8 BIT MCU FAMILIES
EF6801/04/05

8 BIT MCU FAMILIES EF6801/04/05

DATABOOK

1st EDITION

SGS-THOMSON
MICROELECTRONICS

8 BIT MCU FAMILIES EF6801/04/05

DATABOOK

1st EDITION

SEPTEMBER 1989

USE IN LIFE SUPPORT MUST BE EXPRESSLY AUTHORIZED

SGS-THOMSON' PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF SGS-THOMSON Microelectronics. As used herein:

1. Life support devices to systems are devices or systems which, are intended for surgical implant into the body to support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

TABLE OF CONTENTS

ALPHANUMERICAL INDEX

SELECTION GUIDE

EF6801 FAMILY

8 BIT MICROCONTROLLER PRODUCTS FOR HIGH END MARKET.

* **EF6801** : MCU with enhanced capabilities of 6800 family:
faster cycle times, new instructions such as multiplication.

* **EF6801U4** : 4K ROM version of 6801 with enhanced 16 bit timer.

HMOS TECHNOLOGY		6801	6801U4
CLOCK		1 & 2 MHZ	1 & 2 MHZ
MEMORY	RAM	128 x 8	192 x 8
	ROM	2048 x 8	4096 x 8
I/O LINES		32	32
TIMER		• 16 bit free running counter • 1 input capture • 1 output compare	• 16 bit free running counter • 2 input captures • 3 output compares
SERIAL COMMUNICATION INTERFACE		• Full duplex transmitter/receiver independent • Data format: NR2 or biphase • Clock: ext. or int. bit rate • Wake up feature	
PACKAGE		DILP 40 / PLCC 44 / CERDIP 40	
OPERATING TEMP. RANGE		−40°C to +85°C	
COMPATIBILITY		MC6801	MC6801U4

EF6804 FAMILY

8 BIT MICROCONTROLLER PRODUCTS FOR LOW END APPLICATION.

* **EF6804J2** : the highest improvement in cost reduction, a microcontroller at a TTL/LS price level.

* **EF6804P2** : High on-chip feature integration well suited for additional 4 bit application extensions.

* **EF68HC04P3** :Dedicated to power and data saving applications or requiring protection against mains failures (pin compatible with 6804P2)

* **EF68HC04J3** :68HC04P3 with 12 I/O lines

HMOS TECHNOLOGY				**68HC04J3**	**68HC04P3**
HMOS TECHNOLOGY		6804J2	6804P2		
MEMORY	RAM	32 x 8	32 x 8	124 x 8	124 x 8
	ROM	1K x 8	1K x 8	2K x 8	2K x 8
I/O PORT		12	20	12	20
8-bit timer with 7-bit software programmable prescaler		yes	yes	yes	yes
PACKAGE		DIL20/SO20 PLCC28	DIL28/PLCC28	DIL20/SO20	DIL28/PLCC28
OPERATING TEMP. RANGE		−40°C to +85°C	−40°C to +85°C	−40°C to +85°C	−40°C to +85°C
COMPATIBILITY		MC6804J2	MC6804P2	MC68HC04J3	MC68HC04P3

* **Development tool** : Hardware development station INICE
+ Emulator probe EFTMUP4
+ Cross Assembler MSDOS compatible TSR6804

EF6805 FAMILY

8 BIT MICROCONTROLLER PRODUCTS FOR MID-RANGE APPLICATION.

* **EF6805P2/P6** : Low cost single chip MCU well suited for economical design with proven capabilities of 6805 based instruction set. Pin compatible with 6804P2 and 68HC04P3.

* **EF6805U2/U3** : With 32 I/O lines and 4K ROM, it is a super general purpose MCU covering a wide range of applications such as: answering machine, PC keyboard decoder...

* **EF6805R2/R3** : Same as U2/U3 but with A/D converter on board.

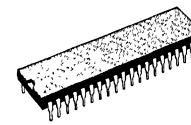
HMOS TECHNOLOGY		6805P2/P6	6805U2/U3	6805R2/R3
MEMORY	RAM	64 x 8	64 x 8 / 112 x 8	64 x 8 / 112 x 8
	ROM	1K x 8 / 1.8K x 8	2K x 8 / 4K x 8	2K x 8 / 4K x 8
I/O PORT	BIDIRECTION	20	24	24
	INPUT ONLY	—	8	8
Timer with 7-bit prescaler	HARD. PROG	yes	yes / —	yes / —
	SOFT. PROG	—	— / yes	— / yes
SPECIAL FEATURES		—	—	A/D
PACKAGE		DIL28 PLCC28	DILP40 PLCC44	DILP40 PLCC44
OPERATING TEMP. RANGE		− 40°C to + 105°C	− 40°C to + 105°C	− 40°C to + 105°C
COMPATIBILITY		MC6805P2/P6	MC6805U2/U3	MC6805R2/R3

* **Development tool** : Hardware development station INICE 4-8: TSTIN48
+ Emulator probe TSTMUP5
+ Cross Assembler MSDOS compatible TSR6805

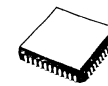
EF6801 FAMILY DATASHEETS

MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

- ENHANCED EF6800 INSTRUCTION SET
- 8 X 8 MULTIPLY INSTRUCTION
- SERIAL COMMUNICATIONS INTERFACE (SCI)
- UPWARD SOURCE AND OBJECT CODE COMPATIBILITY WITH THE 6800
- 16-BIT THREE-FUNCTION PROGRAMMABLE TIMER
- SINGLE-CHIP OR EXPANDED OPERATION TO 64K BYTE ADDRESS SPACE
- BUS COMPATIBILITY WITH THE 6800 FAMILY
- 2048 BYTES OF ROM (EF6801)
- 128 BYTES OF RAM
- 64 BYTES OF RAM RETAINABLE DURING POWERDOWN
- 29 PARALLEL I/O AND TWO HANDSHAKE CONTROL LINES
- INTERNAL CLOCK GENERATOR WITH DIVIDE-BY-FOUR OUTPUT
- – 40°C TO + 85°C TEMPERATURE RANGE
- – 40°C TO + 105°C TEMPERATURE RANGE

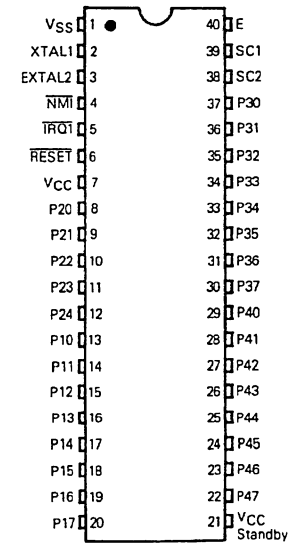


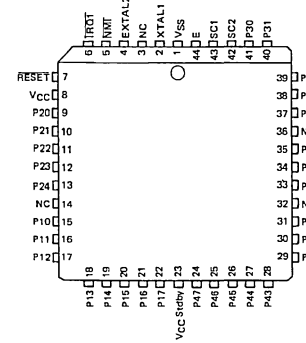
P SUFFIX
(Plastic Package)



FN SUFFIX
(PLCC44)

PIN CONNECTIONS

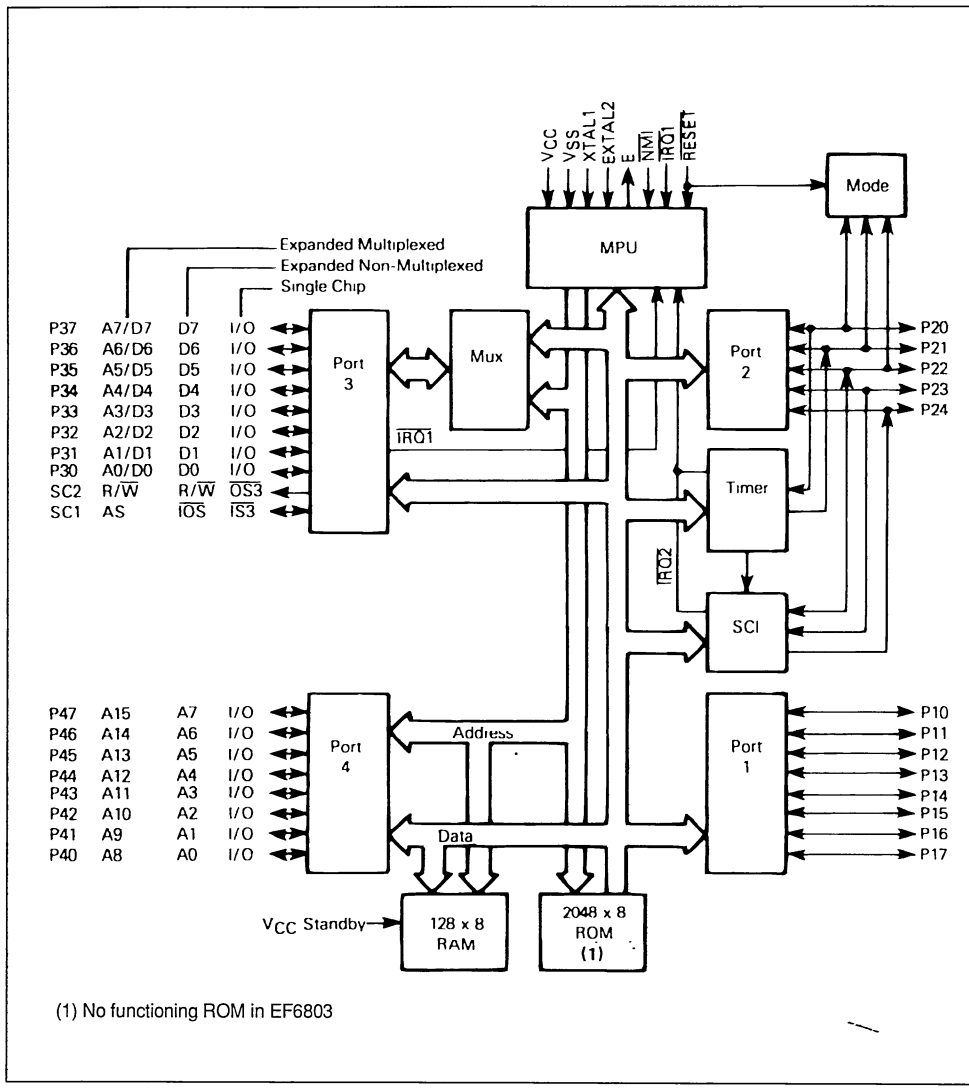




DESCRIPTION

The EF6801 is an 8-bit single-chip microcomputer unit (MCU) which significantly enhances the capabilities of the 6800 family of parts. It includes an upgraded 6800 microprocessor unit (MPU) with upward-source and object-code compatibility. Execution times of key instructions have been improved and several new instructions have been added including an unsigned multiply. The MCU can function as a monolithic microcomputer or can be expanded to a 64K byte address space. It is TTL compatible and requires one + 5V power supply. On-chip resources include 2048 bytes of ROM, 128 bytes of RAM, a Serial Communications Interface (SCI), parallel I/O, and a three function Programmable Timer. The EF6803 can be considered as an EF6801 operating in Modes 2 or 3. EF6801 MCU Family features include :

Figure 1 : 6801/6803 Bock Diagram.



(1) No functioning ROM in EF6803

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where :

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction -to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts - Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} << P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$$P_D = K + (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives :

$$K = PD \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage	– 0.3 to + 7.0	V
T_A	Operating Temperature Range EF6801/03, EF6801/03-1, EF68A01/03, EF68B01/03 EF6801/03, EF6801/03-1 : V Suffix EF6801/03, EF6801/03-1 : A Suffix	T_L to T_H 0 to 70 – 40 to 85 – 40 to 105	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields ; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$. Input protection is enhanced by connecting unused inputs to either V_{DD} or V_{SS}.

THERMAL DATA

θ_{JA}	Thermal Resistance	Plastic PLCC	50 100	°C/W

CONTROL TIMING (V_{CC} = 5.0V ± 5%, V_{SS} = 0, T_A = 0 to 70°C)

Symbol	Parameter	EF6801		EF6801-1		EF68A01		EF68B01		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
f_o	Frequency of Operation	0.5	1.0	0.5	1.25	0.5	1.5	0.5	2.0	MHz
f_{XTAL}	Crystal Frequency	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
$4f_o$	External Oscillator Frequency	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
t_{rc}	Crystal Oscillator Start Up Time		100		100		100		100	ms
t_{PCS}	Processor Control Setup Time	200		170		140		110		ns

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0Vdc ± 5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted)

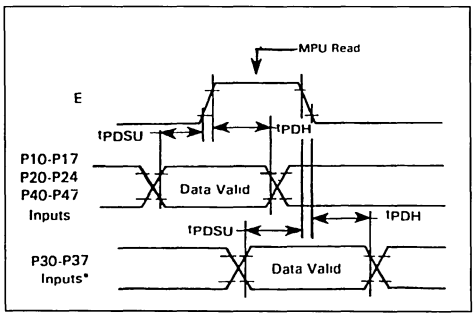
Symbol	Parameter	EF6801/03 0°C to + 70°C		EF6801/03 – 40°C to + 85°C / + 105°C		Unit
		Min.	Max.	Min.	Max.	
V_{IH}	Input High Voltage — $\overline{RESET}$	V_{SS} + 4.0	V_{CC}	V_{SS} + 4.0	V_{CC}	V
	Other Inputs	V_{SS} + 2.0	V_{CC}	V_{SS} + 2.2	V_{CC}	
V_{IL}	Input Low Voltage — All Inputs	V_{SS} – 0.3	V_{SS} + 0.8	V_{SS} – 0.3	V_{SS} + 0.8	V
I_{in}	Input Load Current (V_{in} = 0 to 2.4V) Port 4		0.5		0.8	mA
	SCI		0.8		1.0	
I_{in}	Input Leakage Current (V_{in} = 0 to 5.25V) NMI, IRQ1, $\overline{RESET}$		2.5		5.0	µA
I_{TSI}	Hi-Z (off-state) Input Current (V_{in} = 0.5 to 2.4V) Ports 1, 2, and 3		10		20	µA
V_{OH}	Output High Voltage (I_{Load} = – 65µA, V_{CC} = Min)* E, Port 4, SC1, SC2	V_{SS} + 2.4		V_{SS} + 2.4		V
	(I_{Load} = – 100µA, V_{CC} = Min) Other Outputs	V_{SS} + 2.4		V_{SS} + 2.4		
V_{OL}	Output Low Voltage (I_{Load} = 2.0mA, V_{CC} = Min) All Outputs		V_{SS} + 0.5		V_{SS} + 0.6	V
I_{OH}	Darlington Drive Current (V_O = 1.5V) Port 1	1.0	4.0	1.0	5.0	mA
P_{INT}	Internal Power Dissipation (measured at T_A = T_L in steady-state operation)		1200		1500	mW
C_{in}	Input Capacitance Port 3, Port 4, SC1		12.5		12.5	pF
	(V_{in} = 0, T_A = 25°C, Other Inputs		10		10	
	f_o = 1.0MHz)					
V_{SBB}	V_{CC} Standby — Powerdown	4.0	5.25	4.0	5.25	V
V_{SB}	Powerup	4.75	5.25	4.75	5.25	
I_{SBB}	Standby Current Powerdown		6.0		8.0	mA

* Negociable to –100µA (for further information contact the factory)

PERIPHERAL PORT TIMING (refer to figures 2-5)

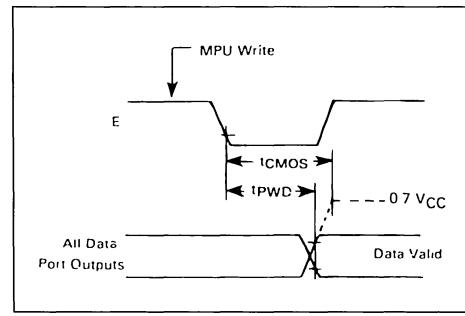
Symbol	Parameter	EF6801 EF6803		EF6801-1 EF6803-1		EF68A01 EF68A03		EF68B01 EF68B03		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t_{PDSU}	Peripheral Data Setup Time	200		200		150		100		ns
t_{PDH}	Peripheral Data Hold Time	200		200		150		100		ns
t_{OSD1}	Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition		350		350		300		250	ns
t_{OSD2}	Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition		350		350		300		250	ns
t_{PWD}	Delay Time, Enable Negative Transition to Peripheral Data Valid		350		350		300		250	ns
t_{CMOS}	Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid		2.0		2.0		2.0		2.0	µs
t_{PWIS}	Input Strobe Pulse Width	200		200		150		100		ns
t_{IH}	Input Data Hold Time	50		50		40		30		ns
t_{IS}	Input Data Setup Time	20		20		20		20		ns

Figure 2 : Data Setup and Hold Times (MPU read).



* Port 3 non-latched operation (latch enable - 0)

Figure 3 : Data Setup and Hold Times (MPU write).



Notes : 1. 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}.
2. Not applicable to P21.
3. Port 4 cannot be pulled above V_{CC}.

Figure 4 : Port 3 Output Strobe Timing (EF6801 single-chip Mode).

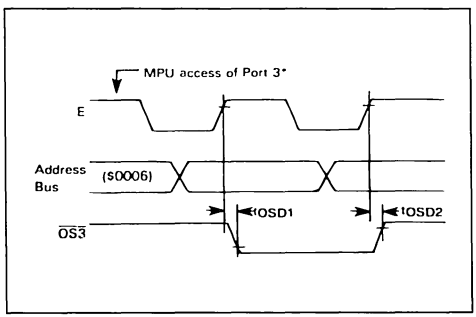
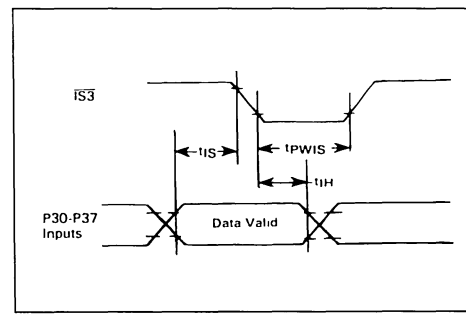


Figure 5 : Port 3 Latch Timing (EF6801 single-chip mode).



* Access matches Output Strobe Select (OSS = 0, a read. OSS = 1, a write).

Note : Timing measurements are referenced to and from a low voltage of 0.8V and a high voltage of 2.0V, unless otherwise noted.

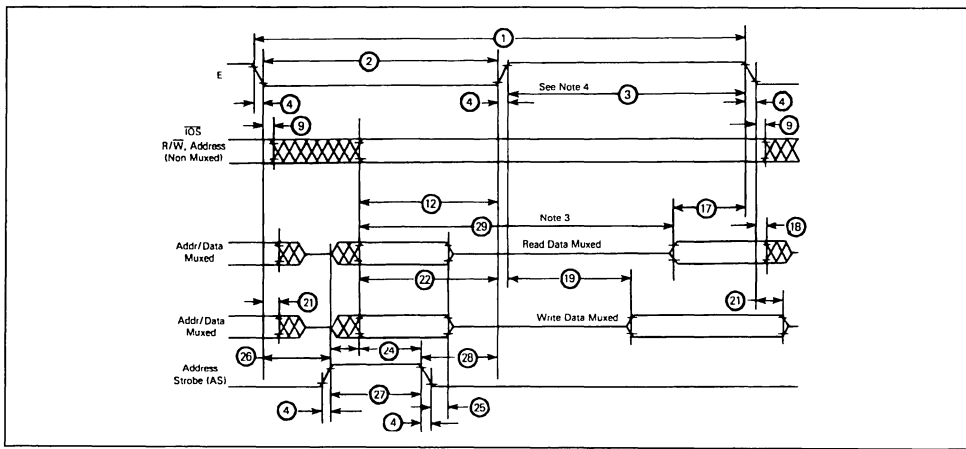
BUS TIMING (see notes 1 and 2)

Ident. Number	Symbol	Parameter	EF6801 EF6803		EF6801-1 EF6803-1		EF68A01 EF68A03		EF68B01 EF68B03		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
1	t_{cyc}	Cycle Time	1.0	2.0	0.8	2.0	0.667	2.0	0.5	2.0	µs
2	PW_{EL}	Pulse Width, E Low	430	1000	360	1000	300	1000	210	1000	ns
3	PW_{EH}	Pulse Width, E High	450	1000	360	1000	300	1000	220	1000	ns
4	t_r, t_f	Clock Rise and Fall Time		25		25		25		20	ns
9	t_{AH}	Address Hold Time	20		20		20		10		ns
12	t_{AV}	Non-muxed Address Valid Time to E*	200		150		115		70		ns
17	t_{DSR}	Read Data Setup Time	80		70		60		40		ns
18	t_{DHR}	Read Data Hold Time	10		10		10		10		ns
19	t_{DDW}	Write Data Delay Time		225		200		170		120	ns
21	t_{DHW}	Write Data Hold Time	20		20		20		10		ns
22	t_{AVM}	Muxed Address Valid Time to E Rise*	200		150		115		80		ns
24	t_{ASL}	Muxed Address Valid Time to AS Fall*	60		50		40		20		ns
25	t_{AHL}	Muxed Address Hold Time	20		20		20		10		ns
26	t_{ASD}	Delay Time, E to AS Rise*	90**		70**		60**		45**		ns
27	PW_{ASH}	Pulse Width, AS High*	220		170		140		110		ns
28	t_{ASED}	Delay Time, AS to E Rise*	90		70		60		45		ns
29	t_{ACC}	Usable Access Time*	595		465		380		270		ns

* At specified cycle time.

* * t_{ASD} parameters listed assume external TTL clock drive with 50% ± 5% duty cycle. Devices driven by an external TTL clock with 50% ± 1% duty cycle or which use a crystal have the following t_{ASD} specifications : 100ns min. (1.0MHz devices), 80ns min. (1.25MHz devices), 65ns min. (1.5MHz devices), 50ns min. (2.0MHz devices).

Figure 6 : Bus Timing.



Notes : 1. Voltage levels shown are $V_L \leq 0.5V$, $V_H \geq 2.4V$, unless otherwise specified.
2 Measurement points shown are 0.8V and 2.0V, unless otherwise specified.
3. Usable access time is computed by 12 + 3 − 17 + 4.
4. Memory devices should be enabled only during E high to avoid Port 3 bus contention.

Figure 7 : CMOS Load.

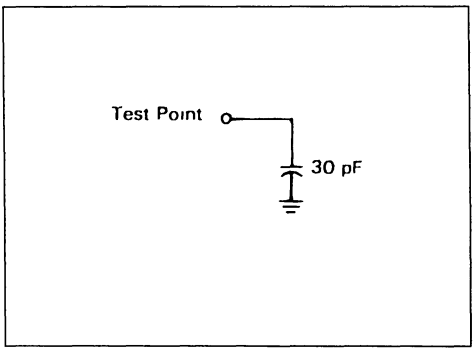
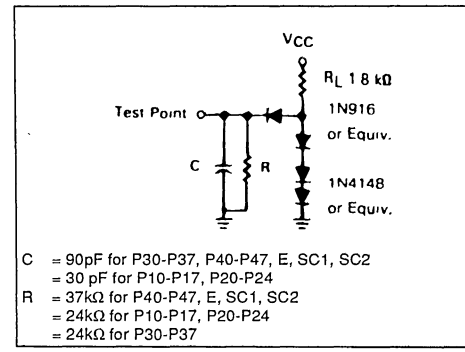


Figure 8 : Timing Test Load Ports 1, 2, 3, 4.



C = 90pF for P30-P37, P40-P47, E, SC1, SC2
= 30 pF for P10-P17, P20-P24
R = 37kΩ for P40-P47, E, SC1, SC2
= 24kΩ for P10-P17, P20-P24
= 24kΩ for P30-P37

INTRODUCTION

The EF6801 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

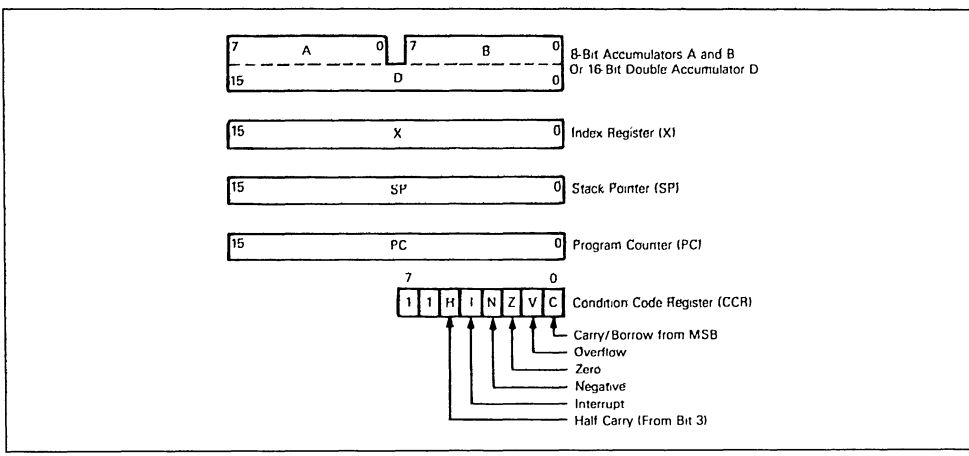
Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a Data Register and a write-only Data Direction Register. The Data Direction Register is used to define whether corresponding bits in the Data Register are configured as an input (clear) or output (set).

The term "port", by itself, refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port", it is controlled by the port Data Direction Register and the programmer has direct access to the port pins using the port Data Register. Port pins are labled as Pij where i identifies one of four ports and j indicates the particular bit.

The Microprocessor Unit (MPU) is an enhanced EF6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the EF6800. The programming model is depicted in figure 9, where Accumulator D is a concatenation of Accumulators A and B. A list of new operations added to the 6800 instruction set are shown in table 1.

The EF6803 can be considered an EF6801 that operates in Modes 2 and 3 only.

Figure 9 : Programming Model.



OPERATING MODES

The EF6801 provides eight different operating modes (modes 0 through 7), the EF6803 provides two operating modes (modes 2 and 3). The operating modes are hardware selectable and determine the device memory map, the configuration of Port 3, Port 4, SC1, SC2, and the physical location of the interrupt vectors.

FUNDAMENTAL MODES

The eight operating modes can be grouped into three fundamental modes which refer to the type of bus it supports : Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Single-Chip modes include 4 and 7, Expanded Non-Multiplexed is Mode 5 and the remaining five are Expanded Multiplexed modes. Table 2 summarizes the characteristics of the operating modes.

EF6801 SINGLE-CHIP MODES (4, 7). In the Single-Chip Mode, the four MCU ports are configured as parallel input/output data ports, as shown in figure 10. The MCU functions as a monolithic microcomputer in these two modes without external address or data buses. A maximum of 29 I/O lines and two Port 3 control lines are provided. Peripherals or another MCU can be interfaced to Port 3 in a loosely coupled dual processor configuration, as shown in figure 11.

Table 1 : New Instructions.

Instruction	Description
ABX	Unsigned addition of accumulator B to index register.
ADDD	Adds (without carry) the double accumulator to memory and leaves the sum in the double accumulator.
ASLD or LSLD	Shifts the double accumulator left (towards MSB) one bit ; the LSB is cleared and the MSB is shifted into the C-bit.
BHS	Branch if higher or same ; unsigned conditional branch (same as BCC).
BLO	Branch if lower ; unsigned conditional branch (same as BCS).
BRN	Branch never.
JSR	Additional addressing mode direct.
LDD	Loads double accumulator from memory.
LSL	Shifts memory or accumulator left (towards MSB) one bit ; the LSB is cleared and the MSB is shifted into the C-bit (same as ASL).
LSRD	Shifts the double accumulator right (towards LSB) one bit ; the MSB is cleared and the LSB is shifted into the C-bit.
MUL	Unsigned multiply ; multiplies the two accumulators and leaves the product in the double accumulator.
PSHX	Pushes the index register to stack.
PULX	Pulls the index register from stack.
STD	Stores the double accumulator to memory.
SUBD	Substracts memory from the double acccumulator and leaves the difference in the double accumulator.
CPX	Internal processing modified to permit its use with any conditional branch instruction.

In Single-Chip Test Mode (4), the RAM responds to $XX80 through $XXFF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using modes 0, 1, 2, or 6. If the MCU is Reset and then programmed into Mode 4, execution will begin at $XXFE : XXFF. Mode 5 can be irreversibly entered from Mode 4 without asserting RESET by setting bit 5 of the Port 2 Data Register. This mode is used primarily to test Ports 3 and 4 in the Single-Chip and Non-Multiplexed Modes.

EF6801 EXPANDED NON-MULTIPLEXED MODE (5). A modest amount of external memory space is provided in the Expanded Non-Multiplexed Mode while significant on-chip resources are retained. Port 3 functions as an 8-bit bidirectional data bus and Port 4 is configured initially as an input data port. Any combination of the eight least-significant address lines may be obtained by writing to the Port 4 Data Direction Register. Stated alternatively, any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Internal pullup resistors pull the Port 4 lines high until the port is configured.

Figure 12 illustrates a typical system configuration in the Expanded Non-Multiplexed Mode. The MCU

interfaces directly with 6800 family parts and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and can be used as a memory page select or chip select line.

Table 2 : Summary of EF6801/03 Operating Modes.

Common to all Modes : Reserved Register Area Port 1 Port 2 Programmable Timer Serial Communications Interface
Single-chip Mode 7 : 128 Bytes of RAM, 2048 Bytes of ROM Port 3 is a parallel I/O port with two control lines. Port 4 is a parallel I/O port. SC1 is input strobe 3 (IS3). SC2 is output strobe (OS3).
Expanded Non-multiplexed Mode 5 : 128 Bytes of RAM, 2048 Bytes of ROM 256 Bytes of External Memory Space Port 3 is an 8-bit data bus. Port 4 is an input port/address bus. SC1 is input/output select (IOS). SC2 is read/write (R/W).
Expanded Multiplexed Modes 1, 2, 3, 6* : Four memory space options (64K address space). (1) No Internal RAM or ROM (mode 3) (2) Internal RAM, no ROM (mode 2) (3) Internal RAM and ROM (mode 1) (4) Internal RAM, ROM with Partial Address Bus (mode 6) Port 3 is multiplexed address/data bus. Port 4 is an address bus (inputs/address in mode 6). SC1 is address strobe (AS). SC2 is read/write (R/W).
Test Modes 0 and 4 : Expanded multiplexed test mode 0. May be used to test RAM and ROM. Single chip and non-multiplexed test mode 4. (1) May be changed to mode 5 without going through reset. (2) May be used to test ports 3 and 4 as I/O ports.

* The EF6803 operates only in modes 2 and 3

Figure 10 : Single-chip Mode.

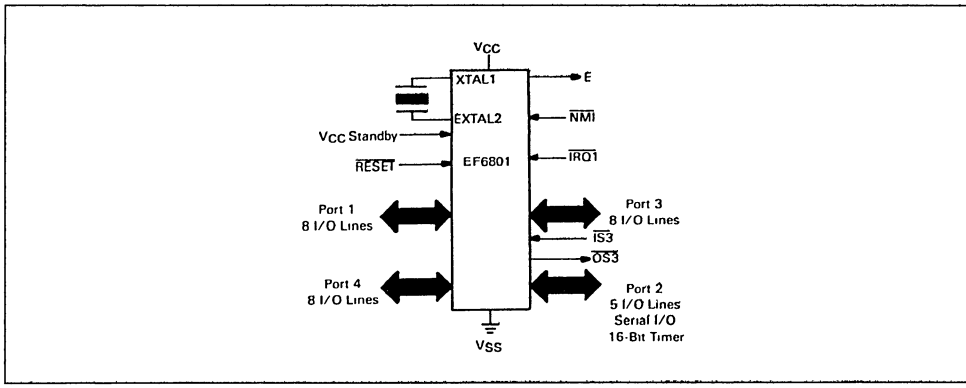


Figure 11 : Single-chip Dual Processor Configuration.

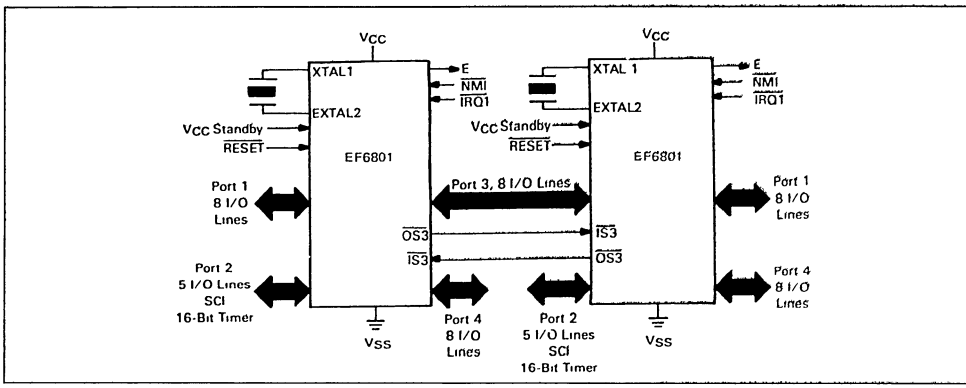
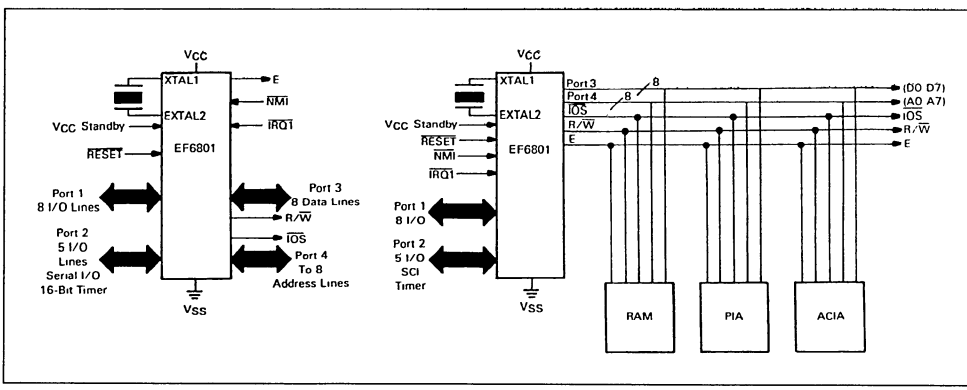


Figure 12 : Expanded Non-multiplexed Configuration.



EXPANDED-MULTIPLEXED MODES (0, 1, 2, 3, 6). A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes Port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS), and data valid while E is high. In Modes 0 to 3, Port 4 provides address lines A8 to A15. In Mode 6, however, Port 4 initially is configured at RESET as an input data port. The port 4 Data Direction Register can then be changed to provide any combination of address lines, A8 to A15. Stated alternatively, any subset of A8 to A15 can be provided while retaining the remaining port 4 lines as input data lines. Internal pullup resistors pull the Port 4 lines high until software configures the port.

In Mode 0, the Reset vector is external for the first two E-cycles after the positive edge of RESET, and internal thereafter. In addition, the internal and external data buses are connected so there must be no memory map overlap in order to avoid potential bus conflicts. Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with the automated test equipment.

Only the EF6801 can operate in each of the expanded multiplexed modes. The EF6803 operates only in Modes 2 and 3.

Figure 13 depicts a typical configuration for the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in figure 14. This allows Port 3 to function as a Data Bus when E is high.

PROGRAMMING THE MODE

The operating mode is determined at RESET by the levels asserted on P22, P21, and P20. These levels are latched into PC2, PC1, and PC0 of the program control register on the positive edge of RESET. The operating mode may be read from the Port 2 Data Register as shown below, and programming levels and timing must be met as shown in figure 15. A brief outline of the operating modes is shown in table 3.

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$0003

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in figure 16 may be used ; otherwise, three-state buffers can be used to provide isolation while programming the mode.

Table 3 : Mode Selection Summary.

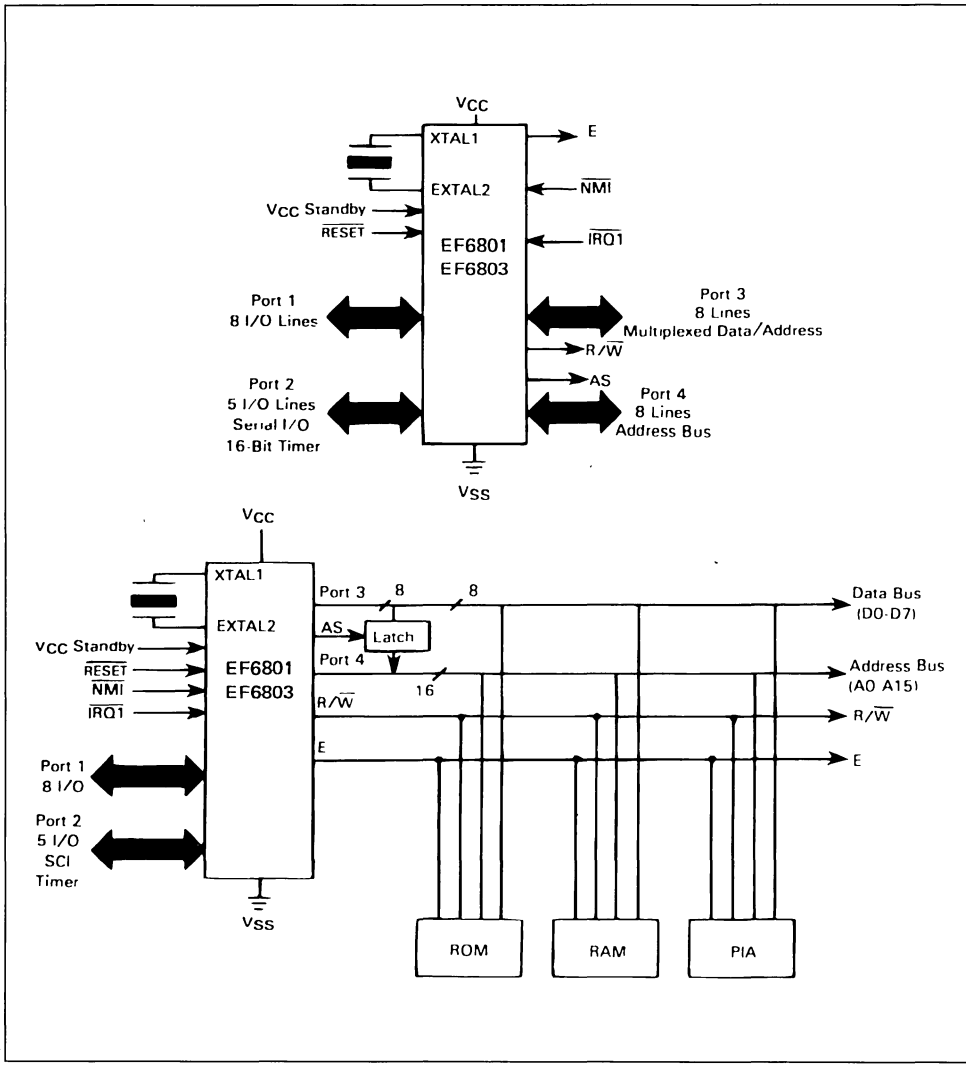
Mode*	P22 PC2	P21 PC1	P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX (5, 6)	Multiplexed/partial Decode
5	H	L	H	I	I	I	NMUX (5, 6)	Non-multiplexed/partial Decode
4	H	L	L	I (2)	I (1)	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX (4)	Multiplexed/no RAM or ROM
2	L	H	L	E	I	E	MUX (4)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX (4)	Multiplexed/RAM & ROM
0	L	L	L	I	I	I (3)	MUX (4)	Multiplexed Test

Legend :
I - Internal
E - External
MUX - Multiplexed
NMUX - Non-Multiplexed
L - Logic "0"
H - Logic "1"

Notes : (1) Internal RAM is addressed at $XX80
(2) Internal ROM is disabled
(3) RESET vector is external for 2 cycles after RESET goes high
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
(5) Addresses associated with Port 3 are considered external in Modes 5 and 6
(6) Port 4 default is user data input, address output is optional by writing to Port 4 Data Direction Register

* The EF6803 operates only in Modes 2 and 3

Figure 13 : Expanded Multiplexed Configuration.



Note : To avoid data bus (Port 3) contention in the expanded multiplexed modes, memory devices should be enabled only during E high time

Figure 14 : Typical Latch Arrangement.

