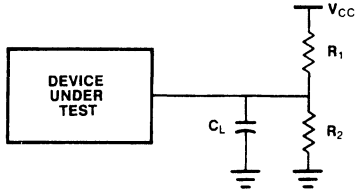
1. This parameter is periodically sampled and not 100% tested.

Symbol	Parameter	Limits			Unit
		Min	Typ(1)	Max	
C_{IN}	Input Capacitance		8	12	pF
C_{OUT}	Output Capacitance Control Signals		7	15	pF
I/O	I/O Capacitance (D or DB)		8	15	pF

A.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5V ±5%

Symbol	Parameter	Limits		Unit	Conditions
		Min	Max		
t_{PW}	Width of Status Strobe	22		ns	
t_{SS}	Setup Time, Status Inputs D_0–D_7	8		ns	
t_{SH}	Hold Time, Status Inputs D_0–D_7	5		ns	
t_{DC}	Delay from $\overline{STSTB}$ to any Control Signal	20	60	ns	C_L = 100 pF
t_{RR}	Delay from DBIN to Control Outputs		30	ns	C_L = 100 pF
t_{RE}	Delay from DBIN to Enable/Disable 8080 Bus		45	ns	C_L = 25 pF
t_{RD}	Delay from System Bus to 8080 Bus during Read		30	ns	C_L = 25 pF
t_{WR}	Delay from $\overline{WR}$ to Control Outputs	5	45	ns	C_L = 100 pF
t_{WE}	Delay to Enable System Bus DB_0–DB_7 after $\overline{STSTB}$		30	ns	C_L = 100 pF
t_{WD}	Delay from 8080 Bus D_0–D_7 to System Bus DB_0–DB_7 during Write	5	40	ns	C_L = 100 pF
t_E	Delay from $\overline{\text{System Bus Enable}}$ to System Bus DB_0–DB_7		30	ns	C_L = 100 pF
t_{HD}	HLDA to Read Status Outputs		25	ns	
t_{DS}	Setup Time, System Bus Inputs to HLDA	10		ns	
t_{DH}	Hold Time, System Bus Inputs to HLDA	20		ns	C_L = 100 pF

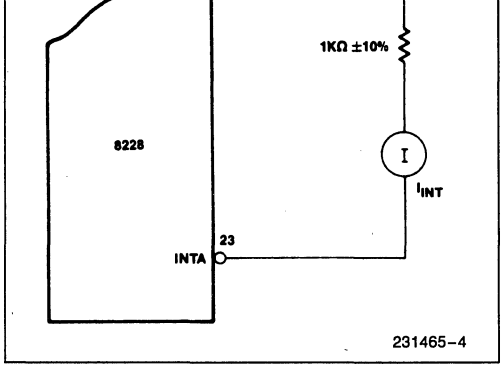
AC TESTING LOAD CIRCUIT



231465-3

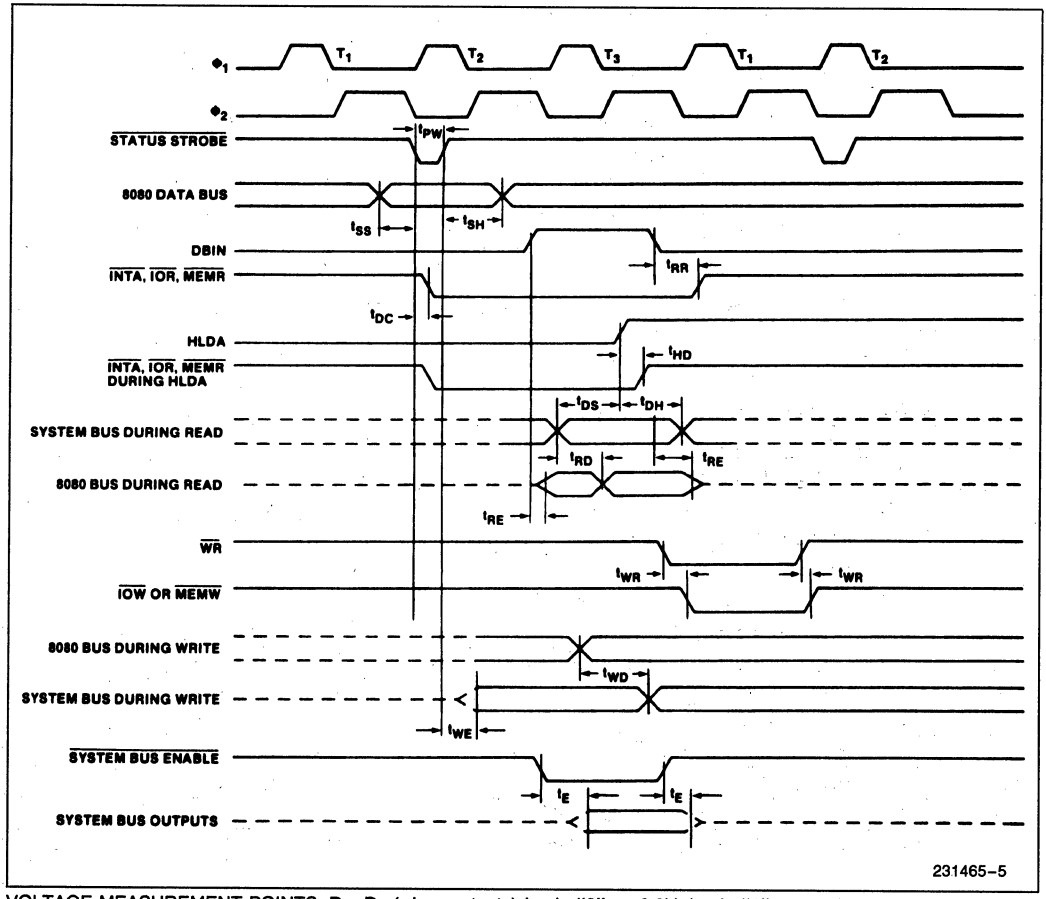
For D_0–D_7; R_1 = 4 KΩ, R_2 = ∞Ω, C_L = 25 pF.
For all other outputs: R_1 = 500Ω, R_2 = 1 KΩ, C_L = 100 pF.

INTA Test Circuit (for RST 7)



231465-4

WAVEFORMS



VOLTAGE MEASUREMENT POINTS: D_0–D_7 (when outputs) Logic "0" = 0.8V, Logic "1" = 3.0V. All other signals measured at 1.5V.

8755A
16,384-BIT EPROM WITH I/O

- **2048 Words x 8 Bits**
- **Single +5V Power Supply (V_{CC})**
- **Directly Compatible with 8085AH**
- **U.V. Erasable and Electrically Reprogrammable**
- **Internal Address Latch**
- **2 General Purpose 8-Bit I/O Ports**
- **Each I/O Port Line Individually Programmable as Input or Output**
- **Multiplexed Address and Data Bus**
- **40-Pin DIP**
- **Available in EXPRESS**
  - **— Standard Temperature Range**
  - **— Extended Temperature Range**

The Intel 8755A is an erasable and electrically reprogrammable ROM (EPROM) and I/O chip to be used in the 8085AH microprocessor systems. The EPROM portion is organized as 2048 words by 8 bits. It has a maximum access time of 450 ns to permit use with no wait states in an 8085AH CPU.

The I/O portion consists of 2 general purpose I/O ports. Each I/O port has 8 port lines, and each I/O port line is individually programmable as input or output.