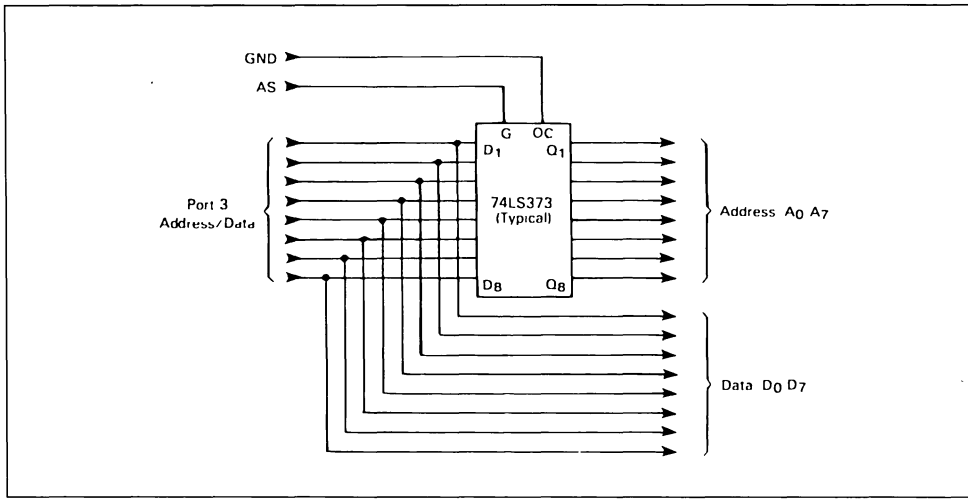
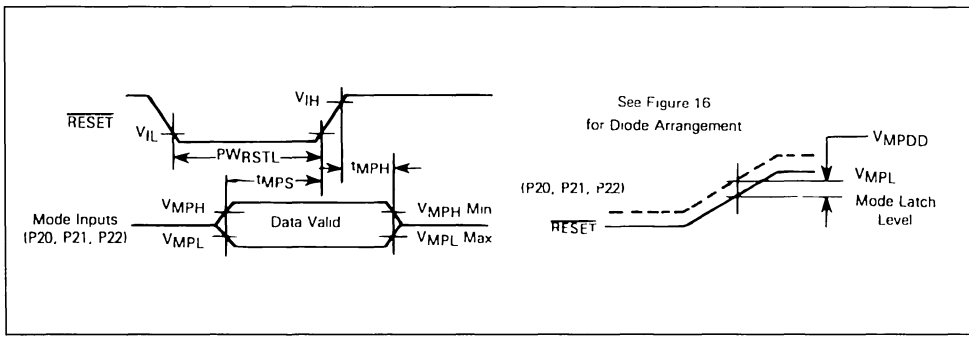


Figure 15 : Mode Programming Timing.

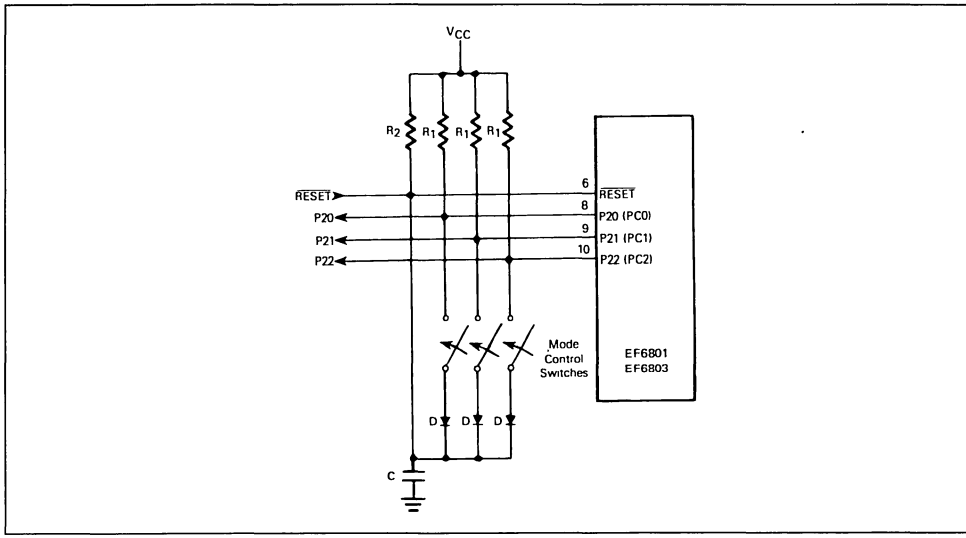


MODE PROGRAMMING (refer to figure 15)

Symbol	Parameter	Min.	Max.	Unit
V_{MPL}	Mode Programming Input Voltage Low*		1.8	V
V_{MPH}	Mode Programming Input Voltage High	4.0		V
V_{MPDD}	Mode Programming Diode Differential (if diodes are used)	0.6		V
PW_{RSTL}	$\overline{RESET}$ Low Pulse Width	3.0		E-Cycles
t_{MPS}	Mode Programming Setup Time	2.0		E-Cycles
t_{MPH}	Mode Programming Hold Time $\overline{RESET}$ Rise Time ≥ 1µs $\overline{RESET}$ Rise Time < 1µs	0 100		ns

* For $T_A = -40°C$ to $+105°C$, $V_{MPL} = 1.7V$.

Figure 16 : Typical Mode Programming Circuit.



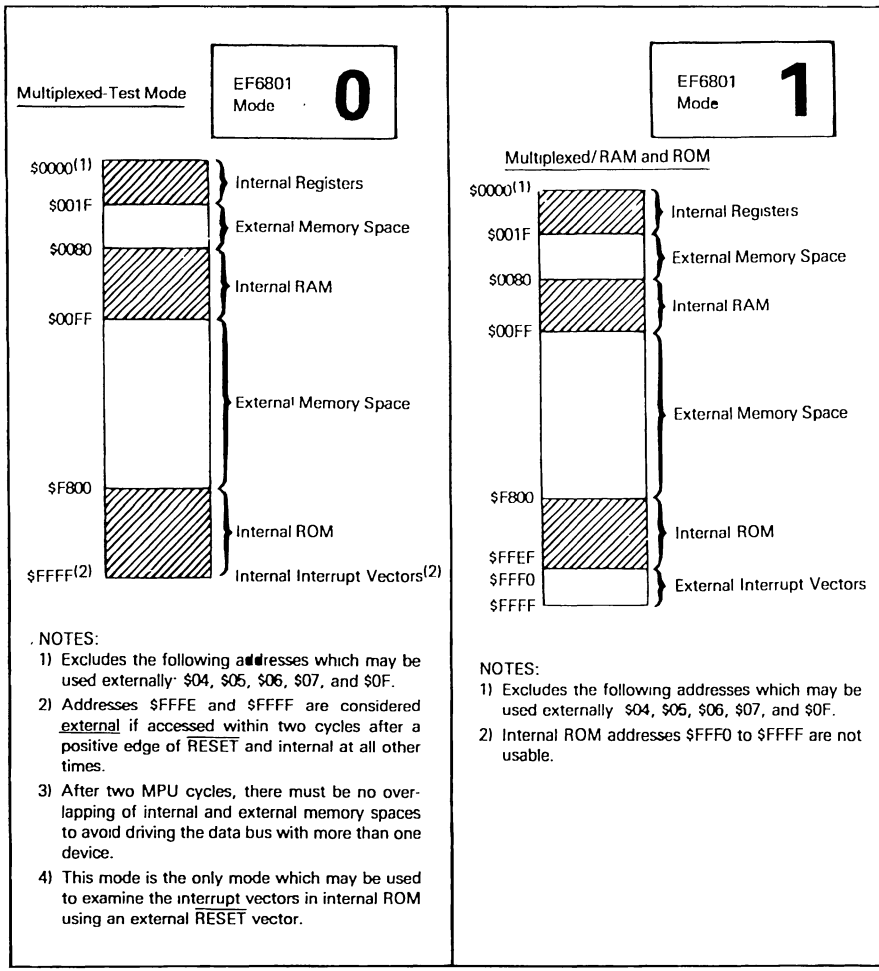
Notes : 1. Mode 7 as shown
2. $R_2.C$ = Reset time constant
3. R_1 = 10k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min

MEMORY MAPS

The 6801 Family can provide up to 64K byte address space depending on the operating mode. A memory map for each operating mode is shown in figure 17.

The first 32 locations of each map are reserved for the internal register area, as shown in table 4, with exceptions as indicated.

Figure 17 : EF6801/03 Memory Maps (sheet 1 of 3).



NOTES:

1) Excludes the following addresses which may be used externally· $04, $05, $06, $07, and $0F.
2) Addresses $FFFE and $FFFF are considered external if accessed within two cycles after a positive edge of $\overline{\text{RESET}}$ and internal at all other times.
3) After two MPU cycles, there must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which may be used to examine the interrupt vectors in internal ROM using an external $\overline{\text{RESET}}$ vector.

NOTES:

1) Excludes the following addresses which may be used externally $04, $05, $06, $07, and $0F.
2) Internal ROM addresses $FFF0 to $FFFF are not usable.

Figure 17 : EF6801/03 Memory Maps (sheet 2 of 3).

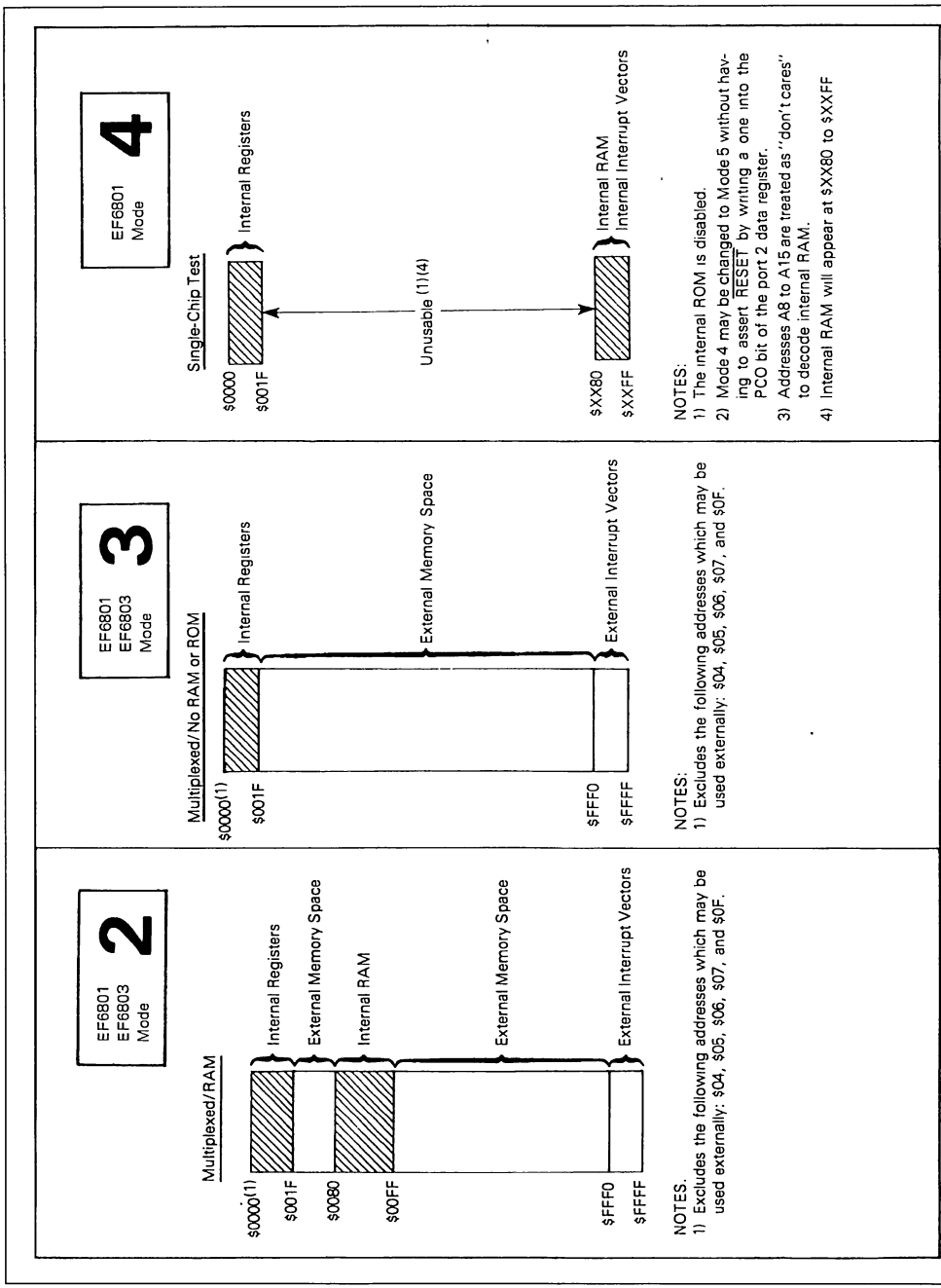
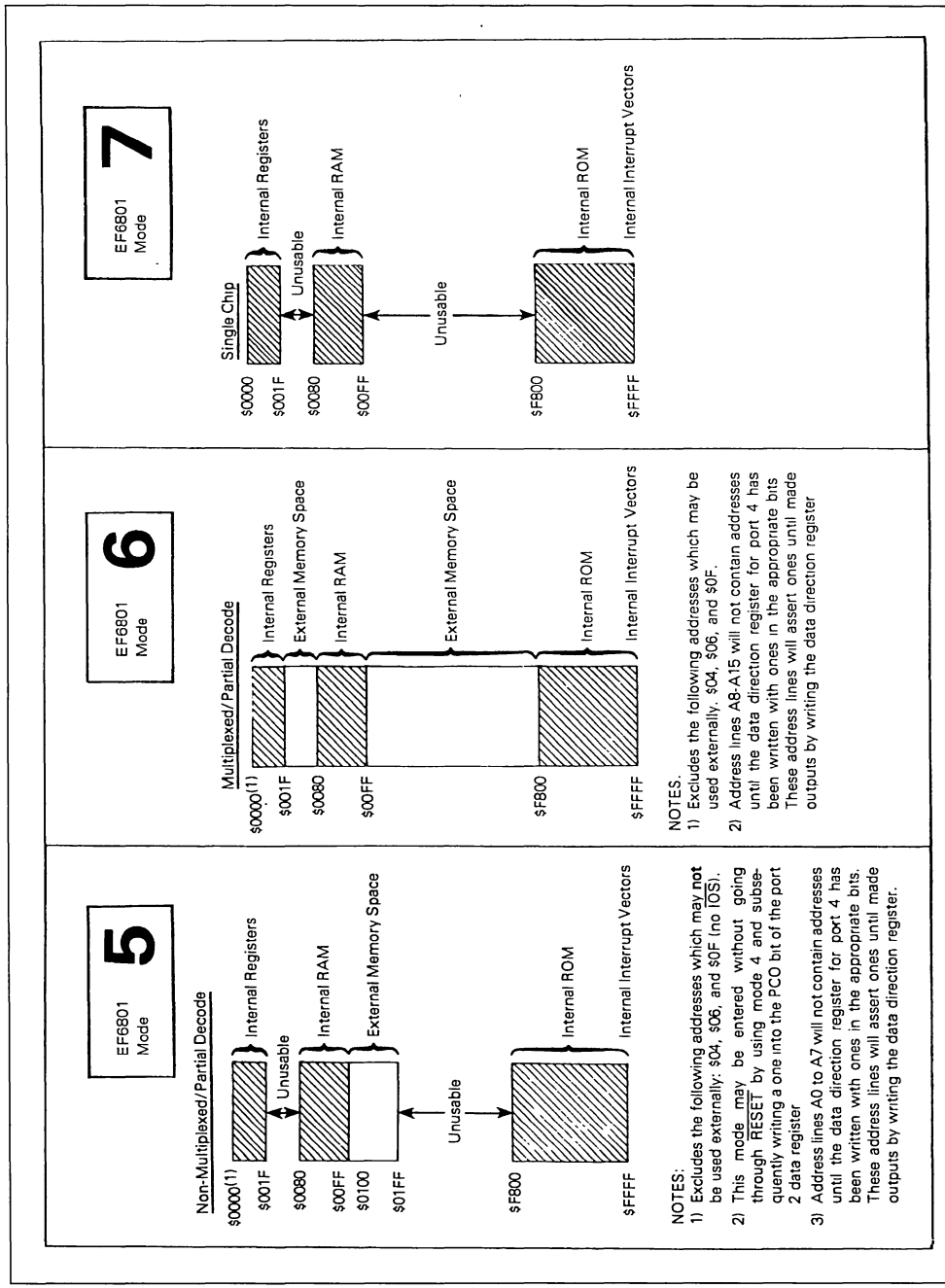


Figure 17 : EF6801/03 Memory Maps (sheet 3 of 3).



EF6801/03 INTERRUPTS

The 6801 Family supports two types of interrupt requests : maskable and non-maskable. A Non-Maskable Interrupt (NMI) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the Condition Code Register I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types : IRQ1 and IRQ2. The Programmable Timer and Serial Communications Interface use an internal IRQ2 interrupt line, as shown in figure 1. External devices (and IS3) use IRQ1. An IRQ1 interrupt is serviced before IRQ2 if both are pending.

All IRQ2 interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order and each is vectored to a separate location. All interrupt vector locations are shown in table 5.

The Interrupt flowchart is depicted in figure 18 and is common to every interrupt excluding reset. During interrupt servicing the Program Counter, Index Register, A Accumulator, B Accumulator, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. Interrupt and RESET timing are illustrated in figures 19 and 20.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power to a large portion of the MCU. The power supply should provide + 5 volts (± 5%) to V_{CC}, and V_{SS} should be tied to ground. Total power dissipation (including V_{CC} Standby), will not exceed P_D milliwatts.

V_{CC} STANDBY

V_{CC} Standby provides power to the standby portion ($80 through $BF) of the RAM and the STBY PWR and RAME bits of the RAM Control Register. Voltage requirements depend on whether the device is in a powerup or powerdown state. In the powerup state, the power supply should provide + 5 volts (± 5%) and must reach V_{SB} volts before RESET reaches 4.0 volts. During powerdown, V_{CC} Standby must remain above V_{SBB} (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} Standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during powerdown operation. V_{CC} Standby should be tied to ground in Mode 3.

Table 4 : Internal Register Area.

Register	Address
Port 1 Data Direction Register***	00
Port 2 Data Direction Register***	01
Port 1 Data Register	02
Port 2 Data Register	03
Port 3 Data Direction Register***	04*
Port 4 Data Direction Register***	05**
Port 3 Data Register	06*
Port 4 Data Register	07**
Timer Control and Status Register	08
Counter (high byte)	09
Counter (low byte)	0A
Output Compare Register (high byte)	0B
Output Compare Register (low byte)	0C
Input Capture Register (high byte)	0D
Input Capture Register (low byte)	0E
Port 3 Control and Status Register	0F*
Rate and Mode Control Register	10
Transmit/receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External addresses in Modes 0, 1, 2, 3, 5, 6 ; cannot be accessed in Mode 5 (No $\overline{IOS}$)
** External addresses in Modes 0, 1, 2, 3
*** 1 = Output, 0 = Input

Table 5 : MCU Interrupt Vector Locations.

MSB	LSB	Interrupt
FFFE	FFFF	$\overline{\text{RESET}}$
FFFC	FFFD	$\overline{\text{NMI}}$
FFFA	FFFB	Software Interrupt (SWI)
FFF8	FFF9	$\overline{\text{IRQ1}}$ (or $\overline{\text{IS3}}$)
FFF6	FFF7	ICF (input capture)*
FFF4	FFF5	OCF (output compare)*
FFF2	FFF3	TOF (timer overflow)*
FFF0	FFF1	SCI (RDRF + ORFE + TDRE)*

* IRQ2 Interrupt

Figure 18 : Interrupt Flowchart.

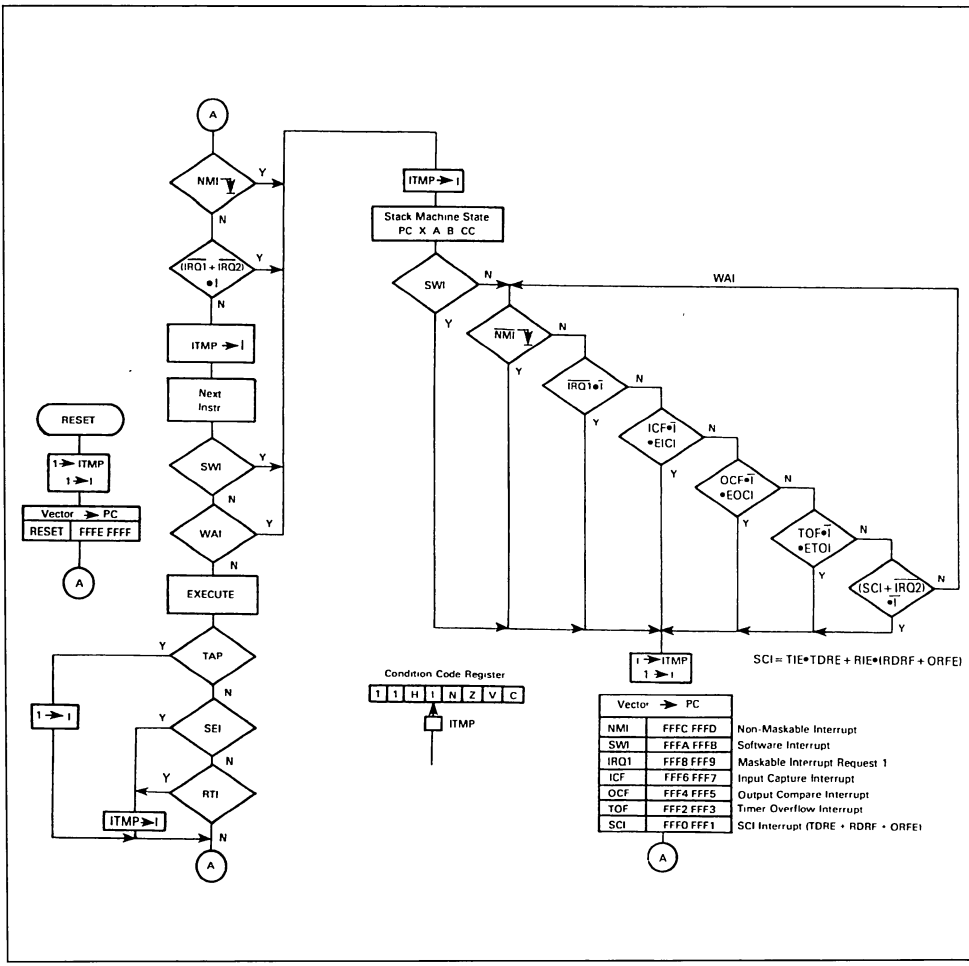


Figure 19 : Interrupt Sequence.

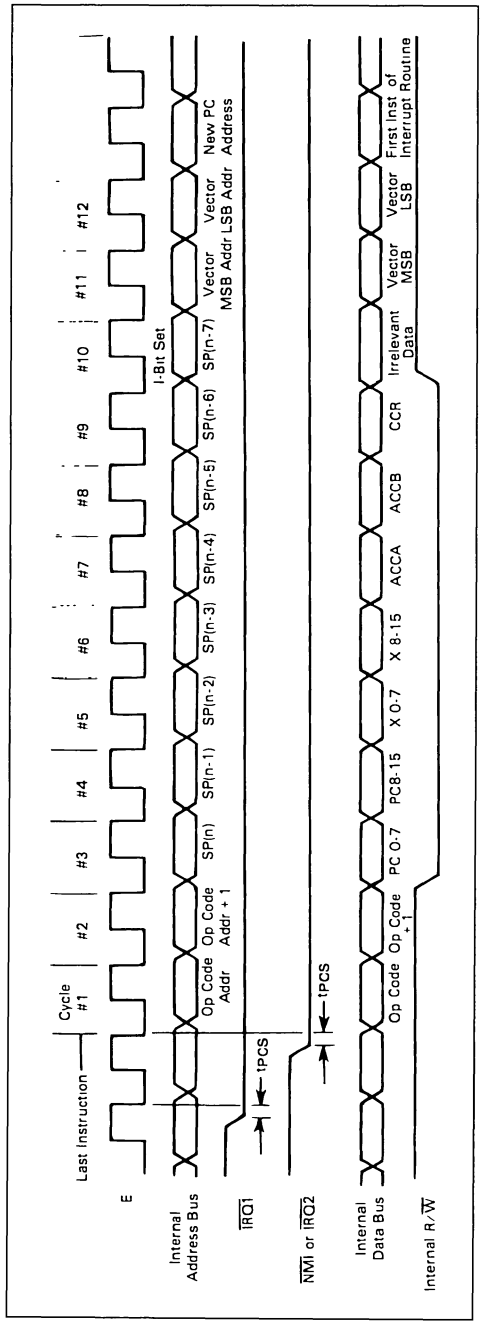
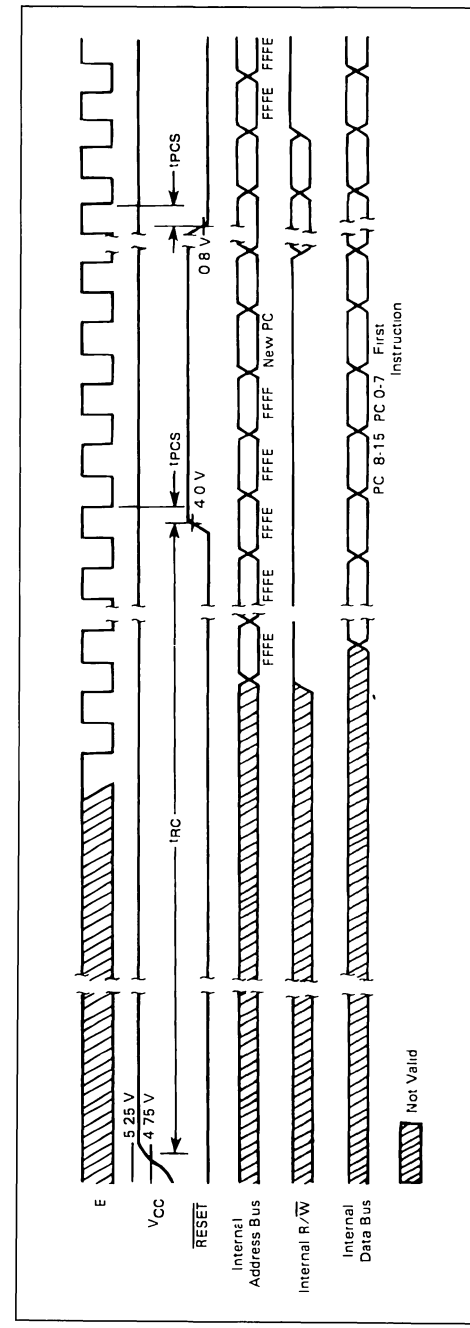


Figure 20 : Reset Timing.



XTAL1 AND EXTAL2

These two input pins interface either a crystal or TTL compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58MHz or 4.4336MHz Color Burst TV crystals. A 20pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL compatible clock at $4f_o$ with a duty cycle of 50% (± 5%) with XTAL1 connected to ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for f_{XTAL}. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.* The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in figure 21.

$\overline{\text{RESET}}$

This input is used to reset the internal state of the device and provide an orderly startup procedure. During powerup, $\overline{\text{RESET}}$ must be held below 0.8 volts : (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} Standby reaches 4.75 volts. $\overline{\text{RESET}}$ must be held low at least three E-cycles if asserted during powerup operation.

E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the device input clock frequency. It will drive one Schottky TTL load and 90pF, and all data given in cycles is referenced to this clock unless otherwise noted.

$\overline{\text{NMI}}$ (NON-MASKABLE INTERRUPT)

An $\overline{\text{NMI}}$ negative edge requests an MCU interrupt sequence, but the current instruction will be completed before it responds to the request. The MCU will then begin an interrupt sequence. Finally, a vector is fetched from $FFFC and $FFFD, transferred to the Program Counter and instruction execution is resumed. $\overline{\text{NMI}}$ typically requires a 3.3kΩ (nominal) resistor to V_{CC}. There is no internal $\overline{\text{NMI}}$ pullup resistor. $\overline{\text{NMI}}$ must be held low for at least one E-cycle to be recognized under all conditions.

$\overline{\text{IRQ1}}$ (MASKABLE INTERRUPT REQUEST 1)

$\overline{\text{IRQ1}}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the MCU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9, transferred to the Program Counter, and instruction execution is resumed.

$\overline{\text{IRQ1}}$ typically requires an external 3.3kΩ (nominal) resistor to V_{CC} for wire-OR applications. IRQ1 has no internal pullup resistor.

SC1 AND SC2 (STROBE CONTROL 1 AND 2)

The function of SC1 and SC2 depends on the operating mode. SC1 is configured as an output in all modes except single chip mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90pF.

SC1 AND SC2 IN SINGLE-CHIP MODE. In Single-Chip Mode, SC1 and SC2 are configured as an input and output, respectively, and both function as Port 3 control lines. SC1 functions as $\overline{\text{IS3}}$ and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with $\overline{\text{IS3}}$ are controlled by Port 3 Control and Status Register and are discussed in the Port 3 description. If unused, $\overline{\text{IS3}}$ can remain unconnected.

SC2 is configured as $\overline{\text{OS3}}$ and can be used to strobe output data or acknowledge input data. It is controlled by Output Strobe Select (OSS) in the Port 3 Control and Status Register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the Port 3 Data Register. $\overline{\text{OS3}}$ timing is shown in figure 4.

SC1 AND SC2 IN EXPANDED NON-MULTIPLEXED MODE. In the Expanded Non-Multiplexed Mode, both SC1 and SC2 are configured as outputs. SC1 functions as Input/Output Select ($\overline{\text{IOS}}$) and is asserted only when $0100 through $01FF is sensed on the internal address bus.

SC2 is configured as Read/Write and is used to control the direction of data bus transfers. An MPU read is enabled when Read/Write and E are high.

SC1 AND SC2 IN EXPANDED MULTIPLEXED MODE. In the Expanded Multiplexed Modes, both SC1 and SC2 are configured as outputs. SC1 functions as Address Strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by Address Strobe captures address on the negative edge, as shown in figure 14.

SC2 is configured as Read/Write and is used to control the direction of data bus transfers. An MPU read is enabled when Read/Write and E are high.

P10-P17 (PORT 1)

Port 1 is a mode independent 8-bit I/O port with each line an input or output as defined by the Port 1 Data Direction Register. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30pF, Darlington transistors, or CMOS devices using external pullup resistors. It is configured as a data input port by RESET. Unused lines can remain unconnected.

P20-P24 (PORT 2)

Port 2 is a mode-independent, 5-bit, multipurpose I/O port. The voltage levels present on P20, P21, and P22 on the rising edge of RESET determine the operating mode of the MCU. The entire port is then configured as a data input port. The Port 2 lines can be selectively configured as data output lines by setting the appropriate bits in the Port 2 Data Direction Register. The Port 2 Data Register is used to move data through the port. However, if P21 is configured as an output, it will be tied to the timer Output Compare function and cannot be used to provide output from the Port 2 Data Register.

Port 2 can also be used to provide an interface for the Serial Communications Interface and the timer Input Edge function. These configurations are described in the Programmable Timer and Serial Communications Interface (SCI) section.

The Port 2 three-state, TTL-compatible output buffers are capable of driving one Schottky TTL load and 30pF, or CMOS devices using external pullup resistors.

PORT2 DATA REGISTER

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$0003

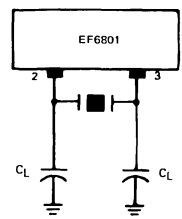
Figure 21 : 6801 Family Oscillator Characteristics.

Nominal Crystal Parameters*

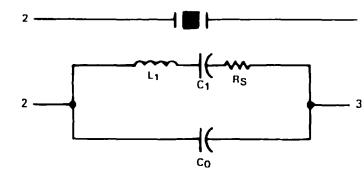
	3.58MHz	4.00MHz	5.0MHz	6.0MHz	8.0MHz
RS	60Ω	50Ω	30-50Ω	30-50Ω	20-40Ω
C0	3.5pF	6.5pF	4-6pF	4-6pF	4-6pF
C1	0.015pF	0.025pF	0.01-0.02pF	0.01-0.02pF	0.01-0.02pF
Q	> 40K	> 30K	> 20K	> 20K	> 20K

Note : These are representative AT-cut crystal parameters only. Crystals of other types of cut may also be used.

(a) Nominal Recommended Crystal Parameters

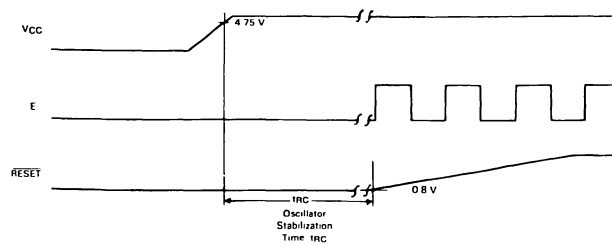


C_L = 20 pF (typical)



Equivalent Circuit

(b) Oscillator Stabilization Time (t_{RC})



P30-P37 (PORT 3)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address/data bus depending on the operating mode. The TTL compatible three-state output buffers can drive one Schottky TTL load and 90pF. Unused lines can remain unconnected.

PORT 3 IN SINGLE-CHIP MODE. Port 3 is an 8-bit I/O port in the Single-Chip Mode, with each line configured by the Port 3 Data Direction Register. There are also two lines, IS3 and OS3, which can be used to control Port 3 data transfers.

Three Port 3 options are controlled by the Port 3 Control and Status Register and are available only in Single-Chip Mode : (1) Port 3 input data can be latched using IS3 as a control signal, (2) OS3 can be generated by either an MPU read or write to the Port 3 Data Register, and (3) an IRQ1 interrupt can be enabled by an IS3 negative edge. Port 3 latch timing is shown in figure 5.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3	IS3 IRQ1 Enable	X	OSS	Enable	X	X	X	$000F

Bit 0-2 Not used.

Bit 3 LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an IS3 negative edge. The latch is transparent after a read of the Port 3 Data Register. LATCH ENABLE is cleared during reset.

Bit 4 OSS (Output Strobe Select). This bit determines whether OS3 will be generated by a read or write of the Port 3 Data Register. When clear, the strobe is generated by a read ; when set, it is generated by a write. OSS is cleared during reset.

Bit 5 Not used.

Bit 6 IS3 IRQ1 ENABLE. When set, an IRQ1 interrupt will be enabled whenever IS3 FLAG is set ; when clear, the interrupt is inhibited. This bit is cleared during reset.

Bit 7 IS3 FLAG. This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the Port 3 Control and Status Register (with IS3 FLAG set) followed by a read or write to the Port 3 Data Register or during reset.

PORT 3 IN EXPANDED NON-MULTIPLEXED MODE. Port 3 is configured as a bidirectional data bus (D7-D0) in the Expanded Non-Multiplexed Mode. The direction of data transfers is controlled by Read/Write (SC2). Data is clocked by E (Enable).

PORT 3 IN EXPANDED MULTIPLEXED MODE. Port 3 is configured as a time multiplexed address (A0-A7) and data bus (D7-D0) in the Expanded Multiplexed Modes, where Address Strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high impedance state between valid address and data to prevent bus conflicts.

P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90pF and is the only port with internal pullup resistors. Unused lines can remain unconnected.

PORT 4 IN SINGLE-CHIP MODE. In Single-Chip Mode, Port 4 functions as an 8-bit I/O port with each line configured by the Port 4 Data Direction Register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

PORT 4 IN EXPANDED NON-MULTIPLEXED MODE. Port 4 is configured from reset as an 8-bit input port, where the Port 4 Data Direction Register can be written to provide any or all of eight address lines, A0 to A7. Internal pullup resistors pull the lines high until the Port 4 Data Direction Register is configured.

PORT 4 IN EXPANDED MULTIPLEXED MODE. In all Expanded Multiplexed modes except Mode 6, Port 4 functions as half of the address bus and provides A8 to A15. In Mode 6, the port is configured from reset as an 8-bit parallel input port, where the Port 4 Data Direction Register can be written to provide any or all of upper address lines A8 to A15. Internal pullup resistors pull the lines high until the Port 4 Data Direction Register is configured, where bit 0 controls A8.

RESIDENT MEMORY

The EF6801 provides 2048 bytes of on-board ROM and 128 bytes of on-board RAM.

One half of the RAM is powered through the V_{CC} standby pin and is maintainable during V_{CC} powerdown. This standby portion of the RAM consists of 64 bytes located from $80 through $BF.

Power must be supplied to V_{CC} standby if the internal RAM is to be used regardless of whether standby power operation is anticipated.

The RAM is controlled by the RAM Control Register.

RAM CONTROL REGISTER ($14)

The RAM Control Register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during powerdown operation. It is intended that RAME be cleared and STBY PWR be set as part of a powerdown procedure.

RAM CONTROL REGISTER

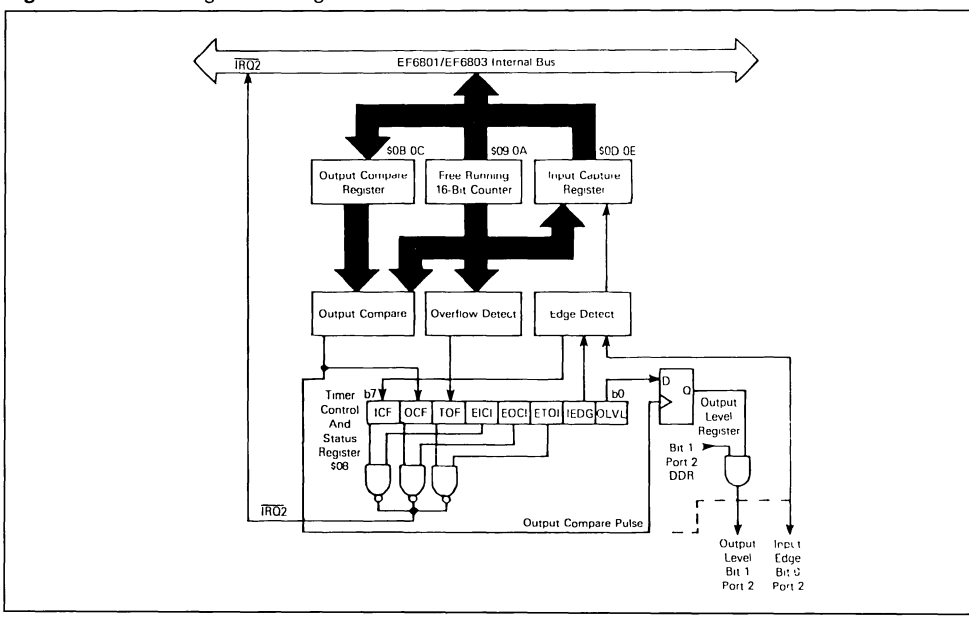
7	6	5	4	3	2	1	0
STY PWR	RAME	X	X	X	X	X	X

Bit 0-5 Not used

Bit 6 RAME RAM Enable. This read/write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of RESET. If RAME is clear, any access to a RAM address is external. If RAME is set and not in mode 3, the RAM is included in the internal map.

Bit 7 STBY PWR Standby Power. This bit is a read/write status bit which, when once set, remains set as long as V_{CC} standby remains above V_{SBB} (minimum). As long as this bit is set following a period of standby operation, the standby power supply has adequately preserved the data in the standby RAM. If this bit is cleared during a period of standby operation, it indicates that V_{CC} standby had fallen to a level sufficiently below V_{SBB} (minimum) to suspect that data in the standby RAM is not valid. This bit can be set only by software and is not affected during reset.

Figure 22 : Block Diagram of Programmable Timer.



PROGRAMMABLE TIMER

The programmable timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in figure 22.

COUNTER ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (enable). It is cleared during reset and is read-only with one exception : a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all ones.

OUTPUT COMPARE REGISTER ($0B:0C)

The output compare register is a 16-bit read/write register used to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E cycle. When a match occurs, OCF is set and OLVL is clocked to an output level register. If port 2, bit 1, is configured as an output, OLVL will appear at P21 and the output compare register and OLVL can then be changed for the next and OLVL is clocked to an output level register. If Port 2, bit 1, is configured as an output, OLVL will appear at P21 and the Output Compare Register and OLVL can then be changed for the next compare. The function is inhibited for one cycle after a write to its high byte ($0B) to ensure a valid compare. The Output Compare Register is set to $FFFF at RESET.

INPUT CAPTURE REGISTER ($0D:0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20 even when configured as an output. An input capture can occur independently of ICF : the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTER ($08)

The Timer Control and Status Register (TCSR) is an 8-bit register of which all bits are readable, while only bits 0-4 can be written. The three most significant bits provide the timer status and indicate if :

- a proper level transition has been detected,
- a match has occured between the free-running counter and the output compare register, and
- the free-running counter has overflowed.

Each of the three events can generate an $\overline{IRQ2}$ interrupt and is controlled by an individual enable bit in the TCSR.

TIMER CONTROL AND STATUS REGISTER (TCSR)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$008

Bit 0 — OLVL Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P21 if Bit 1 of the Port 2 Data Direction Register is set. It is cleared during reset.