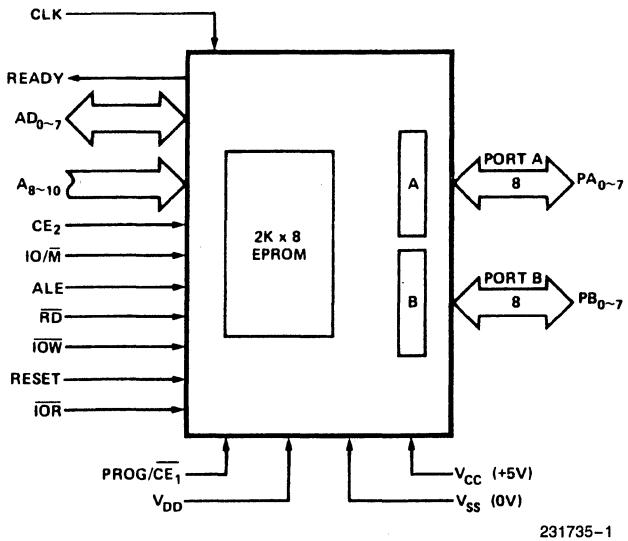


Figure 1. Block Diagram

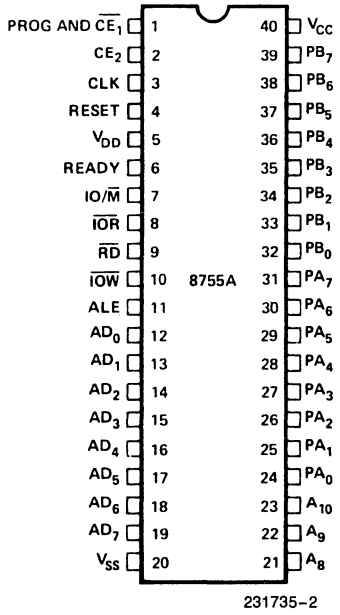


Figure 2. Pin Configuration

November 1986
Order Number: 231735-002

Table 1. Pin Description

Symbol	Type	Name and Function
ALE	I	**ADDRESS LATCH ENABLE:** When Address Latch Enable goes *high*, AD_{0-7}, IO/$\overline{M}$, A_{8-10}, CE_2, and $\overline{CE_1}$ enter the address latches. The signals, (AD, IO/$\overline{M}$, AD_{8-10}, CE_2, $\overline{CE_1}$) are latched in at the trailing edge of ALE.
AD_{0-7}	I	**BIDIRECTIONAL ADDRESS/DATA BUS:** The lower 8 bits of the PROM or I/O address are applied to the bus lines when ALE is high. During an I/O cycle, Port A or B is selected based on the latched value of AD_0. IF $\overline{RD}$ or $\overline{IOR}$ is low when the latched Chip Enables are active, the output buffers present data on the bus.
AD_{8-10}	I	**ADDRESS BUS:** These are the high order bits of the PROM address. They do not affect I/O operations.
PROG/$\overline{CE_1}$ CE_2	I	**CHIP ENABLE INPUTS:** $\overline{CE_1}$ is active low and CE_2 is active high. The 8755A can be accessed only when *both* Chip Enables are active at the time the ALE signal latches them up. If either Chip Enable input is not active, the AD_{0-7}, and READY ouputs will be in a high impedance state. $\overline{CE_1}$ is also used as a programming pin. (See section on programming.)
IO/$\overline{M}$	I	**I/O MEMORY:** If the latched IO/$\overline{M}$ is high when $\overline{RD}$ is low, the output data comes from an I/O port. If it is low the output data comes from the PROM.
$\overline{RD}$	I	**READ:** If the latched Chip Enables are active when $\overline{RD}$ goes low, the AD_{0-7} output buffers are enabled and output either the selected PROM location or I/O port. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD_{0-7} output buffers are 3-stated.
$\overline{IOW}$	I	**I/O WRITE:** If the latched Chip Enables are active, a low on $\overline{IOW}$ causes the output port pointed to by the latched value of AD_0 to be written with the data on AD_{0-7}. The state of IO/$\overline{M}$ is ignored.
CLK	I	**CLOCK:** The CLK is used to force the READY into its high impedance state after it has been forced low by $\overline{CE_1}$ low, CE_2 high, and ALE high.
READY	O	**READY** is a 3-state output controlled by $\overline{CE_1}$, CE_2, ALE and CLK. READY is forced low when the Chip Enables are active during the time ALE is high, and remains low until the rising edge of the next CLK. (See Figure 6c.)
PA_{0-7}	I/O	**PORT A:** These are general purpose I/O pins. Their input/output direction is determined by the contents of Data Direction Register (DDR). Port A is selected for write operations when the Chip Enables are active and $\overline{IOW}$ is low and a 0 was previously latched from AD_0, AD_1. Read Operation is selected by either $\overline{IOR}$ low and active Chip Enables and AD_0 and AD_1 low, *or* IO/$\overline{M}$ high, $\overline{RD}$ low, active Chip Enables, and AD_0 and AD_1 low.
PB_{0-7}	I/O	**PORT B:** The general purpose I/O port is identical to Port A except that it is selected by a 1 latched from AD_0 and a 0 from AD_1.
RESET	I	**RESET:** In normal operation, an input high on RESET causes all pins in Ports A and B to assume input mode (clear DDR register).
$\overline{IOR}$	I	**I/O READ:** When the Chip Enables are active, a low on $\overline{IOR}$ will output the selected I/O port onto the AD bus. $\overline{IOR}$ low performs the same function as the combination of IO/$\overline{M}$ high and $\overline{RD}$ low. When $\overline{IOR}$ is not used in a system, $\overline{IOR}$ should be tied to V_{CC} ("1").
V_{CC}		**POWER:** +5V supply.
V_{SS}		**GROUND:** Reference.
V_{DD}		**POWER SUPPLY:** V_{DD} is a programming voltage, and must be tied to V_{CC} when the 8755A is being read. For programming, a high voltage is supplied with V_{DD} = 25V, typical. (See section on programming.)

FUNCTIONAL DESCRIPTION

PROM Section

The 8755A contains an 8-bit address latch which allows it to interface directly to MCS®-48 and MCS®-85 processors without additional hardware.

The PROM section of the chip is addressed by the 11-bit address and the Chip Enables. The address, $\overline{CE_1}$ and CE_2 are latched into the address latches on the falling edge of ALE. If the latched Chip Enables are active and IO/$\overline{M}$ is low when $\overline{RD}$ goes low, the contents of the PROM location addressed by the latched address are put out on the AD_{0-7} lines (provided that V_{DD} is tied to V_{CC}).

I/O Section

The I/O section of the chip is addressed by the latched value of AD_{0-1}. Two 8-bit Data Direction Registers (DDR) in 8755A determine the input/output status of each pin in the corresponding ports. A "0" in a particular bit position of a DDR signifies that the corresponding I/O port bit is in the input mode. A "1" in a particular bit position signifies that the corresponding I/O port bit is in the output mode. In this manner the I/O ports of the 8755A are bit-by-bit programmable as inputs or outputs. The table summarizes port and DDR designation. DDR's cannot be read.

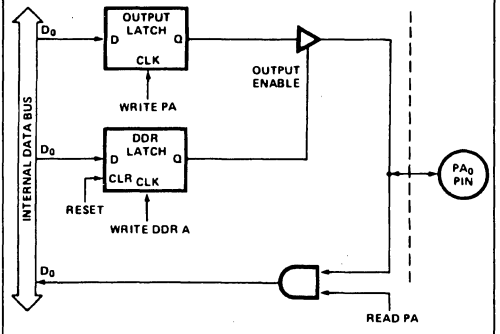
AD_1	AD_0	Selection
0	0	Port A
0	1	Port B
1	0	Port A Data Direction Register (DDR A)
1	1	Port B Data Direction Register (DDR B)

When $\overline{IOW}$ goes low and the Chip Enables are active, the data on the AD_{0-7} is written into I/O port selected by the latched value of AD_{0-1}. During this operation all I/O bits of the selected port are affected, regardless of their I/O mode and the state of IO/$\overline{M}$. The actual output level does not change until $\overline{IOW}$ returns high. (Glitch free output.)

A port can be read out when the latched Chip Enables are active and either $\overline{RD}$ goes low with IO/$\overline{M}$ high, or $\overline{IOR}$ goes low. Both input and output mode bits of a selected port will appear on lines AD_{0-7}.

To clarify the function of the I/O Ports and Data Direction Registers, the following diagram shows the configuration of one bit of PORT A and DDR A. The same logic applies to PORT B and DDR B.

8755A ONE BIT OF PORT A AND DDR A



231735-3

WRITE PA = ($\overline{IOW}$ = 0) • (CHIP ENABLES ACTIVE) • (PORT A ADDRESS SELECTED)
WRITE DDR A = ($\overline{IOW}$ = 0) • (CHIP ENABLES ACTIVE) • (DDR A ADDRESS SELECTED)
READ PA = {(IO/$\overline{M}$ = 1) • ($\overline{RD}$ = 0) + ($\overline{IOR}$ = 0)} • (CHIP ENABLES ACTIVE) • (PORT A ADDRESS SELECTED)

NOTE:
Write PA is not qualified by IO/$\overline{M}$.

Note that hardware RESET or writing a zero to the DDR latch will cause the output latch's output buffer to be disabled, preventing the data in the Output Latch from being passed through to the pin. This is equivalent to putting the port in the input mode. Note also that the data can be written to the Output Latch even though the Output Buffer has been disabled. This enables a port to be initialized with a value prior to enabling the output.

The diagram also shows that the contents of PORT A and PORT B can be read even when the ports are configured as outputs.

ERASURE CHARACTERISTICS

The erasure characteristics of the 8755A are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8755A in approximately 3 years while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 8755A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8755A window to prevent unintentional erasure.

The recommended erasure procedure for the 8755A is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The 8755A should be placed within one inch from the lamp tubes during erasure. Some lamps have a filter on their tubes and this filter should be removed before erasure.

PROGRAMMING

Initially, and after each erasure, all bits of the EPROM portions of the 8755A are in the "1" state. Information is introduced by selectively programming "0" into the desired bit locations. A programmed "0" can only be changed to a "1" by UV erasure.

The 8755A can be programmed on the Intel Universal Programmer (iUP), and iUPF8744A programming module.

The program mode itself consists of programming a single address at a time, giving a single 50 msec pulse for every address. Generally, it is desirable to have a verify cycle after a program cycle for the same address as shown in the attached timing diagram. **In the verify cycle (i.e., normal memory read cycle) 'V_{DD}' should be at +5V.**

SYSTEM APPLICATIONS

System Interface with 8085AH

A system using the 8755A can use either one of the two I/O Interface techniques:

- Standard I/O
- Memory Mapped I/O

If a standard I/O technique is used, the system can use the feature of both CE_2 and $\overline{CE_1}$. By using a combination of unused address lines A_{11-15} and the Chip Enable inputs, the 8085AH system can use up to 5 8755A's without requiring a CE decoder. See Figure 4.

If a memory mapped I/O approach is used the 8755A will be selected by the combination of both the Chip Enables and IO/$\overline{M}$ using AD_{8-15} address lines. See Figure 3.

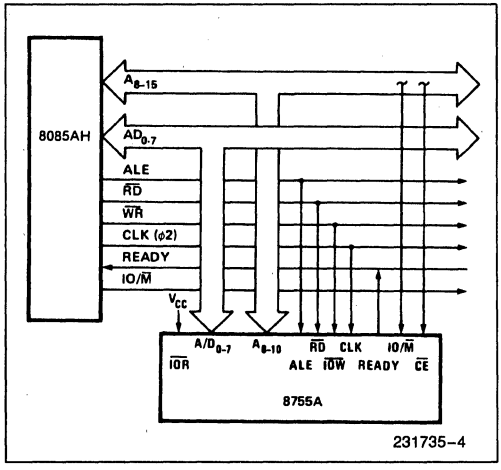
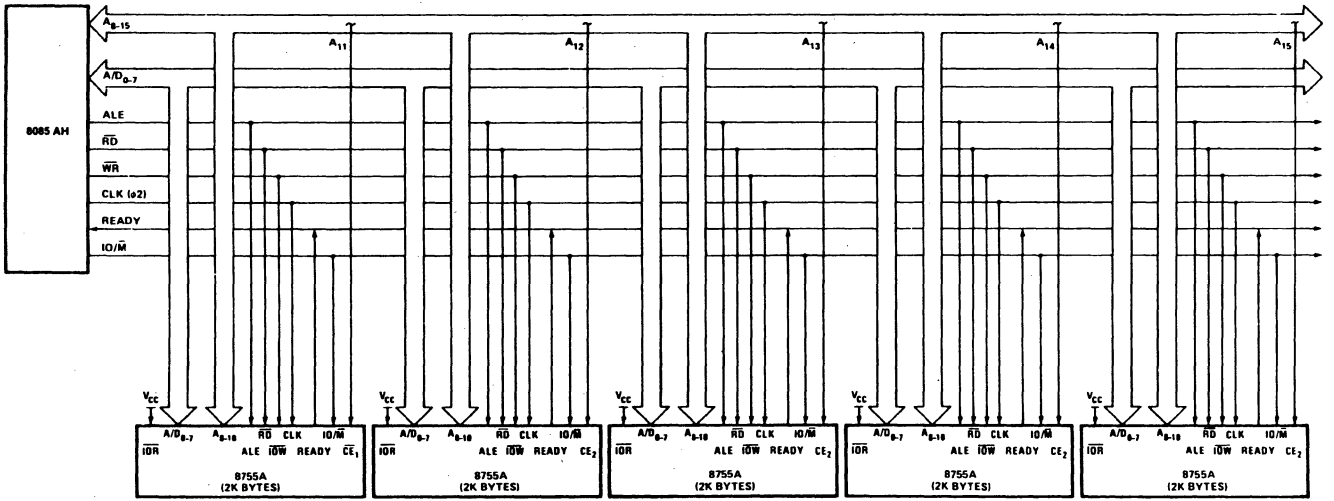


Figure 3. 8755A in 8085AH System (Memory-Mapped I/O)



231735-6

NOTE:
Use $\overline{CE_1}$ for the first 8755A in the system, and CE_2 for the other 8755A's. Permits up to 5-8755A's in a system without CE decoder.