Bit 1 — EIDG Input Edge. IEDG is cleared during reset and controls which level transition will trigger a counter transfer to the Input Capture Register :

IEDG = 0 Transfer on a negative-edge

IEDG = 1 Transfer on a positive-edge

Bit 2 — ETOI Enable Timer Overflow Interrupt. When set, an IRQ2 interrupt is enabled for a timer overflow ; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 3 EOCI — Enable Output Compare Interrupt. When set, an IRQ2 interrupt is enabled for an output compare ; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 4 EICI — Enable Input Capture Interrupt. When set, an IRQ2 interrupt is enabled for an input capture ; when clear, the interrupt is inhibited. It is cleared during reset.

Bit 5 TOF — Timer Overflow Flag. TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) then reading the counter high byte ($09), or during reset.

Bit 6 OCF — Output Compare Flag. OCF is set when the Output Compare Register matches the free-running counter. It is cleared by reading the TCSR (with OCF set) and then writing to the Output Compare Register ($0B or $0C), or during reset.

Bit 7 ICF — Input Capture Flag. ICF is set to indicate a proper level transition ; it is cleared by reading the TCSR (with ICF set) and then the Input Capture Register High Byte ($0D), or during reset.

SERIAL COMMUNICATIONS INTERFACE (SCI)

A full-duplex asynchronous Serial Communications Interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and Biphase and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

In a typical serial loop multi-processor configuration, the software protocol will usually identify the addresse(s) at the beginning of the message. In order to permit uninterested MPU's to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 1's or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable :

- format : standard mark/space (NRZ) or Bi-phase
- clock : external or internal bit rate clock
- Baud : one of 4 per E-clock frequency, or external clock (x 8 desired baud)
- wake-up feature : enabled or disabled
- interrupt requests : enabled individually for transmitter and receiver
- clock output : internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The Serial Communications Interface includes four addressable registers as depicted in figure 23. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and a read-only Receive Register. The shift registers are not accessible to software.

RATE AND MODE CONTROL REGISTER (RMCR) ($10). The Rate and Mode Control Register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of four write-only bits which are cleared during reset. The two least significant bits control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER (RMCR)

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	$0010

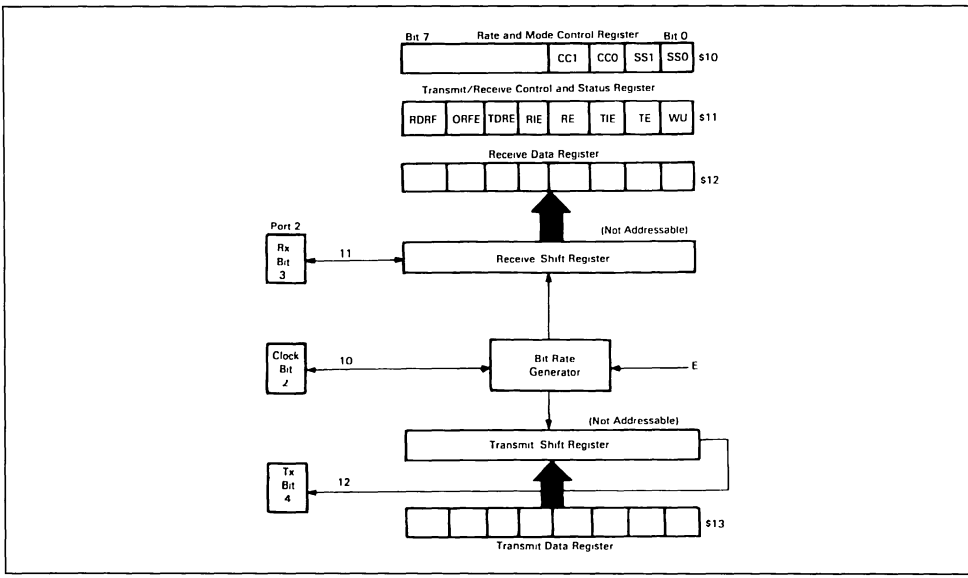
Bit 1 : Bit 0 — SS1: SS0 Speed Select. These two-bits select the Baud rate when using the internal clock. Four rates may be selected which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

Bit 3 : Bit 2 — CC1 : CC0 Clock Control and Format Select. These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P22 at eight times (8X) the desired bit rate, but not greater than E, with a duty cycle of 50% (± 10%). If CC1 : CC0 = 10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE : The source of SCI internal bit rate clock is the timer free running counter. An MPU write to the counter can disturb serial operations.

Figure 23 : SCI Registers.



TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR) ($11). The Transmit/Receive Control and Status Register controls the transmitter, receiver, wake-up feature, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by RESET.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$001

Bit 0 WU — "Wake-up" on Idle Line. When set, WU enables the wake-up function ; it is cleared by ten consecutive 1's or during reset. WU will not set if the line is idle.

Bit 1 TE — Transmit Enable. When set, P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive 1's is transmitted. TE is cleared during reset.

Bit 2 TIE — Transmit Interrupt Enable. When set, an IRQ2 interrupt is enabled when TDRE is set ; when clear, the interrupt is inhibited. TE is cleared during reset.

Bit 3 RE — Receive Enable. When set, the P23 DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared during reset.

Bit 4 RIE — Receiver Interrupt Enable. When set, an IRQ2 interrupt is enabled when RDRF and/or ORFE is set ; when clear, the interrupt is inhibited. RIE is cleared during reset.

Bit 5 TDRE — Transmit Data Register Empty. TDRE is set when the Transmit Data Register is transferred to the output serial shift register or during reset. It is cleared by reading the TRCSR (with TDRE set) and then writing to the Transmit Data Register. Additional data will be transmitted only if TDRE has been cleared.

Bit 6 ORFE — Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the Receiver Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF : if RDRF is set, then an overrun has occurred ; otherwise a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun condition. Unframed data causing a framing error is transferred to the Receive Data Register. However, subsequent data transfer is blocked until the framing error flag is cleared.* ORFE is cleared by reading the TRCSR (with ORFE set) then the Receive Data Register, or during reset.

Bit 7 RDRF — Receive Data Register Full. RDRF is set when the input serial shift register is transferred to the Receive Data Register. It is cleared by reading the TRCSR (with RDRF set), and then the Receive Data Register, or during reset.

Table 6 : SCI Bit Times and Rates.

SS1	SS0	4 f_o → E	2.4576MHz 614.4kHz	4.0MHz 1.0MHz	4.9152MHz 1.2288MHz
0	0	+ 16	26µs/38.400 Baud	16µs/62.500 Baud	13.0µs/76.800 Baud
0	1	+ 128	208µs/4.800 Baud	128µs/7812.5 Baud	104.2µs/9.600 Baud
1	0	+ 1024	1.67ms/600 Baud$	1.024ms/976.6 Baud	833.3µs/1.200 Baud
1	1	+ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud	3.33ms/300 Baud
*External (P22)			13.0µs/76.800 Baud	8.0µs/125.000 Baud	6.5µs/153.600 Baud

* Using maximum clock rate

Table 7 : SCI Format and Clock Source Control.

CC1:CC0	Format	Clock Source	Port 2 Bit 2
00	Bi-phase	Internal	Not used
01	NRZ	Internal	Not used
10	NRZ	Internal	Output
11	NRZ	External	Input

SERIAL OPERATIONS

The SCI is initialized by writing control bytes first to the Rate and Mode Control Register and then to the Transmit/Receive Control and Status Register. When TE is set, the output of the transmit serial shift register is connected to P24 and serial output is initiated by transmitting a 9-bit preamble of 1's.

At this point one of two situations exist : 1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of 1's will be sent indicating an idle line, or 2) if a byte has been written to the Transmit-Data Register (TDRE = 0), it will be transferred to the output serial shift register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0) and a stop bit (1), will be transmitted. If TDRE is still set when the next byte transfer should occur, 1's will be sent until more data is provided. In Bi-phase format, the output toggles at the start of each bit and at half-bit time when a "1" is sent. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. SCI data formats are illustrated in figure 24.

INSTRUCTION SET

The EF6801/03 is upward source and object code compatible with the EF6800. Execution times of key instructions have been reduced and several new instructions have been added, including a hardware multiply. A list of new operations added to the EF6800 instruction set is shown in table 1.

In addition, two new special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the Program Counter to increment like a 16-bit counter, causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in table 8. There are 220 valid machine codes, 34 unassigned codes, and 2 codes reserved for test purposes.

PROGRAMMING MODEL

A programming model for the EF6801/03 is shown in figure 10. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows :

PROGRAM COUNTER. The program counter is a 16-bit register which always points to the next instruction.

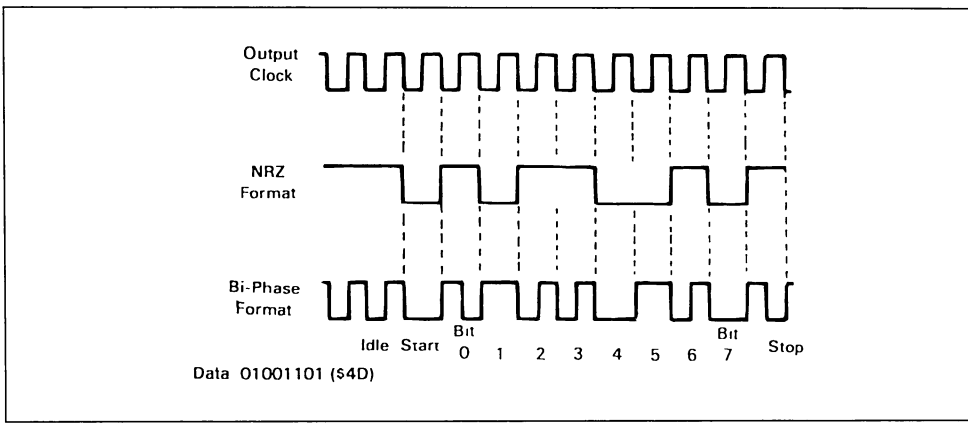
STACK POINTER. The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location defined by the programmer.

INDEX REGISTER. The Index Register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

ACCUMULATORS. The MPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

CONDITION CODE REGISTERS. The condition code register indicates the results of an instruction and includes the following five condition bits : Negative (N), Zero (Z), Overflow (V), Carry/Borrow from MSB (C), and Half Carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7, are read as ones.

Figure 24 : SCI Data Formats.



ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of addressing modes for all instructions is presented in table 9, 10, 11, and 12, where execution times are provided in E-cycles. Instruction execution times are summarized in table 13. With an input frequency of 4MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in table 14 and a description of selected instructions is shown in figure 25.

IMMEDIATE ADDRESSING. The operand or "immediate byte(s)" is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

DIRECT ADDRESSING. The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, the 256-byte area is reserved for frequently referenced data.

EXTENDED ADDRESSING. The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

INDEXED ADDRESSING. The unsigned offset contained in the second byte of the instruction is added with carry to the Index Register and used to reference memory without changing the Index Register. These are two byte instructions.

INHERENT ADDRESSING. The operand(s) are registers and no memory reference is required. These are single byte instructions.

RELATIVE ADDRESSING. Relative addressing is used only for branch instructions. If the branch condition is true, the Program Counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current Program Counter. This provides a branch range of – 126 to 129 bytes from the first byte of the instruction. These are two byte instructions.

Table 8 : CPU Instruction Map.

OP	MNEM	MODE	~	#
00	*			
01	NOP	INHER	2	1
02	*			
03	*			
04	LSRD		3	1
05	ASLD		3	1
06	TAP		2	1
07	TPA		2	1
08	INX		3	1
09	DEX		3	1
0A	CLV		2	1
0B	SEV		2	1
0C	CLC		2	1
0D	SEC		2	1
0E	CLI		2	1
0F	SEI		2	1
10	SBA		2	1
11	CBA		2	1
12	*			
13	*			
14	*			
15	*			
16	TAB		2	1
17	TBA		2	1
18	*			
19	DAA	INHER	2	1
1A	*			
1B	ABA	INHER	2	1
1C	*			
1D	*			
1E	*			
1F	*			
20	BRA	REL	3	2
21	BRN		3	2
22	BHI		3	2
23	BLS		3	2
24	BCC		3	2
25	BCS		3	2
26	BNE		3	2
27	BEQ		3	2
28	BVC		3	2
29	BVS		3	2
2A	BPL		3	2
2B	BMI		3	2
2C	BGE		3	2
2D	BLT		3	2
2E	BGT		3	2
2F	BLE	REL	3	2
30	TSX	INHER	3	1
31	INS		3	1
32	PULA		4	1
33	PULB		4	1
34	DES	INHER	3	1
35	TXS		3	1
36	PSHA		3	1
37	PSHB		3	1
38	PULX		5	1
39	RTS		5	1
3A	ABX		3	1
3B	RTI		10	1
3C	PSHX		4	1
3D	MUL		10	1
3E	WAI		9	1
3F	SWI		12	1
40	NEGA		2	1
41	*			
42	*			
43	COMA		2	1
44	LSRA		2	1
45	*			
46	RORA		2	1
47	ASRA		2	1
48	ASLA		2	1
49	ROLA		2	1
4A	DECA		2	1
4B	*			
4C	INCA		2	1
4D	TSTA		2	1
4E	T			
4F	CLRA		2	1
50	NEGB		2	1
51	*			
52	*			
53	COMB		2	1
54	LSRB		2	1
55	*			
56	RORB		2	1
57	ASRB		2	1
58	ASLB		2	1
59	ROLB		2	1
5A	DECB		2	1
5B	*			
5C	INCB		2	1
5D	TSTB		2	1
5E	T			
5F	CLRB	INHER	2	1
60	NEG	INDXD	6	2
61	*			
62	*			
63	COM		6	2
64	LSR		6	2
65	*			
66	ROR		6	2
67	ASR	INDXD	6	2
68	ASL	INDXD	6	2
69	ROL		6	2
6A	DEC		6	2
6B	*			
6C	INC		6	2
6D	TST		6	2
6E	JMP		3	2
6F	CLR	INDXD	6	2
70	NEG	EXTND	6	3
71	*			
72	*			
73	COM		6	3
74	LSR		6	3
75	*			
76	ROR		6	3
77	ASR		6	3
78	ASL		6	3
79	ROL		6	3
7A	DEC		6	3
7B	*			
7C	INC		6	3
7D	TST		6	3
7E	JMP		3	3
7F	CLR	EXTND	6	3
80	SUBA	IMMED	2	2
81	CMPA		2	2
82	SBCA		2	2
83	SUBD		4	3
84	ANDA		2	2
85	BITA		2	2
86	LDAA		2	2
87	*			
88	EORA		2	2
89	ADCA		2	2
8A	ORAA		2	2
8B	ADDA		2	2
8C	CPX	IMMED	4	3
8D	BSR	REL	6	2
8E	LDS	IMMED	3	3
8F	*			
90	SUBA	DIR	3	2
91	CMPA		3	2
92	SBCA		3	2
93	SUBD		5	2
94	ANDA		3	2
95	BITA		3	2
96	LDAA		3	2
97	STAA		3	2
98	EORA		3	2
99	ADCA		3	2
9A	ORAA		3	2
9B	ADDA		3	2
9C	CPX	DIR	5	2
9D	JSR		5	2
9E	LDS		4	2
9F	STS	DIR	4	2
A0	SUBA	INDXD	4	2
A1	CMPA		4	2
A2	SBCA		4	2
A3	SUBD		6	2
A4	ANDA		4	2
A5	BITA		4	2
A6	LDAA		4	2
A7	STAA		4	2
A8	EORA		4	2
A9	ADCA		4	2
AA	ORAA		4	2
AB	ADDA		4	2
AC	CPX		6	2
AD	JSR		6	2
AE	LDS		5	2
AF	STS	INDXD	5	2
B0	SUBA	EXTND	4	3
B1	CMPA		4	3
B2	SBCA		4	3
B3	SUBD		6	3
B4	ANDA		4	3
B5	BITA		4	3
B6	LDAA		4	3
B7	STAA		4	3
B8	EORA		4	3
B9	ADCA		4	3
BA	ORAA		4	3
BB	ADDA		4	3
BC	CPX		6	3
BD	JSR		6	3
BE	LDS		5	3
BF	STS	EXTND	5	3
C0	SUBB	IMMED	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB		2	2
C6	LDAB		2	2
C7	*			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORAB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	*			
CE	LDX	IMMED	3	3
CF	*			
D0	SUBB	DIR	3	2
D1	CMPB		3	2
D2	SBCB		3	2
D3	ADDD		5	2
D4	ANDB		3	2
D5	BITB		3	2
D6	LDAB		3	2
D7	STAB		3	2
D8	EORB		3	2
D9	ADCB		3	2
DA	ORAB		3	2
DB	ADDB		3	2
DC	LDD		4	2
DD	STD		4	2
DE	LDX		4	2
DF	STX	DIR	4	2
E0	SUBB	INDXD	4	2
E1	CMPB		4	2
E2	SBCB		4	2
E3	ADDD		6	2
E4	ANDB		4	2
E5	BITB		4	2
E6	LDAB		4	2
E7	STAB		4	2
E8	EORB		4	2
E9	ADCB		4	2
EA	ORAB		4	2
EB	ADDB		4	2
EC	LDD		5	2
ED	STD		5	2
EE	LDX		5	2
EF	STX	INDXD	5	2
F0	SUBB	EXTND	4	3
F1	CMPB		4	3
F2	SBCB		4	3
F3	ADDD		6	3
F4	ANDB		4	3
F5	BITB		4	3
F6	LDAB		4	3
F7	STAB		4	3
F8	EORB		4	3
F9	ADCB		4	3
FA	ORAB		4	3
FB	ADDB		4	3
FC	LDD		5	3
FD	STD		5	3
FE	LDX		5	3
FF	STX	EXTND	5	3

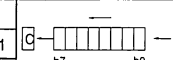
* UNDEFINED OP CODE

Notes : 1. Addressing Modes
INHER = Inherent INDXD = Indexed IMMED = Immediate
REL = Relative EXTND = Extended DIR = Direct
2 Unassigned opcodes are indicated by " " and should not be executed
3 Codes marked by "T" force the PC to function as a 16-bit counter.

Table 9 : Index Register and Stack Manipulation Instructions.

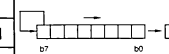
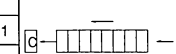
Pointer Operations	Mnem	Immed			Direct			Index			Extnd			Inherent			Boolean/ Arithmetic Operation	Condition Codes					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		5	4	3	2	1	0
																		H	I	N	Z	V	C
Compare Index Register	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X – M : M + 1	•	•	↕	↕	↕	↕
Decrement Index Register	DEX													09	3	1	X – 1 → X	•	•	•	↕	•	•
Decrement Stack Pointer	DES													34	3	1	SP – 1 → SP	•	•	•	•	•	•
Increment Index Register	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pointer	INS													31	3	1	1 SP + 1 → SP	•	•	•	•	•	•
Load Index Register	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	↕	↕	R	•
Load Stack Pointer	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	•	•	↕	↕	R	•
Store Index Register	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	↕	↕	R	•
Store Stack Pointer	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	↕	↕	R	•
Index Reg → Stack Pointer	TXS													35	3	1	X – 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Register	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP – 1 → SP X_H → M_{SP}, SP – 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•

Table 10 : Accumulator and Memory Instructions.

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00 : B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M : M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A – M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B – M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕

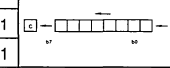
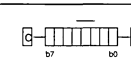
The Condition Code Register notes are listed after Table 12

Table 10 : Accumulator and Memory Instructions (continued).

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Shift Left Dbl	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A – M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B – M	•	•	↕	↕	R	•
Compare Acmltrs	CBA													11	2	1	A – B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A – M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B – M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A} \rightarrow A$	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B} \rightarrow B$	•	•	↕	↕	R	S
Decimal Adj., A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M – 1→ M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A – 1→ A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B – 1→ B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M M + 1 → D	•	•	↕	↕	R	•
Logical Shift, Left	LSL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	LSLA													48	2	1		•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	2		•	•	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R	↕	↕	↕
	LSRA													44	2	1		•	•	R	↕	↕	↕
	LSRB													54	2	1		•	•	R	↕	↕	↕
	LSRD													04	3	1		•	•	R	↕	↕	↕

The Condition Code Register notes are listed after Table 12

Table 10 : Accumulator and Memory Instructions (continued).

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Multiply	MUL													3D	10	1	A x B → D	•	•	•	•	•	↕
2's Complement (negate)	NEG							60	6	2	70	6	3				00 – M → M	•	•	↕	↕	↕	↕
	NEGA													40	2	1	00 – A → A	•	•	↕	↕	↕	↕
	NEGB													50	2	1	00 – B → B	•	•	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	↕	↕
	ROLA													49	2	1		•	•	↕	↕	↕	↕
	ROLB													59	2	1		•	•	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	↕	↕
	RORA													46	2	1		•	•	↕	↕	↕	↕
	RORB													56	2	1		•	•	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A – B → A	•	•	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A – M – C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B – M – C → B	•	•	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
	STD				DD	4	2	ED	5	2	FD	5	3				D → M . M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A – M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B – M → B	•	•	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D – M · M + 1 → D	•	•	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M – 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A – 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B – 00	•	•	↕	↕	R	R

The Condition Code Register notes are listed after table 12.

Table 11 : Jump and Branch Instructions.

Operations	MNEM	Direct			Relative			Index			Extend			Inherent			Branch Test	Cond. Code Reg.					
																		5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch if Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•
Branch if Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•
Branch if = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•
Branch if ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•
Branch if > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch if Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•
Branch if Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•
Branch if ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch if Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•
Branch if Lower or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•
Branch if < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•
Branch if Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•
Branch if not Equal Zero	BNE				26	3	2										Z = 0	•	•	•	•	•	•
Branch if Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•
Branch if Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•
Branch if Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•
Branch to Subroutine	BSR				8D	6	2										See Special Operations- figure 26	•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump to Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return from Interrupt	RTI													3B	10	1	See Special Operations- figure 26	↕	↕	↕	↕	↕	↕
Return from Subroutine	RTS													39	5	1		•	•	•	•	•	•
Sofware Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI													3E	9	1		•	•	•	•	•	•

The Condition Code Register notes are listed after Table 12

Table 12 : Condition Code Register Manipulation Instructions.

Operations	Inherent				Boolean Operation	Condition Code Register					
						5	4	3	2	1	0
	Mnem	OP	~	#		H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	A → CCR	↕	↕	↕	↕	↕	↕
CCR → Accumulator A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

LEGEND
- OP Operation Code (Hexadecimal)
- ~ Number of MPU Cycles
- M_{SP} Contents of memory location pointed to by Stack Pointer
- # Number of Program Bytes
- + Arithmetic Plus
- – Arithmetic Minus
- • Boolean AND
- X Arithmetic Multiply
- + Boolean Inclusive OR
- ⊕ Boolean Exclusive OR
- $\overline{M}$ Complement of M
- → Transfer Into
- 0 Bit = Zero
- 00 Byte = 0

CONDITION CODE SYMBOLS
- H Half-carry from bit 3
- I Interrupt mask
- N Negative (sign bit)
- Z Zero (byte)
- V Overflow, 2's complement
- C Carry/Borrow from MSB
- R Reset Always
- S Set Always
- ↕ Affected
- • Not Affected

Table 13 : Instruction Execution Times In E-cycles.

	Addressing Mode					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
ABA	●	●	●	●	2	●
ABX	●	●	●	●	3	●
ADC	2	3	4	4	●	●
ADD	2	3	4	4	●	●
ADDD	4	5	6	6	●	●
AND	2	3	4	4	●	●
ASL	●	●	6	6	2	●
ASLD	●	●	●	●	3	●
ASR	●	●	6	6	2	●
BCC	●	●	●	●	●	3
BCS	●	●	●	●	●	3
BEQ	●	●	●	●	●	3
BGE	●	●	●	●	●	3
BGT	●	●	●	●	●	3
BHI	●	●	●	●	●	3
BHS	●	●	●	●	●	3
BIT	2	3	4	4	●	●
BLE	●	●	●	●	●	3
BLO	●	●	●	●	●	3
BLS	●	●	●	●	●	3
BLT	●	●	●	●	●	3
BMI	●	●	●	●	●	3
BNE	●	●	●	●	●	3
BPL	●	●	●	●	●	3
BRA	●	●	●	●	●	3
BRN	●	●	●	●	●	3
BSR	●	●	●	●	●	6
BVC	●	●	●	●	●	3
BVS	●	●	●	●	●	3
CBA	●	●	●	●	2	●
CLC	●	●	●	●	2	●
CLI	●	●	●	●	2	●
CLR	●	●	6	6	2	●
CLV	●	●	●	●	2	●
CMP	2	3	4	4	●	●
COM	●	●	6	6	2	●
CPX	4	5	6	6	●	●
DAA	●	●	●	●	2	●
DEC	●	●	6	6	2	●
DES	●	●	●	●	3	●
DEX	●	●	●	●	3	●
EOR	2	3	4	4	●	●
INC	●	●	6	6	●	●
INS	●	●	●	●	3	●

	Addressing Mode					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
INX	●	●	●	●	3	●
JMP	●	●	3	3	●	●
JSR	●	5	6	6	●	●
LDA	2	3	4	4	●	●
LDD	3	4	5	5	●	●
LDS	3	4	5	5	●	●
LDX	3	4	5	5	●	●
LSL	●	●	6	6	2	●
LSLD	●	●	●	●	3	●
LSR	●	●	6	6	2	●
LSRD	●	●	●	●	3	●
MUL	●	●	●	●	10	●
NEG	●	●	6	6	2	●
NOP	●	●	●	●	2	●
ORA	2	3	4	4	●	●
PSH	●	●	●	●	3	●
PSHX	●	●	●	●	4	●
PUL	●	●	●	●	4	●
PULX	●	●	●	●	5	●
ROL	●	●	6	6	2	●
ROR	●	●	6	6	2	●
RTI	●	●	●	●	10	●
RTS	●	●	●	●	5	●
SBA	●	●	●	●	2	●
SBC	2	3	4	4	●	●
SEC	●	●	●	●	2	●
SEI	●	●	●	●	2	●
SEV	●	●	●	●	2	●
STA	●	3	4	4	●	●
STD	●	4	5	5	●	●
STS	●	4	5	5	●	●
STX	●	4	5	5	●	●
SUB	2	3	4	4	●	●
SUBD	4	5	6	6	●	●
SWI	●	●	●	●	12	●
TAB	●	●	●	●	2	●
TAP	●	●	●	●	2	●
TBA	●	●	●	●	2	●
TPA	●	●	●	●	2	●
TST	●	●	6	6	2	●
TSX	●	●	●	●	3	●
TXS	●	●	●	●	3	●
WAI	●	●	●	●	9	●

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 14 provides a detailed description of the information present on the address bus, data bus, and the read/write (R/W) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Note that during MPU reads of internal locations, the resultant value will not appear on the external data bus except in mode 0. "High order" byte refers to the most-significant byte of a 16-bit value.

Table 14 : Cycle-by-cycle Operation (sheet 1 of 5)

IMMEDIATE

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Operand Data
LDS LDX LDD	3	1 2 3	Opcode Address Opcode Address + 1 Opcode Address + 2	1 1 1	Opcode Operand Data (high order byte) Operand Data (low order byte)
CPX SUBD ADDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Opcode Address + 2 Address Bus FFFF	1 1 1 1	Opcode Operand Data (high order byte) Operand Data (low order byte) Low Byte of Restart Vector

DIRECT

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Opcode Address Opcode Address + 1 Address of Operand	1 1 1	Opcode Address of Operand Operand Data
STA	3	1 2 3	Opcode Address Opcode Address + 1 Destination Address	1 1 0	Opcode Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Opcode Address Address of Operand Operand Data (high order byte) Operand Data (low order byte)
STS STX STD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Opcode Address Address of Operand Operand Data (high order byte) Operand Data (low order byte)
CPX SUBD ADDD	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Opcode Address of Operand Operand Data (high order byte) Operand Data (low order byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Subroutine Address Stack Pointer Stack Pointer – 1	1 1 1 0 0	Opcode Irrelevant Data First Subroutine Opcode Return Address (low order byte) Return Address (high order byte)

Table 14 : Cycle-by-cycle Operation (sheet 2 of 5)

EXTENDED

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Jump Address (high order byte)
		3	Opcode Address + 2	1	Jump Address (low order byte)
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Address of Operand
AND ORA		3	Opcode Address + 2	1	Address of Operand (low order byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Destination Address (high order byte)
		3	Opcode Address + 2	1	Destination Address (low order byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Address of Operand (high order byte)
LDD		3	Opcode Address + 2	1	Address of Operand (low order byte)
		4	Address of Operand	1	Operand Data (high order byte)
		5	Address of Operand + 1	1	Operand Data (low order byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Address of Operand (high order byte)
STD		3	Opcode Address + 2	1	Address of Operand (low order byte)
		4	Address of Operand	0	Operand Data (high order byte)
		5	Address of Operand + 1	0	Operand Data (low order byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Address of Operand (high order byte)
CLR ROL		3	Opcode Address + 2	1	Address of Operand (low order byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Operand Address (high order byte)
ADDD		3	Opcode Address + 2	1	Operand Address (low order byte)
		4	Operand Address	1	Operand Data (high order byte)
		5	Operand Address + 1	1	Operand Data (low order byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector

* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF

Table 14 : Cycle-by-cycle Operation (sheet 2 of 5)

EXTENDED (continued)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Subroutine (high order byte)
		3	Opcode Address + 2	1	Address of Subroutine (low order byte)
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (low order byte)
		6	Stack Pointer – 1	0	Return Address (high order byte)

* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

Table 14 : Cycle-by-cycle Operation (sheet 3 of 5)

INDEXED

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (high order byte)
		5	Index Register Plus Offset + 1	1	Operand Data (low order byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (high order byte)
		5	Index Register Plus Offset + 1	0	Operand Data (low order byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (high order byte)
		5	Index Register + Offset + 1	1	Operand Data (low order byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (low order byte)
		6	Stack Pointer – 1	0	Return Address (high order byte)

* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

Table 14 : Cycle-by-cycle Operation (sheet 4 of 5)

INHERENT

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Opcode Irrelevant Data Index Register (low order byte) Index Register (high order byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (low order byte) Index Register (high order byte)

Table 14 : Cycle-by-cycle Operation (sheet 4 of 5)

INHERENT (continued)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
RTS	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Address of Next Instruction (high order byte)
		5	Stack Pointer + 2	1	Address of Next Instruction (low order byte)
WAI	9	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Opcode of Next Instruction
		3	Stack Pointer	0	Return Address (low order byte)
		4	Stack Pointer – 1	0	Return Address (high order byte)
		5	Stack Pointer – 2	0	Index Register (low order byte)
		6	Stack Pointer – 3	0	Index Register (high order byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Condition Code Register

Table 14 : Cycle-by-cycle Operation (sheet 5 of 5)

INHERENT

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
MUL	10	1	Opcode Addres	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (high order byte)
		8	Stack Pointer + 5	1	Index Register from Stack (low order byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (high order byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (low order byte)

Table 14 : Cycle-by-cycle Operation (sheet 5 of 5)

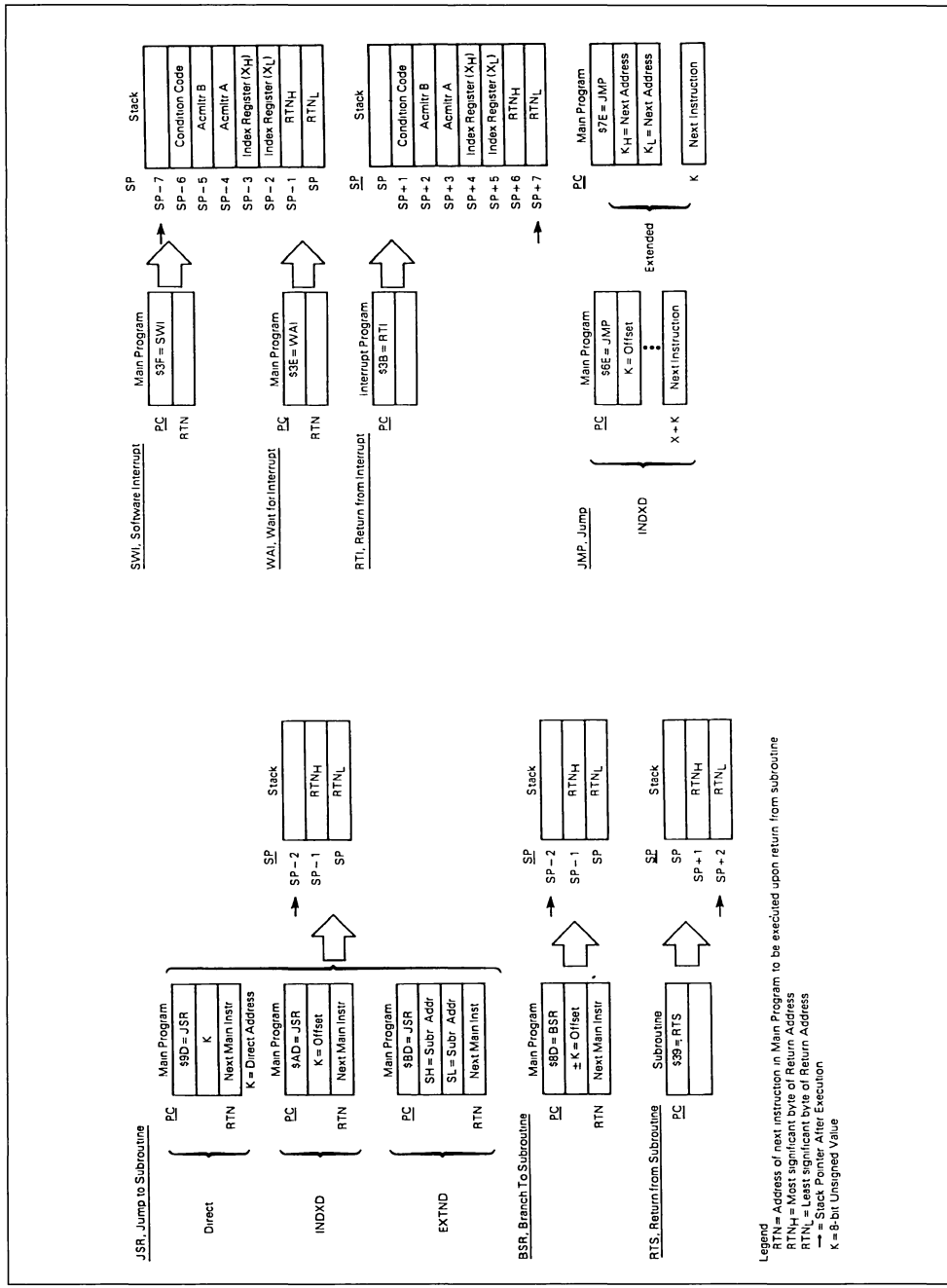
INHERENT (continued)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
SWI	12	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (low order byte)
		4	Stack Pointer – 1	0	Return Address (high order byte)
		5	Stack Pointer – 2	0	Index Register (low order byte)
		6	Stack Pointer – 3	0	Index Register (high order byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Condition Code Register
		10	Stack Pointer – 7	1	Irrelevant Data
		11	Vector Address FFFA (hex)	1	Address of Subroutine (high order byte)
		12	Vector Address FFFB (hex)	1	Address of Subroutine (low order byte)

RELATIVE

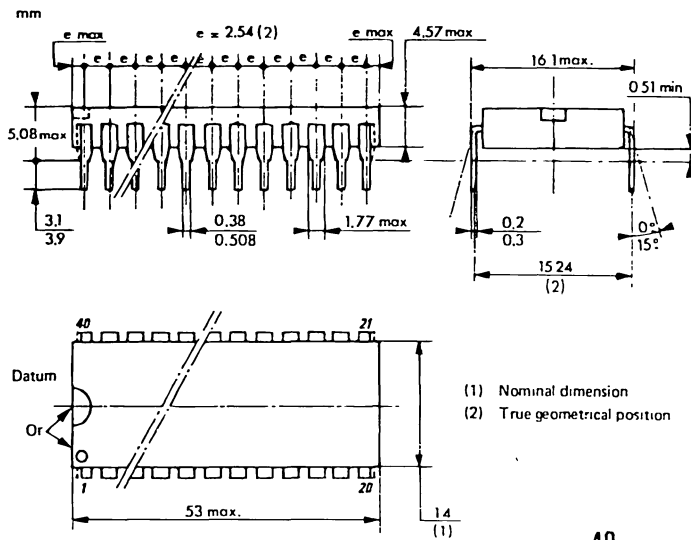
Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMI BVS	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
BSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (low order byte)
		6	Stack Pointer – 1	0	Return Address (high order byte)

Figure 25 : Special Operations.

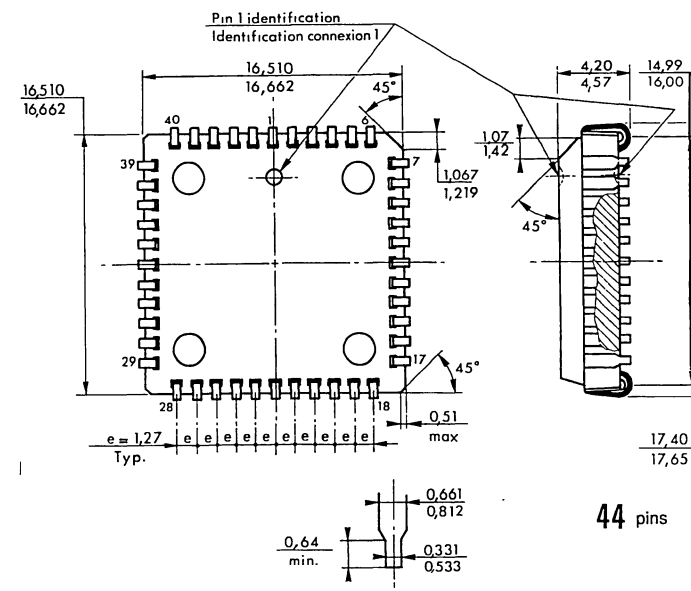


PACKAGE MECHANICAL DATA

P SUFFIX - PLASTIC PACKAGE



FN SUFFIX - PLASTIC PACKAGE



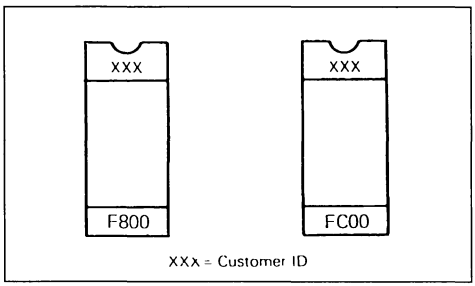
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS THOMSON on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local SGS THOMSON representative or distributor.

EPROMs

Two 2708 or one 2716 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



After the EPROM (s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS THOMSON. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, SGS-THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by SGS THOMSON. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6801 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS THOMSON factory representatives.

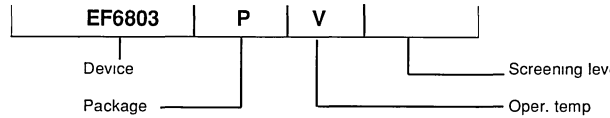
EFDOS is SGS THOMSON Disk Operating System available on development systems such as DEVICE, ...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser,...

* Requires prior factory approval

Whenever ordering a custom MCU is required, please contact your local SGS THOMSON representative or SGS THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS THOMSON Microelectronics representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device		Package					Oper. Temp			Screening Level			
		C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
1.0MHz	EF6801/03			●		●	●			●			
				●				●		●			
	EF6803							●		●			
1.25MHz	EF6801/03-1			●		●	●			●			
				●				●		●			
	EF6803-1							●		●			
1.5MHz	EF68A01/03			●			●			●			
	EF68A03							●		●			
2.0MHz	EF68B01/03			●			●			●			
Examples : EF6801P, EF6801FN, EF6801PV.													

Package : C : Ceramic DIL, **J** : Cerdip DIL, **P** Plastic DIL, **E** : LCCC, **FN** . PLCC;
Oper. temp. : L* : 0°C to 70°C, **V** : –40°C to +85°C, **M** : –55°C to +125°C, * . may be omitted
Screening level : Std : (no-end suffix), **D** : NFC 96883 level D,
G/B : NFC 96883 level G, **B/B** : NFC 96883 level B and MIL-STD-883C level B.

EXORciser is a registered trade mark of MOTOROLA Inc.

EF6801 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : |_|_|_|_|_|_|_|_|

Customer's marking : |_|_|_|_|_|_|_|_|

Application : .....

.....

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
- ❑ SGS-THOMSON Microelectronics reference
  .....
- ❑ Special customer data reference*
  .....

ROM capacity required : |_|_|_|_| bytes

Number of interrupt vector : |_|_|

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 40°C / + 85°C
- ❑ – 40°C / + 105°C

Quality level :
- ❑ STD
- ❑ D
- ❑ Other* (customer's quality specification ref.) :

.....

Package :
- ❑ Plastic
- ❑ PLCC

Software developped by :
- ❑ SGS-THOMSON Microelectronics application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
- ❑ EPROM Reference :
- ❑ EFDOS/MDOS* disk file
  - ❑ 8" floppy
  - ❑ 5" 1/4 floppy
- ❑ Other *

OPTION LIST

-Internal max. clock frequency :
- ❑ 1.0MHz
- ❑ 1.25MHz
- ❑ 1.5MHz
- ❑ 2.0MHz

* Requires prior factory approval

Yearly quantity forecast :

- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME : DATE : SIGNATURE :

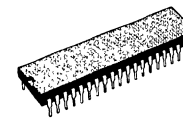
SGS-THOMSON MICROELECTRONICS

EF6801U4
EF6803U4

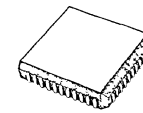
MICROCOMPUTER/MICROPROCESSOR

ADVANCE DATA

- ENHANCED EF6800 INSTRUCTION SET
- UPWARD SOURCE AND OBJECT CODE COMPATIBILITY WITH THE EF6800 AND EF6801
- BUS COMPATIBILITY WITH THE EF6800 FAMILY
- 8 x 8 MULTIPLY INSTRUCTION
- SINGLE-CHIP OR EXPANDED OPERATION TO 64K BYTE ADDRESS SPACE
- INTERNAL CLOCK GENERATOR WITH DIVIDE-BY-FOUR OUTPUT
- SERIAL COMMUNICATIONS INTERFACE (SCI)
- 16-BIT SIX-FUNCTION PROGRAMMABLE TIMER
- THREE OUTPUT COMPARE FUNCTIONS
- TWO INPUT CAPTURE FUNCTIONS
- COUNTER ALTERNATE ADDRESS
- 4096 BYTES OF ROM (EF6801U4)
- 192 BYTES OF RAM
- 32 BYTES OF RAM RETAINABLE DURING POWER DOWN
- 29 PARALLEL I/O AND TWO HANDSHAKE CONTROL LINES
- $\overline{NMI}$ INHIBITED UNTIL STACK LOAD
- – 40°C TO 85°C TEMPERATURE RANGE

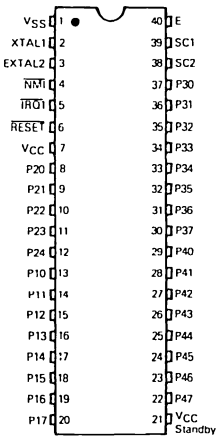


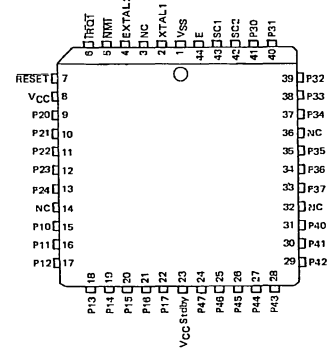
P SUFFIX
(Plastic Package)



FN SUFFIX
(PLCC 44)

PIN CONNECTIONS

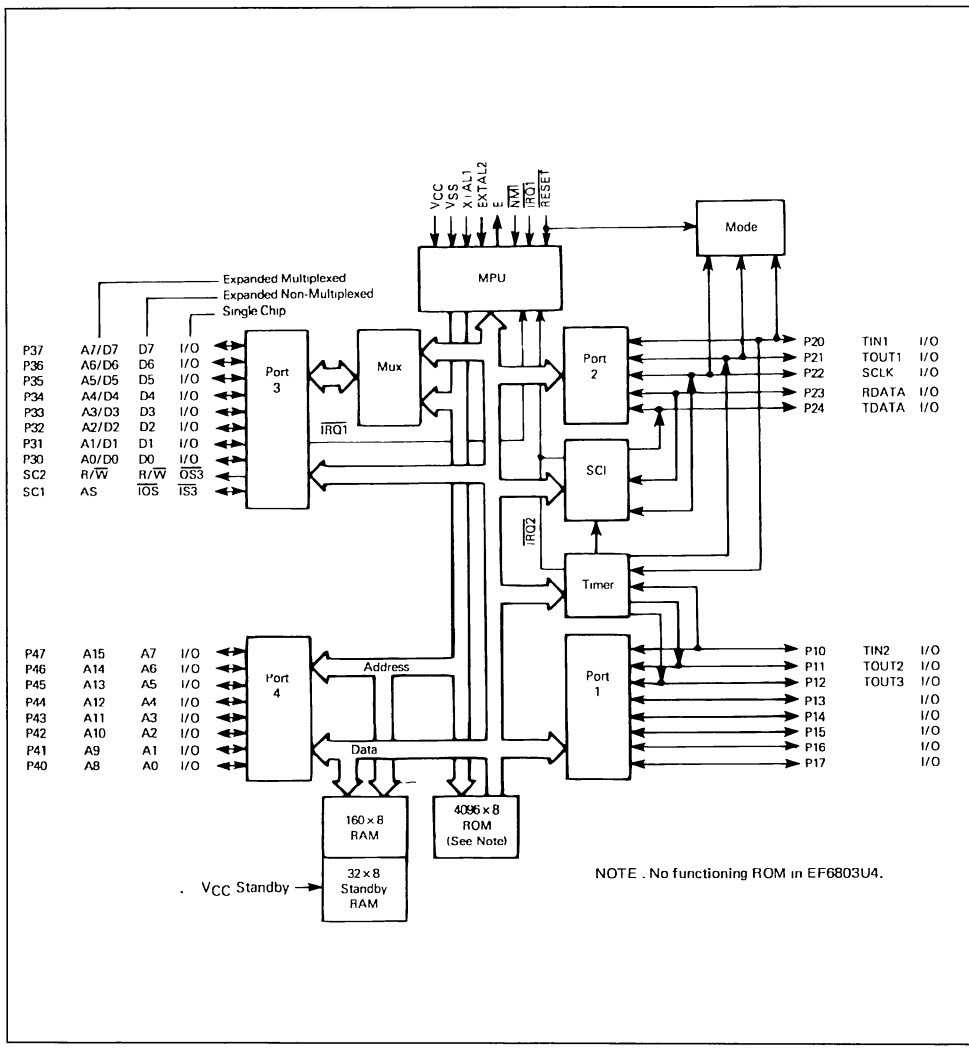




DESCRIPTION

The EF6801U4 is an 8-bit single-chip microcomputer unit (MCU) which enhances the capabilities of the EF6801 and significantly enhances the capabilities of the EF6800 Family of parts. It includes an EF6801 microprocessor unit (MPU) with direct object-code compatibility and upward object-code compatibility with the EF6800. Execution times of key instructions have been improved over the EF6800 and the new instructions found on the EF6801 are included. The MCU can function as a monolithic microcomputer or can be expanded to a 64K byte address space. It is TTL compatible and requires one + 5-volt power supply. On-chip resources include 4096 bytes of ROM, 192 bytes of RAM, a serial communications interface (SCI), parallel I/O, and a 16-bit six-function programmable timer. The EF6803U4 can be considered as an EF6801U4 operating in modes 2 or 3 ; i.e., those that do not use internal ROM.

EF6801U4 MICROCOMPUTER FAMILY BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage	– 0.3 to + 7.0	V
T_A	Operating Temperature Range EF6801/03U4, EF6801/03U4-1, EF68A01/03U4 EF6801/03U4, EF6801/03U4-1 : V Suffix	T_H to T_L 0 to 70 – 40 to 85	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields , however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. Input protection is enhanced by connecting unused inputs to either V_{DD} or V_{SS}.

THERMAL DATA

θ_{JA}	Thermal Resistance	Plastic PLCC	50 100	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} << P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$$P_D = K + (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives :

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

CONTROL TIMING (V_{CC} = 5.0V ± 5%, V_{SS} = 0, T_A = 0 to 70°C)

Symbol	Parameter	EF6801U4 EF6803U4		EF6801U4-1 EF6803U4-1		EF68A01U4 EF68A03U4		EF68B01U4 EF68B03U4		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
f_o	Frequency of Operation	0.5	1.0	0.5	1 25	0.5	1.5	0.5	2.0	MHz
f_{XTAL}	Crystal Frequency	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
$4f_o$	External Oscillator Frequency	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
t_{rc}	Crystal Oscillator Startup Time		100		100		100		100	ms
t_{PCS}	Processor Control Setup Time	200		170		140		110		ns

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0Vdc ± 5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted)

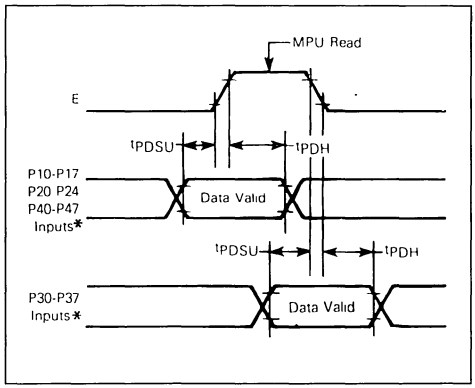
Symbol	Parameter		EF6801/03U4 (1) 0 to + 70°C		EF6801U4/6803U4 – 40 to + 85°C		Unit
			Min.	Max.	Min.	Max.	
V_{IH}	Input High Voltage	$\overline{RESET}$	V_{SS} + 4.0	V_{CC}	V_{SS} + 4.0	V_{CC}	V
		Other Inputs*	V_{SS} + 2.0	V_{CC}	V_{SS} + 2.2	V_{CC}	
V_{IL}	Input Low Voltage	All Inputs*	V_{SS} – 0.3	V_{SS} + 0.8	V_{SS} – 0.3	V_{SS} + 0.8	V
I_{in}	Input Load Current (V_{in} = 0 to 2.4V)	Port 4		0.5		0.8	mA
		SCI		0.8		1.0	
I_{in}	Input Leakage Current (V_{in} = 0 to 5.5V)	$\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$		2.5		5.0	µA
I_{TSI}	Hi-Z (off-state) Input Current (V_{in} = 0.5 to 2.4V)	Port 1, Port 2, Port 3		10		20	µA
V_{OH}	Output High Voltage (I_{Load} = – 65µA, V_{CC} = Min)	Port 4, SC1, SC2	V_{SS} + 2.4		V_{SS} + 2.4		V
	(I_{Load} = – 100µA, V_{CC} = Min)	Other Outputs	V_{SS} + 2.4		V_{SS} + 2.4		
V_{OL}	Output Low Voltage (I_{Load} = 2.0mA, V_{CC} = Min)	All Outputs		V_{SS} + 0.5		V_{SS} + 0.6	V
I_{OH}	Darlington Drive Current (V_O = 1.5V)	Port 1	1.0	4.0	1.0	5.0	mA
P_{INT}	Internal Power Dissipation (measured at T_A = T_L in steady-state operation)***			1200		1500	mW
C_{in}	Input Capacitance (V_{in} = 0, T_A = 25°C, f_o = 1.0MHz)	Port 3, Port 4, SC1		12.5		12.5	pF
		Other Inputs		10.0		10.0	
V_{SBB}	V_{CC} Standby	Powerdown	4.0	5.25	4.0	5.25	V
V_{SB}		Powerup	4.75	5.25	4.75	5.25	
I_{SBB}	Standby Current	Powerdown		3.0		3.5	mA

* Except mode programming levels ; see figure 16.
** Negotiable to – 100µA (for further information contact the factory).
*** For the EF6801U4/EF6803U4 T_L = 0°C and the the EF6801U4/EF6803U4 : V suffix T_L = – 40°C.
(1) Same values for EF6801/03U4-1, EF68A01/03U4 and EF68B01/03U4.

PERIPHERAL PORT TIMING (refer to figures 1.4)

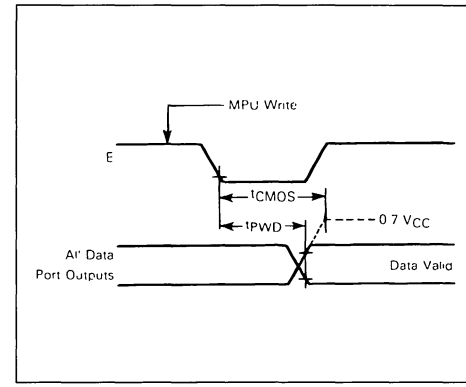
Symbol	Parameter	EF6801/03U4 EF6801/03U4-1		EF68A01/03U4		EF68B01U4 EF68B03U4		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{PDSU}	Peripheral Data Setup Time	200		150		100		ns
t_{PDH}	Peripheral Data Hold Time	200		150		100		ns
t_{OSD1}	Delay Time, Enable Positive Transition to OS3 Negative Transition		350		300		250	ns
t_{OSD2}	Delay Time, Enable Positive Transition to OS3 Positive Transition		350		300		250	ns
t_{PWD}	Delay Time, Enable Negative Transition to Peripheral Data Valid Port 1 Port 2, 3, 4		 350 350		 300 300		 250	ns
t_{CMOS}	Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid		2.0		2.0		2.0	µs
t_{PWIS}	Input Strobe Pulse Width	200		150		100		ns
t_{IH}	Input Data Hold Time	50		40		30		ns
t_{IS}	Input Data Setup Time	20		20		20		ns

Figure 1 : Data Setup and Hold Times (MPU Read).



* Port 3 non-latched operation (latch enable = 0).

Figure 2 : Data Setup and Hold Times (MPU Write).



Notes : 1.10k pullup resistor required for port 2 to reach 0.7 V_{CC}
2.Not applicable to P21.
3.Port 4 cannot be pulled above V_{CC}

Figure 3 : Port 3 Output Strobe Timing (EF6801U4 single-chip mode).

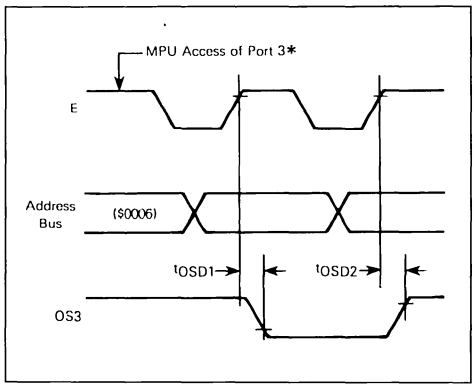
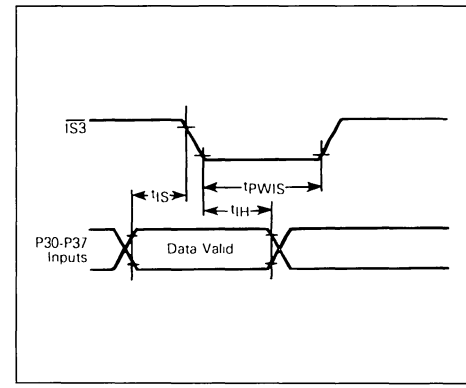


Figure 4 : Port 3 Latch Timing (EF6801U4 single-chip mode).



* Access matches output strobe select (OSS = 0, a read ; OSS = 1, a write)

Note : Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2 0 volts, unless otherwise noted

Figure 5 : CMOS Load.

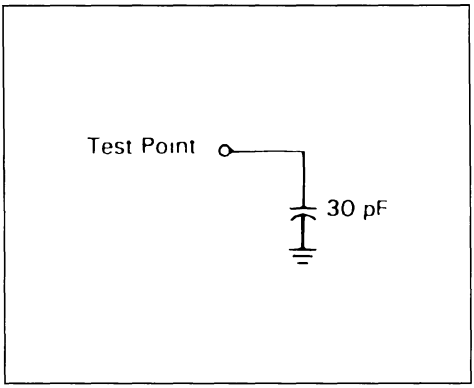
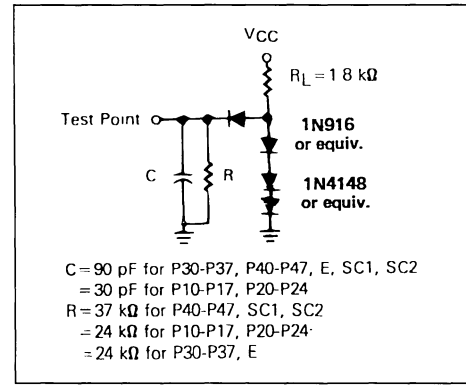


Figure 6 : Timing Test Load Ports 1, 2, 3, and 4.

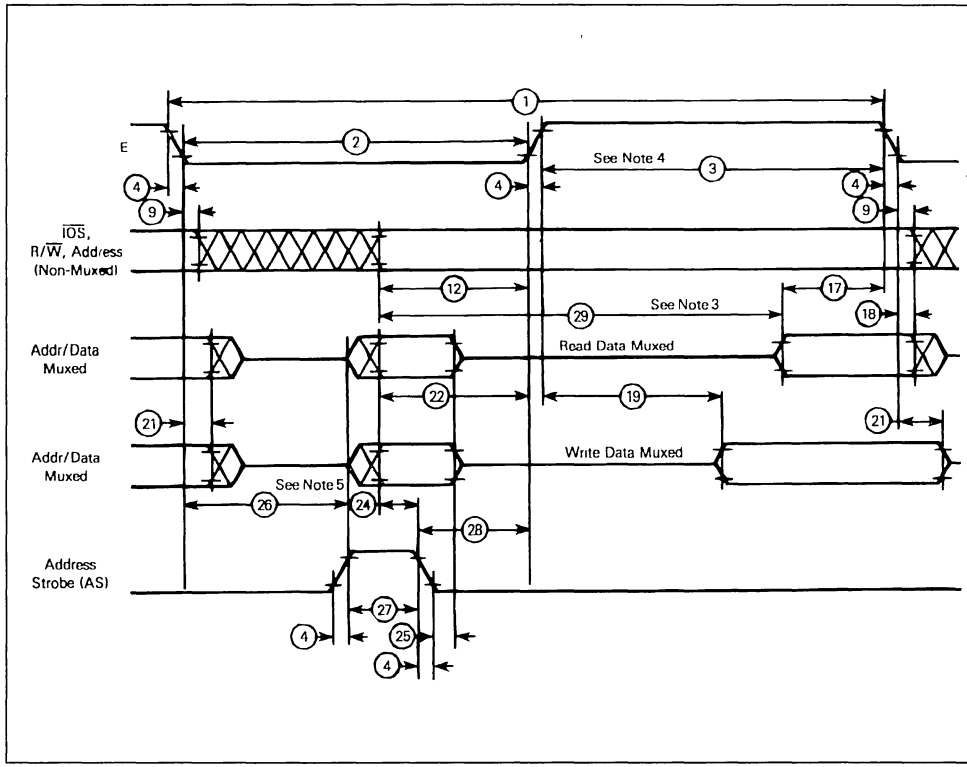


BUS TIMING (see notes 1 and 2, and figure 7)

Ident. Number	Symbol	Parameter	EF6801U4 EF6803U4		EF6801U4-1 EF6803U4-1		EF68A01U4 EF68A03U4		EF68B01U4 EF68B03U4		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
1	t_{cyc}	Cycle Time	1.0	2.0	0.8	2.0	0.66	2.0	0.5	2.0	µs
2	PW_{EL}	Pulse Width, E Low	430	1000	360	1000	30	1000	210	1000	ns
3	PW_{EH}	Pulse Width, E High	450	1000	360	1000	300	1000	220	1000	ns
4	t_r, t_f	Clock Rise and Fall Time		25		25		25		20	ns
9	t_{AH}	Address Hold Time	20		20		20		10		ns
12	t_{AV}	Non-muxed Address Valid Time to E*	200		150		115		70		ns
17	t_{DSR}	Read Data Setup Time	80		70		60		40		ns
18	t_{DHR}	Read Data Hold Time	10		10		10		10		ns
19	t_{DDW}	Write Data Delay Time		225		200		160		120	ns
21	t_{DHW}	Write Data Hold Time	20		20		20		10		ns
22	t_{AVM}	Muxed Address Valid Time to E Rise*	160		120		100		80		ns
24	t_{ASL}	Muxed Address Valid Time to AS Fall*	40		30		30		20		ns
25	t_{AHL}	Muxed Address Hold Time	20		20		20		10		ns
26	t_{ASD}	Delay Time, E to AS Rise*	200		170		130		110		ns
27	PW_{ASH}	Pulse Width, AS High*	100		80		60		45		ns
28	t_{ASED}	Delay Time, AS to E Rise*	90		70		60		45		ns
29	t_{ACC}	Usable Access Time* (see note 3)	555		435		385		270		ns

* At specified cycle time

Figure 7 : Bus Timing.



Notes : 1. Voltage levels shown are $V_L \leq 0.5V$, $V_H \geq 2.4V$, unless otherwise specified.
2. Measurement points shown are 0.8V and 2.0V, unless otherwise specified.
3. Usable access time is computed by 22 + 3 – 17 + 4.
4. Memory devices should be enabled only during E high to avoid port 3 bus contention
5. Item 26 is different from the EF6801 but it is upward compatible.

INTRODUCTION

The EF6801U4 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a data register and a write-only data direction register. The data direction register is used to define whether corresponding bits in the data register are configured as an input (clear) or output (set).

The term "port" by itself refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port", it is controlled by the port data direction register and the programmer has direct access to the port pins using the port data register. Port pins are labeled as Pij where i identifies one of four ports and j indicates the particular bit.

The microprocessor unit (MPU) is an enhanced EF6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the EF6800 and the EF6801. The programming model is depicted in figure 8

where accumulator D is a concatenation of accumulators A and B. A list of new operations added to the EF6800 instruction set are shown in table 1.

The EF6803U4 can be considered an EF6801U4 that operates in modes 2 and 3 only.

Figure 8 : Programming Model.

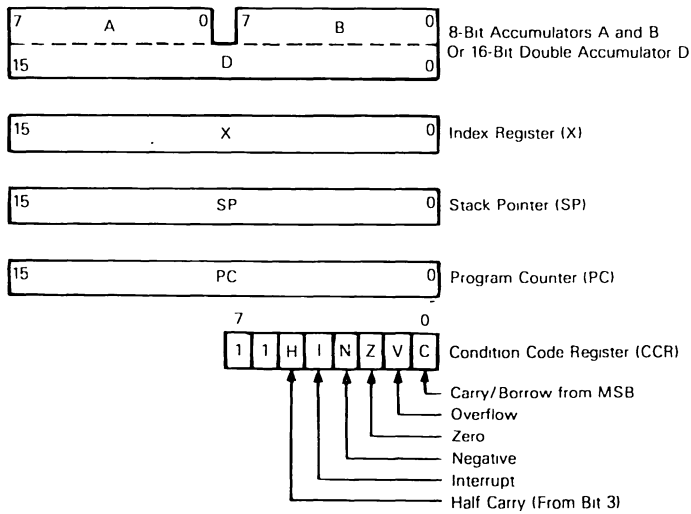


Table 1 : New Instructions.

Instruction	Description
ABX	Unsigned addition of accumulator B to index register.
ADDD	Adds (without carry) the double accumulator to memory and leaves the sum in the double accumulator.
ASLD or LSLD	Shifts the double accumulator left (towards MSB) one bit, the LSB is cleared, and the MSB is shifted into the C bit.
BHS	Branch if higher or same, unsigned conditional branch (same as BCC).
BLO	Branch if lower, unsigned conditional branch (same as BCS).
BRN	Branch never.
JSR	Additional addressing mode direct.
LDD	Loads double accumulator from memory.
LSL	Shifts memory or accumulator left (towards MSB) one bit, the LSB is cleared, and the MSB is shifted into the C bit (same as ASL).
LSRD	Shifts the double accumulator right (towards LSB) one bit, the MSB is cleared, and the LSB is shifted into the C bit.
MUL	Unsigned multiply, multiplies the two accumulators and leaves the product in the double accumulator.
PSHX	Pushes the index register to stack.
PULX	Pulls the index register from stack.
STD	Stores the double accumulator to memory.
SUBD	Substracts memory from the double acccumulator and leaves the difference in the double accumulator.
CPX	Internal processing modified to permit its use with any conditional branch instruction.

OPERATING MODES

The EF6801U4 provides seven different operating modes (modes 0 through 3 and 5 through 7) and the EF6803U4 provides two operating modes (modes 2 and 3). The operating modes are hardware selectable and determine the device memory map, the configuration of port 3, port 4, SC1, SC2, and the physical location of the interrupt vectors.

FUNDAMENTAL MODES

The seven operating modes (0-3, 5-7) can be grouped into three fundamental modes which refer to the type of bus it supports : single chip, expanded non-multiplexed, and expanded multiplexed. Single chip is mode 7, expanded non-multiplexed is mode 5, and the remaining 5 are expanded multiplexed modes. Table 2 summarizes the characteristics of the operating modes.

EF6801U4 SINGLE-CHIP MODE (7) - In the single-chip mode, the four MCU ports are configured as parallel input/output data ports, as shown in figure 9. The MCU functions as a monolithic microcomputer in this mode without external address or data buses. A maximum of 29 I/O lines and two port 3 control lines are provided. Peripherals or another MCU can be interfaced to port 3 in a loosely coupled dual-processor configuration, as shown in figure 10.

Table 2 : Summary of EF6801U4/EF6803U4 Operating Modes.

Single-chip (mode 7)
- 192 Bytes of RAM, 4096 Bytes of ROM
- Port 3 is a parallel I/O port with two control lines.
- Port 4 is a parallel I/O port.

Expanded Non-multiplexed (mode 5)
- 192 Bytes of RAM, 4096 Bytes of ROM
- 256 Bytes of External Memory Space
- Port 3 is an 8-bit data bus.
- Port 4 is an input port/address bus.

Expanded Multiplexed (modes 0, 1, 2, 3, 6*)
- Four Memory Space Options (total 64K address space)
  - (1) Internal RAM and ROM with Partial Address Bus (mode 1)
  - (2) Internal RAM, no ROM (mode 2)
  - (3) Extended Addressing of Internal I/O and RAM
  - (4) Internal RAM and ROM with Partial Address Bus (mode 6)
- Port 3 is multiplexed address/data bus.
- Port 4 is address bus (inputs/address in mode 6).
- Test Mode (mode 0) :
  - May be used to test internal RAM and ROM.
  - May be used to test ports 3 and 4 as I/O ports by writing into mode 7.
- Only modes 5, 6, and 7 can be irreversibly entered from mode 0.

Ressources Common to All Modes
- Reserved Register Area
- Port 1 Input/output Operation
- Port 2 Input/output Operation
- Timer Operation
- Serial Communications Interface Operation

* The EF6803U4 operates only in modes 2 and 3

Figure 9 : Single-chip Mode.

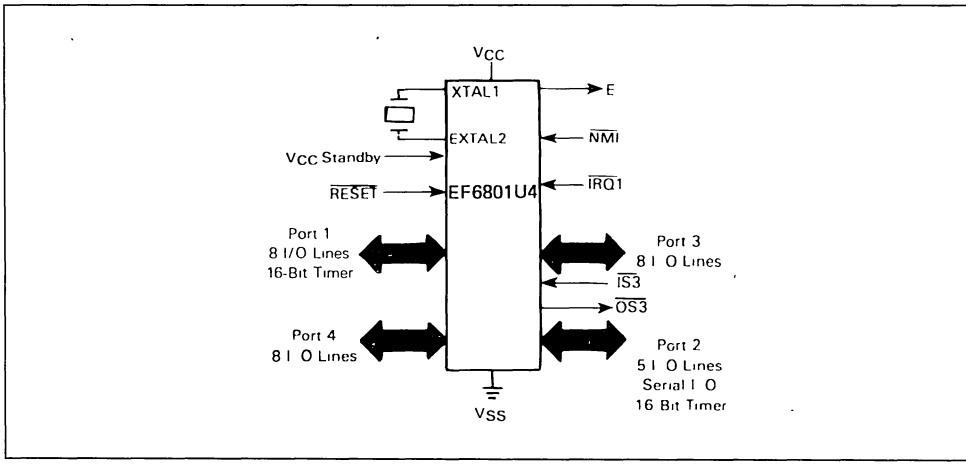
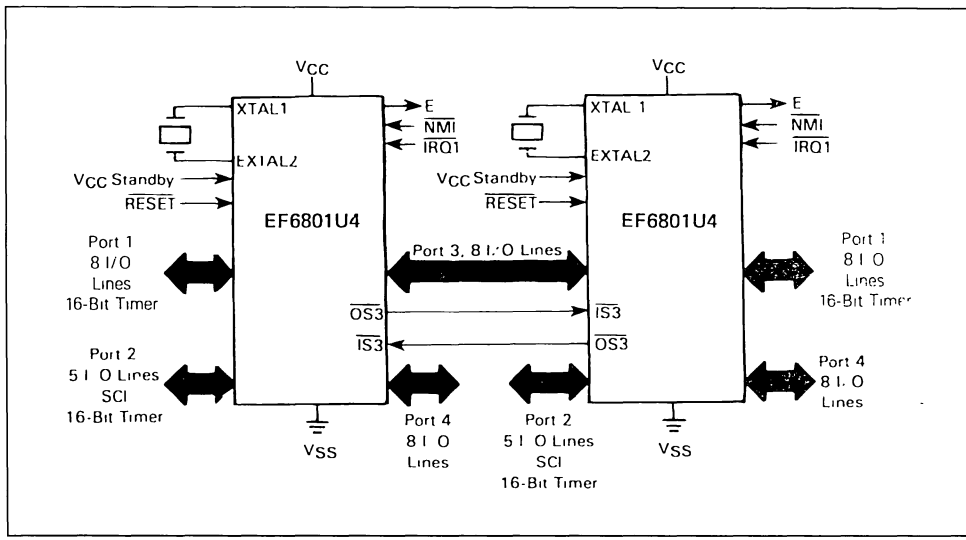


Figure 10 : Single-chip Dual Processor Configuration.



EF6801U4 EXPANDED NON-MULTIPLEXED MODE (5) - A modest amount of external memory space is provided in the expanded non-multiplexed mode while significant on-chip resources are retained. Port 3 functions as an 8-bit bidirectional data bus and port 4 is configured initially as an input data port. Any combination of the eight least significant address lines may be obtained by writing to the port 4 data direction register. Stated alternatively, any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Internal pullup resistors pull the port 4 lines high until the port is configured.

Figure 11 illustrates a typical system configuration in the expanded non-multiplexed mode. The MCU interfaces directly with EF6800 family parts and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and can be used as a memory-page select or chip-select line.

EXPANDED MULTIPLEXED MODES (0, 1, 2, 3, 6) - A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes, port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of address strobe (AS) and data valid while E is high. In modes 0, 2, and 3, port 4 provides address lines A8 to A15. In modes 1 and 6, however, port 4 initially is configured at reset as an input data port. The port 4 data direction register can then be changed to provide any combination of address lines A8 to A15. Stated alternatively, any subset of A8 to A15 can be provided while retaining the remaining port 4 lines as input data lines. Internal pullup resistors pull the port 4 lines high until software configures the port. In mode 1, the internal pullup resistors will hold the upper address lines high producing a value of $FFXX for a reset vector. A simple method of getting the desired address lines configured as outputs is to have an external EPROM not fully decoded so it appears at two address locations (i.e., $FXXX and $BXXX). Then, when the reset vector appears as $FFFE, the EPROM will be accessed and can point to an address in the top $100 bytes of the internal or external ROM/EPROM that will configure port 4 as desired.

In mode 0, the reset and interrupt vectors are located at $BFF0-$BFFF. In addition, the internal and external data buses are connected, so there must be no memory map overlap in order to avoid potential bus conflicts. By writing the PC0-PC2 bits in the port 2 data register, modes 5, 6, and 7 can be irreversibly entered from mode 0. Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with the automated test equipment.

Only the EF6801U4 can operate in each of the expanded multiplexed modes. The EF6803U4 operates only in modes 2 and 3.

Figure 12 depicts a typical configuration for the expanded multiplexed modes. Address strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in figure 13. This allows port 3 to function as a data bus when E is high.

Figure 11 : Expanded Non-multiplexed Configuration.

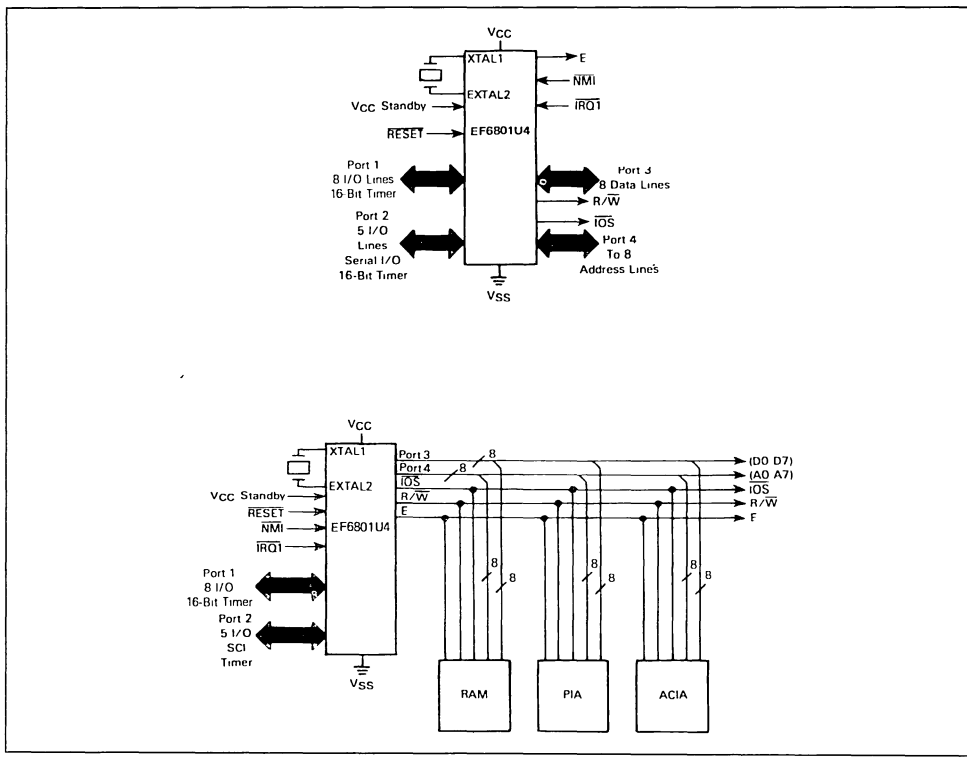
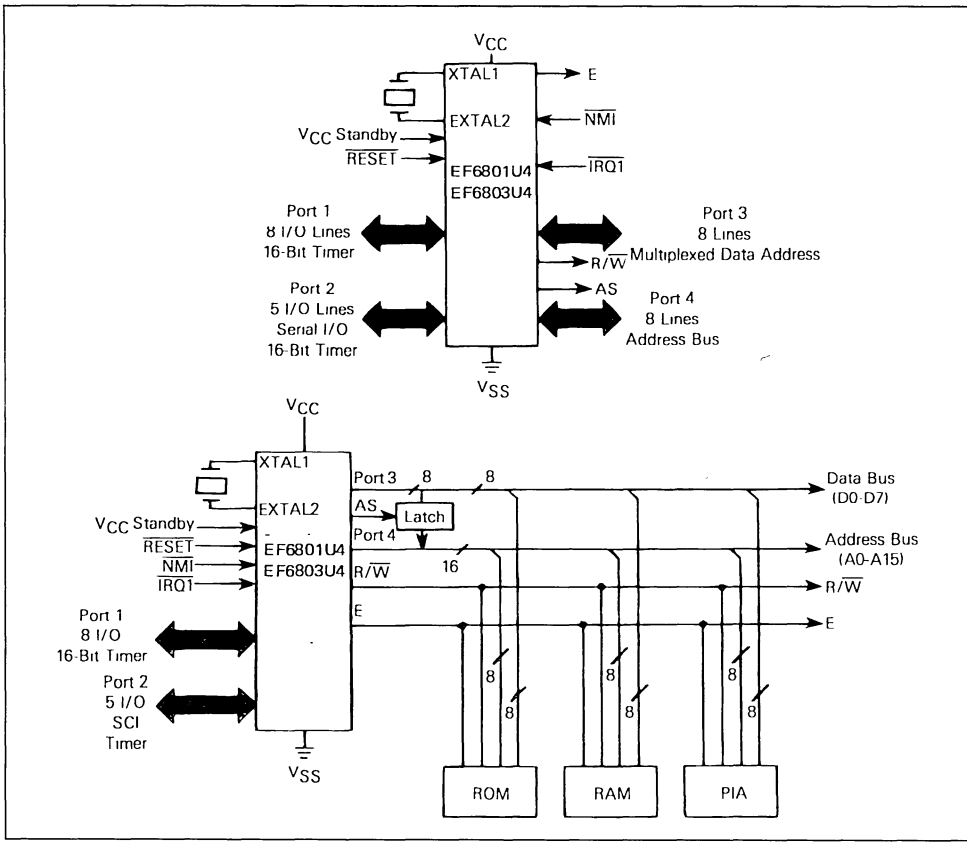
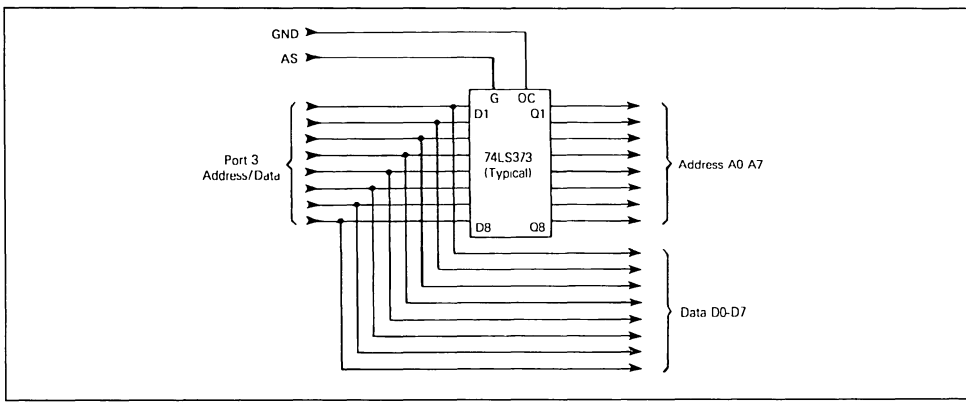


Figure 12 : Expanded Multiplexed Configuration.



Note : To avoid data bus (port 3) contention in the expanded multiplexed modes, memory devices should be enabled only during E high time.

Figure 13 : Typical Latch Arrangement.



PROGRAMMING THE MODE

The operating mode is determined at $\overline{\text{RESET}}$ by the levels asserted on P22, P21, and P20. These levels are latched into PC2, PC1, and PC0 of the program control register on the positive edge of $\overline{\text{RESET}}$. The operating mode may be read from the port 2 data register, as shown below, and programming levels and timing must be met as shown in figure 14. A brief outline of the operating modes is shown in table 3.

PORT 2 DATA REGISTER

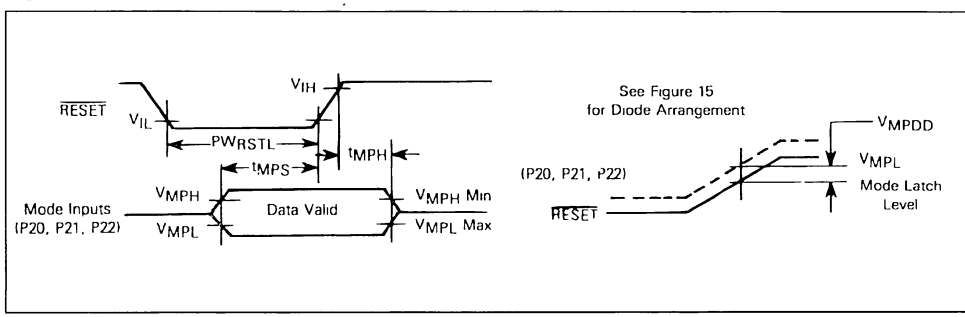
7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in figure 15 may be used ; otherwise, three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

The EF6801U4/EF6803U4 can provide up to 64K byte address space depending on the operating mode. A memory map for each operating mode is shown in figure 16. The first 32 locations of each map are reserved for the internal register area, as shown in table 4, with exceptions as indicated.

Figure 14 : Mode Programming Timing.



MODE PROGRAMMING (refer to figure 14)

Symbol	Parameter	Min.	Max.	Unit
V_{MPL}	Mode Programming Input Voltage Low		1.8	V
V_{MPH}	Mode Programming Input Voltage High	4.0		V
V_{MPDD}	Mode Programming Diode Differential (if diodes are used)	0.6		V
PW_{RSTL}	$\overline{\text{RESET}}$ Low Pulse Width	3.0		E Cycles
t_{MPS}	Mode Programming Setup Time	2.0		E Cycles
t_{MPH}	Mode Programming Hold Time $\overline{\text{RESET}}$ Rise Time ≥ 1µs $\overline{\text{RESET}}$ Rise Time < 1µs	 0 100		ns

Table 3 : Mode Selection Summary.