Figure 4. 8755A in 8085AH System (Standard I/O)

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias .....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on any Pin
with Respect to Ground.....−0.5V to +7V
Power Dissipation.....1.5W

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = V_{DD} = 5V ±5%

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	V_{CC} = 5.0V
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	V_{CC} = 5.0V
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{IL}	Input Leakage		10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{LO}	Output Leakage Current		±10	μA	$0.45V \le V_{OUT} \le V_{CC}$
I_{CC}	V_{CC} Supply Current		180	mA	
I_{DD}	V_{DD} Supply Current		30	mA	$V_{DD} = V_{CC}$
C_{IN}	Capacitance of Input Buffer		10	pF	f_C = 1 μHz
$C_{I/O}$	Capacitance of I/O Buffer		15	pF	f_C = 1 μHz

D.C. CHARACTERISTICS—PROGRAMMING

T_A = 0°C to 70°C, V_{CC} = 5V ±5%, V_{SS} = 0V, V_{DD} = 25V ±1V

Symbol	Parameter	Min	Typ	Max	Unit
V_{DD}	Programming Voltage (during Write to EPROM)	24	25	26	V
I_{DD}	Prog Supply Current		15	30	mA

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ±5%

Symbol	Parameter	8755A		Unit
		Min	Max	
t_{CYC}	Clock Cycle Time	320		ns
T_1	CLK Pulse Width	80		ns
T_2	CLK Pulse Width	120		ns
t_f, t_r	CLK Rise and Fall Time		30	ns
t_{AL}	Address to Latch Set Up Time	50		ns
t_{LA}	Address Hold Time after Latch	80		ns
t_{LC}	Latch to READ/WRITE Control	100		ns
t_{RD}	Valid Data Out Delay from READ Control*		170	ns
t_{AD}	Address Stable to Data Out Valid**		450	ns
t_{LL}	Latch Enable Width	100		ns
t_{RDF}	Data Bus Float after READ	0	100	ns
t_{CL}	READ/WRITE Control to Latch Enable	20		ns
t_{CC}	READ/WRITE Control Width	250		ns
t_{DW}	Data in Write Set Up Time	150		ns
t_{WD}	Data in Hold Time after WRITE	30		ns
t_{WP}	WRITE to Port Output		400	ns
t_{PR}	Port Input Set Up Time	50		ns
t_{RP}	Port Input Hold Time to Control	50		ns
t_{RYH}	READY HOLD Time to Control	0	160	ns
t_{ARY}	ADDRESS (CE) to READY		160	ns
t_{RV}	Recovery Time between Controls	300		ns
t_{RDE}	READ Control to Data Bus Enable	10		ns

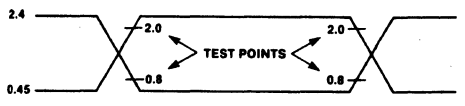
NOTES:
C_{LOAD} = 150 pF.
*Or $T_{AD} - (T_{AL} + T_{LC})$, whichever is greater.
**Defines ALE to Data Out Valid in conjunction with T_{AL}.

A.C. CHARACTERISTICS—PROGRAMMING

T_A = 0°C to 70°C, V_{CC} = 5V ±5%, V_{SS} = 0V, V_{DD} = 25V ±1V

Symbol	Parameter	Min	Typ	Max	Unit
t_{PS}	Data Setup Time	10			ns
t_{PD}	Data Hold Time	0			ns
t_S	Prog Pulse Setup Time	2			μs
t_H	Prog Pulse Hold Time	2			μs
t_{PR}	Prog Pulse Rise Time	0.01	2		μs
t_{PF}	Prog Pulse Fall Time	0.01	2		μs
t_{PRG}	Prog Pulse Width	45	50		ms

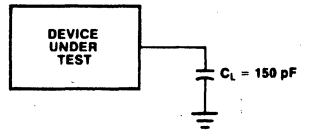
A.C. TESTING INPUT, OUTPUT WAVEFORM



231735-7

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT

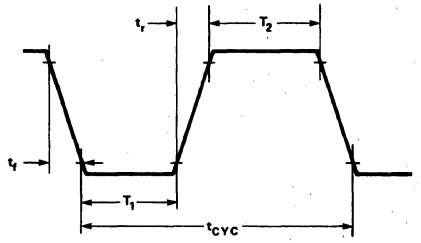


231735-8

C_L = 150 pF
C_L Includes Jig Capacitance

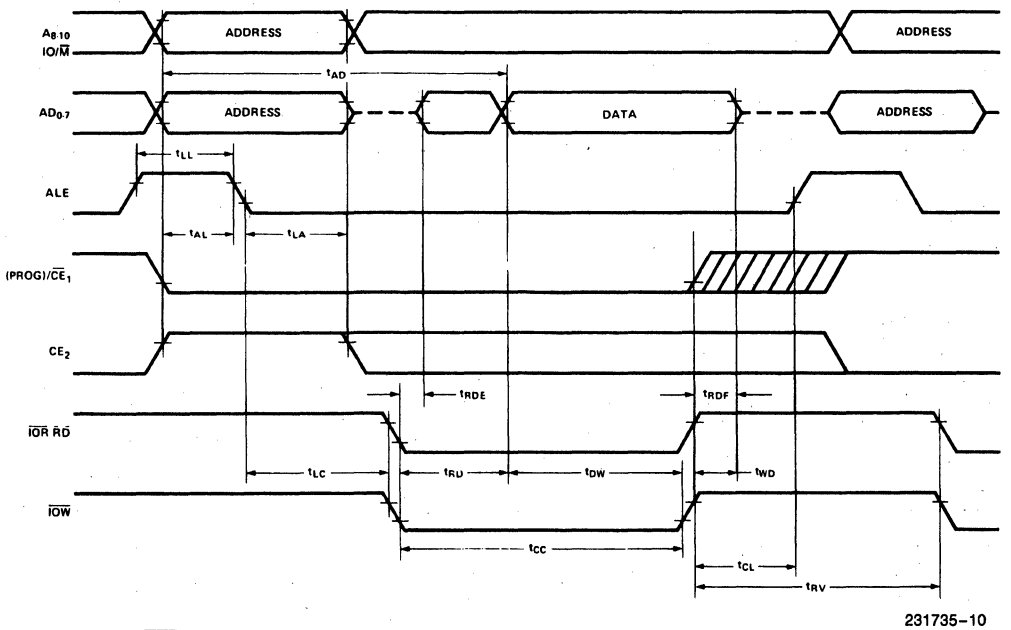
WAVEFORMS

CLOCK SPECIFICATION FOR 8755A



231735-9

PROM READ, I/O READ AND WRITE

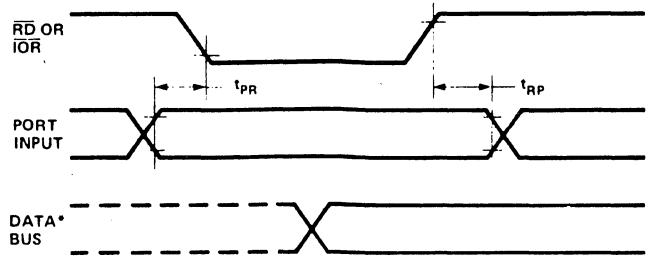


231735-10

Please note that $\overline{CE_1}$ **must remain low for the entire cycle.**

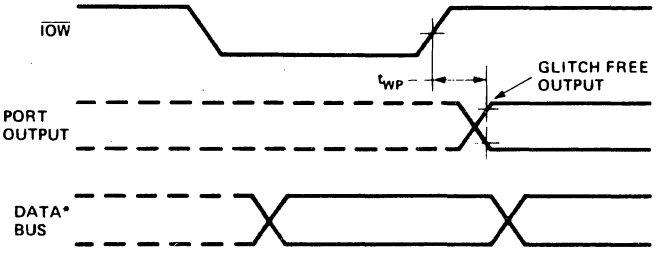
WAVEFORMS (Continued)

I/O PORT



231735-11

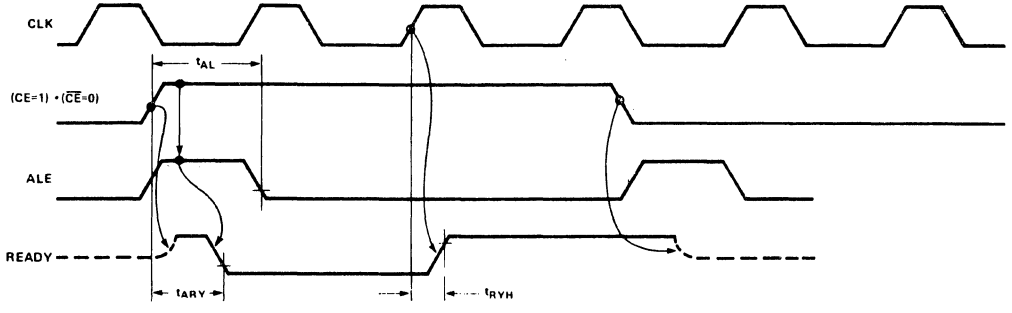
A. Input Mode



231735-12

B. Output Mode

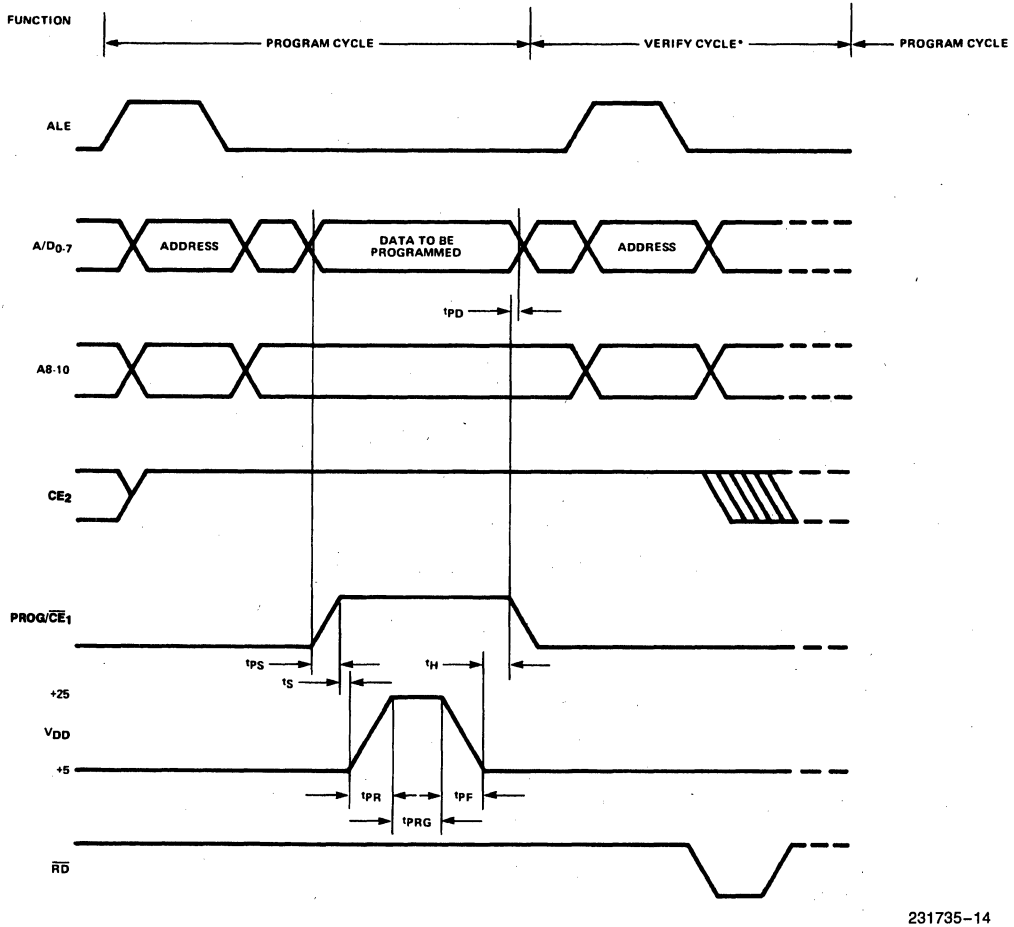
WAIT STATE (READY = 0)



231735-13

WAVEFORMS (Continued)

8755A PROGRAM MODE



*Verify cycle is a regular Memory Read Cycle (with V_{DD} = +5V for 8755A).