Mode*	P22 PC2	P21 PC1	P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX (2, 3)	Multiplexed/partial Decode
5	H	L	H	I	I	I	NMUX (2, 3)	Non-multiplexed/partial Decode
4	H	L	L					Undefined (4)
3	L	H	H	E	I	E	MUX (1, 5)	Multiplexed/RAM
2	L	H	L	E	I	E	MUX (1)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX (1, 3)	Multiplexed/RAM and ROM
0	L	L	L	I	I	E	MUX (1)	Multiplexed Test

Legend
I - Internal
E - External
MUX - Multiplexed
NMUX - Non-Multiplexed
L - Logic "0"
H - Logic "1"

Notes : 1. Addresses associated with ports 3 and 4 are considered external in modes 0, 1, 2, and 3.
2. Addresses associated with port 3 are considered external in modes 5 and 6
3. Port 4 default is user data input ; address output is optional by writing to port 4 data direction register.
4. Mode 4 is a non-user mode and should not be used as an operating mode.
5. Mode 3 has the internal RAM and internal registers relocated at $D000-$D0FF.

* The EF6803U4 operates only in modes 2 and 3.

Figure 15 : Typical Mode Programming Circuit.

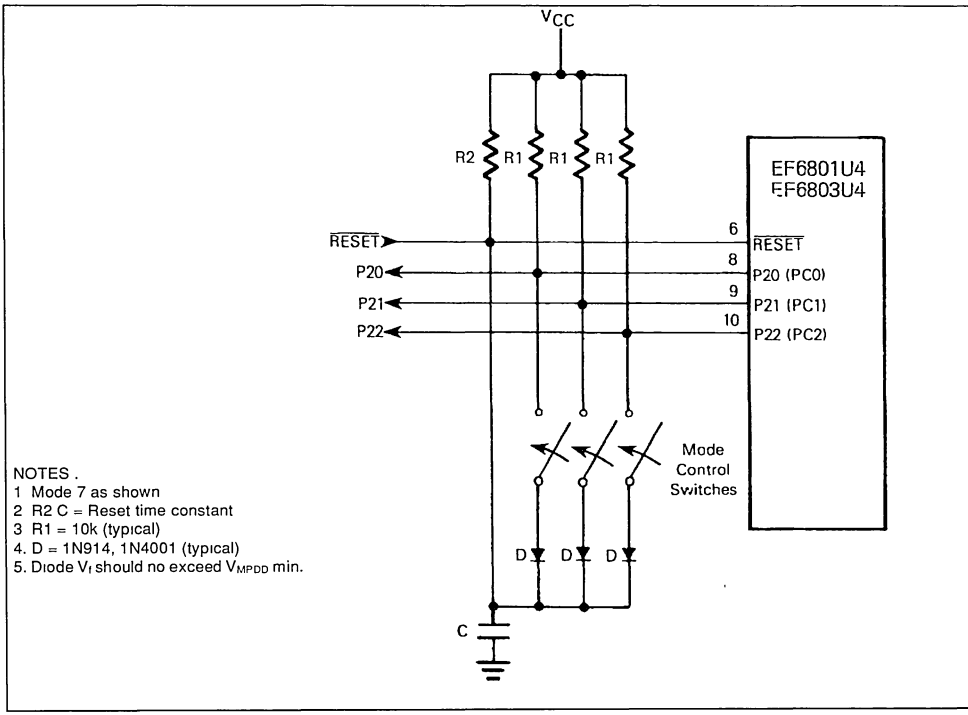
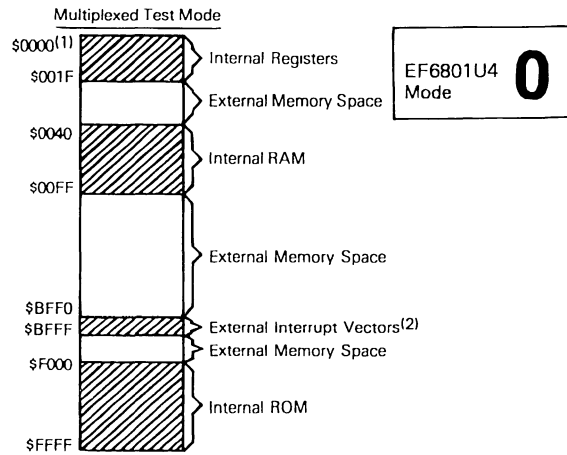
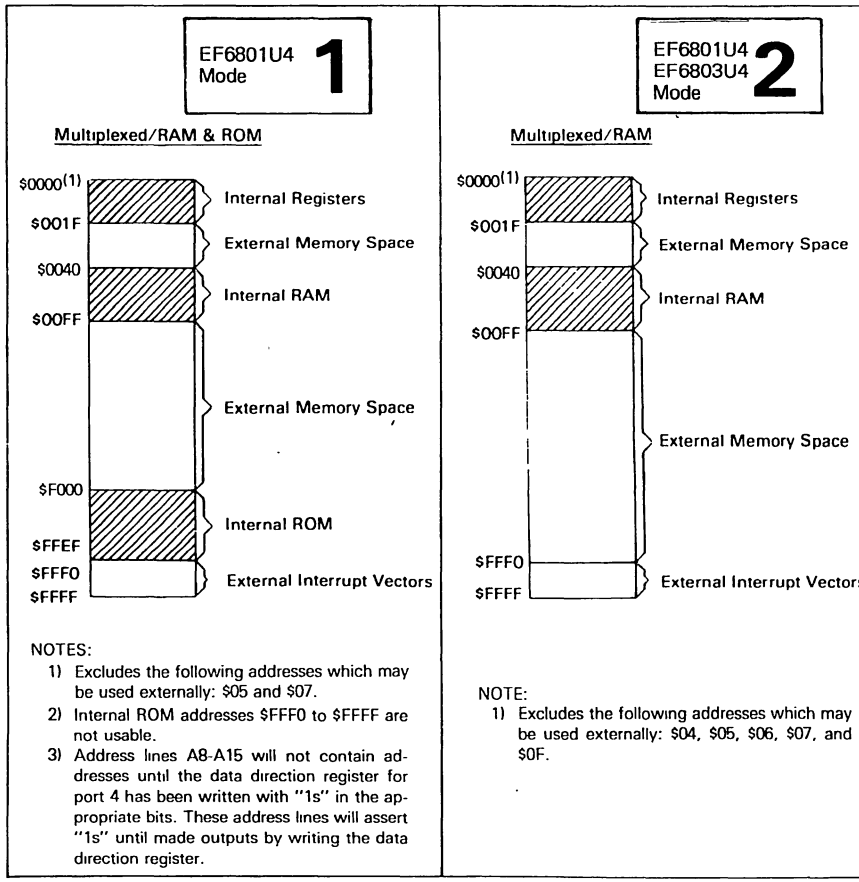


Figure 16 : EF6801U4/EF6803U4 Memory Maps (sheet 1 of 4).



Notes : 1. Excludes the following addresses which may be used externally : $04, $05, $06, $07, and $0F.
2. The interrupt vectors are at $BFF0-$BFFF.
3. There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4. This mode is the only mode which may be used to examine the entire ROM using an external $\overline{\text{RESET}}$ vector.
5. Modes 5-7 can be irreversibly entered from mode 0 by writing to the PC0-PC2 bits of the port 2 data register.

Figure 16 : EF6801U4/EF6803U4 Memory Maps (sheet 2 of 4).



NOTES:

1) Excludes the following addresses which may be used externally: $05 and $07.
2) Internal ROM addresses $FFF0 to $FFFF are not usable.
3) Address lines A8-A15 will not contain addresses until the data direction register for port 4 has been written with "1s" in the appropriate bits. These address lines will assert "1s" until made outputs by writing the data direction register.

NOTE:

1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

Figure 16 : EF6801U4/EF6803U4 Memory Maps (sheet 3 of 4).

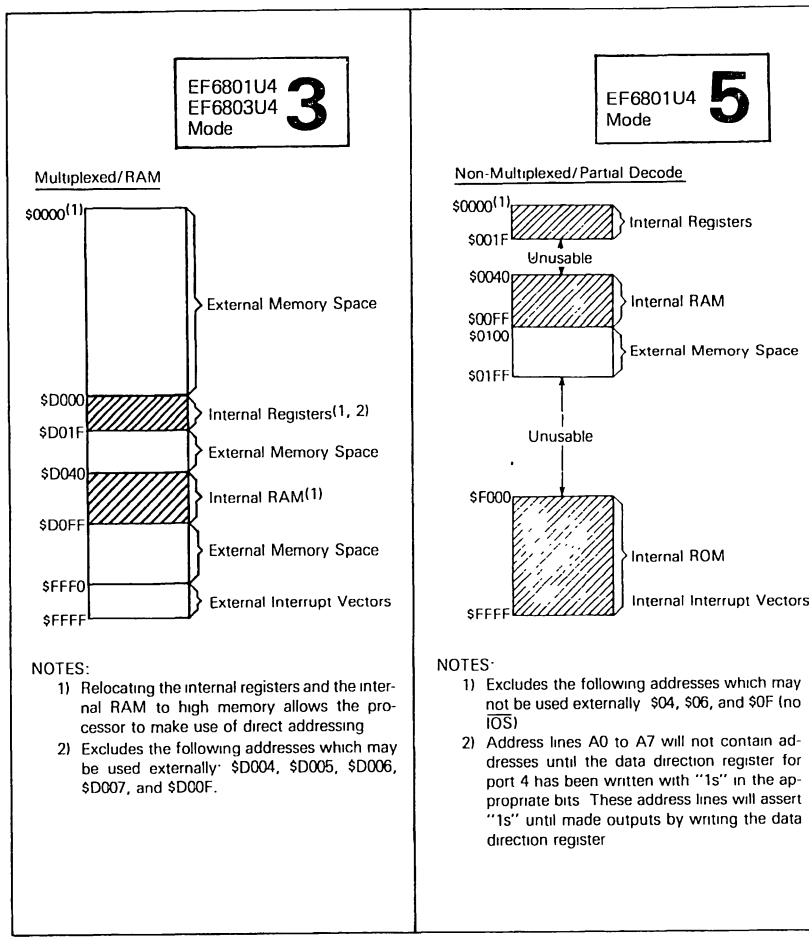
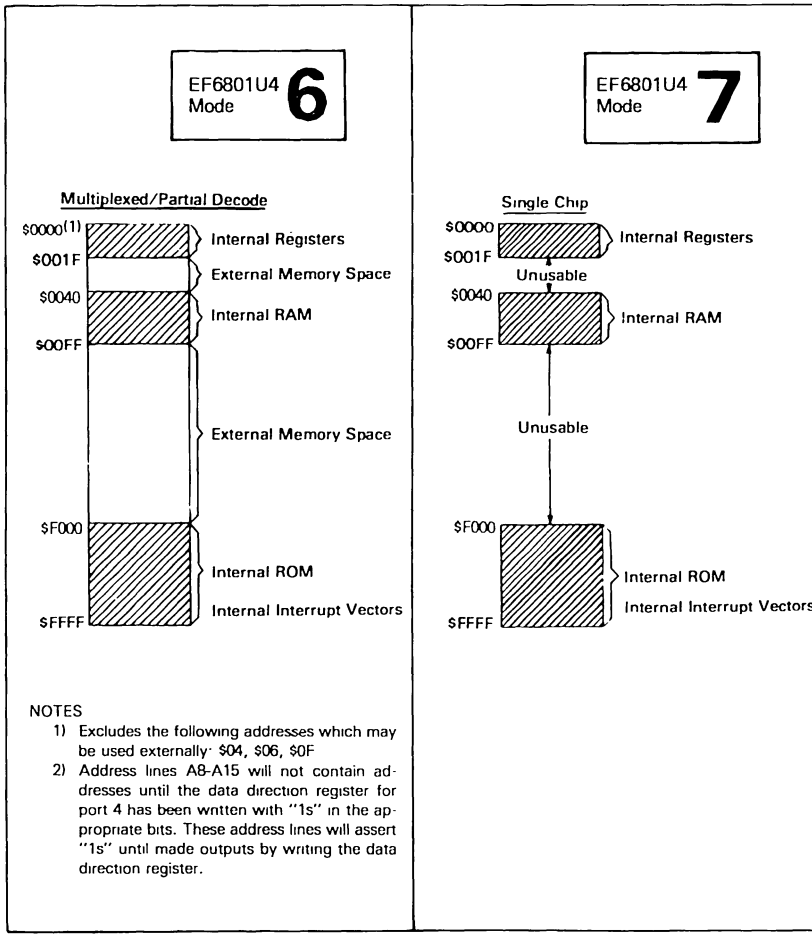


Figure 16 : EF6801U4/EF6803U4 Memory Maps (sheet 4 of 4).



NOTES

1) Excludes the following addresses which may be used externally· $04, $06, $0F
2) Address lines A8-A15 will not contain addresses until the data direction register for port 4 has been written with "1s" in the appropriate bits. These address lines will assert "1s" until made outputs by writing the data direction register.

Table 4 : Internal Register Area.

Register	Address	
	Other Modes	Mode 3
Port 1 Data Direction Register***	0000	D000
Port 2 Data Direction Register***	0001	D001
Port 1 Data Register	0002	D002
Port 2 Data Register	0003	D003
Port 3 Data Direction Register***	0004*	D004*
Port 4 Data Direction Register***	0005**	D005**
Port 3 Data Register	0006*	D006*
Port 4 Data Register	0007**	D007**
Timer Control and Status Register	0008	D008
Counter (high byte)	0009	D009
Counter (low byte)	000A	D00A
Output Compare Register (high byte)	000B	D00B
Output Compare Register (low byte)	000C	D00C
Input Capture Register (high byte)	000D	D00D
Input Capture Register (low byte)	000E	D00E
Port 3 Control and Status Register	000F*	D00F*
Rate and Mode Control Register	0010	D010
Transmit/receive Control and Status Register	0011	D011
Receive Data Register	0012	D012
Transmit Data Register	0013	D013
RAM Control Register	0014	D014
Counter Alternate Address (high byte)	0015	D015
Counter Alternate Address (low byte)	0016	D016
Timer Control Register 1	0017	D017
Timer Control Register 2	0018	D018
Timer Status Register	0019	D019
Output Compare Register 2 (high byte)	001A	D01A
Output Compare Register 2 (low byte)	001B	D01B
Output Compare Register 3 (high byte)	001C	D01C
Output Compare Register 3 (low byte)	001D	D01D
Input Capture Register 2 (high byte)	001E	D01E
Input Capture Register 2 (low byte)	001F	D01F

* External addresses in modes 0, 1, 2, 3, 5 and 6 cannot be accessed in mode 5 (no IOS).
** External Addresses in Modes 0, 2, and 3.
*** 1 = Output, 0 = Input.

EF6801U4/EF6803U4 INTERRUPTS

The EF6801 Family supports two types of interrupt requests : maskable and non-maskable. A non-maskable interrupt ($\overline{NMI}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the condition code register I bit and by individual enable bits. The I bit controls all maskable interrupts. Of the maskable interrupts, there are two types : $\overline{IRQ1}$ and $\overline{IRQ2}$. The programmable timer and serial communications interface use an internal $\overline{IRQ2}$ interrupt line, as shown in the block diagram. External devices and IS3 use $\overline{IRQ1}$. An $\overline{IRQ1}$ interrupt is serviced before $\overline{IRQ2}$ if both are pending.

NOTE

After reset, an $\overline{NMI}$ will not be serviced until the first program load of the stack pointer. Any $\overline{NMI}$ generated before this load will be remembered by the processor and serviced subsequent to the stack pointer load.

All $\overline{IRQ2}$ interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order and each is vectored to a separate location. All interrupt vector locations are shown in table 5. In mode 0, reset and interrupt vectors are defined as $BFF0-$BFFF.

The interrupt flowchart is depicted in figure 17 and is common to every interrupt excluding reset. During interrupt servicing, the program counter, index register, A accumulator, B accumulator, and condition code register are pushed to the stack. The I bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the program counter and instruction execution is resumed. Interrupt and $\overline{RESET}$ timing are illustrated in figures 18 and 19.

Table 5 : MCU Interrupt Vector Locations.

Mode 0		Modes 1-3, 5-7		Interrupt***
MSB	LSB	MSB	LSB	
BFFE	BFFF	FFFE	FFFF	$\overline{\text{RESET}}$
BFFC	BFFD	FFFC	FFFD	Non-maskable Interrupt**
BFFA	BFFB	FFFA	FFFB	Software Interrupt
BFF8	BFF9	FFF8	FFF9	Maskable Interrupt Request 1
BFF6	BFF7	FFF6	FFF7	Input Capture Flag*
BFF4	BFF5	FFF4	FFF5	Output Compare Flag*
BFF2	BFF3	FFF2	FFF3	Timer Overflow Flag*
BFF0	BFF1	FFF0	FFF1	Serial Communications Interface*

* $\overline{\text{IRQ2}}$ interrupt
** $\overline{\text{NMI}}$ must be armed (by accessing stack pointer) before an $\overline{\text{NMI}}$ is executed.
*** Mode 4 interrupt vectors are undefined.

Figure 17 : Interrupt Flowchart.

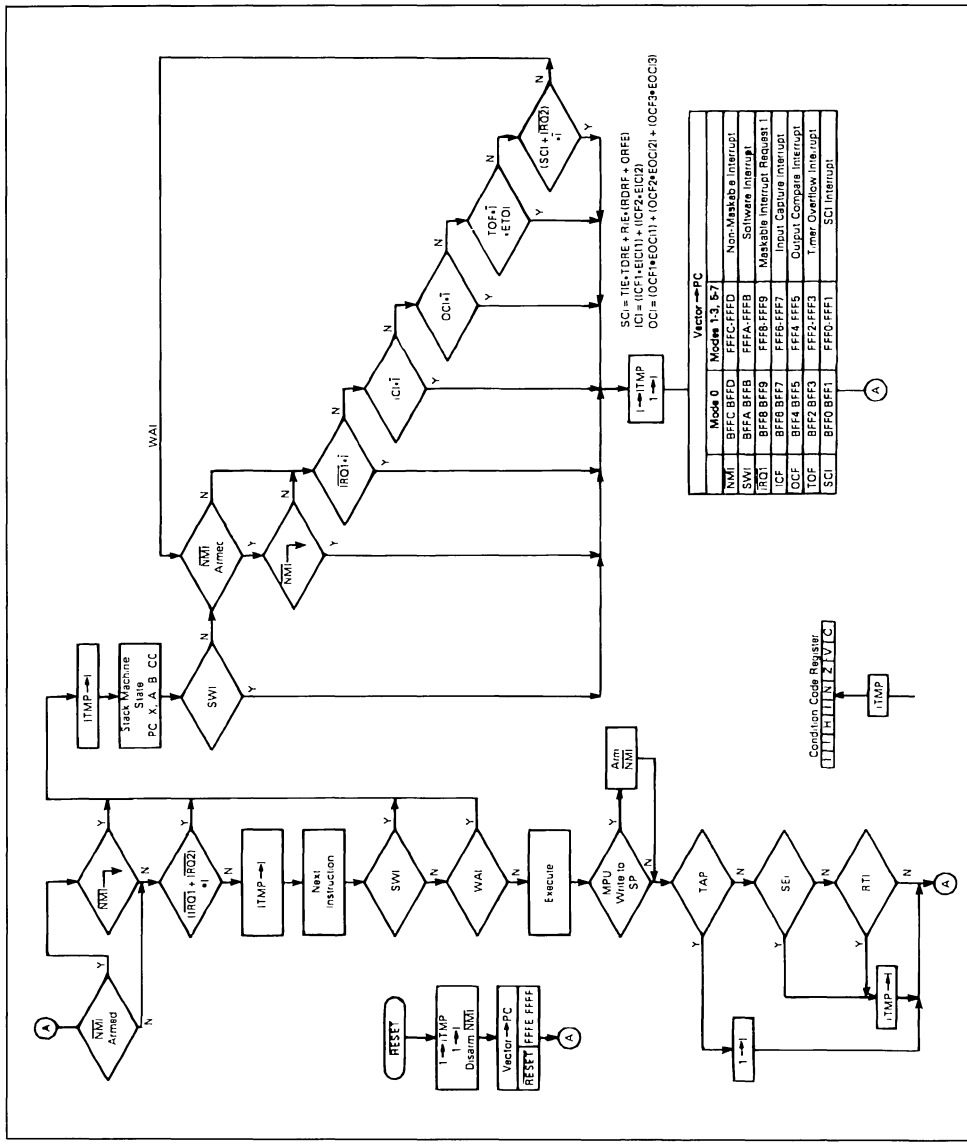


Figure 18 : Interrupt Sequence.

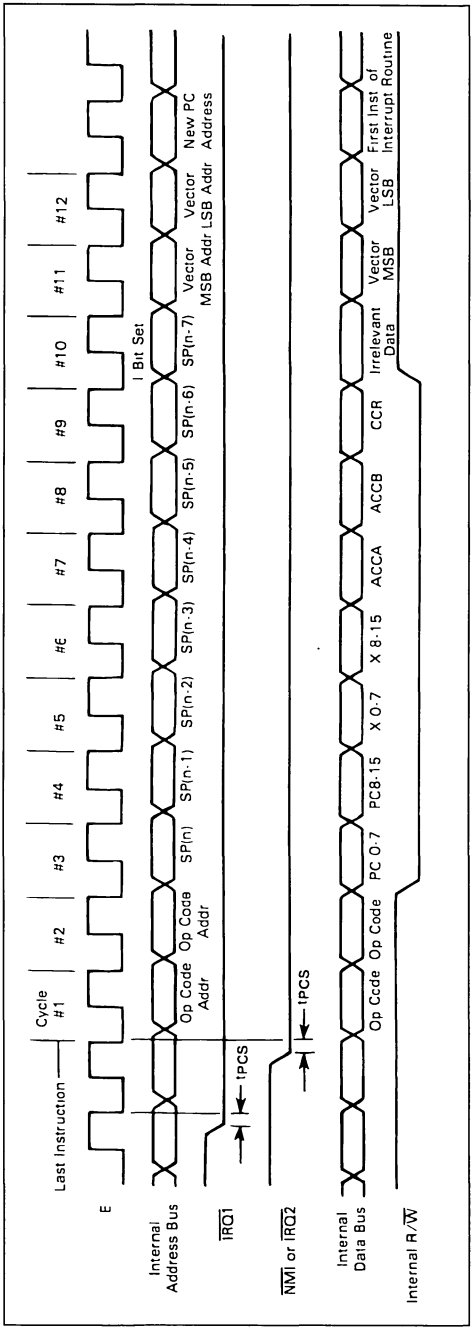
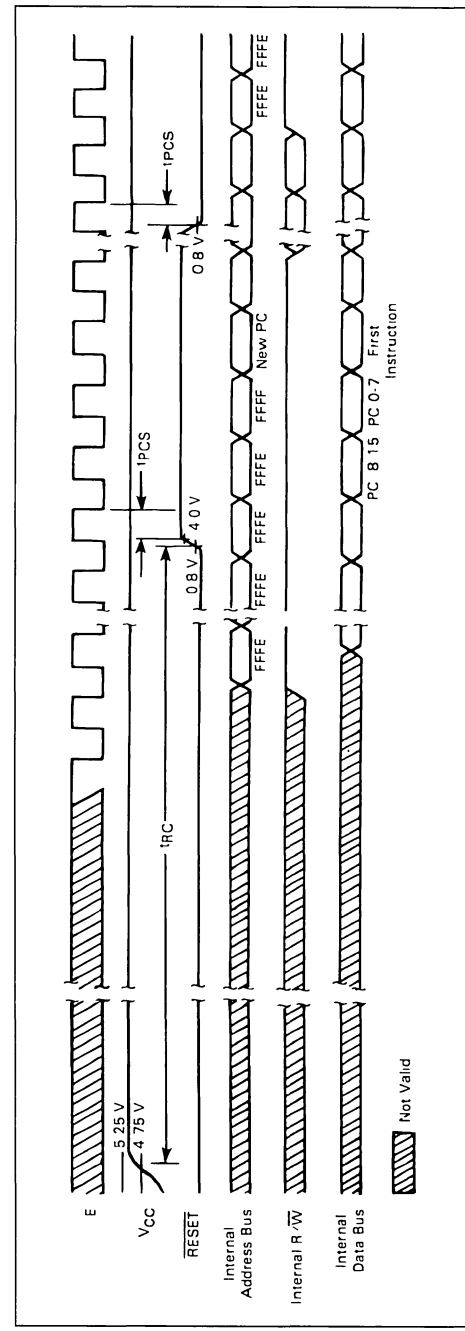


Figure 19 : RESET Timing.



FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power to a large portion of the MCU. The power supply should provide + 5 volts (± 5%) to V_{CC} and V_{SS} should be tied to ground. Total power dissipation (including V_{CC} standby) will not exceed P_D milliwatts.

V_{CC} STANDBY

V_{CC} standby provides power to the standby portion ($40 through $5F in all modes except mode 3 which is $D040 through $D05F) of the RAM and the STBY PWR and RAME bits of the RAM control register. Voltage requirements depend on whether the device is in a power-up or power-down state. In the power-up state, the power supply should provide + 5 volts (± 5%) and must reach V_{SB} volts before RESET reaches 4.0 volts. During power down, V_{CC} standby must remain above V_{SBB} (minimum) to sustain the standby RAM and STBY PWR bit. While in power-down operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during power-down operation.

XTAL1 AND EXTAL2

These two input pins interface either a crystal or TTL-compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58MHz or 4.4336MHz color burst TV crystals. A 20pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL-compatible clock at 4 f_o with a duty cycle of 50% (± 5%) with XTAL1 connected ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for f_{XTAL}. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in figure 20.

$\overline{\text{RESET}}$

This input is used to reset the internal state of the device and provide an orderly startup procedure. During power up, $\overline{\text{RESET}}$ must be held below 0.8 volt : (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} standby reaches 4.75 volts. $\overline{\text{RESET}}$ must be held low at least three E cycles if asserted during power-up operation.

E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the device input clock frequency. It will drive one Schottky TTL load and 90pF, and all data given in cycles is referenced to this clock unless otherwise noted.

$\overline{\text{NMI}}$ (NON-MASKABLE INTERRUPT)

An $\overline{\text{NMI}}$ negative edge requests an MCU interrupt sequence, but the current instruction will be completed before it responds to the request. The MCU will then begin an interrupt sequence. Finally, a vector is fetched from $FFFC and $FFFD ($BFFC and $BFFD in mode 0), transferred to the program counter, and instruction execution is resumed. NMI typically requires a 3.3kΩ (nominal) resistor to V_{CC}. There is no internal NMI pullup resistor. NMI must be held low for at least one E cycle to be recognized under all conditions.

NOTE

After reset, an $\overline{\text{NMI}}$ will not be serviced until the first program load of the stack pointer. Any NMI generated before this load will remain pending by the processor.

$\overline{\text{IRQ1}}$ (MASKABLE INTERRUPT REQUEST 1)

$\overline{\text{IRQ1}}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I bit) in the condition code register is clear, the MCU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9 ($BFF8 and $BFF9 in mode 0), transferred to the program counter, and instruction execution is resumed.

$\overline{\text{IRQ1}}$ typically requires an external 3.3kΩ (nominal) resistor to V_{CC} for wire-OR applications. IRQ1 has no internal pullup resistor.

SC1 AND SC2 (STROBE CONTROL 1 AND 2)

The function of SC1 and SC2 depends on the operating mode. SC1 is configured as an output in all modes except single-chip mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90pF.

SC1 AND SC2 IN SINGLE-CHIP MODE - In single-chip mode, SC1 and SC2 are configured as an input and output, respectively, and both function as port 3 control lines. SC1 functions as IS3 and can

be used to indicate that port 3 input data is ready or output data has been accepted. Three options associated with IS3 are controlled by the port 3 control and status register and are discussed in the port 3 description ; refer to **P30-P37** (PORT 3). If unused, IS3 can remain unconnected.

SC2 is configured as OS3 and can be used to strobe output data or acknowledge input data. It is controlled by output strobe select (OSS) in the port 3 control and status register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the port 3 data register. OS3 timing is shown in figure 3.

SC1 AND SC2 IN EXPANDED NON-MULTIPLEXED MODE - In the expanded non-multiplexed mode, both SC1 and SC2 are configured as outputs. SC1 functions as input/output select (IOS) and is asserted only when $0100 through $01FF is sensed on the internal address bus.

SC2 is configured as read/write and is used to control the direction of data bus transfers. An MPU read is enabled when read/write and E are high.

SC1 AND SC2 IN EXPANDED MULTIPLEXED MODE - In the expanded multiplexed modes, both SC1 and SC2 are configured as outputs. SC1 functions as address strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by address strobe captures the lower address on the negative edge, as shown in figure 13.

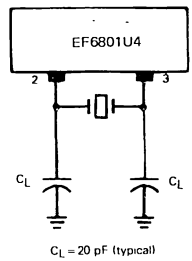
Figure 20 : EF6801U4/EF6803U4 Family Oscillator Characteristics.

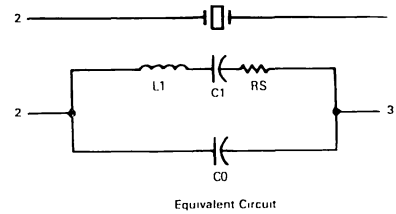
(a) Nominal Recommended Crystal Parameters

Nominal Crystal Parameters*

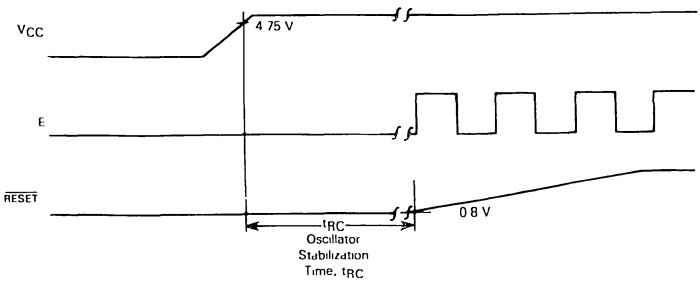
	3.58MHz	4.00MHz	5.0MHz
RS	60Ω	50Ω	30-50Ω
C0	3.5pF	6.5pF	4-6pF
C1	0.015pF	0.025pF	0.01-0.02pF
Q	> 40K	> 30K	> 20K

* **Note :** These are representative AT-cut crystal parameters only. Crystals of other types of cut may also be used





(b) Oscillator Stabilization Time (t_{RC})



SC2 is configured as read/write and is used to control the direction of data bus transfers. An MPU read is enabled when read/write and E are high.

P10-P17 (PORT 1)

Port 1 is a mode independent 8-bit I/O and timer port. Each line can be configured as either an input or output as defined by the port 1 data direction register. Port 1 bits 0, 1, and 2 (P10, P11, and P12) can also be used to exercise one input edge function and two output compare functions of the timer. The TTL compatible three-state buffers can drive one Schottky TTL load and 30pF, Darlington transistors, or CMOS devices using external pullup resistors. It is configured as a data input port during RESET. Unused pins can remain unconnected.

P20-P24 (PORT 2)

Port 2 is a mode-independent, 5-bit, multipurpose I/O port. The voltage levels present on P20, P21, and P22 on the rising edge of RESET determine the operating mode of the MCU. The entire port is then configured as a data input port. The port 2 lines can be selectively configured as data output lines by setting the appropriate bits in the port 2 data direction register. The port 2 data register is used to move data through the port. However, if P21 is configured as an output, it is tied to the timer output compare 1 function and cannot be used to provide output from the port 2 data register unless output enable 1 (OE1) is cleared in timer control register 1.

Port 2 can also be used to provide an interface for the serial communications interface and the timer input edge function. These configurations are described in **SERIAL COMMUNICATIONS INTERFACE and PROGRAMMABLE TIMER.**

The port 2 three-state TTL-compatible output buffers are capable of driving one Schottky TTL load and 30pF, or CMOS devices using external pullup resistors.

PORT 2 DATA REGISTER

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

P30-P37 (PORT 3)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address/data bus depending on the operating mode. The TTL compatible three-state output buffers can drive one Schottky TTL load and 90pF. Unused lines can remain unconnected.

PORT 3 IN SINGLE-CHIP MODE - Port 3 is an 8-bit I/O port in the single-chip mode with each line configured by the port 3 data direction register. There are also two lines, IS3 and OS3, which can be used to control port 3 data transfers. Three port 3 options are controlled by the port 3 control and status register and are available only in single-chip mode : 1) port 3 input data can be latched using IS3 as a control signal, 2) OS3 can be generated by either an MPU read or write to the port 3 data register, and 3) an IRQ1 interrupt can be enabled by an IS3 negative edge. Port 3 latch timing is shown in figure 4.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3 Flag	IS3 IRQ1	X	OSS	Latch Enable	X	X	X	$0F

Bits 0-2 Not Used.

Bit 3 **Latch Enable** - This bit controls the input latch for port 3. If set, input data is latched by an IS3 negative edge. The latch is transparent after a read of the port 3 data register. Latch enable is cleared during reset.

Bit 4 **OSS (Output Strobe Select)** - This bit determines whether OS3 will be generated by a read or write of the port 3 data register. When clear, the strobe is generated by a read ; when set, it is generated by a write. OSS is cleared during reset.

Bit 5 Not used.

Bit 6 **IS3 IRQ1 Enable** - When set, an IRQ1 interrupt will be enabled whenever the IS3 flag is set ; when clear, the interrupt is inhibited. This bit is cleared during reset.

Bit 7 **IS3 Flag** - This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the port 3 data register or during reset.

PORT 3 IN EXPANDED NON-MULTIPLEXED MODE - Port 3 is configured as a bidirectional data bus (D7-D0) in the expanded non-multiplexed mode. The direction of data transfers is controlled by read/write (SC2). Data is clocked by E (enable).

PORT 3 IN EXPANDED MULTIPLEXED MODE - Port 3 is configured as a time multiplexed address (A7-A0) and data bus (D7-D0) in the expanded multiplexed mode where address strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent bus conflicts.

P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90pF, and is the only port with external pullup resistors. Unused lines can remain unconnected.

PORT 4 IN SINGLE-CHIP MODE - In single-chip mode, port 4 functions as an 8-bit I/O port with each line configured by the port 4 data direction register. Internal pullup resistors allow the port to directly interface with CMOS at 5-volts levels. External pullup resistors to more than 5 volts, however, cannot be used.

PORT 4 IN EXPANDED NON-MULTIPLEXED MODE - Port 4 is configured from reset as an 8-bit input port where the port 4 data direction register can be written to provide any or all of eight address lines A0 to A7. Internal pullup resistors pull the lines high until the port 4 data direction register is configured.

PORT 4 IN EXPANDED MULTIPLEXED MODE - In all expanded multiplexed modes except modes 1 and 6, port 4 functions as half of the address bus and provides A8 to A15. In modes 1 and 6, the port is configured from reset as an 8-bit parallel input port where the port 4 data direction register can be written to provide any or all of upper address lines A8 to A15. Internal pullup resistors pull the lines high until the port 4 data direction register is configured where bit 0 controls A8.

RESIDENT MEMORY

The EF6801U4 provides 4096 bytes of on-chip ROM and 192 bytes of on-chip RAM.

Thirty-two bytes of the RAM are powered through the V_{CC} standby pin and are maintainable during V_{CC} power down. This standby portion of the RAM consists of 32 bytes located from $40 through $5F in all modes except mode 3 which is $D040 through $D05F.

Power must be supplied to V_{CC} standby if the internal RAM is to be used regardless of whether standby power operation is anticipated.

The RAM is controlled by the RAM control register.

RAM CONTROL REGISTER ($14)

The RAM control register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during power-down operation. It is intended that RAME be cleared and STBY PWR be set as part of a power-down procedure.

RAM CONTROL REGISTER

7	6	5	4	3	2	1	0	
STBY PWR	RAM	X	X	X	X	X	X	$14

Bits 0-5 Not Used.

Bit 6 **RAM Enable** - This read/write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of RESET. If RAME is clear, any access to a RAM address is external. If RAME is set, the RAM is included in the internal map.

Bit 7 **Standby Power** - This bit is a read/write status bit which when cleared indicates that V_{CC} standby has decreased sufficiently below V_{SBB} (minimum) to make data in the standby RAM suspect. It can be set only by software and is not affected during reset.

PROGRAMMABLE TIMER

The programmable timer can be used to perform measurements on two separate input waveforms while independently generating three output waveforms. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in figure 21.

COUNTER ($09:0A), ($15, $16)

The key timer element is a 16-bit free-running counter which is incremented by E (enable). It is cleared during reset and is read-only with one exception : in mode 0 a write to the counter ($09) will configure it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. The TOF is set whenever the counter contains all ones. If ETOI is set, an interrupt will occur when the TOF is set. The counter may also be read as $15 and $16 to avoid inadvertently clearing the TOF.

OUTPUT COMPARE REGISTERS ($0B:0C), ($1A:1B), ($1C:1D)

The three output compare registers are 16-bit read/write registers, each used to control an output waveform or provide an arbitrary time-out flag. They are compared with the free-running counter during the negative half of each E cycle. When a match occurs, the corresponding output compare flag (OCF) is set and the corresponding output level (OLVL) is

clocked to an output level register. If both the corresponding output enable bit and data direction register bit are set, the value represented in the output level register will appear on the corresponding port pin. The appropriate OLVL bit can then be changed for the next compare.

The function is inhibited for one cycle after a write to its high byte ($0B, $1A, or $1A, or $1C) to ensure a valid compare after a double byte write. Writes can be made to either byte of the output compare register without affecting the other byte. The OLVL value will be clocked out independently of whether the OCF has previously been cleared. The output compare registers are set to $FFFF during reset.

INPUT CAPTURE REGISTERS ($0D:0E), ($1E:1F)

The two input capture registers are 16-bit read-only registers used to store the free-running counter when a "proper" input transition occurs as defined by the corresponding input edge bit (IEDG1 or IEDG2). The input pin's data direction register should be configured as an input, but the edge detect circuit always senses P10 and P20 even when configured as an output. The counter value will be latched into the input capture registers on the second negative edge of the E clock following the transition.

An input capture can occur independently of ICF ; the register always contains the most current value. Counter transfer is inhibited, however, between accessed of a double byte MPU read. The input pulse width must be at least two E cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTERS

Four registers are used to provide the EF6801U4/EF6803U4 with control and status information about the three output compare functions, the timer overflow function, and the two input edge functions of the timer. They are :

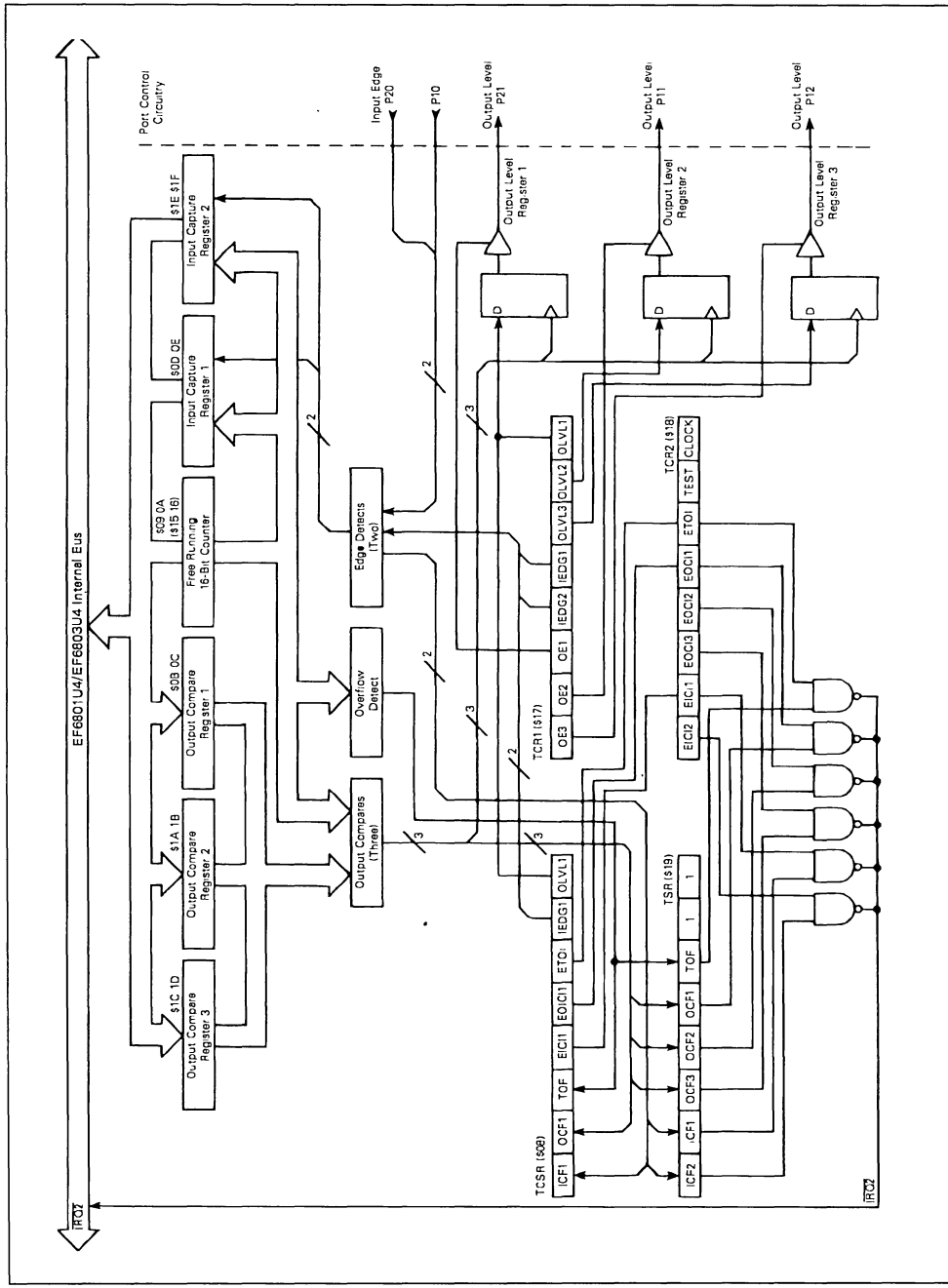
Timer Control and Status Register (TCSR)

Timer Control Register 1 (TCR1)

Timer Control Register 2 (TCR2)

Timer Status Register (TSR)

Figure 21 : Block Diagram of Programmable Timer.



TIMER CONTROL AND STATUS REGISTER (TCSR) ($08) - The timer control and status register is an 8-bit register of which all bits are readable, while only bits 0-4 can be written. All the bits in this register are also accessible through the two timer control registers and the timer status register. The three most significant bits provide the timer status and indicate if :

1. a proper level transition has been detected at P20,
2. a match has occurred between the free-running counter and output compare register 1, or
3. the free-running counter has overflowed.

Each of the three events can generate an IRQ2 interrupt and is controlled by an individual enable bit in the TCSR.

TIMER CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
ICF1	OCF1	TOF	EICI1	EOCI1	ETOI	IEDG1	OLVL1	$08

Bit 0 **Output Level 1** - OLVL1 is clocked to output level register 1 by a successful output compare and will appear at P21 if bit 1 of the port 2 data direction register is set and the OE1 control bit in timer control register 1 is set. OLVL1 and output level register 1 are cleared during reset. Refer to **TIMER CONTROL REGISTER 1** (TCR1) ($17).

Bit 1 **Input Edge 1** - IEDG1 is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register 1 :
IEDG1 = 0 transfer on a negative-edge
IEDG1 = 1 transfer on a positive-edge
Refer to **TIMER CONTROL REGISTER 1** (TCR1) ($17).

Bit 2 **Enable Timer Overflow Interrupt** - When set, an IRQ2 interrupt will be generated when the timer overflow flag is set ; when clear, the interrupt is inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL REGISTER 2** (TCR2) ($18).

Bit 3 **Enable Output Compare Interrupt 1** - When set, an IRQ2 interrupt will be generated when output compare flag 1 is set ; when clear, the interrupt is inhibited. EOCI1 is cleared during reset. Refer to **TIMER CONTROL REGISTER 2** (TCR2) ($18).

Bit 4 **Enable Input Capture Interrupt 1** - When set, an IRQ2 interrupt will be generated when input capture flag 1 is set ; when clear, the interrupt is inhibited. EICI1 is cleared during reset. Refer to **TIMER CONTROL REGISTER 2** (TCR2) ($18).

Bit 5 **Timer Overflow Flag** - The TOF is set when the counter contains all ones ($FFFF). It is cleared by reading the TCSR or the TSR (with TOF set) and the counter high byte ($09), or during reset. Refer to **TIMER STATUS REGISTER** (TSR) ($19).

Bit 6 **Output Compare Flag 1** - OCF1 is set when output compare register 1 matches the free-running counter. OCF1 is cleared by reading the TCSR or the TSR (with OCF1 set) and then writing to output compare register 1 ($0B or $0C), or during reset. Refer to **TIMER STATUS REGISTER** (TSR) ($19).

Bit 7 **Input Capture Flag** - ICF1 is set to indicate that a proper level transition has occurred ; it is cleared by reading the TCSR or the TSR (with ICF1 set) and the input capture register 1 high byte ($0D), or during reset. Refer to **TIMER STATUS REGISTER** (TSR) ($19).

TIMER CONTROL REGISTER 1 (TCR1) ($17) - Timer control register 1 is an 8-bit read/write register which contains the control bits for interfacing the output compare and input capture registers to the corresponding I/O pins.

TIMER CONTROL REGISTER 1

7	6	5	4	3	2	1	0	
OE3	OE2	OE1	IEDG2	IEDG1	OLVL3	OLVL2	OLVL1	$17

Bit 0 **Output Level 1** - OLVL1 is clocked to output level register 1 by a successful output compare and will appear at P21 if bit 1 of the port 2 data direction register is set and the OE1 control bit is set. OLVL1 and output level register 1 are cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER** (TCSR) ($08).

Bit 1 **Output Level 2** - OLVL2 is clocked to output level register 2 by a successful output compare and will appear at P11 if bit 1 of port 1 data direction register is set and the OE2 control bit is set. OLVL2 and output level register 2 are cleared during reset.

Bit 2 **Output Level 3** - OLVL3 is clocked to output level register 3 by a successful output compare and will appear at P12 if bit 2 of port 1 data direction register is set and the

OE3 control bit is set. OLVL3 and output level register 3 are cleared during reset.

Bit 3 **Input Edge 1** - IEDG1 is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register 1.
IEDG1 = 0 transfer on a negative-edge
IEDG1 = 1 transfer on a positive-edge
Refer to **TIMER CONTROL AND STATUS REGISTER** (TCSR) ($08).

Bit 4 **Input Edge 2** - IEDG2 is cleared during reset and controls which level transition on P20 will trigger a counter transfer to input capture register 2.
IEDG2 = 0 transfer on a negative-edge
IEDG2 = 1 transfer on a positive-edge

Bit 5 **Output Enable 1** - OE1 is set during reset and enables the contents of output level register 1 to be connected to P21 when bit 1 of port 2 data direction register is set.
OE1 = 0 port 2 bit 1 data register output
OE1 = 1 output level register 1

Bit 6 **Output Enable 2** - OE2 is cleared during reset and enables the contents of output level register 2 to be connected to P11 when bit 1 of port 1 data direction register is set.
OE2 = 0 port 1 bit 1 data register output
OE2 = 1 output level register 2

Bit 7 **Output Enable 3** - OE3 is cleared during reset and enables the contents of output level register 3 to be connected to P12 when bit 2 of port 1 data direction register is set
OE3 = 0 port 1 bit 2 data register output
OE3 = 1 output level register 3

TIMER CONTROL REGISTER 2 (TCR2) ($18) - Timer control register 2 is an 8-bit read/write register (except bits 0 and 1) which enable the interrupts associated with the free-running counter, the output compare registers, and the input capture registers. In test mode 0, two more bits (clock and test) are available for checking the timer.

TIMER CONTROL REGISTER 2 (non-test modes)

7	6	5	4	3	2	1	0	
EICI2	EICI1	EOCI3	EOCI2	EOCI1	ETOI	1	1	$18

Bits 0-1 **Read-only Bits** - When read, these bits return a value of 1. Refer to **TIMER CONTROL REGISTER 2** (test mode).

Bit 2 **Enable Timer Overflow Interrupt** - When set, an IRQ2 interrupt will be generated when the timer overflow flag is set ; when clear, the interrupt is inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER** (TCSR) ($08).

Bit 3 **Enable Output Compare Interrupt 1** - When set, an IRQ2 interrupt will be generated when the output compare flag 1 is set ; when clear, the interrupt is inhibited. EOCI1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER** (TCSR) ($08).

Bit 4 **Enable Output Compare Interrupt 2** - When set, an IRQ2 interrupt will be generated when the output compare flag 2 is set ; when clear, the interrupt is inhibited. EOCI2 is cleared during reset.

Bit 5 **Enable Output Compare Interrupt 3** - When set, an IRQ2 interrupt will be generated when the output compare flag 3 is set ; when clear, the interrupt is inhibited. EOCI3 is cleared during reset.

Bit 6 **Enable Input Capture Interrupt 1** - When set, an IRQ2 interrupt will be generated when the input capture flag 1 is set ; when clear, the interrupt is inhibited. EICI1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER** (TCSR) ($08).

Bit 7 **Enable Input Capture Interrupt 2** - When set, an IRQ2 interrupt will be generated when the input capture flag 2 is set ; when clear, the interrupt is inhibited. EICI2 is cleared during reset.

The timer test bits (test and clock) allow the free-running counter to be tested as two separate 8-bit counters to speed testing.

TIMER CONTROL REGISTER 2 (test mode)

7	6	5	4	3	2	1	0	
EICI2	EICI1	EOCI3	EOCI2	EOCI1	ETOI	TEST	CLOCK	$18

Bit 0 **CLOCK** - The CLOCK control bit selects which half of the 16-bit free-running counter (MSB or LSB) should be clocked with E. The CLOCK bit is a read/write bit only in mode 0 and is set during reset.
CLOCK = 0 - Only the eight most significant bits of the three-running counter run with TEST = 0.
CLOCK = 1 - Only the eight least significant bits of the free-running counter run when TEST = 0.

Bit 1 **TEST** - the TEST control bit enables the timer test mode. TEST is a read/write bit in mode 0 and is set during reset.
TEST = 0 - Timer test mode enabled :
a) The timer LSB latch is transparent which allows the LSB to be read independently of the MSB.
b) Either the MSB or the LSB of the timer is clocked by E, as defined by the CLOCK bit.
TEST = 1 – Timer test mode disabled.

Bits 2-7 See **TIMER CONTROL REGISTER 2** (non-test modes). (these bits function the same as in the non-test modes).

TIMER STATUS REGISTER (TSR) ($19) - The timer status register is an 8-bit read-only register which contains the flags associated with the free-running counter, the output compare registers, and the input capture registers.

TIMER STATUS REGISTER

7	6	5	4	3	2	1	0	
ICF2	ICF1	OCF3	OCF2	OCF1	TOF	1	1	$19

Bits 0-1 Not used.

Bit 2 **Timer Overflow Flag** - The TOF is set when the counter contains all ones ($FFFF). It is cleared by reading the TSR or the TCSR (with TOF set) and then the counter high byte ($09), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER** (TCSR) ($08).

Bit 3 **Output Compare Flag 1** - OCF1 is set when output compare register 1 matches the free-running counter. OCF1 is cleared by reading the TSR or the TCSR (with OCF1 set) and then writing to output compare register 1 ($0B or $0C), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER** (TCSR) ($08).

Bit 4 **Output Compare Flag 2** - OCF2 is set when output compare register 2 matches the free-running counter. OCF2 is cleared by reading the TSR (with OCF2 set) and then writing to output compare register 2 ($1A or $1B), or during reset,

Bit 5 **Output Compare Flag 3** - OCF3 is set when output compare register 3 matches the free-running counter. OCF3 is cleared by reading the TSR (with OCF3 set) and then writing to output compare register 3 ($1C or $1D), or during reset.

Bit 6 **Input Capture Flag 1** - ICF1 is set to indicate that a proper level transition has occurred ; it is cleared by reading the TSR or the TCSR (with ICF1 set) and the input capture register 1 high byte ($0D), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER** (TCSR) ($08).

Bit 7 **Input Capture Flag 2** - ICF2 is set to indicate that a proper level transition has occurred ; it is cleared by reading the TSR (with ICF2 set) and the input capture register 2 high byte ($1E), or during reset.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous serial communications interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and biphase and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

In a typical serial loop multiprocessor configuration, the software protocol will usually identify the addresse(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive ones or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

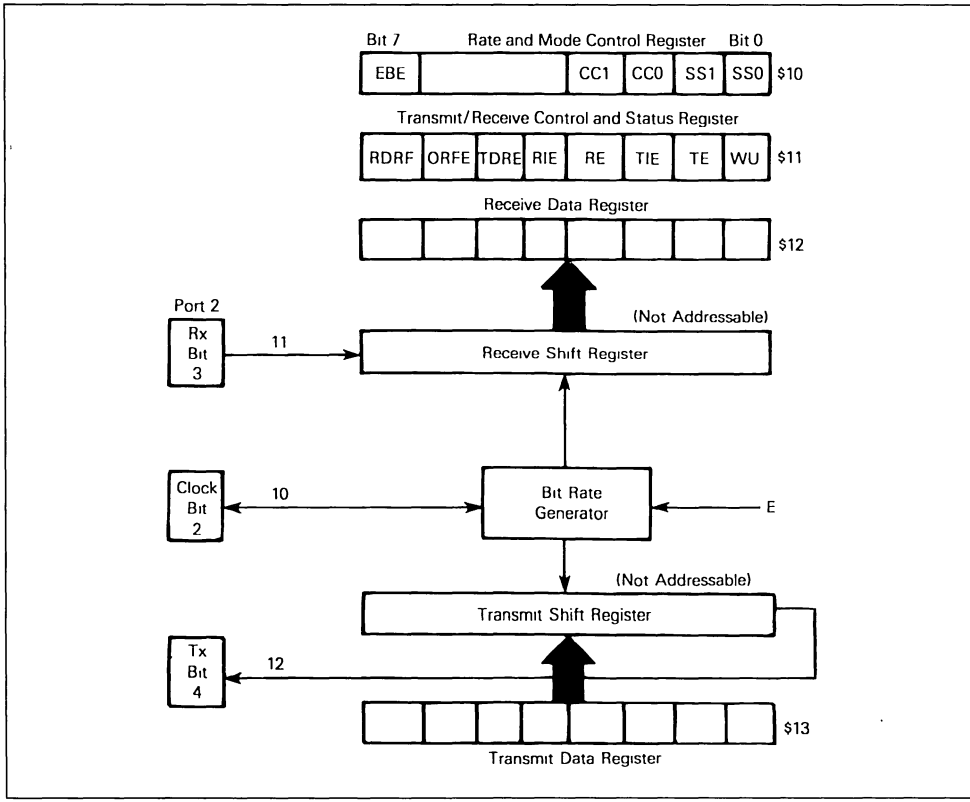
The following features of the SCI are programmable :

- Format : standard mark/space (NRZ) or bi-phase
- Clock : external or internal bit rate clock
- Baud : one of eight per E clock frequency or external clock (x 8 desired baud)
- Wake-up Feature : enabled or disabled
- Interrupt Requests : enabled individually for transmitter and receiver
- Clock Output : Internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The serial communications interface includes four addressable registers as depicted in figure 22. It is controlled bythe rate and mode control register and the transmit/receive control and status register. Data is transmitted and received utilizing a write-only transmit register and a read-only receive register. The shift registers are not accessible to software.

Figure 22 : SCI Registers.



RATE AND MODE CONTROL REGISTER (RMCR) ($10)

The rate and mode control register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of five write-only bits in conjunction with bit 7 control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER

7	6	5	4	3	2	1	0	
EBE	X	X	X	CC1	CC0	SS1	SS0	$10

Bit 1 : Bit 0 **SS1 : SS0 Speed Select** - These two bits select the baud when using the internal clock. Eight rates may be selected (in conjunction with bit 7) which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

Bit 3 : Bit 2 **CC1 : CC0 Clock Control and Format Select** - These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is

cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

Bits 4-6 Not used.

Bit 7 **EBE Enhanced Baud Enable** - EBE selects the standard EF6801 baud rates when clear and the additional baud rates when set (table 6). This bit is cleared by reset and is a write-only control bit.
EBE = 0 standard EF6801 baud rates
EBE = 1 additional baud rates

Table 6 : SCI Bit Times and Rates.

EBE	SS1:SS0		4 f_o → E	2.4576MHz 614.4kHz		4.0MHz 1.0MHz		4.9152MHz 1.2288MHz	
				Baud	Time	Baud	Time	Baud	Time
0	0	0	÷ 16	38400.0	26µs	62500.0	16.0µs	76800.0	13.0µs
0	0	1	÷ 128	4800.0	208.3µs	7812.5	128.0µs	9600.0	104.2µs
0	1	0	÷ 1024	600.0	1.67ms	976.6	1.024ms	1200.0	833.3µs
0	1	1	÷ 4096	150.0	6.67ms	244.1	4.096ms	300.0	3.33ms
1	0	0	÷ 64	9600.0	104.2µs	15625.0	64µs	19200.0	52.0µs
1	0	1	÷ 256	2400.0	416.6 µs	3906.3	256µs	4800.0	208.3µs
1	1	0	÷ 512	1200.0	833.3µs	1953.1	512µs	2400.0	416.6µs
1	1	1	÷ 2048	300.0	3.33ms	488.3	2.05ms	600.0	1.67ms
External (P22)*				76800.0	13.0µs	125000.0	8.0µs	153600.0	6.5µs

* Using maximum clock rate

Table 7 : SCI Format and Clock Source Control.

CC1:CC0	Format	Clock Source	Port 2 Bit 2
00	Bi-phase	Internal	Not used
01	NRZ	Internal	Not used
10	NRZ	Internal	Output
11	NRZ	External	Input

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR) ($11) - The transmit/receive control and status register controls the transmitter, receiver, wake-up feature, and two individual interrupts, and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by RESET.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$11

If both CC1 and CC0 are set, an external TTL-compatible clock must be connected to P22 at eight times (8 x) the desired bit rate, but not greater than E, with a duty cycle of 50% (± 10%). If CC1 : CC0 = 10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE

The source of SCI internal bit rate clock is the timer free-running counter. An MPU write to the counter in mode 0 can disturb serial operations.

Bit 0 **"Wake-Up" on Idle Line** - When set, WU enables the wake-up function ; it is cleared by ten consecutive ones or during reset. WU will not be set if the line is idle. Refer to **WAKE-UP FEATURE**.

Bit 1 **Transmit Enable** - When set, P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive ones is transmitted. TE is cleared during reset.

Bit 2 **Transmit Interrupt Enable** - When set, an IRQ2 is set ; when clear, the interrupt is inhibited. TE is cleared during reset.

Bit 3 **Receive Enable** - When set, the P23 DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared during reset.

Bit 4 **Receiver Interrupt Enable** - When set, an IRQ2 interrupt is enabled when RDRF and/or ORFE is set ; when clear, the interrut is inhibited. RIE is cleared during reset.

Bit 5 **Transmit Data Register Empty** - TDRE is set when the transmit data register is transferred to the output serial shift register or during reset. It is cleared by reading the RCSR (with TDRE set) and then writing to the transmit data register. Additional data will be transmitted only if TDRE has been cleared.

Bit 6 **Overrun Framing Error** - If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF : if RDRF is set, then an overrun has occurred ; otherwise, a framing error has been detected. Data is not transferred to the receive data register in an overrun condition. Unframed data causing a framing error is transferred to the receive data register. However, subsequent data transfer is blocked until the framing error flag is cleared. ORFE is cleared by reading the TRCSR (with ORFE set) then the receive data register, or during reset.

Bit 7 **Receive Data Register Full** - RDRF is set when the input serial shift register to the receive data registers, or during reset.

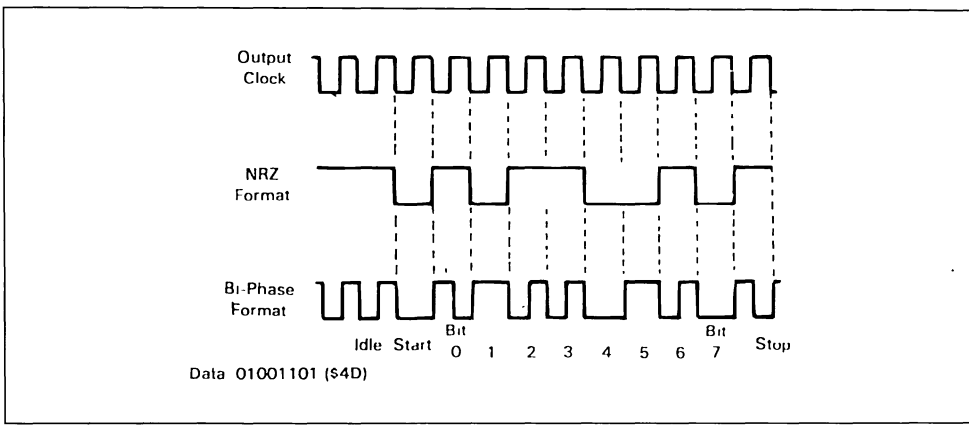
SERIAL OPERATIONS

The SCI is initialized by writing control bytes first to the rate and mode control register and then to the transmit/receive control and status register. When TE is set, the output of the transmit serial shift register is connected to P24 and serial output is initiated by transmitting a 9-bit preamble of ones.

At this point, one of two situations exists : 1) if the transmit data register is empty (TDRE =1), a continuous string of ones will be sent indicating an idle line ; or 2) if a byte has been written to the transmit data register (TDRE = 0), it will be transferred to the output serial shift register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0), and a stop bit (1) will be transmitted. If TDRE is still set when the next byte transfer occurs, ones will be sent until more data is provided. In bi-phase format, the output toggles at the start of each bit and at half-bit time when a one is sent. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. SCI data formats are illustrated in figure 23.

Figure 23 : SCI Data Formats.



INSTRUCTION SET

The EF6801U4/EF6803U4 is directly source compatible with the EF6801 and upward source and object code compatible with the EF6800. Execution times of key instructions have been reduced and several instructions have been added, including a hardware multiply. A list of new operations added to theEF6800 instruction set is shown in table 1.

In addition, two special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the program counter to increment like a 13-bit counter causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

The coding of the first (or only) byte corresponding to an executable instructions is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in table 8. There are 220 valid machine codes, 34 unassigned codes, and 2 codes reserved for test purposes.

Table 8 : CPU Instruction Map.

OP	MNEM	MODE	~	#
00	*			
01	NOP	INHER	2	1
02	*			
03	*			
04	LSRD		3	1
05	ASLD		3	1
06	TAP		2	1
07	TPA		2	1
08	INX		3	1
09	DEX		3	1
0A	CLV		2	1
0B	SEV		2	1
0C	CLC		2	1
0D	SEC		2	1
0E	CLI		2	1
0F	SEI		2	1
10	SBA		2	1
11	CBA		2	1
12	*			
13	*			
14	*			
15	*			
16	TAB		2	1
17	TBA		2	1
18	*			
19	DAA	INHER	2	1
1A	*			
1B	ABA	INHER	2	1
1C	*			
1D	*			
1E	*			
1F	*			
20	BRA	REL	3	2
21	BRN		3	2
22	BHI		3	2
23	BLS		3	2
24	BCC		3	2
25	BCS		3	2
26	BNE		3	2
27	BEQ		3	2
28	BVC		3	2
29	BVS		3	2
2A	BPL		3	2
2B	BMI		3	2
2C	BGE		3	2
2D	BLT		3	2
2E	BGT		3	2
2F	BLE	REL	3	2
30	TSX	INHER	3	1
31	INS		3	1
32	PULA		4	1
33	PULB		4	1
34	DES	INHER	3	1
35	TXS		3	1
36	PSHA		3	1
37	PSHB		3	1
38	PULX		5	1
39	RTS		5	1
3A	ABX		3	1
3B	RTI		10	1
3C	PSHX		4	1
3D	MUL		10	1
3E	WAI		9	1
3F	SWI		12	1
40	NEGA		2	1
41	*			
42	*			
43	COMA		2	1
44	LSRA		2	1
45	*			
46	RORA		2	1
47	ASRA		2	1
48	ASLA		2	1
49	ROLA		2	1
4A	DECA		2	1
4B	*			
4C	INCA		2	1
4D	TSTA		2	1
4E	T			
4F	CLRA		2	1
50	NEGB		2	1
51	*			
52	*			
53	COMB		2	1
54	LSRB		2	1
55	*			
56	RORB		2	1
57	ASRB		2	1
58	ASLB		2	1
59	ROLB		2	1
5A	DECB		2	1
5B	*			
5C	INCB		2	1
5D	TSTB		2	1
5E	T			
5F	CLRB	INHER	2	1
60	NEG	INDXD	6	2
61	*			
62	*			
63	COM		6	2
64	LSR		6	2
65	*			
66	ROR		6	2
67	ASR	INDXD	6	2
68	ASL	INDXD	6	2
69	ROL		6	2
6A	DEC		6	2
6B	*			
6C	INC		6	2
6D	TST		6	2
6E	JMP		3	2
6F	CLR	INDXD	6	2
70	NEG	EXTND	6	3
71	*			
72	*			
73	COM		6	3
74	LSR		6	3
75	*			
76	ROR		6	3
77	ASR		6	3
78	ASL		6	3
79	ROL		6	3
7A	DEC		6	3
7B	*			
7C	INC		6	3
7D	TST		6	3
7E	JMP		3	3
7F	CLR	EXTND	6	3
80	SUBA	IMMED	2	2
81	CMPA		2	2
82	SBCA		2	2
83	SUBD		4	3
84	ANDA		2	2
85	BITA		2	2
86	LDAA		2	2
87	*			
88	EORA		2	2
89	ADCA		2	2
8A	ORAA		2	2
8B	ADDA		2	2
8C	CPX	IMMED	4	3
8D	BSR	REL	6	2
8E	LDS	IMMED	3	3
8F	*			
90	SUBA	DIR	3	2
91	CMPA		3	2
92	SBCA		3	2
93	SUBD		5	2
94	ANDA		3	2
95	BITA		3	2
96	LDAA		3	2
97	STAA		3	2
98	EORA		3	2
99	ADCA		3	2
9A	ORAA		3	2
9B	ADDA		3	2
9C	CPX	DIR	5	2
9D	JSR		5	2
9E	LDS		4	2
9F	STS	DIR	4	2
A0	SUBA	INDXD	4	2
A1	CMPA		4	2
A2	SBCA		4	2
A3	SUBD		6	2
A4	ANDA		4	2
A5	BITA		4	2
A6	LDAA		4	2
A7	STAA		4	2
A8	EORA		4	2
A9	ADCA		4	2
AA	ORAA		4	2
AB	ADDA		4	2
AC	CPX		6	2
AD	JSR		6	2
AE	LDS		5	2
AF	STS	INDXD	5	2
B0	SUBA	EXTND	4	3
B1	CMPA		4	3
B2	SBCA		4	3
B3	SUBD		6	3
B4	ANDA		4	3
B5	BITA		4	3
B6	LDAA		4	3
B7	STAA		4	3
B8	EORA		4	3
B9	ADCA		4	3
BA	ORAA		4	3
BB	ADDA		4	3
BC	CPX		6	3
BD	JSR		6	3
BE	LDS		5	3
BF	STS	EXTND	5	3
C0	SUBB	IMMED	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB		2	2
C6	LDAB		2	2
C7	*			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORAB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	*			
CE	LDX	IMMED	3	3
CF	*			
D0	SUBB	DIR	3	2
D1	CMPB		3	2
D2	SBCB		3	2
D3	ADDD		5	2
D4	ANDB		3	2
D5	BITB		3	2
D6	LDAB		3	2
D7	STAB		3	2
D8	EORB		3	2
D9	ADCB		3	2
DA	ORAB		3	2
DB	ADDB		3	2
DC	LDD		4	2
DD	STD		4	2
DE	LDX		4	2
DF	STX	DIR	4	2
E0	SUBB	INDXD	4	2
E1	CMPB		4	2
E2	SBCB		4	2
E3	ADDD		6	2
E4	ANDB		4	2
E5	BITB		4	2
E6	LDAB		4	2
E7	STAB		4	2
E8	EORB		4	2
E9	ADCB		4	2
EA	ORAB		4	2
EB	ADDB		4	2
EC	LDD		5	2
ED	STD		5	2
EE	LDX		5	2
EF	STX	INDXD	5	2
F0	SUBB	EXTND	4	3
F1	CMPB		4	3
F2	SBCB		4	3
F3	ADDD		6	3
F4	ANDB		4	3
F5	BITB		4	3
F6	LDAB		4	3
F7	STAB		4	3
F8	EORB		4	3
F9	ADCB		4	3
FA	ORAB		4	3
FB	ADDB		4	3
FC	LDD		5	3
FD	STD		5	3
FE	LDX		5	3
FF	STX	EXTND	5	3

* UNDEFINED OP CODE

Notes : 1. Addressing Modes
INHER = Inherent INDXD = Indexed IMMED = Immediate
REL = Relative EXTND = Extended DIR = Direct
2 Unassigned opcodes are indicated by "*" and should not be executed.
3. Codes marked by "T" force the PC to function as a 16-bit counter.

PROGRAMMING MODEL

A programming model for the EF6801U4/ EF6803U4 is shown in figure 8. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulators A and/or B. Other registers are defined as follows :

PROGRAM COUNTER - The program counter is a 16-bit register which always points to the next instruction.

STACK POINTER - The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random-access memory at a location defined by the programmer.

INDEX REGISTER - The index register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

ACCUMULATORS - The MPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

CONDITION CODE REGISTER - The condition code register indicates the results on an instruction and includes the following five condition bits : negative (N), zero (Z), overflow (V), carry/borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7, are read as ones.

ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of addressing modes for all instructions is presented in tables 9, 10, 11, and 12 where execution times are provided in E cycles. Instruction execution times are summarized in table 13. With an input frequency of 4MHz, one E cycle is equivalent to one microsecond. A cycle-by-cycle description of bus activity for each instruction is provided in table 14 and descriptions of selected instructions are shown in figure 24.

IMMEDIATE ADDRESSING - The operand or "immediate byte(s)" is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

DIRECT ADDRESSING - The least significant byte of the operand address in contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, the 256-byte area is reserved for frequently referenced data.

EXTENDED ADDRESSING - The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

INDEXED ADDRESSING - The unsigned offset contained in the second byte of the instruction is added with carry to the index register and is used to reference memory without changing the index register. These are two byte instructions.

INHERENT ADDRESSING - The operand(s) is a register and no memory reference is required. These are single byte instructions.

RELATIVE ADDRESSING - Relative addressing is used only for branch instructions. If the branch condition is true, the program counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current program counter. This provides a branch range of – 126 to + 129 bytes from the first byte of the instruction. These are two byte instructions.

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 14 provides a detailed description of the information presednt on the address bus, data bus, and the read/write (R/W) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

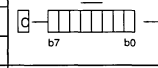
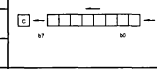
Note that during MPU reads of internal locations, the resultant value will not appear on the external data bus except in mode 0. "High order" byte refers to the most significant byte of a 16-bit value.

Table 9 : Index Register and Stack Manipulation Instructions.

Pointer Operations	MNEM	Immed			Direct			Index			Extnd			Inherent			Boolean/ Arithmetic Operation	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		5	4	3	2	1	0
																		H	I	N	Z	V	C
Compare Index Register	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X – M : M + 1	•	•	↕	↕	↕	↕
Decrement Index Register	DEX													09	3	1	X – 1 → X	•	•	•	↕	•	•
Decrement Stack Pointer	DES													34	3	1	SP – 1 → SP	•	•	•	•	•	•
Increment Index Register	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pointer	INS													31	3	1	1 SP + 1 → SP	•	•	•	•	•	•
Load Index Register	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	↕	↕	R	•
Load Stack Pointer	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	•	•	↕	↕	R	•
Store Index Register	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	↕	↕	R	•
Store Stack Pointer	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	↕	↕	R	•
Index Reg → Stack Pointer	TXS													35	3	1	X – 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Register	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP – 1 → SP X_H → M_{SP}, SP – 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•

The condition code register notes are listed after Table 12.

Table 10 : Accumulator and Memory Instructions (sheet 1 of 3).

Accumulator and Memory Operations	MNEM	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
																		5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00 . B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M : M +1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A • M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B • M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕
Shift Left Double	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A • M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B • M	•	•	↕	↕	R	•
Compare Accumulators	CBA													11	2	1	A – B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A – M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B – M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				M → M	•	•	↕	↕	R	S
	COMA													43	2	1	A → A	•	•	↕	↕	R	S
	COMB													53	2	1	B → B	•	•	↕	↕	R	S

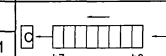
The condition code register notes are listed after Table 12

Table 10 : Accumulator and Memory Instructions (sheet 2 of 3).

Accumulator and Memory Operations	MNEM	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
																		5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Decimal Adjust, A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M – 1→ M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A – 1→ A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B – 1→ B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M +1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Accumulators	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M : M + 1 → D	•	•	↕	↕	R	•
Logical Shift, Left	LSL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	LSLA													48	2	1		•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	2		•	•	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R	↕	↕	↕
	LSRA													44	2	1		•	•	R	↕	↕	↕
	LSRB													54	2	1		•	•	R	↕	↕	↕
	LSRD													04	3	1		•	•	R	↕	↕	↕
Multiply	MUL													3D	10	1	A x B → D	•	•	•	•	•	↕
2's Complement (negate)	NEG							60	6	2	70	6	3				00 – M → M	•	•	↕	↕	↕	↕
	NEGA													40	2	1	00 – A → A	•	•	↕	↕	↕	↕
	NEGB													50	2	1	00 – B → B	•	•	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	BA	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Pusch Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•

The condition code register notes are listed after Table 12.

Table 10 : Accumulator and Memory Instructions (sheet 3 of 3).

Accumulator and Memory Operations	MNEM	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
																		5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	↕	↕
	ROLA													49	2	1		•	•	↕	↕	↕	↕
	ROLB													59	2	1		•	•	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	↕	↕
	RORA													46	2	1		•	•	↕	↕	↕	↕
	RORB													56	2	1		•	•	↕	↕	↕	↕
Subtract Accumulator	SBA													10	2	1	A – B → A	•	•	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A – M – C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B – M – C → B	•	•	↕	↕	↕	↕
Store Accumulators	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
	STD				DD	4	2	ED	5	2	FD	5	3				D → M : M +1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A – M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B – M → B	•	•	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D – M : M + 1 → D	•	•	↕	↕	↕	↕
Transfer Accumulator	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M – 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A – 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B – 00	•	•	↕	↕	R	R

The condition code register notes are listed after Table 12.

Table 11 : Jump and Branch Instructions.

Operations	MNEM	Direct			Relative			Index			Extend			Inherent			Branch Test	Condition Code Reg.					
																		5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch if Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•
Branch if Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•
Branch if = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•
Branch if ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•
Branch if > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch if Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•
Branch if Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•
Branch if ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch if Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•
Branch if Lower or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•
Branch if < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•
Branch if Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•
Branch if not Equal Zero	BNE				26	3	2										Z = 0	•	•	•	•	•	•
Branch if Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•
Branch if Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•
Branch if Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•
Branch to Subroutine	BSR				8D	6	2										See Special Operations-figure 24	•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump to Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return from Interrupt	RTI													3B	10	1	See Special Operations-figure 24	↕	↕	↕	↕	↕	↕
Return from Subroutine	RTS													39	5	1		•	•	•	•	•	•
Sofware Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI													3E	9	1		•	•	•	•	•	•

Table 12 : Condition Code Register Manipulation Instructions.

Operations	Inherent				Boolean Operation	Condition Code Register					
						5	4	3	2	1	0
	MNEM	Op	~	#		H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	A → CCR	↕	↕	↕	↕	↕	↕
CCR → Accumulator A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

LEGEND

Op Operation Code (hexadecimal)
~ Number of MPU Cycles
M_{SP} Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
– Arithmetic Minus
• Boolean AND
X Arithmetic Multiply
+ Boolean Inclusive OR
• Boolean Exclusive OR
M Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow 2's complement
C Carry/borrow from MSB
R Reset Always
S Set Always
↕ Affected
• Not Affected

Table 13 : Instruction Execution Times in E-cycles.

	Addressing Mode					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
ABA	●	●	●	●	2	●
ABX	●	●	●	●	3	●
ADC	2	3	4	4	●	●
ADD	2	3	4	4	●	●
ADDD	4	5	6	6	●	●
AND	2	3	4	4	●	●
ASL	●	●	6	6	2	●
ASLD	●	●	●	●	3	●
ASR	●	●	6	6	2	●
BCC	●	●	●	●	●	3
BCS	●	●	●	●	●	3
BEQ	●	●	●	●	●	3
BGE	●	●	●	●	●	3
BGT	●	●	●	●	●	3
BHI	●	●	●	●	●	3
BHS	●	●	●	●	●	3
BIT	2	3	4	4	●	●
BLE	●	●	●	●	●	3
BLO	●	●	●	●	●	3
BLS	●	●	●	●	●	3
BLT	●	●	●	●	●	3
BMI	●	●	●	●	●	3
BNE	●	●	●	●	●	3
BPL	●	●	●	●	●	3
BRA	●	●	●	●	●	3
BRN	●	●	●	●	●	3
BSR	●	●	●	●	●	6
BVC	●	●	●	●	●	3
BVS	●	●	●	●	●	3
CBA	●	●	●	●	2	●
CLC	●	●	●	●	2	●
CLI	●	●	●	●	2	●
CLR	●	●	6	6	2	●
CLV	●	●	●	●	2	●
CMP	2	3	4	4	●	●
COM	●	●	6	6	2	●
CPX	4	5	6	6	●	●
DAA	●	●	●	●	2	●
DEC	●	●	6	6	2	●
DES	●	●	●	●	3	●
DEX	●	●	●	●	3	●
EOR	2	3	4	4	●	●
INC	●	●	6	6	●	●
INS	●	●	●	●	3	●

	Addressing Mode					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
INX	●	●	●	●	3	●
JMP	●	●	3	3	●	●
JSR	●	5	6	6	●	●
LDA	2	3	4	4	●	●
LDD	3	4	5	5	●	●
LDS	3	4	5	5	●	●
LDX	3	4	5	5	●	●
LSL	●	●	6	6	2	●
LSLD	●	●	●	●	3	●
LSR	●	●	6	6	2	●
LSRD	●	●	●	●	3	●
MUL	●	●	●	●	10	●
NEG	●	●	6	6	2	●
NOP	●	●	●	●	2	●
ORA	2	3	4	4	●	●
PSH	●	●	●	●	3	●
PSHX	●	●	●	●	4	●
PUL	●	●	●	●	4	●
PULX	●	●	●	●	5	●
ROL	●	●	6	6	2	●
ROR	●	●	6	6	2	●
RTI	●	●	●	●	10	●
RTS	●	●	●	●	5	●
SBA	●	●	●	●	2	●
SBC	2	3	4	4	●	●
SEC	●	●	●	●	2	●
SEI	●	●	●	●	2	●
SEV	●	●	●	●	2	●
STA	●	3	4	4	●	●
STD	●	4	5	5	●	●
STS	●	4	5	5	●	●
STX	●	4	5	5	●	●
SUB	2	3	4	4	●	●
SUBD	4	5	6	6	●	●
SWI	●	●	●	●	12	●
TAB	●	●	●	●	2	●
TAP	●	●	●	●	2	●
TBA	●	●	●	●	2	●
TPA	●	●	●	●	2	●
TST	●	●	6	6	2	●
TSX	●	●	●	●	3	●
TXS	●	●	●	●	3	●
WAI	●	●	●	●	9	●

Table 14 : Cycle-by-cycle Operation (sheet 1 of 6).

IMMEDIATE

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Operand Data
LDS LDX LDD	3	1 2 3	Opcode Address Opcode Address + 1 Opcode Address + 2	1 1 1	Opcode Operand Data (high order byte) Operand Data (low order byte)
CPX SUBD ADDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Opcode Address + 2 Address Bus FFFF	1 1 1 1	Opcode Operand Data (high order byte) Operand Data (low order byte) Low Byte of Restart Vector

DIRECT

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Opcode Address Opcode Address + 1 Address of Operand	1 1 1	Opcode Address of Operand Operand Data
STA	3	1 2 3	Opcode Address Opcode Address + 1 Destination Address	1 1 0	Opcode Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Opcode Address of Operand Operand Data (high order byte) Operand Data (low order byte)
STS STX STD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Opcode Address of Operand Register Data (high order byte) Register Data (low order byte)
CPX SUBD ADDD	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Opcode Address of Operand Operand Data (high order byte) Operand Data (low order byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Subroutine Address Stack Pointer Stack Pointer – 1	1 1 1 0 0	Opcode Irrelevant Data First Subroutine Opcode Return Address (low order byte) Return Address (high order byte)

Table 14 : Cycle-by-cycle Operation (sheet 2 of 6).