Indexes

MCS®-48 INDEX

MCS®-51 INDEX

8XC451 Architectural Index

DOMESTIC SALES OFFICES

ALABAMA

†Intel Corp.
5015 Bradford Dr., #2
Huntsville 35805
Tel: (205) 830-4010

ARIZONA

†Intel Corp.
11225 N. 28th Dr.
Suite D-214
Phoenix 85029
Tel: (602) 869-4980

†Intel Corp.
1161 N. El Dorado Place
Suite 301
Tucson 85715
Tel: (602) 299-6815

CALIFORNIA

†Intel Corp.
21515 Vanowen Street
Suite 116
Canoga Park 91303
Tel: (818) 704-8500

†Intel Corp.
2250 E. Imperial Highway
Suite 218
El Segundo 90245
Tel: (213) 640-6040

†Intel Corp.
1510 Arden Way, Suite 101
Sacramento 95815
Tel: (916) 920-8096

†Intel Corp.
4350 Executive Drive
Suite 105
San Diego 92121
Tel: (619) 452-5880

†Intel Corp.*
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114

†Intel Corp.*
San Tomas 4
2700 San Tomas Expressway
2nd Floor
Santa Clara 95051
Tel: (408) 986-8086
TWX: 910-338-0255
FAX: 408-727-2620

COLORADO

†Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (719) 594-6822

†Intel Corp.*
650 S. Cherry St., Suite 915
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

CONNECTICUT

†Intel Corp.
26 Mill Plain Road
2nd Floor
Danbury 06811
Tel: (203) 748-3130
TWX: 710-456-1199

FLORIDA

†Intel Corp.
6363 N.W. 6th Way, Suite 100
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407
FAX: 305-772-8193

†Intel Corp.
5850 T.G. Lee Blvd.
Suite 340
Orlando 32822
Tel: (407) 240-8000
FAX: 407-240-8097

†Intel Corp.
11300 4th Street North
Suite 170
St. Petersburg 33716
Tel: (813) 577-2413
FAX: 813-578-1607

GEORGIA

†Intel Corp.
3280 Pointe Parkway
Suite 200
Norcross 30092
Tel: (404) 449-0541

ILLINOIS

†Intel Corp.*
300 N. Martingale Road, Suite 400
Schaumburg 60173
Tel: (312) 605-8031
FAX: 312-605-9762

INDIANA

†Intel Corp.
8777 Purdue Road
Suite 125
Indianapolis 46268
Tel: (317) 875-0623

IOWA

Intel Corp.
1930 St. Andrews Drive N.E.
2nd Floor
Cedar Rapids 52402
Tel: (319) 393-5510

KANSAS

†Intel Corp.
10985 Cody St.
Suite 140, Bldg. D
Overland Park 66210
Tel: (913) 345-2727

MARYLAND

†Intel Corp.*
7321 Parkway Drive South
Suite C
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

†Intel Corp.
7833 Walker Drive
Suite 550
Greenbelt 20770
Tel: (301) 441-1020

MASSACHUSETTS

†Intel Corp.*
Westford Corp. Center
3 Carlisle Road
2nd Floor
Westford 01886
Tel: (508) 692-3222
TWX: 710-343-6333

MICHIGAN

†Intel Corp.
7071 Orchard Lake Road
Suite 100
West Bloomfield 48322
Tel: (313) 851-8096

MINNESOTA

†Intel Corp.
3500 W. 80th St., Suite 360
Bloomington 55431
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI

†Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY

†Intel Corp.*
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel: (201) 747-2233

†Intel Corp.
280 Corporate Center
75 Livingston Avenue
First Floor
Roseland 07068
Tel: (201) 740-0111
FAX: 201-740-0626

NEW MEXICO

†Intel Corp.
8500 Menaul Boulevard N.E.
Suite B 295
Albuquerque 87112
Tel: (505) 292-8086

NEW YORK

Intel Corp.
127 Main Street
Binghamton 13905
Tel: (607) 773-0337
FAX: 607-723-2677

†Intel Corp.*
850 Cross Keys Office Park
Fairport 14450
Tel: (716) 425-2750
TWX: 510-253-7391

†Intel Corp.*
2950 Expressway Dr., South
Suite 130
Islandia 11722
Tel: (516) 231-3300
TWX: 510-227-6236

†Intel Corp.
Westage Business Center
Bldg. 300, Route 9
Fishkill 12524
Tel: (914) 897-3860
FAX: 914-897-3125

NORTH CAROLINA

†Intel Corp.
5800 Executive Center Dr.
Suite 105
Charlotte 28212
Tel: (704) 568-8966
FAX: 704-535-2236

†Intel Corp.
2700 Wycliff Road
Suite 102
Raleigh 27607
Tel: (919) 781-8022

OHIO

†Intel Corp.*
3401 Park Center Drive
Suite 220
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528

†Intel Corp.*
25700 Science Park Dr., Suite 100
Beachwood 44122
Tel: (216) 464-2736
TWX: 810-427-9298

OKLAHOMA

†Intel Corp.
6801 N. Broadway
Suite 115
Oklahoma City 73162
Tel: (405) 848-8086

OREGON

†Intel Corp.
15254 N.W. Greenbrier Parkway
Building B
Beaverton 97006
Tel: (503) 645-8051
TWX: 910-467-8741

PENNSYLVANIA

†Intel Corp.*
455 Pennsylvania Avenue
Suite 230
Fort Washington 19034
Tel: (215) 641-1000
TWX: 510-661-2077

Intel Corp.*
400 Penn Center Blvd., Suite 610
Pittsburgh 15235
Tel: (412) 823-4970

PUERTO RICO

†Intel Microprocessor Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

†Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

†Intel Corp.*
12000 Ford Road
Suite 400
Dallas 75234
Tel: (214) 241-8087
FAX: 214-484-1180

†Intel Corp.*
7322 S.W. Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8086
TWX: 910-881-2490

UTAH

†Intel Corp.
428 East 6400 South
Suite 104
Murray 84107
Tel: (801) 263-8051

VIRGINIA

†Intel Corp.
1504 Santa Rosa Road
Suite 108
Richmond 23288
Tel: (804) 282-5668

WASHINGTON

†Intel Corp.
155 108th Avenue N.E.
Suite 386
Bellevue 98004
Tel: (206) 453-8086
TWX: 910-443-3002

†Intel Corp.
408 N. Mullan Road
Suite 102
Spokane 99206
Tel: (509) 928-8086

WISCONSIN

†Intel Corp.
330 S. Executive Dr.
Suite 102
Brookfield 53005
Tel: (414) 784-8087
FAX: (414) 796-2115

CANADA

BRITISH COLUMBIA

Intel Semiconductor of Canada, Ltd.
4585 Canada Way, Suite 202
Burnaby V5G 4L6
Tel: (604) 298-0387
FAX: (604) 298-8234

ONTARIO

†Intel Semiconductor of Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel: (613) 829-9714
TLX: 053-4115

†Intel Semiconductor of Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel: (416) 675-2105
TLX: 06983574
FAX: (416) 675-2438

QUEBEC

†Intel Semiconductor of Canada, Ltd.
620 St. John Boulevard
Pointe Claire H9R 3K2
Tel: (514) 694-9130
TWX: 514-694-9134

†Sales and Service Office
*Field Application Location

DOMESTIC DISTRIBUTORS

ALABAMA

Arrow Electronics, Inc.
1015 Henderson Road
Huntsville 35805
Tel: (205) 837-6955

†Hamilton/Avnet Electronics
4940 Research Drive
Huntsville 35805
Tel: (205) 837-7210
TWX: 810-726-2162

Pioneer/Technologies Group, Inc.
4825 University Square
Huntsville 35805
Tel: (205) 837-9300
TWX: 810-726-2197

ARIZONA

†Hamilton/Avnet Electronics
505 S. Madison Drive
Tempe 85281
Tel: (602) 231-5140
TWX: 910-950-0077

Hamilton/Avnet Electronics
30 South McKiemy
Chandler 85226
Tel: (602) 961-6669
TWX: 910-950-0077

Arrow Electronics, Inc.
4134 E. Wood Street
Phoenix 85040
Tel: (602) 437-0750
TWX: 910-951-1550

Wyle Distribution Group
17855 N. Black Canyon Hwy.
Phoenix 85023
Tel: (602) 249-2232
TWX: 910-951-4282

CALIFORNIA

Arrow Electronics, Inc.
10824 Hope Street
Cypress 90630
Tel: (714) 220-6300

Arrow Electronics, Inc.
19748 Dearborn Street
Chatsworth 91311
Tel: (213) 701-7500
TWX: 910-493-2086

†Arow Electronics, Inc.
521 Weddell Drive
Sunnyvale 94086
Tel: (408) 745-6600
TWX: 910-339-9371

Arrow Electronics, Inc.
9511 Ridgehaven Court
San Diego 92123
Tel: (619) 565-4800
TWX: 888-064

†Arrow Electronics, Inc.
2961 Dow Avenue
Tustin 92680
Tel: (714) 838-5422
TWX: 910-595-2860

†Avnet Electronics
350 McCormick Avenue
Costa Mesa 92626
Tel: (714) 754-6071
TWX: 910-595-1928

†Hamilton/Avnet Electronics
1175 Bordeaux Drive
Sunnyvale 94086
Tel: (408) 743-3300
TWX: 910-339-9332

†Hamilton/Avnet Electronics
4545 Ridgeview Avenue
San Diego 92123
Tel: (619) 571-7500
TWX: 910-595-2638

†Hamilton/Avnet Electronics
9650 Desoto Avenue
Chatsworth 91311
Tel: (818) 700-1161

†Hamilton Electro Sales
10950 W. Washington Blvd.
Culver City 20230
Tel: (213) 558-2458
TWX: 910-340-6364

Hamilton Electro Sales
1361B West 190th Street
Gardena 90248
Tel: (213) 217-6700

†Hamilton/Avnet Electronics
3002 'G' Street
Ontario 91761
Tel: (714) 989-9411

†Avnet Electronics
20501 Plummer
Chatsworth 91351
Tel: (213) 700-6271
TWX: 910-494-2207

CALIFORNIA (Cont'd.)