EXTENDED

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Jump Address (high order byte)
		3	Opcode Address + 2	1	Jump Address (low order byte)
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Address of Operand
AND ORA		3	Opcode Address + 2	1	Address of Operand (low order byte)
BIT SBC					
CMP SUB		4	Address of Operand	1	Operand Data
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Destination Address (high order byte)
		3	Opcode Address + 2	1	Destination Address (low order byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Address of Operand (high order byte)
LDD		3	Opcode Address + 2	1	Address of Operand (low order byte)
		4	Address of Operand	1	Operand Data (high order byte)
		5	Address of Operand + 1	1	Operand Data (low order byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Address of Operand (high order byte)
STD		3	Opcode Address + 2	1	Address of Operand (low order byte)
		4	Address of Operand	0	Operand Data (high order byte)
		5	Address of Operand + 1	0	Operand Data (low order byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Address of Operand (high order byte)
CLR ROL		3	Opcode Address + 2	1	Address of Operand (low order byte)
COM ROR					
DEC TST*		4	Address of Operand	1	Current Operand Data
INC		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address of Operand	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Operand Address (high order byte)
ADDD		3	Opcode Address + 2	1	Operand Address (low order byte)
		4	Operand Address	1	Operand Data (high order byte)
		5	Operand Address + 1	1	Operand Data (low order byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Subroutine (high order byte)
		3	Opcode Address + 2	1	Address of Subroutine (low order byte)
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (low order byte)
		6	Stack Pointer – 1	0	Return Address (high order byte)

* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

Table 14 : Cycle-by-cycle Operation (sheet 3 of 6).

INDEXED

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS LDX LDD	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (high order byte)
		5	Index Register Plus Offset + 1	1	Operand Data (low order byte)
STS STX STD	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (high order byte)
		5	Index Register Plus Offset + 1	0	Operand Data (low order byte)
ASL LSR ASR NEG CLR ROL COM ROR DEC TST* INC	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Current Operand Data
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Index Register Plus Offset	0	New Operand Data
CPX SUBD ADDD	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (high order byte)
		5	Index Register + Offset + 1	1	Operand Data (low order byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (low order byte)
		6	Stack Pointer – 1	0	Return Address (high order byte)

* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

Table 14 : Cycle-by-cycle Operation (sheet 4 of 6).

INHERENT

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Opcode Irrelevant Data Index Register (low order byte) Index Register (high order byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (low order byte) Index Register (high order byte)

Table 14 : Cycle-by-cycle Operation (sheet 5 of 6).

INHERENT

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
RTS	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Address of Next Instruction (high order byte)
		5	Stack Pointer + 2	1	Address of Next Instruction (low order byte)
WAI	9	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Opcode of Next Instruction
		3	Stack Pointer	0	Return Address (low order byte)
		4	Stack Pointer – 1	0	Return Address (high order byte)
		5	Stack Pointer – 2	0	Index Register (low order byte)
		6	Stack Pointer – 3	0	Index Register (high order byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Condition Code Register
MUL	10	1	Opcode Addres	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (high order byte)
		8	Stack Pointer + 5	1	Index Register from Stack (low order byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (high order byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (low order byte)

Table 14 : Cycle-by-cycle Operation (sheet 6 of 6).

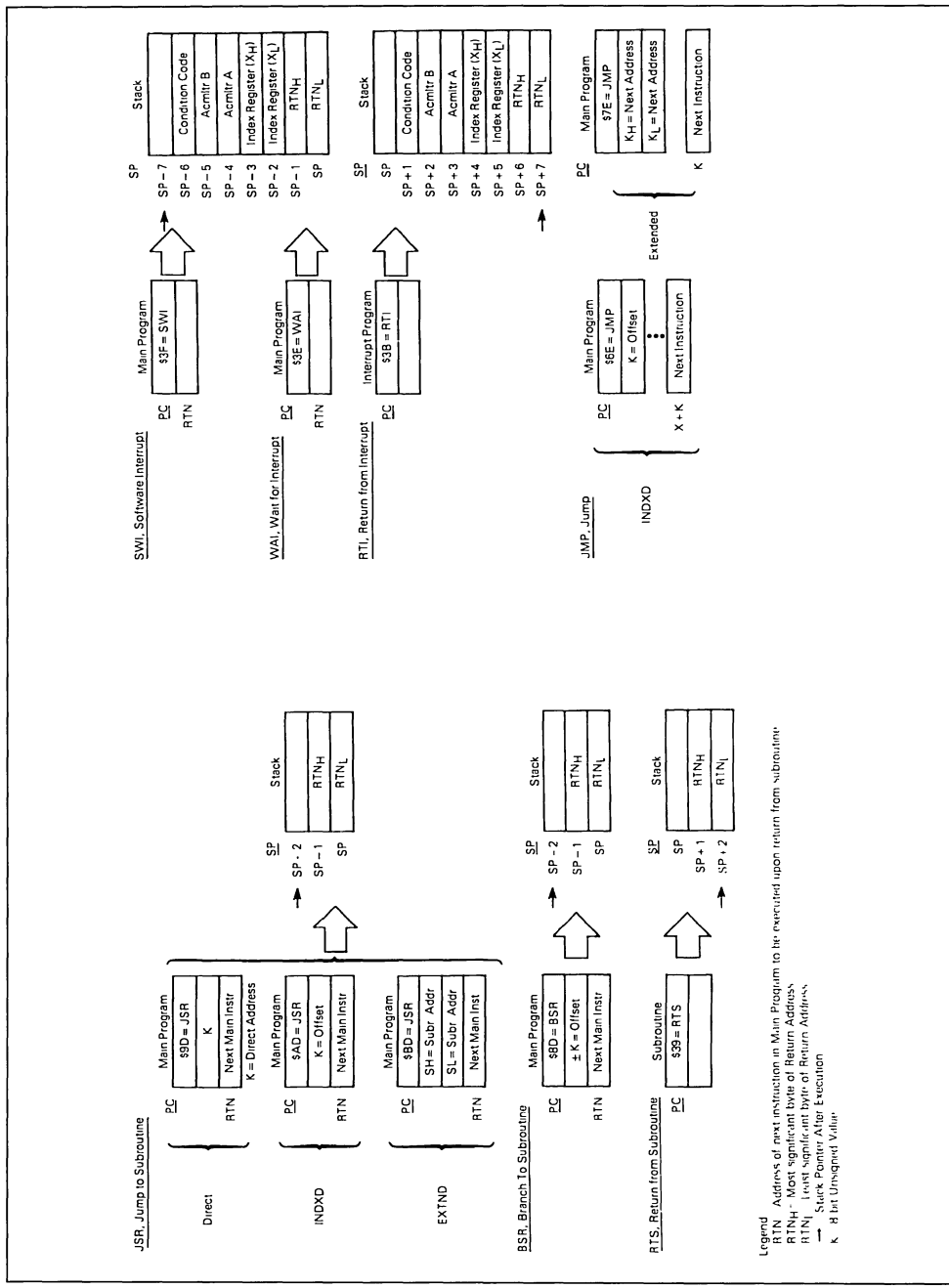
INHERENT (continued)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
SWI	12	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (low order byte)
		4	Stack Pointer – 1	0	Return Address (high order byte)
		5	Stack Pointer – 2	0	Index Register (low order byte)
		6	Stack Pointer – 3	0	Index Register (high order byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Condition Code Register
		10	Stack Pointer – 7	1	Irrelevant Data
		11	Vector Address FFFA (hex)	1	Address of Subroutine (high order byte)
		12	Vector Address FFFB (hex)	1	Address of Subroutine (low order byte)

RELATIVE

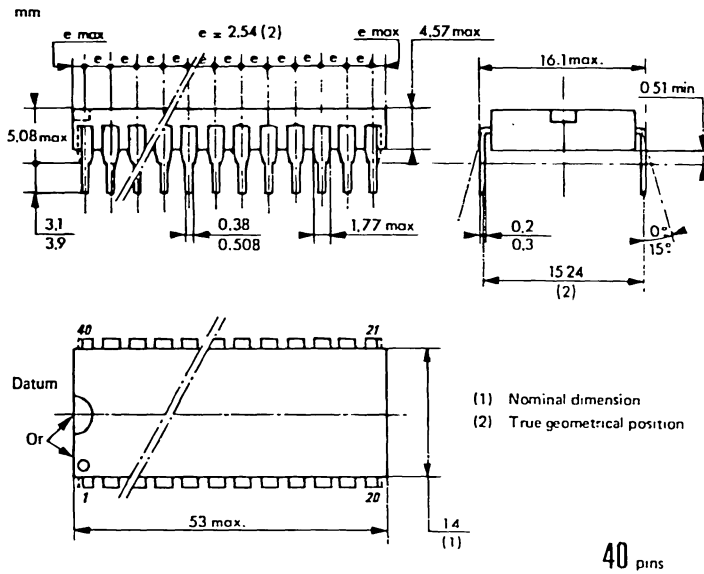
Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMI BVS	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
BSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (low order byte)
		6	Stack Pointer – 1	0	Return Address (high order byte)

Figure 24 : Special Operations.

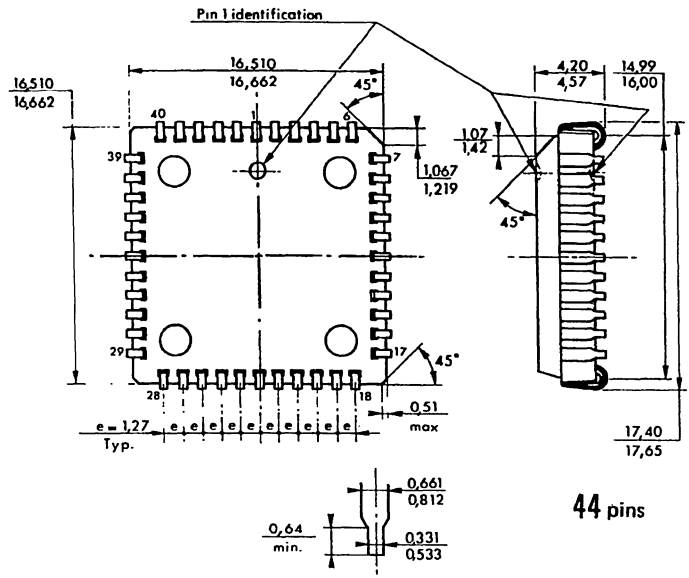


PACKAGE MECHANICAL DATA

DIL–PLASTIC PACKAGE



PLASTIC CHIP CARRIER



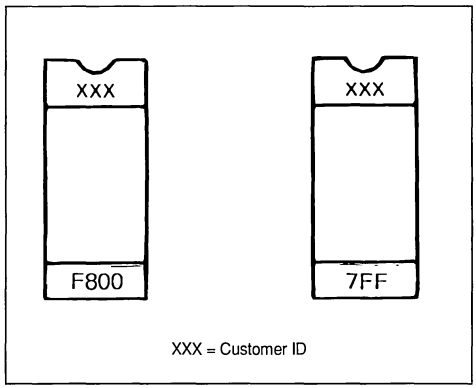
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS-THOMSON on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is ecessary to first contact your local SGS-THOMSON representative or distributor.

EPROMs

Two ET2716 or one ET2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed signed, and returned to SGS-THOMSON. The signed verification form consitutes the contractual agreement for creation of the customer mask. If desired, SGS-THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verifications process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not produce parts. The RVUs are thus not guaranteed by SGS-THOMSON. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6801 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process inhouse if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

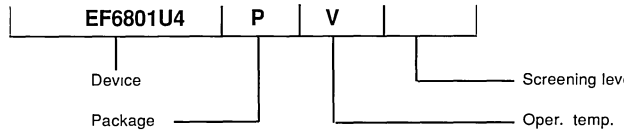
EFDOS is SGS-THOMSON' Disk Operating System available on development systems such as DEVICE...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser...

* Requires prior factory approval

Whenever ordering a custom MCU is required, please contact your local SGS-THOMSON representative or SGS-THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS-THOMSON Microelectronics representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device	Package					Oper. Temp			Screening Level			
	C	J	P	E	FN	L*	V	M	Std	D	G/B	B/B
EF6801/03U4 (1.0MHz)			●		●	●			●			
			●				●		●			
EF6801/03U4-1 (1.25MHz)			●		●	●			●			
			●				●		●			
EF68A01/03U4 (1.5MHz)			●			●			●			
Examples : EF6801P, EF6801FN, EF6801PV												

Package : C : Ceramic DIL, **J** : Cerdip DIL, **P** · Plastic DIL, **E** : LCCC, **FN** : PLCC.
Oper. temp. : L* · 0°C to 70°C, **V** : –40°C to +85°C, **M** : –55°C to +125°C, * : may be omitted.
Screening level : Std : (no-end suffix), **D** : NFC 96883 level D,
G/B : NFC 96883 level, G, **B/B** : NFC 96883 level B and MIL-STD-883C level B.

EXORciser is a registered trade mark of MOTOROLA Inc.

EF6801 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | | |

Customer's marking : | | | | | | | | | |

Application : .....

.....

ROM capacity required : | | | | bytes

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 40°C / + 85°C
- ❑ – 40°C / + 105°C

Package :
- ❑ Plastic
- ❑ PLCC

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
- ❑ SGS-THOMSON Microelectronics reference
  .....
- ❑ Special customer data reference*
  .....

Number of interrupt vector : | |

Quality level :
- ❑ STD
- ❑ D
- ❑ Other* (customer's quality specification ref.) :
  .....

Software developped by :
- ❑ SGS-THOMSON Microelectronics application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
- ❑ EPROM Reference :
- ❑ EFDOS/MDOS* disk file
  - ❑ 8" floppy
  - ❑ 5" 1/4 floppy
- ❑ Other *

OPTION LIST

-Internal max. clock frequency :
- ❑ 1.0MHz
- ❑ 1.25MHz
- ❑ 1.5MHz*
- ❑ 2.0MHz

* Requires prior factory approval

Yearly quantity forecast :
- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME DATE : SIGNATURE :

EF6804 FAMILY DATASHEETS

SGS-THOMSON MICROELECTRONICS

EF6804J2

8 BIT MICROCOMPUTER

HARDWARE FEATURES

- 5-VOLT SINGLE SUPPLY
- 32 BYTES OF RAM
- MEMORY MAPPED I/O
- 1012 BYTES OF PROGRAM ROM
- 64 BYTES OF DATA ROM
- 12 BIDIRECTIONAL I/O LINES (eight lines with high current sink capability)
- ON-CHIP CLOCK GENERATOR
- SELF-TEST MODE
- MASTER RESET
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE ®
- SOFTWARE PROGRAMMABLE 8-BIT TIMER CONTROL REGISTER AND TIMER PRESCALER (7 bits, 2^n)
- TIMER PIN IS PROGRAMMABLE AS INPUT OR OUTPUT
- ON-CHIP CIRCUIT FOR ROM VERIFY

SOFTWARE FEATURES

- SOFTWARE FEATURES
- SIMILAR TO EF6805 HMOS FAMILY
- BYTE EFFICIENT INSTRUCTION SET
- EASY TO PROGRAM
- TRUE BIT MANIPULATION
- BIT TEST AND BRANCH INSTRUCTION
- SEPARATE FLAGS FOR INTERRUPT AND NORMAL PROCESSING
- VERSATILE INDIRECT REGISTERS
- CONDITIONAL BRANCHES
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- TRUE LIFO STACK ELIMINATES STACK POINTER
- NINE POWERFUL ADDRESSING MODES
- ANY BIT IN DATA SPACE MEMORY MAY BE TESTED
- ANY BIT IN DATA SPACE MEMORY CAPABLE OF BEING WRITTEN TO MAY BE SET OR CLEARED

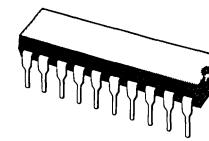
USER SELECTABLE OPTIONS

- 12 BIDIRECTIONAL I/O LINES WITH LSTTL, LSTTL/CMOS, OR OPEN-DRAIN INTERFACE
- CRYSTAL OR LOW-COST RESISTOR-CAPACITOR OSCILLATOR
- MASK SELECTABLE EDGE- OR LEVEL- SENSITIVE INTERRUPT PIN

INICE © is SGS-THOMSON development/emulation tool

DESCRIPTION

The EF6804J2 microcomputer unit (MCU) is a member of the EF6804 Family of very low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the EF6800-based instruction set.



P
(Plastic Package)

PIN CONNECTIONS

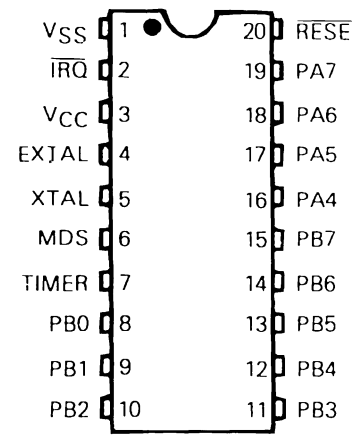
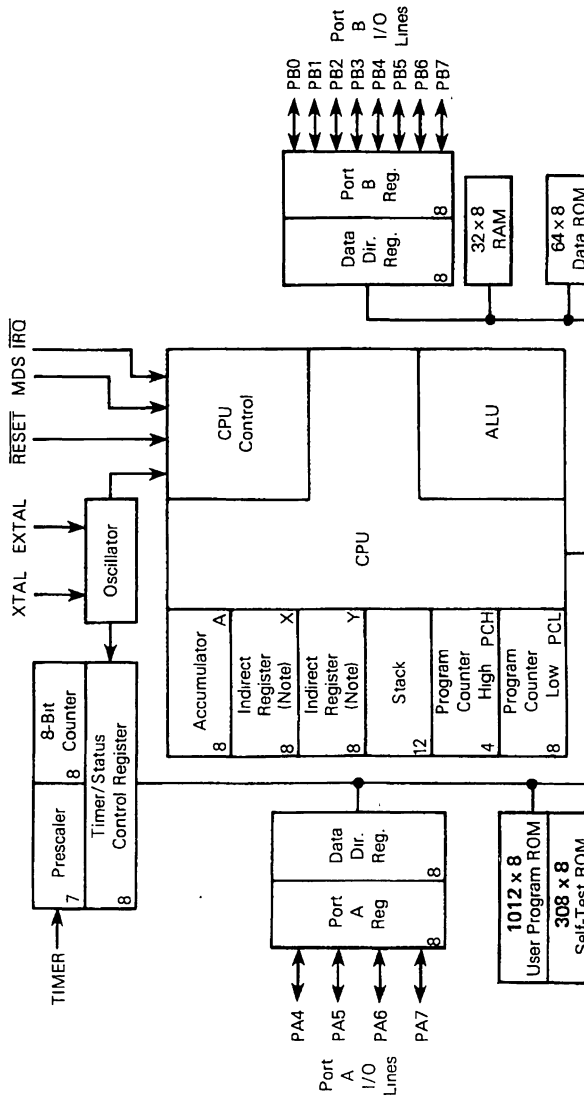


Figure 1.1 : EF6804J2 MCU Block Diagram.



Note : 8-bit indirect registers X and Y , although shown as part of the CPU, are actually located in the 32 x 8 RAM at locations $80 and $81.

SECTION 2

FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS

This section provides a description of the functional pins, memory spaces, the central processing unit (CPU), and the various registers and flags.

2.1. FUNCTIONAL PIN DESCRIPTION

2.1.1. V_{CC} AND V_{SS}. Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

2.1.2. $\overline{IRQ}$. This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **4.1. INTERRUPT** for additional information.

2.1.3. XTAL AND EXTAL. These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor and capacitor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to 4.4. Internal Clock Generator Options for recommendations concerning these inputs.

2.1.4. TIMER. In the input mode, the timer pin is connected to the prescaler input and serves as the timer clock. In the output mode, the timer pin signals that a time out of the timer has occurred. Refer to Section 3 Timer for additional information.

2.1.5. $\overline{RESET}$. The $\overline{RESET}$ pin is used to restart the processor of the EF6804J2 to the beginning of a program. This pin, together with the MDS pin, is also used to select the operating mode of the EF6804J2. If the MDS pin is at zero volts, the normal mode is selected and the program counter is loaded with the user restart vector. However, if the MDS pin is at + 5 volts, then pins PA6 ad PA7 are decoded to allow selection of the operating mode. Refer to 4.3. Reset for additional information.

2.1.6. MDS. The MDS (mode select) pin is used to place the MCU into special operating modes. If MDS is held at + 5 volts at the exit of the reset state, the decoded state of PA6 and PA7 is latched to determine the operating mode (single-chip, self-test, or ROM verify). However, if MDS is held at zero volts at the exit of the reset state, the single-chip operating mode is automatically selected (regardless of PA6 and PA7 state).

For those users familiar with the EF6801 microcomputer, mode selection is similar but much less complex in the EF6804J2. No special external diodes, switches, transistors, etc. are required in the EF6804J2.

2.1.7. INPUT/OUTPUT LINES (PA4-PA7, PB0-PB7). These 12 lines are arranged into one 4-bit port (A) and one 8-bit port (B). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to Section 5 Input/output Ports for additional information.

2.2. MEMORY

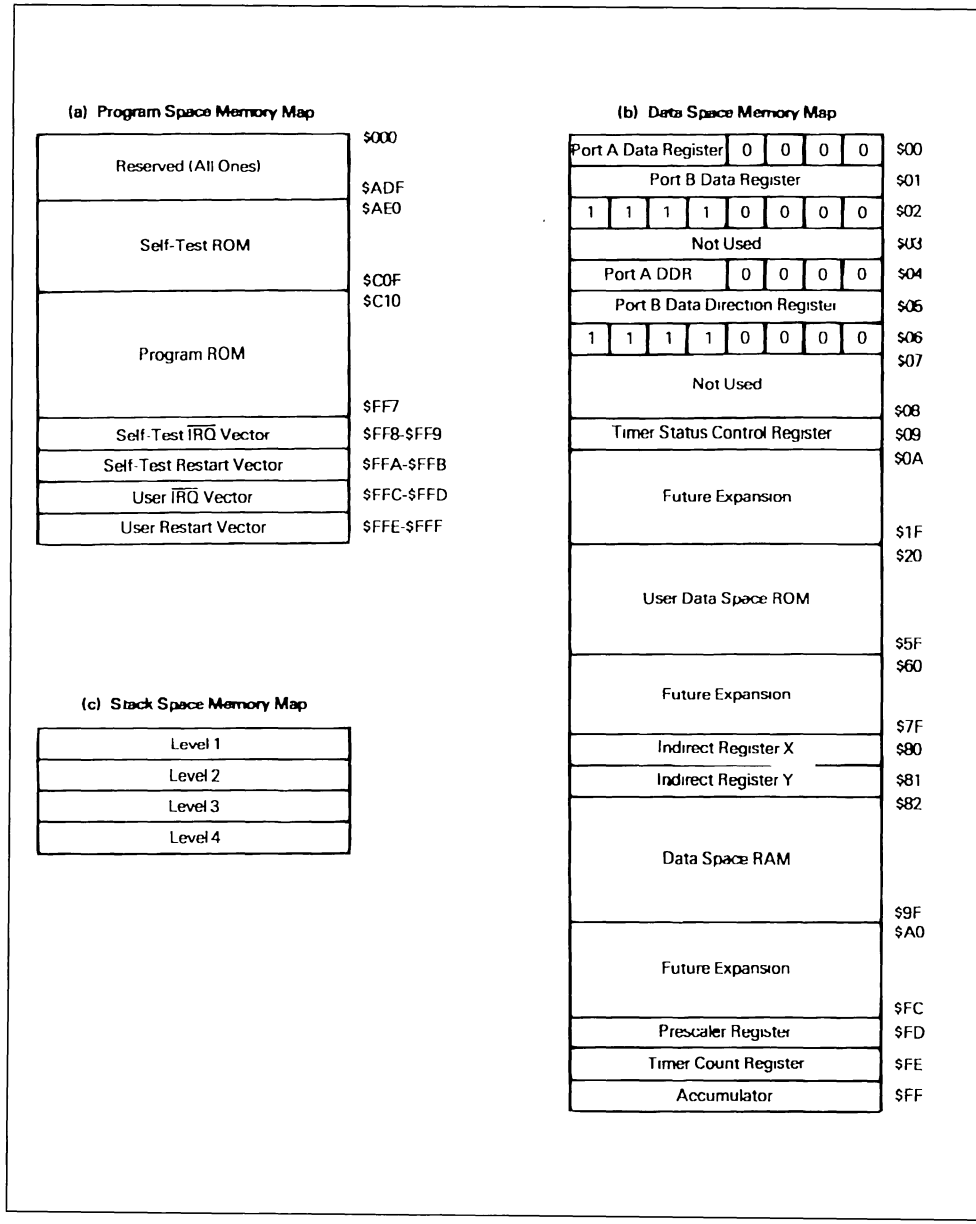
The MCU operates in three different memory spaces : program space, data space, and stack space. A representation of these memory spaces is shown in figure 2.1. The program space (figure 2.1a) contains all of the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, and the self-test and user vectors. The data space (figure 2.1b) contains all of the RAM locations, plus I/O locations and some ROM used for storage of tables ad constants. The stack space (figure 2.1c) contains RAM which is used for stacking subroutine and interrupt return addresses.

The MCU is capable of addressing 4096 bytes of program space memory with its program counter and 256 bytes of data space memory with its instructions. The data space memory contains three bytes for port data registers, three bytes for port data direction registers, one byte for timer status/control, 64 bytes ROM, 32 bytes RAM (which includes two bytes for X and Y indirect registers), two bytes for timer prescaler and count registers, and one byte for the accumulator. The program space section includes 304 bytes of self-test ROM, 1008 bytes program ROM, and eight bytes of vectors for self-test and user programs.

2.3. CENTRAL PROCESSING UNIT

The PCU of the EF6804 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control buses.

Figure 2.1 : EF6804J2 MCU Address Map.



2.4. REGISTERS

The EF6804 Family CPU has four registers and two flags available to the programmer. They are shown in figure 2.2. and are explained in the following paragraphs.

2.4.1. ACCUMULATOR (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is implemented as the highest RAM location ($FF) in data space and thus implies that several instructions exist which are not explicitly implemented. Refer to 6.3. Implied Instructions for additional information.

2.4.2. INDIRECT REGISTERS (X, Y). These two indirect registers are used to maintain pointers to other memory locations in data space. They are used in the register-indirect addressing mode, and can be accessed with the direct, indirect, short direct, or bit set/clear addressing modes. These registers are implemented as two of the 32 RAM locations ($80, $81) and as such generate implied instructions and may be manipulated in a manner similar to any RAM memory location in data space. Refer to 6.3. Implied Instructions for additional information.

2.4.3. PROGRAM COUNTER (PC). The program counter is a 12-bit register that contains the address of the next ROM word to be used (may be opcode, operand, or address of operand). The 12-bit program counter is contained in PCL (low byte) and PCH (high nibble).

2.4.4. FLAGS (C, Z). The carry (C) bit is set on a carry or a borrow out of the ALU. It is cleared if the result of an arithmetic operation does not result in a carry or a borrow. The (C) bit is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

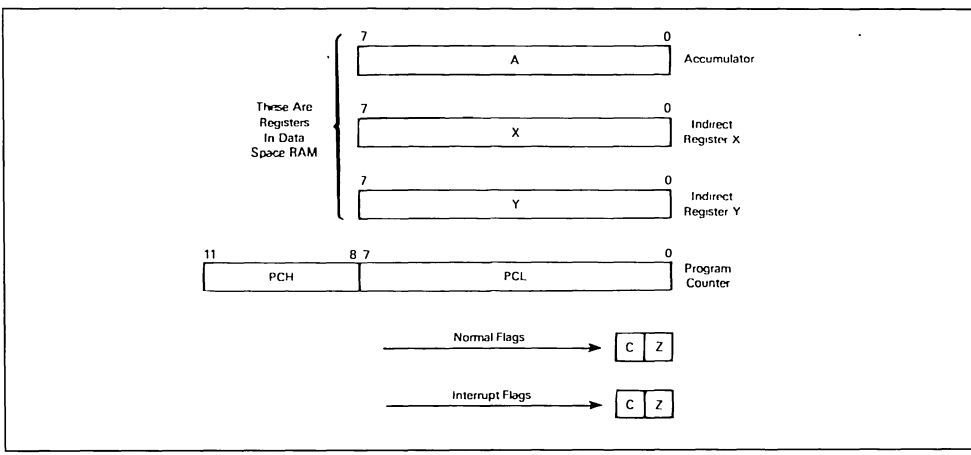
The zero (Z) bit is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

There are two sets of these flags, one set is for interrupt processing, the other for all other routines. When an interrupt occurs, a context switch is made from the program flags to the interrupt flags (interrupt mode). An RTI forces the context switch back to the program flags (program mode). While in either mode only the flags for that mode are available. Further, the interrupt flags will not be cleared upon entering the interrupt mode. Instead, the flags will be as they were at the exit of the last interrupt mode. Both sets of flags are cleared by reset.

2.4.5. STACK. There is a true LIFO stack incorporated in the EF6804J2 which eliminates the need for a stack pointer. Stack space is implemented in separate RAM (12-bits wide) shown in figure 2.1c. Whenever a subroutine call (or interrupt) occurs, the contents of the PC are shifted into the top register of the stack. At the same time (same cycle), the top register is shifted to the next level deeper. This happens to all registers with the bottom register falling out the bottom of the stack.

Whenever a subroutine or interrupt return occurs, the top register is shifted into the PC and all lower registers are shifted up one level higher. The stack RAM is four levels deep. If the stack is pulled more than four times without any pushes, the address that was stored in the bottom level will be shifted into the PC.

Figure 2.2 : Programming Model.



SECTION 3

TIMER

3.1. INTRODUCTION

A block diagram of the EF6804J2 timer circuitry is shown in figure 3.1. The timer logic in the MCU is comprised of a simple 8-bit counter (timer count register, TCR) with a 7-bit prescaler, and a timer status/control register (TSCR). The timer count register, which may be loaded under program control, is decremented towards zero by a clock input (prescaler output). The prescaler is used to extend the maximum interval of the overall timer. The prescaler tap is selected by bits 0-2 (PS0-PS2) of the timer status/control register. Bits PS0-PS2 control the actual division of the prescaler within the range of divide-by-1 (2^0) to divide-by-128 (2^7). The timer count register (TCR) and prescaler are decremented on rising clock edges. The coding of the TCSR PS0-PS2 bits produce a division in the prescaler as shown in table 3.1.

Table 3.1 : Prescaler Coding Table.

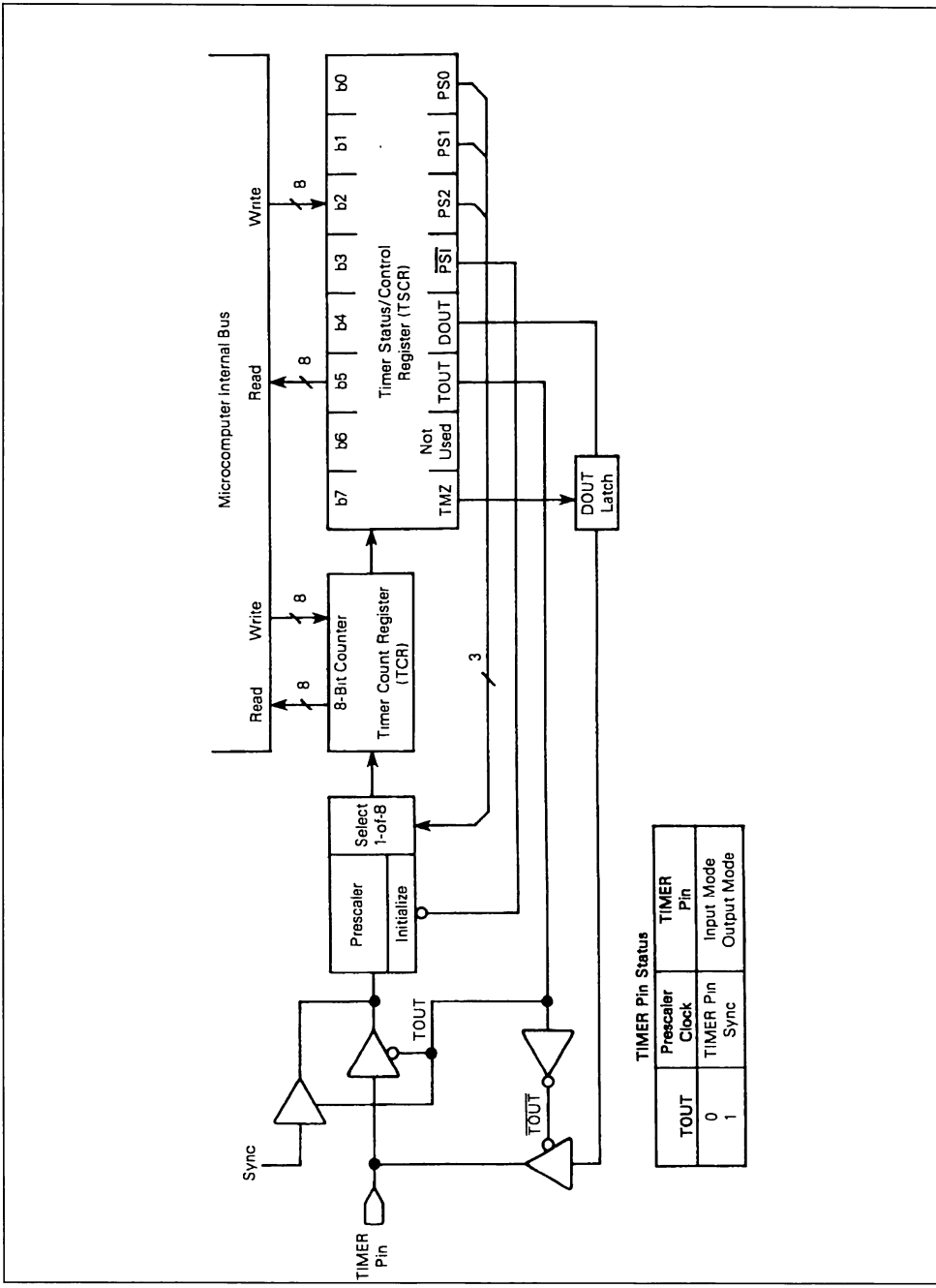
PS2	PS1	PS0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The TIMER pin may be programmed as either an input or an output depending on the status of TOUT (TSCR bit 5). Refer to figure 3.1. In the input mode, TOUT is a logic zero and the TIMER pin is connected directly to the prescaler input. Therefore, the timer prescaler is clocked by the signal applied from the TIMER pin. The prescaler then divides its clock input by a value determined by the coding of the TSCR bits PS0-PS2 as shown in table 3.1. The divided prescaler output then clocks the 8-bit timer count register (TCR). When the TCR is decremented to zero, it sets the TMZ bit in the timer status/control register (TSCR). The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The frequency of the external clock applied to the TIMER pin must be less than t_{byte} ($f_{osc}/48$).

In the output mode, TOUT is a logic one and the TIMER pin is connected to the DOUT latch. Therefore, the timer prescaler is clocked by the internal sync pulse (divide-by-48 of the internal oscillator). Operation is similar to that described above for the input mode. However, in the output mode, the low-to-high TMZ bit transition is used to latch the DOUT bit ofhte TSCR and provide it as output of the TIMER pin.

NOTE :
TMZ is normally set to logic one when the timer times out (TCR decrements to $00) ; however, it may be set by a write of $00 to the TCR or by a write to bit 7 of the TSCR.

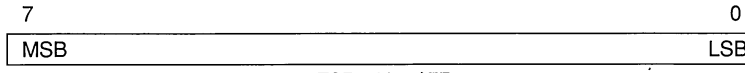
Figure 3.1 : Timer Block Diagram.



During reset, the timer count register and prescaler are set to $FF, while the timer status/control register is cleared to $00 and the DOUT LATCH (TIMER pin is in the high-impedance input mode) is forced to a logic high. The prescaler and timer count register are implemented in data space RAM locations ($FD, $FE) ; therefore, they are both readable and writeable. A write to either will predominate over the TCR decrement-to-$00 function ; i.e., if a write and a TCR decrement-to-$00 occur simultaneously, the write will take precedence, and the TMZ bit is not set until the next timer time out.

3.2. TIMER REGISTERS

3.2.1. TIMER COUNT REGISTER (TCR). The timer count register indicates the state of the internal 8-bit counter.



TCR addr = $FE

3.2.2. TIMER STATUS/CONTROL REGISTER (TSCR).

7	6	5	4	3	2	1	0
TMZ	Not Used	TOUT	DOUT	PSI	PS2	PS1	PS0

TSCR address = $09

b7, TMZ. Low-to-high transition indicates the timer count register has decremented to zero since the timer status/control register was last read. Cleared by a read of TSCR register if TMZ was read as a logic one.

b6. Not used.

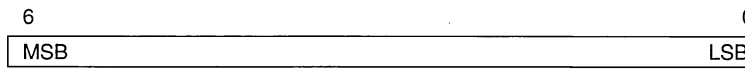
b5, TOUT. When low, this bit selects the input mode for the timer. When high, the output mode is selected.

b4, DOUT. Data sent to the timer output pin when TMZ is set high (output mode only).

b3, $\overline{PSI}$. Used to initialize the prescaler and inhibit its counting while $\overline{PSI}$ = 0. The initialized value is set to $FF. The timer count register will also be inhibited (contents unchanged). When PSI = 1 the prescaler begins to count downward.

b0, b1, b2, PS0-PS1-PS2. These bits are used to select the prescaler divide-by ratio ; therefore, effecting the clock input frequency to the timer count register.

3.2.3. TIMER PRESCALER REGISTER. The timer prescaler register indicates the state of the internal 7-bit prescaler. This 7-bit prescaler divide ratio is normally determined by bits PS0-PS2 of the timer status/control register (see table 3.1).



TPR Address = $FD

SECTION 4

INTERRUPT, SELF–TEST, RESET AND INTERNAL CLOCK GENERATOR

4.1. INTERRUPT

The EF6804J2 can be interrupted by applying a logic low signal to the $\overline{IRQ}$ pin ; however, a mask option selected at the time of manufacture determines whether the negative-going edge or the actual low level is sensed to indicate an interrupt.

4.1.1. EDGE-SENSITIVE OPTION. When the $\overline{IRQ}$ pin is pulled low, the internal interrupt request latch is set. Prior to each instruction fetch, the interrupt request latch is tested and, if its output is high, an interrupt sequence is initiated at the end of the current instruction (provided the interrupt mask is cleared). Figure 4.1 contains a flowchart which illustrates both the reset and interrupt sequence. The interrupt sequence consists of one cycle during which : the interrupt request latch is cleared, the interrupt mode flags are selected, the PC is saved on the stack, the interrupt mask is set, and the $\overline{IRQ}$ vector (single chip mode = $FFC/$FFD, self-test mode = $FF8/$FF9) is loaded into the PC. Internal processing of the interrupt continues until an RTI (return from interrupt) instruction is processed. During the RTI instruction, the interrupt mask is cleared and the program mode flags are selected. The next instruction of the program is then fetched and executed. Once the interrupt was initially detected and the interrupt sequence started, the interrupt request latch is cleared so that the next (second) interrupt may be detected even while the previous (first) one is being serviced. However, even though the second interrupt sets the interrupt request latch during processing of the first interrupt, the second interrupt sequence will not be initiated until completion of the interrupt service routine for the first interrupt. Completion of an interrupt service routine is always accomplished using an RTI instruction to return to the main program. The interrupt mask (which is not directly available to the programmer) is cleared during the last cycle of the RTI instruction.

4.1.2. LEVEL-SENSITIVE OPTION. The actual operation of the level-sensitive and edge-sensitive options are similar except that the level-sensitive option does not have an interrupt request latch. With no interrupt request latch, the logic level of the $\overline{IRQ}$ pin is checked for detection of the interrupt. Also, in the interrupt sequence, there is no need to clear the interrupt request latch. These differences are illustrated in the flowchart of figure 4.1.

4.1.3. POWER UP AND TIMING. During the power-up sequence the interrupt mask is set to preclude any false or "ghost" interrupts from occurring. To clear the interrupt mask, the programmer should write a JSR (instead of a JMP) instruction to an initialization routine as the first instruction in a program. The initialization routine should end with an RTI (instead of RTS). Maximum interrupt response time is eight machine (t_{byte}) cycles (see 4.4. Internal Clock Generator Options). This includes five machine cycles for the longest instruction, plus one machine cycle for stacking the PC and switching flags, plus two machine cycles for synchronization of the $\overline{IRQ}$ input with the internal clock. Minimum response time is one machine cycle for stacking PC and switching flags (see 2.4.4. flags (C, Z)).

4.2. SELF-TEST

The EF6804J2 MCU has a unique internal ROM-based off-line self-test capability using signature analysis techniques. A test program stored in the on-chip ROM is initiated by configuring pins PA6 and PA7 during reset. The test results are sampled on a cycle-by-cycle basis by a 16-bit on-chip signature analysis register configured as a linear feedback shift register (LFSR) using the standard CCITT CRC16 polynomial. A schematic diagram of the self-test connections is shown in figure 4.2. To perform a test of the MCU, connect it as shown in figure 4.2a and monitor the LEDs for a 1101 ($D) pattern.