†Hamilton Electro Sales
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4150
TWX: 910-595-2638

†Hamilton/Avnet Electronics
4103 Northgate Blvd.
Sacramento 95834
Tel: (916) 920-3150

Wyle Distribution Group
124 Maryland Street
El Segundo 90254
Tel: (213) 322-8100

Wyle Distribution Group
7382 Lampson Ave.
Garden Grove 92641
Tel: (714) 891-1717
TWX: 910-348-7140 or 7111

Wyle Distribution Group
11151 Sun Center Drive
Rancho Cordova 95670
Tel: (916) 638-5282

†Wyle Distribution Group
9525 Chesapeake Drive
San Diego 92123
Tel: (619) 565-9171
TWX: 910-335-1590

†Wyle Distribution Group
3000 Bowers Avenue
Santa Clara 95051
Tel: (408) 727-2500
TWX: 910-338-0296

†Wyle Distribution Group
17872 Cowan Avenue
Irvine 92714
Tel: (714) 863-9953
TWX: 910-595-1572

Wyle Distribution Group
26677 W. Agoura Rd.
Calabasas 91302
Tel: (818) 880-9000
TWX: 372-0232

COLORADO

Arrow Electronics, Inc.
7060 South Tucson Way
Englewood 80112
Tel: (303) 790-4444

†Hamilton/Avnet Electronics
8765 E. Orchard Road
Suite 708
Englewood 80111
Tel: (303) 740-1017
TWX: 910-935-0787

†Wyle Distribution Group
451 E. 124th Avenue
Thornton 80241
Tel: (303) 457-9953
TWX: 910-936-0770

CONNECTICUT

†Arrow Electronics, Inc.
12 Beaumont Road
Wallingford 06492
Tel: (203) 265-7741
TWX: 710-476-0162

Hamilton/Avnet Electronics
Commerce Industrial Park
Commerce Drive
Danbury 06810
Tel: (203) 797-2800
TWX: 710-456-9974

†Pioneer Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515
TWX: 710-468-3373

FLORIDA

†Arrow Electronics, Inc.
400 Fairway Drive
Suite 102
Deerfield Beach 33441
Tel: (305) 429-8200
TWX: 510-955-9456

Arrow Electronics, Inc.
37 Skyline Drive
Suite 3101
Lake Marv 32746
Tel: (407) 323-0252
TWX: 510-959-6337

†Hamilton/Avnet Electronics
6801 N.W. 15th Way
Ft. Lauderdale 33309
Tel: (305) 971-2900
TWX: 510-956-3097

†Hamilton/Avnet Electronics
3197 Tech Drive North
St. Petersburg 33702
Tel: (813) 576-3930
TWX: 810-863-0374

FLORIDA (Cont'd.)

†Hamilton/Avnet Electronics
6947 University Boulevard
Winter Park 32792
Tel: (305) 628-3888
TWX: 810-853-0322

†Pioneer/Technologies Group, Inc.
337 S. Lake Blvd.
Alta Monte Springs 32701
Tel: (407) 834-9090
TWX: 810-853-0284

Pioneer/Technologies Group, Inc.
674 S. Military Trail
Deerfield Beach 33442
Tel: (305) 428-8877
TWX: 510-955-9653

GEORGIA

†Arrow Electronics, Inc.
3155 Northwoods Parkway
Suite A
Norcross 30071
Tel: (404) 449-8252
TWX: 810-766-0439

†Hamilton/Avnet Electronics
5825 D Peachtree Corners
Norcross 30092
Tel: (404) 447-7500
TWX: 810-766-0432

Pioneer/Technologies Group, Inc.
3100 F Northwoods Place
Norcross 30071
Tel: (404) 448-1711
TWX: 810-766-4515

ILLINOIS

Arrow Electronics, Inc.
1140 W. Thorndale
Itasca 60143
Tel: (312) 250-0500
TWX: 312-250-0916

†Hamilton/Avnet Electronics
1130 Thorndale Avenue
Bensenville 60106
Tel: (312) 860-7780
TWX: 910-227-0060

MTI Systems Sales
1100 W. Thorndale
Itasca 60143
Tel: (312) 773-2300

†Pioneer Electronics
1551 Carmen Drive
Elk Grove Village 60007
Tel: (312) 437-9680
TWX: 910-222-1834

INDIANA

†Arrow Electronics, Inc.
2495 Directors Row, Suite H
Indianapolis 46241
Tel: (317) 243-9353
TWX: 810-341-3119

Hamilton/Avnet Electronics
485 Gradle Drive
Carmel 46032
Tel: (317) 844-9333
TWX: 810-260-3966

†Pioneer Electronics
6408 Castleplace Drive
Indianapolis 46250
Tel: (317) 849-7300
TWX: 810-260-1794

IOWA

Hamilton/Avnet Electronics
915 33rd Avenue, S.W.
Cedar Rapids 52404
Tel: (319) 362-4757

KANSAS

Arrow Electronics
8208 Melrose Dr., Suite 210
Lenexa 66214
Tel: (913) 541-9542

†Hamilton/Avnet Electronics
9219 Quivera Road
Overland Park 66215
Tel: (913) 888-8900
TWX: 910-743-0005

Pioneer/Tec Gr.
10551 Lockman Rd.
Lenexa 66215
Tel: (913) 492-0500

KENTUCKY

Hamilton/Avnet Electronics
1051 D. Newton Park
Lexington 40511
Tel: (606) 259-1475

MARYLAND

Arrow Electronics, Inc.
8300 Guilford Drive
Suite H, River Center
Columbia 21046
Tel: (301) 995-0003
TWX: 710-236-9005

Hamilton/Avnet Electronics
6822 Oak Hall Lane
Columbia 21045
Tel: (301) 995-3500
TWX: 710-862-1861

†Mesa Technology Corp.
9720 Patuxent Woods Dr.
Columbia 21046
Tel: (301) 290-8150
TWX: 710-828-9702

†Pioneer/Technologies Group, Inc.
9100 Gaither Road
Gaithersburg 20877
Tel: (301) 921-0660
TWX: 710-828-0545

MASSACHUSETTS

Arrow Electronics, Inc.
25 Upton Dr.
Wilmington 01887
Tel: (617) 935-5134

†Hamilton/Avnet Electronics
100 Centennial Drive
Peabody 01960
Tel: (617) 531-7430
TWX: 710-393-0382

MTI Systems Sales
83 Cambridge St.
Burlington 01813

Pioneer Electronics
44 Hartwell Avenue
Lexington 02173
Tel: (617) 861-9200
TWX: 710-326-6617

MICHIGAN

Arrow Electronics, Inc.
755 Phoenix Drive
Ann Arbor 48104
Tel: (313) 971-8220
TWX: 810-223-6020

Hamilton/Avnet Electronics
2215 29th Street S.E.
Space A5
Grand Rapids 49508
Tel: (616) 243-8805
TWX: 810-274-6921

Pioneer Electronics
4504 Broadmoor S.E.
Grand Rapids 49508
FAX: 616-698-1831

†Hamilton/Avnet Electronics
32487 Schoolcraft Road
Livonia 48150
Tel: (313) 522-4700
TWX: 810-282-8775

†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel: (313) 525-1800
TWX: 810-242-3271

MINNESOTA

†Arrow Electronics, Inc.
5230 W. 73rd Street
Edina 55435
Tel: (612) 830-1800
TWX: 910-576-3125

†Hamilton/Avnet Electronics
12400 Whitewater Drive
Minnetonka 55434
Tel: (612) 932-0600

†Pioneer Electronics
7625 Golden Triange Dr.
Suite G
Eden Prairi 55343
Tel: (612) 944-3355

MISSOURI

†Arrow Electronics, Inc.
2380 Schuetz
St. Louis 63141
Tel: (314) 567-6888
TWX: 910-764-0882

†Hamilton/Avnet Electronics
13743 Shoreline Court
Earth City 63045
Tel: (314) 344-1200
TWX: 910-762-0684

NEW HAMPSHIRE

†Arrow Electronics, Inc.
3 Perimeter Road
Manchester 03103
Tel: (603) 668-6968
TWX: 710-220-1684

†Hamilton/Avnet Electronics
444 E. Industrial Drive
Manchester 03103
Tel: (603) 624-9400

NEW JERSEY

†Arrow Electronics, Inc.
Four East Stow Road
Unit 11
Marlton 08053
Tel: (609) 596-8000
TWX: 710-897-0829

†Arrow Electronics
6 Century Drive
Parsipanny 07054
Tel: (201) 538-0900

†Hamilton/Avnet Electronics
1 Keystone Ave., Bldg. 36
Cherry Hill 08003
Tel: (609) 424-0110
TWX: 710-940-0262

†Hamilton/Avnet Electronics
10 Industrial
Fairfield 07006
Tel: (201) 575-5300
TWX: 710-734-4388

†MTI Systems Sales
37 Kulick Rd.
Fairfield 07006
Tel: (201) 227-5552

†Pioneer Electronics
45 Route 46
Pinebrook 07058
Tel: (201) 575-3510
TWX: 710-734-4382

NEW MEXICO

Alliance Electronics Inc.
11030 Cochiti S.E.
Albuquerque 87123
Tel: (505) 292-3360
TWX: 910-989-1151

Hamilton/Avnet Electronics
2524 Baylor Drive S.E.
Albuquerque 87106
Tel: (505) 765-1500
TWX: 910-989-0614

NEW YORK

†Arrow Electronics, Inc.
3375 Brighton Henrietta Townline Rd.
Rochester 14623
Tel: (716) 275-0300
TWX: 510-253-4766

Arrow Electronics, Inc.
20 Oser Avenue
Hauppauge 11788
Tel: (516) 231-1000
TWX: 510-227-6623

Hamilton/Avnet
933 Motor Parkway
Hauppauge 11788
Tel: (516) 231-9800
TWX: 510-224-6166

†Hamilton/Avnet Electronics
333 Metro Park
Rochester 14623
Tel: (716) 475-9130
TWX: 510-253-5470

†Hamilton/Avnet Electronics
103 Twin Oaks Drive
Syracuse 13206
Tel: (315) 437-2088
TWX: 710-541-1560

†MTI Systems Sales
38 Harbor Park Drive
Port Washington 11050
Tel: (516) 621-6200

†Pioneer Electronics
68 Corporate Drive
Binghamton 13904
Tel: (607) 722-9300
TWX: 510-252-0893

Pioneer Electronics
40 Oser Avenue
Hauppauge 11787
Tel: (516) 231-9200

DOMESTIC DISTRIBUTORS (Cont'd.)

NEW YORK (Cont'd.)

†Pioneer Electronics
60 Crossway Park West
Woodbury, Long Island 11797
Tel: (516) 921-8700
TWX: 510-221-2184

†Pioneer Electronics
840 Fairport Park
Fairport 14450
Tel: (716) 381-7070
TWX: 510-253-7001

NORTH CAROLINA

†Arrow Electronics, Inc.
5240 Greensdairy Road
Raleigh 27604
Tel: (919) 876-3132
TWX: 510-928-1856

†Hamilton/Avnet Electronics
3510 Spring Forest Drive
Raleigh 27604
Tel: (919) 878-0819
TWX: 510-928-1836

Pioneer/Technologies Group, Inc.
9801 A-Southern Pine Blvd.
Charlotte 28210
Tel: (919) 527-8188
TWX: 810-621-0366

OHIO

Arrow Electronics, Inc.
7620 McEwen Road
Centerville 45459
Tel: (513) 435-5563
TWX: 810-459-1611

†Arrow Electronics, Inc.
6238 Cochran Road
Solon 44139
Tel: (216) 248-3990
TWX: 810-427-9409

†Hamilton/Avnet Electronics
954 Senate Drive
Dayton 45459
Tel: (513) 439-6733
TWX: 810-450-2531

Hamilton/Avnet Electronics
4588 Emery Industrial Pkwy.
Warrensville Heights 44128
Tel: (216) 349-5100
TWX: 810-427-9452

†Hamilton/Avnet Electronics
777 Brooksedge Blvd.
Westerville 43081
Tel: (614) 882-7004

†Pioneer Electronics
4433 Interpoint Boulevard
Dayton 45424
Tel: (513) 236-9900
TWX: 810-459-1622

†Pioneer Electronics
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600
TWX: 810-422-2211

OKLAHOMA

Arrow Electronics, Inc.
1211 E. 51st Street
Suite 101
Tulsa 74146
Tel: (918) 252-7537

†Hamilton/Avnet Electronics
12121 E. 51st St., Suite 102A
Tulsa 74146
Tel: (918) 252-7297

OREGON

†Almac Electronics Corp.
1885 N.W. 169th Place
Beaverton 97005
Tel: (503) 629-8090
TWX: 910-467-8746

†Hamilton/Avnet Electronics
6024 S.W. Jean Road
Bldg. C, Suite 10
Lake Oswego 97034
Tel: (503) 635-7848
TWX: 910-455-8179

Wyle Distribution Group
5250 N.E. Elam Young Parkway
Suite 600
Hillsboro 97124
Tel: (503) 640-6000
TWX: 910-460-2203

PENNSYLVANIA

Arrow Electronics, Inc.
650 Seco Road
Monroeville 15146
Tel: (412) 856-7000

Hamilton/Avnet Electronics
2800 Liberty Ave.
Pittsburgh 15238
Tel: (412) 281-4150

Pioneer Electronics
259 Kappa Drive
Pittsburgh 15238
Tel: (412) 782-2300
TWX: 710-795-3122

†Pioneer/Technologies Group, Inc.
Delaware Valley
261 Gibralter Road
Horsham 19044
Tel: (215) 674-4000
TWX: 510-665-6778

TEXAS

†Arrow Electronics, Inc.
3220 Commander Drive
Carrollton 75006
Tel: (214) 380-6464
TWX: 910-860-5377

†Arrow Electronics, Inc.
10899 Kinghurst
Suite 100
Houston 77099
Tel: (713) 530-4700
TWX: 910-880-4439

†Arrow Electronics, Inc.
2227 W. Braker Lane
Austin 78758
Tel: (512) 835-4180
TWX: 910-874-1348

†Hamilton/Avnet Electronics
1807 W. Braker Lane
Austin 78758
Tel: (512) 837-8911
TWX: 910-874-1319

TEXAS (Cont'd.)

†Hamilton/Avnet Electronics
2111 W. Walnut Hill Lane
Irving 75038
Tel: (214) 550-6111
TWX: 910-860-5929

†Hamilton/Avnet Electronics
4850 Wright Rd., Suite 190
Stafford 77477
Tel: (713) 240-7733
TWX: 910-881-5523

†Pioneer Electronics
18260 Kramer
Austin 78758
Tel: (512) 835-4000
TWX: 910-874-1323

†Pioneer Electronics
13710 Omega Road
Dallas 75234
Tel: (214) 386-7300
TWX: 910-850-5563

†Pioneer Electronics
5853 Point West Drive
Houston 77036
Tel: (713) 988-5555
TWX: 910-881-1606

Wyle Distribution Group
1810 Greenville Avenue
Richardson 75081
Tel: (214) 235-9953

UTAH

Arrow Electronics
1946 Parkway Blvd.
Salt Lake City 84119
Tel: (801) 973-6913

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City 84119
Tel: (801) 972-2800
TWX: 910-925-4018

Wyle Distribution Group
1325 West 2200 South
Suite E
West Valley 84119
Tel: (801) 974-9953

WASHINGTON

†Almac Electronics Corp.
14360 S.E. Eastgate Way
Bellevue 98007
Tel: (206) 643-9992
TWX: 910-444-2067

Arrow Electronics, Inc.
19540 68th Ave. South
Kent 98032
Tel: (206) 575-4420

†Hamilton/Avnet Electronics
14212 N.E. 21st Street
Bellevue 98005
Tel: (206) 643-3950
TWX: 910-443-2469

Wyle Distribution Group
15385 N.E. 90th Street
Redmond 98052
Tel: (206) 881-1150

WISCONSIN

Arrow Electronics, Inc.
200 N. Patrick Blvd., Ste. 100
Brookfield 53005
Tel: (414) 767-6600
TWX: 910-262-1193

Hamilton/Avnet Electronics
2975 Moorland Road
New Berlin 53151
Tel: (414) 784-4510
TWX: 910-262-1182

CANADA

ALBERTA

Hamilton/Avnet Electronics
2616 21st Street N.E.
Calgary T2E 6Z3
Tel: (403) 230-3586
TWX: 03-827-642

Zentronics
Bay No. 1
3300 14th Avenue N.E.
Calgary T2A 6J4
Tel: (403) 272-1021

BRITISH COLUMBIA

†Hamilton/Avnet Electronics
105-2550 Boundary
Burmalay V5M 3Z3
Tel: (604) 437-6667

Zentronics
108-11400 Bridgeport Road
Richmond V6X 1T2
Tel: (604) 273-5575
TWX: 04-5077-89

MANITOBA

Zentronics
60-1313 Border Unit 60
Winnipeg R3H 0X4
Tel: (204) 694-1957

ONTARIO

Arrow Electronics, Inc.
36 Antares Dr.
Nepean K2E 7W5
Tel: (613) 226-6903

Arrow Electronics, Inc.
1093 Meyerside
Mississauga L5T 1M4
Tel: (416) 673-7769
TWX: 06-218213

†Hamilton/Avnet Electronics
6845 Rexwood Road
Units 3-4-5
Mississauga L4T 1R2
Tel: (416) 677-7432
TWX: 610-492-8867

Hamilton/Avnet Electronics
6845 Rexwood Road
Unit 6
Mississauga L4T 1R2
Tel: (416) 277-0484

ONTARIO (Cont'd.)

†Hamilton/Avnet Electronics
190 Colonnade Road South
Nepean K2E 7L5
Tel: (613) 226-1700
TWX: 05-349-71

†Zentronics
8 Tilbury Court
Brampton L6T 3T4
Tel: (416) 451-9600
TWX: 06-976-78

†Zentronics
155 Colonnade Road
Unit 17
Nepean K2E 7K1
Tel: (613) 226-8840

Zentronics
60-1313 Border St.
Winnipeg R3H 0I4
Tel: (204) 694-7957

QUEBEC

†Arrow Electronics Inc.
4050 Jean Talon Quest
Montreal H4P 1W1
Tel: (514) 735-5511
TWX: 05-25590

Arrow Electronics, Inc.
909 Charest Blvd.
Quebec J1N 2C9
Tel: (418) 687-4231
TWX: 05-13388

Hamilton/Avnet Electronics
2795 Halpern
St. Laurent H2E 7K1
Tel: (514) 335-1000
TWX: 610-421-3731

Zentronics
817 McCaffrey
St. Laurent H4T 1M3
Tel: (514) 737-9700
TWX: 05-827-535

†Microcomputer System Technical Distributor Center

EUROPEAN SALES OFFICES

DENMARK

Intel
Glentevej 61, 3rd Floor
2400 Copenhagen NV
Tel: (45) (01) 19 80 33
TLX: 19567

FINLAND

Intel
Ruosilantie 2
00390 Helsinki
Tel: (358) 0 544 644
TLX: 123332

FRANCE

Intel
1, Rue Edison-BP 303
78054 St. Quentin-en-Yvelines Cedex
Tel: (33) (1) 30 57 70 00
TLX: 699016

Intel
4, Quai des Etroits
69321 Lyon Cedex 05
Tel: (33) (16) 78 42 40 89
TLX: 305153

WEST GERMANY

Intel*
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen
Tel: (49) 089/90992-0
TLX: 5-23177
FAX: 904-3948

Intel
Hohenzollern Strasse 5
3000 Hannover 1
Tel: (49) 0511/344081
TLX: 9-23625

Intel
Abraham Lincoln Strasse 16-18
6200 Wiesbaden
Tel: (49) 06121/7605-0
TLX: 4-186183

Intel
Zettachring 10A
7000 Stuttgart 80
Tel: (49) 0711/7287-0
TLX: 7-254826

ISRAEL

Intel*
Atidim Industrial Park-Neve Sharet
P.O. Box 43202
Tel-Aviv 61430
Tel: (972) 03-498080
TLX: 371215

ITALY

Intel*
Milanofiori Palazzo E
20090 Assago
Milano
Tel: (39) (02) 824 40 71
TLX: 341286

NETHERLANDS

Intel*
Marten Meesweg 93
3068 AV Rotterdam
Tel: (31) (0)10-421.23.77
TLX: 22283

NORWAY

Intel
Hvamveien 4-PO Box 92
2013 Skjetten
Tel: (47) (6) 842 420
TLX: 76018

SPAIN

Intel
Zurbaran, 28
28010 Madrid
Tel: (34) 410 40 04
TLX: 46880

SWEDEN

Intel*
Dalvagen 24
171 36 Solna
Tel: (46) 8 734 01 00
TLX: 12261

SWITZERLAND

Intel
Zuerichstrasse
8185 Winkel-Rueti bei Zuerich
Tel: (41) 01/860 62 62
TLX: 825977

UNITED KINGDOM

Intel*
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel: (44) (0793) 696000
TLX: 444447/8

EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA

Bacher Electronics G.m.b.H.
Rotenmuehlgasse 26
1120 Wien
Tel: (43) (0222) 83 56 46-0
TLX: 131532

BELGIUM

Inelco Belgium S.A.
Av. des Croix de Guerre 94
1120 Bruxelles
Oorlogskruisenlaan, 94
1120 Brussel
Tel: (32) (02) 216 01 60
TLX: 64475

DENMARK

ITT-Multikomponent
Naverland 29
2600 Glostrup
Tel: (45) (0) 2 45 66 45
TLX: 33 355

FINLAND

OY Fintronic AB
Melkonkatu 24A
00210 Helsinki
Tel: (358) (0) 6926022
TLX: 124224

FRANCE

Generim
Z.A. de Courtaboeuf
Av. de la Baltique-BP 88
91943 Les Ulis Cedex
Tel: (33) (1) 69 07 78 78
TLX: 691700

Jermyn
73-79, rue des Solets
Silic 585
94663 Rungis Cedex
Tel: (33) (1) 45 60 04 00
TLX: 260967

Metrologie
Tour d'Asnieres
4, av. Laurent-Cely
92606 Asnieres Cedex
Tel: (33) (1) 47 90 62 40
TLX: 611448

Tekelec-Airtronic
Rue Carle Vernet - BP 2
92315 Sevres Cedex
Tel: (33) (1) 45 34 75 35
TLX: 204552

WEST GERMANY

Electronic 2000 AG
Stahlgruberring 12
8000 Muenchen 82
Tel: (49) 089/42001-0
TLX: 522561

ITT Multikomponent GmbH
Postfach 1265
Bahnhofstrasse 44
7141 Moeglingen
Tel: (49) 07141/4879
TLX: 7264472

Jermyn GmbH
Im Dachstueck 9
6250 Limburg
Tel: (49) 06431/508-0
TLX: 415257-0

Metrologie GmbH
Meglingerstrasse 49
8000 Muenchen 71
Tel: (49) 089/78042-0
TLX: 5213189

Proelectron Vertriebs GmbH
Max Planck Strasse 1-3
6072 Dreieich
Tel: (49) 06103/3040
TLX: 417903

IRELAND

Micro Marketing Ltd.
Glenageary Office Park
Glenageary
Co. Dublin
Tel: (21) (353) (01) 85 63 25
TLX: 31584

ISRAEL

Eastronics Ltd.
11 Rozanis Street
P.O.B. 39300
Tel-Aviv 61392
Tel: (972) 03-475151
TLX: 33638

ITALY

Intesi
Divisione ITT Industries GmbH
Viale Milanofiori
Palazzo E/5
20090 Assago
Milano
Tel: (39) 02/824701
TLX: 311351

Lasi Elettronica S.p.A.
V. le Fulvio Testi, 126
20092 Cinisello Balsamo
Milano
Tel: (39) 02/2440012
TLX: 352040

NETHERLANDS

Koning en Hartman
1 Energieweg
2627 AP Delft
Tel: (31) 15609906
TLX: 38250

NORWAY

Nordisk Elektronikk (Norge) A/S
Postboks 123
Smedsvingen 4
1364 Hvalstad
Tel: (47) (02) 84 62 10
TLX: 77546

PORTUGAL

Ditram
Avenida Marques de Tomar, 46-A
1000 Lisboa
Tel: (351) (1) 73 48 34
TLX: 14182

SPAIN

ATD Electronica, S.A.
Plaza Ciudad de Viena, 6
28040 Madrid
Tel: (34) 234 40 00
TLX: 42754

ITT-SESA
Calle Miguel Angel, 21-3
28010 Madrid
Tel: (34) 419 09 57
TLX: 27461

SWEDEN

Nordisk Elektronik AB
Huvudstagatan 1
Box 1409
171 27 Solna
Tel: (46) (8)-734 97 70
TLX: 105 47

SWITZERLAND

Industrade A.G.
Hertistrasse 31
8304 Wallisellen
Tel: (41) (01) 83 05 04 0
TLX: 56788

TURKEY

EMPA Electronic
Lindwurmstrasse 95A
8000 Muenchen 2
Tel: (49) 089/53 80 570
TLX: 528573

UNITED KINGDOM

Accent Electronic Components Ltd.
Jubilee House, Jubilee Road
Letchworth, Herts SG6 1TL
Tel: (44) (0462) 686666
TLX: 826293

Bytech-Comway Systems
3 The Western Centre
Western Road
Bracknell RG12 1RW
Tel: (44) (0344) 55333
TLX: 847201

Jermyn
Vestry Estate
Otford Road
Sevenoaks
Kent TN14 5EU
Tel: (44) (0732) 450144
TLX: 95142

MMD
Unit 8 Southview Park
Caversham
Reading
Berkshire RG4 0AF
Tel: (44) (0734) 48 16 66
TLX: 846669

Rapid Silicon
Rapid House
Denmark Street
High Wycombe
Buckinghamshire HP11 2ER
Tel: (44) (0494) 442266
TLX: 837931

Rapid Systems
Rapid House
Denmark Street
High Wycombe
Buckinghamshire HP11 2ER
Tel: (44) (0494) 450244
TLX: 837931

YUGOSLAVIA

H.R. Microelectronics Corp.
2005 de la Cruz Blvd., Ste. 223
Santa Clara, CA 95050
U.S.A.
Tel: (1) (408) 988-0286
TLX: 387452

*Field Application Location

INTERNATIONAL SALES OFFICES

AUSTRALIA

Intel Australia Pty. Ltd.*
Spectrum Building
200 Pacific Hwy., Level 6
Crows Nest, NSW, 2065
Tel: (2) 957-2744
TLX: AA 20097
FAX: (2) 923-2632

BRAZIL

Intel Semicondutores do Brasil LTDA
Av. Paulista, 1159-CJS 404/405
01311 - Sao Paulo - S.P.
Tel: 55-11-287-5899
TLX: 1153148 SAPI BR
FAX: 55-11-212-7631

CHINA/HONG KONG

Intel PRC Corporation
15/F, Office 1, Citic Bldg.
Jian Guo Men Wai Street
Beijing, PRC
Tel: (1) 500-4850
TLX: 22947 INTEL CN
FAX: (1) 500-2953

Intel Semiconductor Ltd.*
10/F East Tower
Bond Center
Queensway, Central
Hong Kong
Tel: (5) 8444-555
TLX: 63869 ISHLHK HX
FAX: (5) 8681-989

JAPAN

Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26
Tel: 029747-8511
TLX: 3656-160
FAX: 029747-8450

Intel Japan K.K.*
Daiichi Mitsugi Bldg.
1-8889 Fuchu-cho
Fuchu-shi, Tokyo 183
Tel: 0423-60-7871
FAX: 0423-60-0315

Intel Japan K.K.*
Flower-Hill Shin-machi Bldg.
1-23-9 Shinmachi
Setagaya-ku, Tokyo 154
Tel: 03-426-2231
FAX: 03-427-7620

Intel Japan K.K.*
Bldg. Kumagaya
2-69 Hon-cho
Kumagaya-shi, Saitama 360
Tel: 0485-24-6871
FAX: 0485-24-7518

Intel Japan K.K.*
Mitsui-Seimei Musashi-kosugi Bldg.
915 Shinmaruko, Nakahara-ku
Kawasaki-shi, Kanagawa 211
Tel: 044-733-7011
FAX: 044-733-7010

JAPAN (Cont'd.)

Intel Japan K.K.
Nihon Seimei Atsugi Bldg.
1-2-1 Asahi-machi
Atsugi-shi, Kanagawa 243
Tel: 0462-29-3731
FAX: 0462-29-3781

Intel Japan K.K.*
Ryokuchi-Eki Bldg.
2-4-1 Terauchi
Toyonaka-shi, Osaka 560
Tel: 06-863-1091
FAX: 06-863-1084

Intel Japan K.K.
Shinmaru Bldg.
1-5-1 Marunouchi
Chiyoda-ku, Tokyo 100
Tel: 03-201-3821
FAX: 03-201-6850

Intel Japan K.K.
Green Bldg.
1-16-20 Nishiki
Naka-ku, Nagoya-shi
Aichi 450
Tel: 052-204-1261
FAX: 052-204-1285

KOREA

Intel Technology Asia, Ltd.
Business Center 16th Floor
61, Yoldo-Dong, Young Deung Po-Ku
Seoul 150
Tel: (2) 784-8186, 8286, 8386
TLX: K29312 INTELKO
FAX: (2) 784-8096

SINGAPORE

Intel Singapore Technology, Ltd.
101 Thomson Road #21-06
Goldhill Square
Singapore 1130
Tel: 250-7811
TLX: 39921 INTEL
FAX: 250-9256

TAIWAN

Intel Technology (Far East) Ltd.
Taiwan Branch
10/F, No. 205, Tun Hua N. Road
Taipei, R.O.C.
Tel: 886-2-716-9660
TLX: 13159 INTELTWN
FAX: 886-2-717-2455

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

DAFSYS S.R.L.
Chacabuco, 90-4 PISO
1069-Buenos Aires
Tel: 54-1-334-1871
54-1-334-7726
TLX: 25472

AUSTRALIA

Total Electronics
15-17 Hume Street
Huntingdale, 3166,
Victoria, Australia
Tel: 03-544-8244
TLX: AA 30895
FAX: 03-543-8179

BRAZIL

Elebra Microelectronica
R. Geraldo Flausina Gomes, 78
9 Andar
04575 - Sao Paulo - S.P.
Tel: 011-55-11-534-9637
TLX: 3911125131 ELBR BR
FAX: 55-11-534-9424

CHILE

DIN Instruments
Suecia 2323
Casilla 6055, Correo 22
Santiago
Tel: 56-2-225-8139
TLX: 440422 RUDY CZ

CHINA/HONG KONG

Novel Precision Machinery Co., Ltd.
Flat D, 20 Kingsford Ind. Bldg.
Phase 1, 26 Kwai Hei Street
N.T., Kowloon
Hong Kong
Tel: 852-0-223-222
TWX: 39114 JINMI HX
FAX: 852-0-261-602

INDIA

Micronic Devices
Arun Complex
No. 65 D.V.G. Road
Basavanagudi
Bangalore 560 004
Tel: 91-812-600-631
011-91-812-621-455
TLX: 0845-8332 MD BG IN

Micronic Devices
Flat 403, Gagan Deep
12, Rajendra Place
New Delhi 110 003
Tel: 91-58-97-71
011-91-57-23509
TLX: 03163235 MDND IN

Micronic Devices
No. 516 5th Floor
Swastik Chambers
Sion, Trombay Road
Chembur
Bombay 400 071
Tel: 91-52-39-63
TLX: 9531 171447 MDEV IN

S&S Corporation
Camden Business Center
Suite 8
1610 Blossom Hill Rd.
San Jose, CA 95124
U.S.A.
Tel: (408) 978-6216
TLX: 820281

JAPAN

Asahi Electronics Co. Ltd.
KMM Bldg. 2-14-1 Asano
Kokurakita-ku
Kitakyushu-shi 802
Tel: 093-511-8471
FAX: 093-551-7861

C. Itoh Techno-Science Co., Ltd.
C. Itoh Bldg., 2-5-1 Kita-Aoyama
Minato-ku, Tokyo 107
Tel: 03-497-4900
FAX: 03-497-4879

JAPAN (Cont'd.)

Dia Semicon Systems, Inc.
Wacore 64, 1-37-8 Sangenjaya
Setagaya-ku, Tokyo 154
Tel: 03-487-0388
FAX: 03-487-8088

Okaya Koki
2-4-18 Sakae
Naka-ku, Nagoya-shi 460
Tel: 052-204-2916
FAX: 052-204-2901

Ryoyo Electro Corp.
Konwa Bldg.
1-12-22 Tsukiji
Chuo-ku, Tokyo 104
Tel: 03-546-5011
FAX: 03-546-5044

KOREA

J-Tek Corporation
6th Floor, Government Pension Bldg.
24-3 Yoldo-Dong
Youngdeungpo-ku
Seoul 150
Tel: 82-2-782-8039
TLX: 25299 KODIGIT
FAX: 82-2-784-8391

Samsung Semiconductor &
Telecommunications Co., Ltd.
150, 2-KA, Taipyung-ro, Chung-ku
Seoul 100
Tel: 82-2-751-3987
TLX: 27970 KORSST
FAX: 82-2-753-0967

MEXICO

Dicopel S.A.
Tochtli 368 Fracc. Ind. San Antonio
Azcapotzalco
C.P. 02760-Mexico, D.F.
Tel: 52-5-561-3211
TLX: 1773790 DICOME

NEW ZEALAND

Switch Enterprises
36 Olive Road
Penrose, Auckland
ATTN: Dean Danford
Tel: 64-9-591155
FAX: 64-9-592681

SINGAPORE

Electronic Resources Pte, Ltd.
17 Harvey Road #04-01
Singapore 1336
Tel: 283-0888, 280-1618
TWX: 56541 FRELS
FAX: 2895327

SOUTH AFRICA

Electronic Building Elements, Pty. Ltd.
P.O. Box 4609
Pine Square, 18th Street
Hazelwood, Pretoria 0001
Tel: 27-12-469921
TLX: 3-227786 SA
FAX: 0927-012-46-9221

TAIWAN

Micro Electronics Corporation
No. 585, Ming Shen East Rd.
Taipei, R.O.C.
Tel: 886-2-501-8231
FAX: 886-2-501-4265

Sertek
5FL, 135 Sec. 2
Chien-Kuo N. Rd.
Taipei 10479
R.O.C.
Tel: (02) 5010055
FAX: (02) 5012521
(02) 5058414

VENEZUELA

P. Benavides S.A.
Avilanes a Rio
Residencia Kamarata
Locales 4 A17
La Candelaria, Caracas
Tel: 58-2-571-0396
TLX: 28450 PBVEN VC
FAX: 58-2-572-3321

*Field Application Location

DOMESTIC SERVICE OFFICES

ALABAMA

Intel Corp.
5015 Bradford Dr., #2
Huntsville 35805
Tel: (205) 830-4010

ARIZONA

Intel Corp.
11225 N. 28th Dr.
Suite D-214
Phoenix 85029
Tel: (602) 869-4980

Intel Corp.
500 E. Fry Blvd., Suite M-15
Sierra Vista 85635
Tel: (602) 459-5010

Intel Corp.
1101 N. El Dorado Place
Suite 301
Tucson 85715
Tel: (602) 299-6815

CALIFORNIA

Intel Corp.
21515 Vanowen Street
Suite 116
Canoga Park 91303
Tel: (818) 704-8500

Intel Corp.
2250 E. Imperial Highway
Suite 218
El Segundo 90245
Tel: (213) 640-6040

Intel Corp.
1900 Prairie City Rd.
Folsom 95630-9597
Tel: (916) 351-6143

Intel Corp.
1510 Arden Way, Suite 101
Sacramento 95815
Tel: (916) 920-8096

Intel Corp.
4350 Executive Drive
Suite 105
San Diego 92121
Tel: (619) 452-5880

Intel Corp.*
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.*
San Tomas 4
2700 San Tomas Expressway
2nd Floor
Santa Clara 95051
Tel: (408) 986-8086
TWX: 910-338-0255

COLORADO

Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (303) 594-6622

Intel Corp.*
650 S. Cherry St., Suite 915
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

CONNECTICUT

Intel Corp.
26 Mill Plain Road
2nd Floor
Danbury 06811
Tel: (203) 748-3130
TWX: 710-456-1199

FLORIDA

Intel Corp.
6363 N.W. 6th Way
Suite 100
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407
FAX: 305-772-8193

Intel Corp.
5850 T.G. Lee Blvd.
Suite 340
Orlando 32822
Tel: (305) 240-8000
FAX: 305-240-8097

Intel Corp.
11300 4th Street North
Suite 170
St. Petersburg 33716
Tel: (813) 577-2413
FAX: 813-578-1607

GEORGIA

Intel Corp.
3280 Pointe Parkway
Suite 200
Norcross 30092
Tel: (404) 449-0541

ILLINOIS

Intel Corp.*
300 N. Martingale Road
Suite 400
Schaumburg 60173
Tel: (312) 310-8031

INDIANA

Intel Corp.
8777 Purdue Road
Suite 125
Indianapolis 46268
Tel: (317) 875-0623

IOWA

Intel Corp.
1930 St. Andrews Drive N.E.
2nd Floor
Cedar Rapids 52402
Tel: (319) 393-5510

KANSAS

Intel Corp.
8400 W. 110th Street
Suite 170
Overland Park 66210
Tel: (913) 345-2727

MARYLAND

Intel Corp.*
7321 Parkway Drive South
Suite C
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

Intel Corp.
7833 Walker Drive
Suite 550
Greenbelt 20770
Tel: (301) 441-1020

MASSACHUSETTS

Intel Corp.*
Westford Corp. Center
3 Carlisle Road
2nd Floor
Westford 01886
Tel: (508) 692-3222
TWX: 710-343-6333

MICHIGAN

Intel Corp.
7071 Orchard Lake Road
Suite 100
West Bloomfield 48033
Tel: (313) 851-8096

MINNESOTA

Intel Corp.
3500 W. 80th St., Suite 360
Bloomington 55431
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI

Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY

Intel Corp.
Raritan Plaza III
Raritan Center
Edison 08817
Tel: (201) 225-3000

Intel Corp.
385 Sylvan Avenue
Englewood Cliffs 07632
Tel: (201) 567-0821
TWX: 710-991-8593

Intel Corp.*
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel: (201) 747-2233

†Intel Corp.
280 Corporate Center
75 Livingston Avenue
First Floor
Roseland 07068
Tel: (201) 740-0111
FAX: 201-740-0626

NEW MEXICO

Intel Corp.
8500 Menaul Boulevard N.E.
Suite B 295
Albuquerque 87112
Tel: (505) 292-8086

NEW YORK

Intel Corp.
127 Main Street
Binghamton 13905
Tel: (607) 773-0337
FAX: 607-723-2677

Intel Corp.*
850 Cross Keys Office Park
Fairport 14450
Tel: (716) 425-2750
TWX: 510-253-7391

Intel Corp.*
300 Motor Parkway
Hauppauge 11787
Tel: (516) 231-3300
TWX: 510-227-8236

Intel Corp.
Westage Business Center
Bldg. 300, Route 9
Fishkill 12524
Tel: (914) 897-3860
FAX: 914-897-3125

NORTH CAROLINA

Intel Corp.
5700 Executive Drive
Suite 213
Charlotte 28212
Tel: (704) 568-8966

Intel Corp.
2306 W. Meadowview Road
Suite 206
Greensboro 27407
Tel: (919) 294-1541

Intel Corp.
2700 Wycliff Road
Suite 102
Raleigh 27607
Tel: (919) 781-8022

OHIO

Intel Corp.*
3401 Park Center Drive
Suite 220
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*
25700 Science Park Dr.
Suite 100
Beachwood 44122
Tel: (216) 464-2736
TWX: 810-427-9298

OKLAHOMA

Intel Corp.
6801 N. Broadway
Suite 115
Oklahoma City 73162
Tel: (405) 848-8086

OREGON

Intel Corp.
15254 N.W. Greenbrier Parkway
Building B
Beaverton 97006
Tel: (503) 645-8051
TWX: 910-467-8741

Intel Corp.
5200 N.E. Elam Young Parkway
Hillsboro 97123
Tel: (503) 681-8080

PENNSYLVANIA

Intel Corp.*
455 Pennsylvania Avenue
Suite 230
Fort Washington 19034
Tel: (215) 641-1000
TWX: 510-661-2077

Intel Corp.*
400 Penn Center Blvd.
Suite 610
Pittsburgh 15235
Tel: (412) 823-4970

PUERTO RICO

Intel Microprocessor Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

TEXAS (Cont'd.)

Intel Corp.*
12000 Ford Road
Suite 400
Dallas 75234
Tel: (214) 241-8087
FAX: 214-484-1180

Intel Corp.*
7322 S.W. Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8086
TWX: 910-881-2490

UTAH

Intel Corp.
428 East 6400 South
Suite 104
Murray 84107
Tel: (801) 263-8051

VIRGINIA

Intel Corp.
1504 Santa Rosa Road
Suite 108
Richmond 23288
Tel: (804) 282-5668

WASHINGTON

Intel Corp.
155 108th Avenue N.E.
Suite 386
Bellevue 98004
Tel: (206) 453-8086
TWX: 910-443-3002

Intel Corp.
408 N. Mullan Road
Suite 102
Spokane 99206
Tel: (509) 928-8086

WISCONSIN

Intel Corp.
330 S. Executive Dr.
Suite 102
Brookfield 53005
Tel: (414) 784-8087
FAX: (414) 796-2115

CANADA

BRITISH COLUMBIA

Intel Semiconductor of Canada, Ltd.
4585 Canada Way, Suite 202
Burnaby V5G 4L6
Tel: (604) 298-0387
FAX: (604) 298-8234

ONTARIO

Intel Semiconductor of Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel: (613) 829-9714
TLX: 053-4115

Intel Semiconductor of Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel: (416) 675-2105
TLX: 06983574
FAX: (416) 675-2438

QUEBEC

Intel Semiconductor of Canada, Ltd.
620 St. John Boulevard
Pointe Claire H9R 3K2
Tel: (514) 694-9130
TWX: 514-694-9134

CUSTOMER TRAINING CENTERS

CALIFORNIA

2700 San Tomas Expressway
Santa Clara 95051
Tel: (408) 970-1700

ILLINOIS

300 N. Martingale, #300
Schaumburg 60173
Tel: (312) 310-8700

MASSACHUSETTS

3 Carlisle Road
Westford 01886
Tel: (508) 692-1000

MARYLAND

7833 Walker Dr., 4th Floor
Greenbelt 20770
Tel: (301) 220-3380

SYSTEMS ENGINEERING OFFICES

CALIFORNIA

2700 San Tomas Expressway
Santa Clara 95051
Tel: (408) 986-8086

ILLINOIS

300 N. Martingale, #300
Schaumburg 60173
Tel: (312) 310-8031

NEW YORK

300 Motor Parkway
Hauppauge 11788
Tel: (516) 231-3300

UNITED STATES
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

WEST GERMANY
Intel Semiconductor GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

HONG KONG
Intel Semiconductor Ltd.
10/F East Tower
Bond Center
Queensway, Central

CANADA
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

Printed in U.S.A /0189/70K/RRD JB
Embedded Controller

ISBN 1-55512-048-2