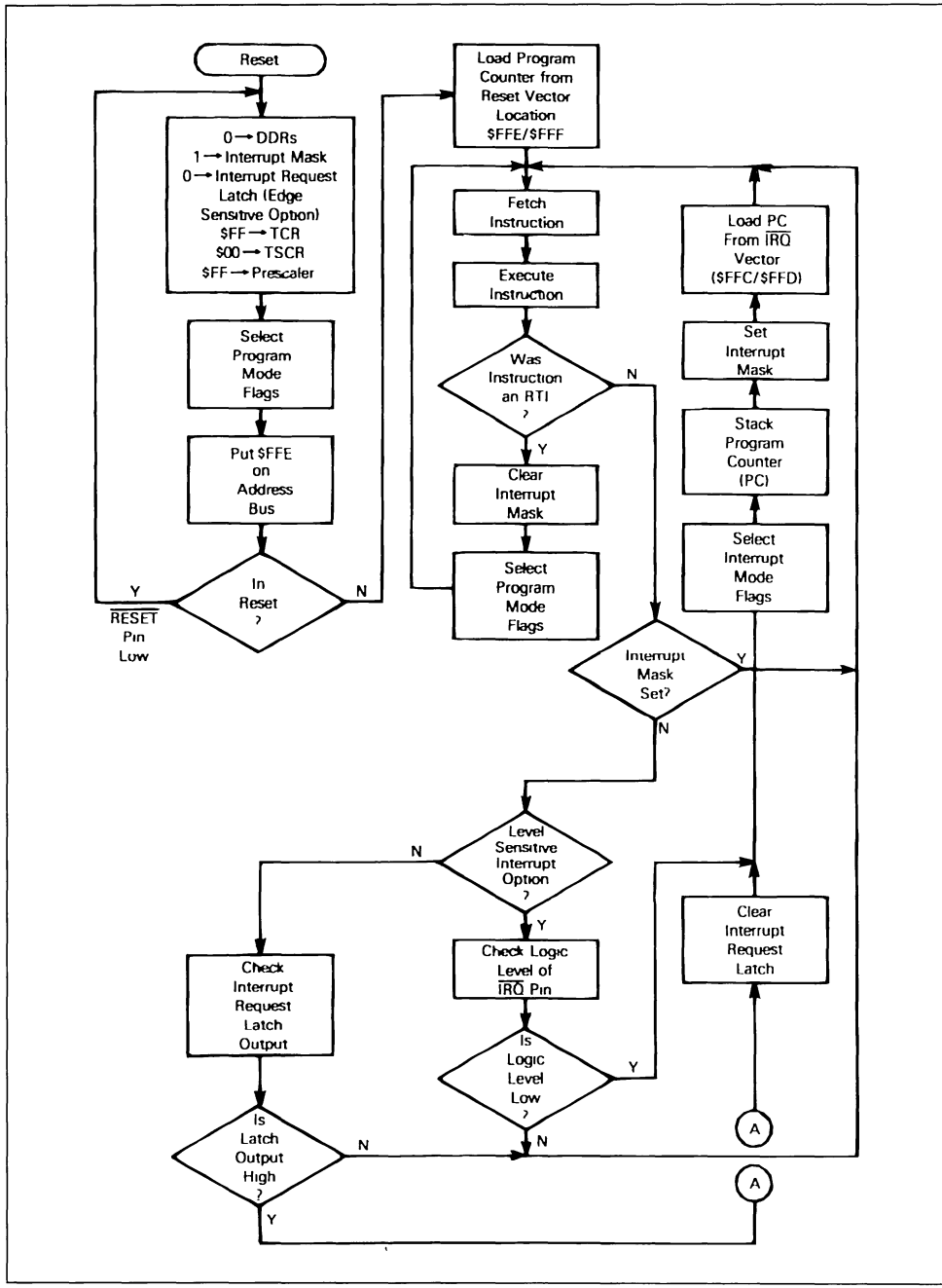
A special ROM self-test utilizing the signature analysis circuitry is also included. To initiate a test of the ROM, connect the circuit as shown in figure 4.2b. This mode also uses the on-chip signature analysis register to verify the contents of the custom ROM by monitoring an internal bus. The "Good" LED indicates that all ROM words have been read and that the result was the correct signature.

The on-chip self-test and the ROM test are the basis of SGS-THOMSON Microelectronics production testing for the EF6804J2. These tests have been fault graded using statistical methods and have been found to provide high fault coverage using automatic test equipment (ATE) or the circuit of figure 4.2.

4.3. RESET

The MCU can be reset in two ways : by initial power up (see figure 4.1) and by the external reset input ($\overline{RESET}$). During power up, a delay of t_{RHL} is needed before allowing the $\overline{RESET}$ input to go high.

Figure 4.1 : Reset and Interrupt Processing Flowchart.



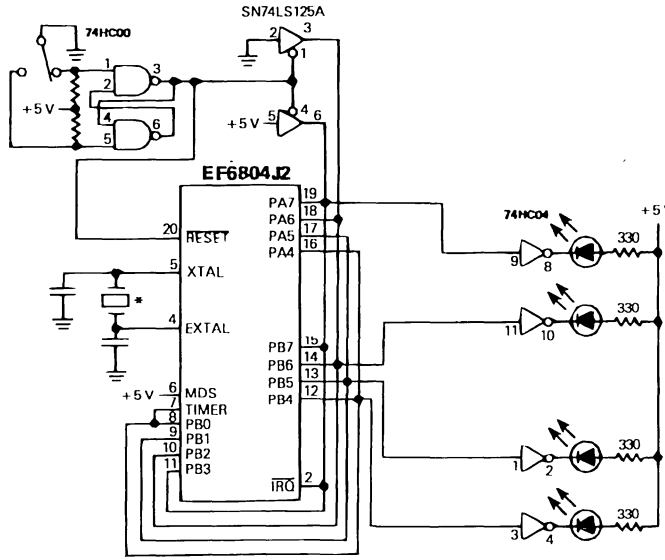
This time delay allows the internal clock generator to stabilize. Connecting a capacitor and resistor to the RESET input, as shown in figure 4.3, typically provides sufficient delay.

4.4. INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor-capacitor, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different clock generator option connection methods are shown in figure 4.4, crystal specifications and suggested PC board layouts are given in figure 4.5, resistor-capacitor selection graph is given in figure 4.6, and a timing diagram is illustrated in figure 4.7. The crystal oscillator startup time is a function of many variables : crystal parameters (especially R_S), oscillator load capacitance (C_L), IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

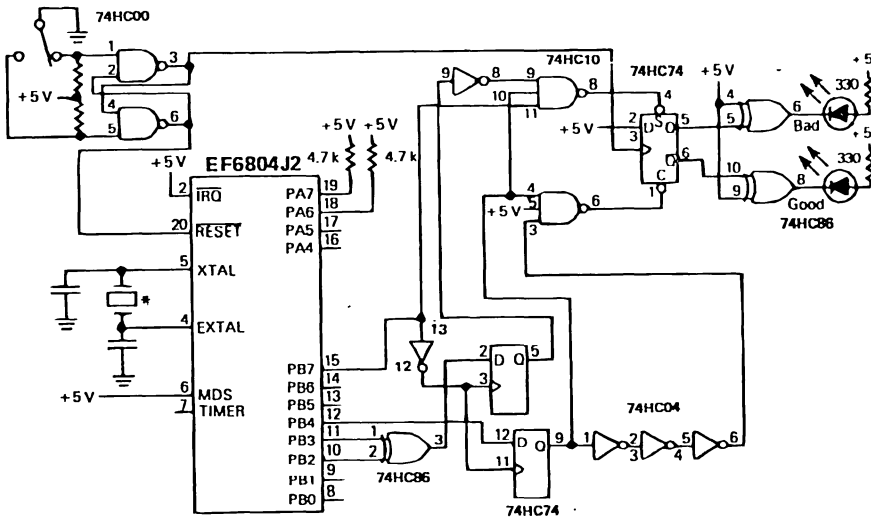
The oscillator output frequency is internally divided by four to produce the internal $\phi 1$ and $\phi 2$ clocks. The $\phi 1$ clock is divided by twelve to produce a machine byte (cycle) clock. A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to execute.

Figure 4.2 : Self–test Circuit.



* Used with crystal option only.

(a) Functional Test



* Used with crystal option only

(b) Simple ROM Verify Test

Figure 4.3 : Power–up Reset Delay Circuit.

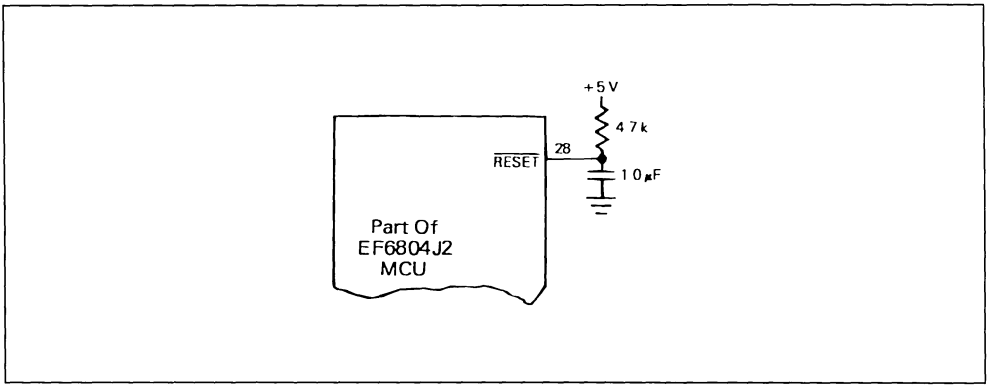


Figure 4.4 : Clock Generator Options.

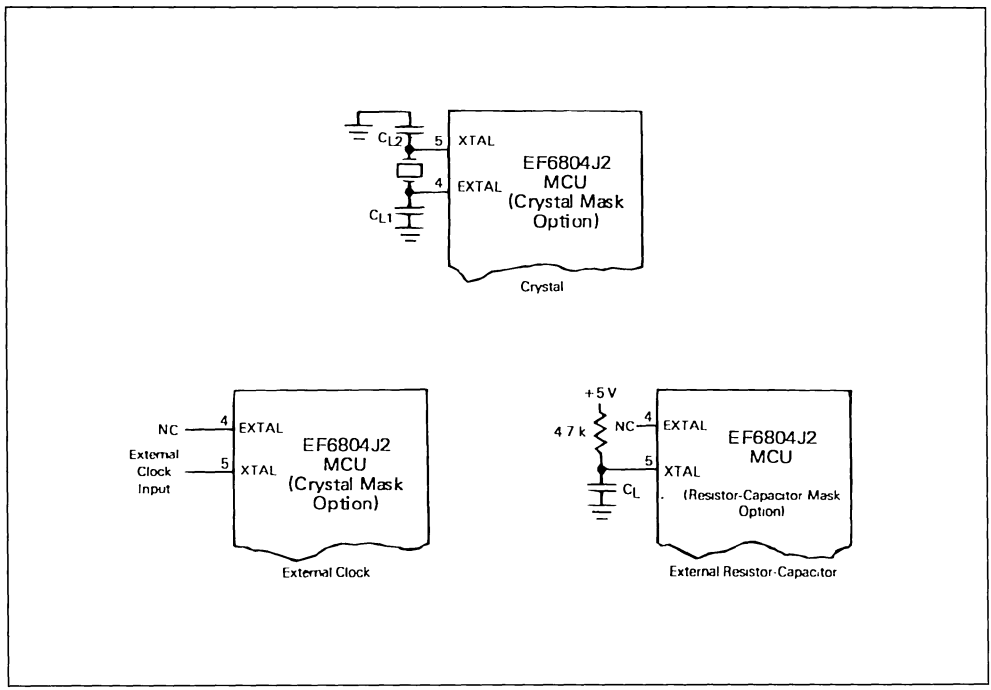
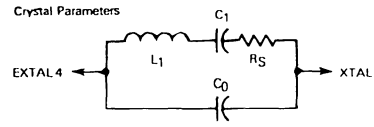
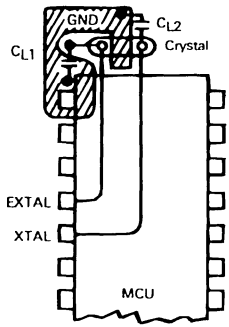
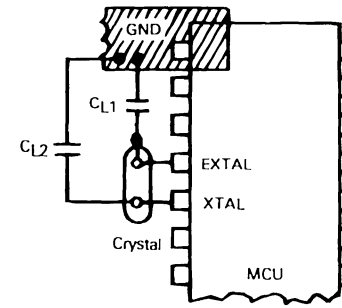


Figure 4.5 : Crystal Motional Arm Parameters and Suggested PC Board Layout.



Crystal Parameters
AT – Cut Parallel Resonance Crystal
C_0 = 7 pF Maximum
Freq = 11MHz
R_S = 50Ω Maximum
Piezoelectric ceramic resonators which have the equivalent specifications may be used instead of crystal oscillators Follow ceramic resonator manufacturer's suggestions for C_0, C_1, and R_S values.





Note · Keep crystal leads and circuit connections as short as possible

Figure 4.6 : Typical Frequency Selection for Resistor–capacitor Oscillator Option (C_L = 17pF).

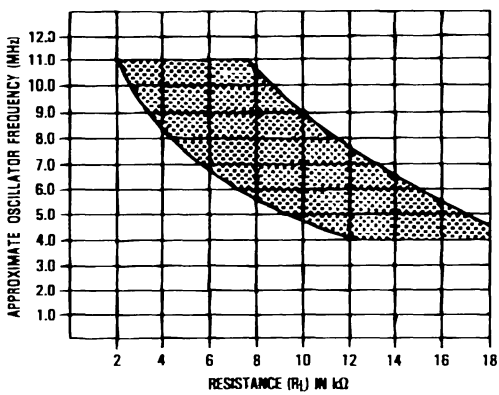
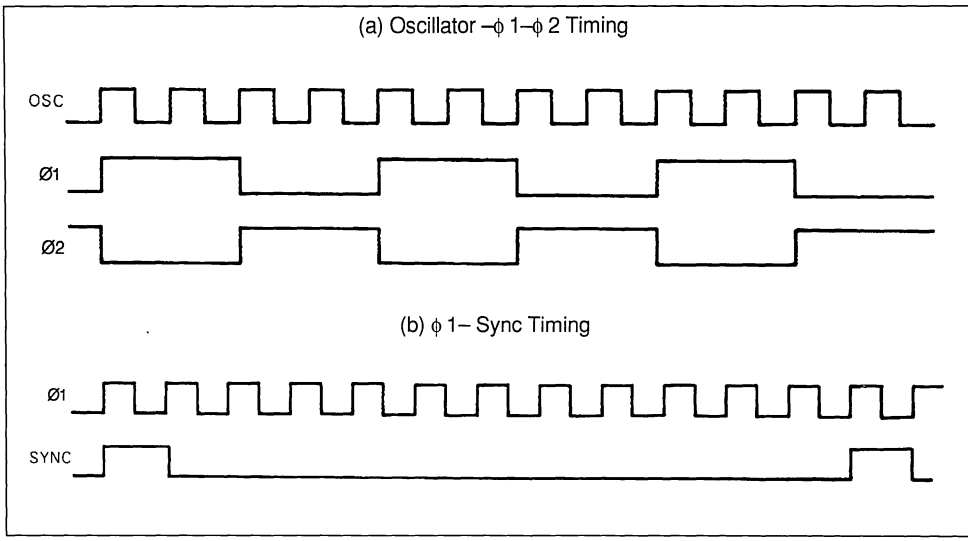


Figure 4.7 : Clock Generator Timing Diagram.



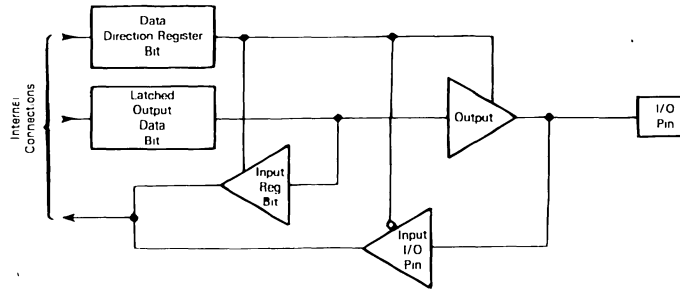
SECTION 5

INPUT/OUTPUT PORTS

5.1. INPUT/OUTPUT

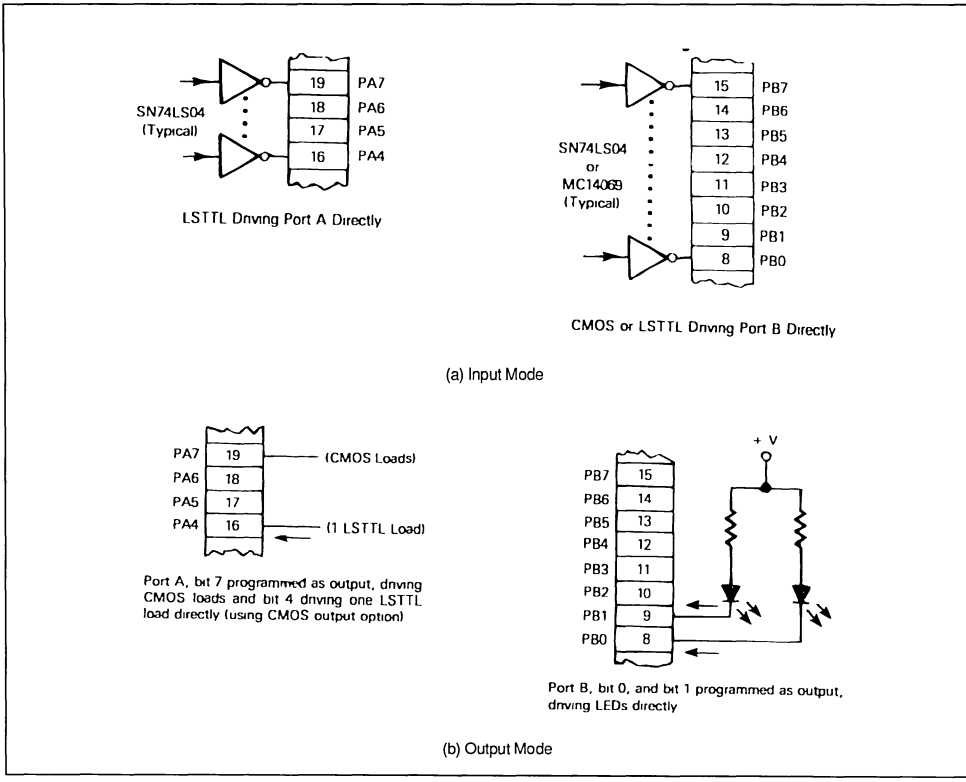
There are 12 input/output pins. All pins (port A and B) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset, all the DDRs are initialized to a logic zero state to put the ports in the input mode. The port output registers are not initialized on reset but should be initialized before changing the DDR bits to avoid undefined levels. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading ; see figure 5.1. All input/output pins are LSTTL compatible as both inputs and outputs. In addition, both ports may have one of two mask options : 1) internal pullup resistor for CMOS output compatibility, or 2) open drain output. The address map in figure 2.1 gives the address of data registers and DDRs. The register configuration is discussed under the registers paragraph below and figure 5.2 provides some examples of port connections.

Figure 5.1 : Typical I/O Port Circuitry.



Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z	Pin

Figure 5.2 : Typical Port Connections.



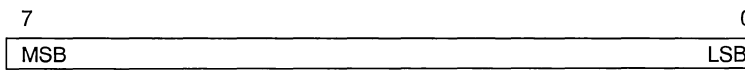
The latched output data bit (see figure 5.1) may always be written. Therefore, any write to a port writes to all of its data bits even though the port DDR is set to input. This may be used to initialize the data registers and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and corresponds to the latched output data when the DDR is an output (1). The 12 bidirectional lines may be configured by port to be LSTTL (standard configuration), LSTTL/CMOS (mask option), or open drain (mask option). Port B outputs are LED compatible.

NOTE
The mask option only allows changes by port. For example, if the customer wishes PA7 to be open drain, then PA4-PA7 must all be open drain.

5.2. REGISTERS

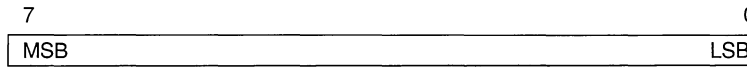
The registers described below are implemented as RAM locations and thus may be read or written.

5.2.1. PORT DATA REGISTER. The source of data read from the port data register will be the port I/O pin or previously latched output data depending upon the contents of the corresponding data direction register (DDR). The destination of data written to the port data register will be an output data latch. If the corresponding data direction register (DDR) for the port I/O pin is programmed as an output, the data will then appear on the port pin.



Port A Address = $00
Port B Address = $01

5.2.2. PORT DATA DIRECTION REGISTER. The port DDRs configure the port pins as either inputs or outputs. Each port pin can be programmed individually to act as an input or an output. A zero in the pins corresponding bit position will program that pin as an input while a one in the pins corresponding bit position will program that pin as an output.



Port A Address = $04
Port B Address = $05

SECTION 6

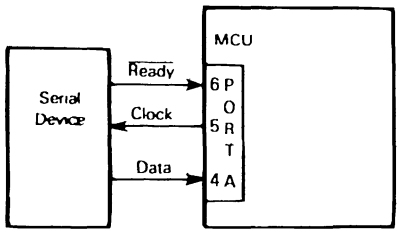
SOFTWARE AND INSTRUCTION SET

6.1. SOFTWARE

6.1.1. BIT MANIPULATION. The EF6804J2 MCU has the ability to set or clear any register or single random access memory (RAM) writable bit with a single instruction (BSET, BCLR). Any bit in data space, including ROM, can be tested, using the BRSET and BRCLR instructions, and the program may branch as a result of its state. The carry bit is set to the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in figure 6.1 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line (to clock data one bit at a time, MSB first, out of the device). The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in the accumulator.

Figure 6.1 : Bit Manipulation Example.




```
                .
SELF    BRSET   6, PORTA, SELF
                .
                .
        BSET    5, PORTA
        BRCLR   4, PORTA, CONT
CONT    BCLR    5, PORTA
        ROLA    .
                .
                .
```

6.1.2. ADDRESSING MODES. The EF6804J2 MCU has nine addressing modes which are explained briefly in the following paragraphs. The EF6804J2 deals with objects in three different address spaces : program space, data space, and stack space. Program sapce contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains all of the RAM lcoatin, X and Y registers, accumulator, timer, I/O locations,and some ROM (for storage of tables and constants). Stack space contains RAM for use in stacking the return addresses for subroutines and interrupts.

The term "Effective Address" (EA) is used in describing the address modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

6.1.2.1. Immediate

In the immediate addressing mode, the operand is located in program ROM and is contained in a byte following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

6.1.2.2. Direct

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the 256 bytes in data space memory with a single two-byte instruction.

6.1.2.3. Short Direct

The MCU also has four locations in data space RAM ($80, $81, $82, $83) which may be used in a short-direct addressing mode. In this mode the lower two bits of the opcode determine the data space. RAM location, and the instruction is only one byte. Short direct addressing is a subset of the direct addressing mode. (The X and Y registers are at locations $80 or $81 respectively).

6.1.2.4. Extended

In the extended addressing mode, the effective address is obtained by concatenating the four least significant bits of the opcode with the byte following the opcode (12-bit address). Instructions using the extended addressing mode (JMP, JSR) are capable of branching anywhere in program space. An extended addressing mode instruction is two bytes long.

6.1.2.5. Relative

The relative addressing mode is only used in conditional branch instructions. In relative addressing, the address is formed by adding the sign extended lower five bits of the opcode (the offset) to the program counter if and only if the condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from - 15 to + 16 from the opcode address. The programmer need not worry about calculating the correct offset when using the Motorola assembler since it calculates the proper offset and checks to see if it is within the span of the branch.

6.1.2.6. Bit Set/Clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any bit in the 256 locations of data space memory, which can be written to, can be set or cleared.

6.1.2.7. Bit Test and Branch

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit which is to be tested is included in the opcode, and the data space address of the byte to be tested is in the single byte immediately following the opcode byte. The third byte is sign extended to twelve bits and becomes the offset added to the program counter if the condition is true. The single three-byte instruction allows the program to branch based on the condition of any bit in data space memory. The span of branching is from - 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry flag.

6.1.2.8. Register-indirect

In the register-indirect addressing mode, the operand is at the address (in data space) pointed to by the contents of one of the indirect registers (X or Y). The particular X or Y register is selected by bit 4 of the opcode. Bit 4 of the opcode is then decoded into an address which selects the desired X or Y register ($80 or $81). A register-indirect instruction is one byte long.

6.1.2.9. Inherent

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

6.2. INSTRUCTION SET

The EF6804J2 MCU has a set of 42 basic instructions, which when combined with nine addressing modes produce 242 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

6.2.1. REGISTER/MEMORY INSTRUCTIONS. Most of these instructions use two operands. One operand is the accumulator and the other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operands. Refer to table 6.1.

6.2.2. READ-MODIFY-WRITE INSTRUCTIONS. These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. There are ten instructions which utilize read-modify-write cycles. All INC and DEC forms along with all bit manipulation instructions use this method. Refer to table 6.2.

6.2.3. BRANCH INSTRUCTIONS. The branch instructions cause a branch from the program when a certain condition is met. Refer to table 6.3.

6.2.4. BIT MANIPULATION INSTRUCTIONS. These instructions are used on any bit in data space memory. One group either sets or clears. The other group performs the bit test branch operations. Refer to table 6.4.

Table 6.1 : Register/memory Instructions.

		Addressing Modes																			
		Indirect				Immediate			Direct			Inherent			Extended			Short-Direct			
Function	Mnem	Opcode X	Opcode Y	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Load A from Memory	LDA	E0	F0	1	4	E8	2	4	F8	2	4	–	–	–	–	–	–	AC-AF	1	4	1
Load XP from Memory	LDXI	–	–	–	–	B0	3	4	–	–	–	–	–	–	–	–	–	–	–	–	4
Load YP from Memory	LDYI	–	–	–	–	B0	3	4	–	–	–	–	–	–	–	–	–	–	–	–	4
Store A in Memory	STA	E1	F1	1	4	–	–	–	F9	2	4	–	–	–	–	–	–	BC-BF	1	4	2
Add to A	ADD	E2	F2	1	4	EA	2	4	FA	2	4	–	–	–	–	–	–	–	–	–	–
Subtract from A	SUB	E3	F3	1	4	EB	2	4	FB	2	4	–	–	–	–	–	–	–	–	–	–
Arithmetic Compare with Memory	CMP	E4	F4	1	4	EC	2	4	FC	2	4	–	–	–	–	–	–	–	–	–	–
AND Memory to A	AND	E5	F5	1	4	ED	2	4	FD	2	4	–	–	–	–	–	–	–	–	–	–
Jump to Subroutine	JSR	–	–	–	–	–	–	–	–	–	–	–	–	–	8 (TAR)	2	4	–	–	–	3
Jump Unconditional	JMP	–	–	–	–	–	–	–	–	–	–	–	–	–	9 (TAR)	2	4	–	–	–	3
Clear A	CLRA	–	–	–	–	–	–	–	FB	2	4	–	–	–	–	–	–	–	–	–	–
Clear XP	CLRX	–	–	–	–	–	–	–	FB	2	4	–	–	–	–	–	–	–	–	–	–
Clear YP	CLRY	–	–	–	–	–	–	–	FB	2	4	–			–	–	–	–	–	–	–
Complement A	COMA	–	–	–	–	–	–	–	–	–	–	B4	1	4	–	–	–	–	–	–	–
Move Immediate Value to Memory	MVI	–	–	–	–	B0	3	4	B0	3	4	–	–	–	–	–	–	–	–	–	5
Rotate A Left and Carry	ROLA	–	–	–	–	–	–	–	–	–	–	B6	1	4	–	–	–	–	–	–	–
Arithmetic Left Shift of A	ASLA	–	–	–	–	–	–	–	FA	2	4	–	–	–	–	–	–	–	–	–	–

SPECIAL NOTES

1. In Short-Direct addressing, the LDA mnemonic represents opcode AC, AD, AE, and AF. This is equivalent to RAM locations $80 (AC), $81 (AD), $82 (AE), and $83 (AF)
2. In Short-Direct addressing, the STA mnemonic represents opcode BC BD, BE, and BF. This is equivalent to RAM locations $80 (BC), $81 (BD), $82 (BE), and $83 (BF).
3. In Extended addressing, the four LSBs of the opcode (Mnemonic JSR and JMP) are formed by the four MSBs of the target address. (TAR)
4. In Immediate addressing, the LDXI and LDYI are mnemonics which are recognized as follows:
   LDXI = MVI $80,data
   LDYI = MVI $81,data Where data is a one-byte hexadecimal number.
5. The MVI instruction refers to both Immediate and Direct addressing.

Table 6.2 : Read-Modify-Write Instructions.

Function	Mnem	Addressing Modes										Special Notes
		Indirect				Direct			Short-Direct			
		Opcode		# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	
		X	Y									
Increment Memory Location	INC	E6	F6	1	4	FE	2	4	A8-AB	1	4	1, 3
Increment A	INCA					FE	2	4				
Increment XP	INCX								A8	1	4	
Increment YP	INCY								A9	1	4	
Decrement Memory Location	DEC	E7	F'	1	4	FF	2	4	B8-BB	1	4	2, 4
Decrement A	DECA					FF	2	4				
Decrement XP	DECX								B8	1	4	
Decrement YP	DECY								B9	1	4	

SPECIAL NOTES

1. In short-direct addressing, the INC mnemonic represents opcode A8, A9, AA and AB. These are equivalent to RAM locations $80 (A8), $81 (A9), $82 (AA) and $83 (AB).
2. In short-direct addressing, the DEC mnemonic represents opcode B8, B9, BA and BB. These are equivalent to RAM locations $80 (B8), $81 (B9), $82 (BA) and $83 (BB).
3. In indirect addressing, the INC mnemonic represents opcode E6 or F6, and causes the location pointed to by X (E6 opcode) or Y (F6 opcode) to be incremented.
4. In indirect addressing, the INC mnemonic represents opcode E7 or F7, and causes the location pointed to by X (E7 opcode) or Y (F7 opcode) to be incremented.

Table 6.3 : Branch Instructions.

		Relative Addressing Mode			
Function	**Mnem**	**Opcode**	**# Bytes**	**# Cycles**	**Special Notes**
Branch if Carry Clear	BCC	40-5F	1	2	1
Branch if Higher or Same	BHS	40-5F	1	2	1, 2
Branch if Carry Set	BCS	60-7F	1	2	1
Branch if Lower	BLO	60-7F	1	2	1, 3
Branch if Not Equal	BNE	00-1F	1	2	1
Branch if Equal	BEQ	20-3F	1	2	1

SPECIAL NOTES
1 Each mnemonic of the Branch Instructions covers a range of 32 opcodes, e g , BCC ranges from 40 through 5F The actual memory location (target address) to which the branch is made is formed by adding the sign extended lower five bits of the opcode to the contents of the program counter.
2 The BHS instruction (shown in parentheses) is identical to the BCC instruction. The C bit is clear if the register was higher or the same as the location in the memory to which it was compared
3 The BLO instruction (shown in parentheses) is identical to the BCS instruction. The C bit is set if the register was lower than the location in memory to which it was compared

Table 6.4 : Bit Manipulation Instructions.

		Addressing Modes						
		Bit Set/Clear			Bit Test and Branch			
Function	**Mnem**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Special Notes**
Branch IFF Bit n is set	BRSET n (n = 0 7)				C8 + n	3	5	1
Branch IFF Bit n is clear	BRCLR n (n = 0 7)				C0 + n	3	5	1
Set Bit n	BSET n (n = 0 7)	D8 + n	2	4				1
Clear Bit n	BCLR n (n = 0 7)	D0 + n	2	4				1

SPECIAL NOTE
1 The opcode is formed by adding the bit number (0-7) to the basic opcode. For example to clear bit six using the BSET6 instruction the opcode becomes DE (D8 + 6), BCLR5 becomes (C0 + 5), etc

6.2.5. CONTROL INSTRUCTIONS. The control instructions control the MCU operations during program execution. Refer to table 6.5.

6.2.6. ALPHABETICAL LISTING. The complete instruction set is given in alphabetical order in table 6.6. There are certain mnemonics recognized by the assembler and converted to other instructions. The fact that all registers and accumulator are in RAM allows many implied instructions to exist. The implied instructions recognized by the assembler are identified in table 6.6.

6.2.7. OPCODE MAP SUMMARY. Table 6.7 contains an opcode map for the instructions used on the MCU.

6.3. IMPLIED INSTRUCTIONS

Since the accumulator and all other registers are located in RAM many implied instructions exist. The assembler-recognized implied instructions are given in table 6.6. Some examples not recognized by the assembler are shown below.

BCLR, 7 $FF	Ensures accumulator is plus
BSET, 7 $FF	Ensures accumulator is minus
BRCLR, 7 $FF	Branch iff accumulator is plus
BRSET, 7 $FF	Branch iff accumulator is minus
BRCLR, 7 $80	Branch iff X is plus (BXPL)
BRSET, 7 $80	Bracnh iff X is minus (BXMI)
BRCLR, 7 $81	Branch iff Y is plus (BYPL)
BRSET, 7 $81	Branch iff Y is minus (BYMI)

Table 6.5 : Control Instructions.

		Addressing Modes									
		Short-Direct			Inherent			Relative			Special Notes
Function	Mnem	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	
Transfer A to X	TAX	BC	1	4							
Transfer A to Y	TAY	BD	1	4							
Transfer X to A	TXA	AC	1	4							
Transfer Y to A	TYA	AD	1	4							
Return from Subroutine	RTS				B3	1	2				
Return from Interrupt	RTI				B2	1	2				
No-operation	NOP										1

SPECIAL NOTE

1. The NOP instruction is equivalent to a branch if equal (BEQ) to the location designated by PC + 1.

Table 6.6 : Instruction Set.

Mnemonic	Addressing Modes									Flags	
	Inherent	Immediate	Direct	Short Direct	Bit/Set Clear	Bit-Test-Branch	Register Indirect	Extended	Relative	Z	C
ADD		X	X				X			^	^
AND		X	X				X			^	•
ASLA	Assembler converts this to "ADD $FF"									•	•
BCC									X	^	^
BCLR					X					•	•
BCS									X	•	•
BEQ									X	•	•
BHS	Assembler converts this to "BCC"									•	•
BLO	Assembler converts this to "BCS"									•	•
BNE									X	•	•
BRCLR						X				•	^
BRSET						X				•	^
BSET					X					•	•
CLRA	Assembler converts this to "SUB $FF"									^	^
CLRX	Assembler converts this to "MVI = 0, $80"									•	•
CLRY	Assembler converts this to "MVI = 0, $81"									•	•
CMP		X	X				X			^	^
COMA	X									^	^
DEC			X	X			X			^	•
DECA	Assembler converts this to "DEC $FF"									^	•
DECX	Assembler converts this to "DEC $80"									^	•
DECY	Assembler converts this to "DEC $81"									^	•
INC			X	X			X			^	•
INCA	Assembler converts this to "INC $FF"									^	•
INCX	Assembler converts this to "INC $80"									^	•
INCY	Assembler converts this to "INC $81"									^	•
JMP								X		•	•
JSR								X		•	•
LDA		X	X	X			X			^	•
LDXI	Assembler converts this to "MVI DATA, $80"									•	•
LDYI	Assembler converts this to "MVI DATA, $81"									•	•
MVI		X	X							•	•
NOP	Assembler converts this to "BEQ (PC) + 1"									•	•
ROLA	X									^	^
RTI	X									^	^
RTS	X									•	•
STA			X	X			X			^	•
SUB		X	X				X			^	^
TAX	Assembler converts this to "STA $80"									^	•
TAY	Assembler converts this to "STA $81"									^	•
TXA	Assembler converts this to "LDA $80"									^	•
TYA	Assembler converts this to "LDA $81"									^	•

Flag Symbols Z = Zero, C = Carry/borrow, ^ = Test and Set if True, Cleared Otherwise, • = Not affected

Table 6.7 : EF6804P2 Microcomputer Instruction Set Opcode Map.

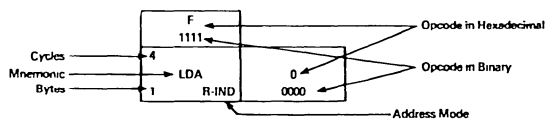
	Branch Instructions							
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
1 0001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
2 0010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
3 0011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
4 0100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
5 0101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
6 0110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
7 0111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
8 1000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
9 1001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
A 1010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
B 1011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
C 1100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
D 1101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
E 1110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
F 1111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL

Abbreviations for Address Modes

INH	Inherent
S-D	Short Direct
B-T-B	Bit Test and Branch
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
R-IND	Register Indirect
*	Indicates Instruction Reserved for Future Use
#	Indicates Illegal Instruction

Table 6.7 : (continued).

Register/Memory, Control, and Read/Modify/Write Instructions				Bit Manipulation Instructions		Register/Memory and Read/Modify/Write		
8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 MVI 3 IMM	5 BRCLR0 3 B-T-B	5 BCLR0 3 BSC	4 LDA 1 R-IND	4 LDA 1 R-IND	0 0000
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR1 3 B-T-B	5 BCLR1 3 BSC	4 STA 1 R-IND	4 STA 1 R-IND	0 0001
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTI 1 INH	5 BRCLR2 3 B-T-B	5 BCLR2 3 BSC	4 ADD 1 R-IND	4 ADD 1 R-IND	2 0010
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTS 1 INH	5 BRCLR3 3 B-T-B	5 BCLR2 3 BSC	4 SUB 1 R-IND	4 SUB 1 R-IND	3 0011
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 COMA 1 INH	5 BRCLR4 3 B-T-B	5 BCLR4 3 BSC	4 CMP 1 R-IND	4 CMP 1 R-IND	4 0100
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 ROLA 1 INH	5 BRCLR5 3 B-T-B	5 BCLR5 3 BSC	4 AND 1 R-IND	4 AND 1 R-IND	5 0101
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR6 3 B-T-B	5 BCLR6 3 BSC	4 INC 1 R-IND	4 INC 1 R-IND	6 0110
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR7 3 B-T-B	5 BCLR7 3 BSC	4 DEC 1 R-IND	4 DEC 1 R-IND	7 0111
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET0 3 B-T-B	5 BSET0 3 BSC	4 LDA 2 IMM	4 LDA 2 DIR	8 1000
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET1 3 B-T-B	5 BSET1 3 BSC	#	4 STA 2 DIR	9 1001
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET2 3 B-T-B	5 BSET2 3 BSC	4 ADD 2 IMM	4 ADD 1 DIR	A 1010
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET3 3 B-T-B	5 BSET3 3 BSC	4 SUB 2 IMM	4 SUB 2 DIR	B 1011
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET4 3 B-T-B	5 BSET4 3 BSC	4 CMP 2 IMM	4 CMP 2 DIR	C 1100
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET5 3 B-T-B	5 BSET5 3 BSC	4 AND 2 IMM	4 AND 2 DIR	D 1101
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET6 3 B-T-B	5 BSET6 3 BSC	#	4 INC 2 DIR	E 1110
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET7 3 B-T-B	5 BSET7 3 BSC	#	4 DEC 2 DIR	F 1111



SECTION 7

ELECTRICAL SPECIFICATIONS

7.1. INTRODUCTION

This section contains the electrical specifications and associated timing for the EF6804J2.

7.2. ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage	– 0.3 to + 7.0	V
T_A	Operating Temperature Range Standard or L Suffix V Suffix T Suffix	TL to TH 0 to + 70 – 40 to + 85 – 40 to + 105	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_J	Junction Temperature Range Plastic	150	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages of electric fields , however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit For proper operation it is recommended that V_{in} and V_{out} be constrained to the range V_{SS} (V_{in} or V_{out}) V_{CC} Reliability of operation is enhanced if unused input except EXTAL are connected to an appropriate logic voltage level (e g., either V_{SS} or V_{CC})

7.3. THERMAL DATA

θ_{JA}	Thermal Resistance Plastic	90	°C/W

Figure 7.3: LSTTL Equivalent Test Load (port A and TIMER).

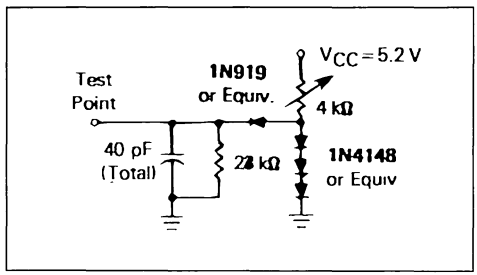


Figure 7.2 : CMOS Equivalent Test Load (port A and B).

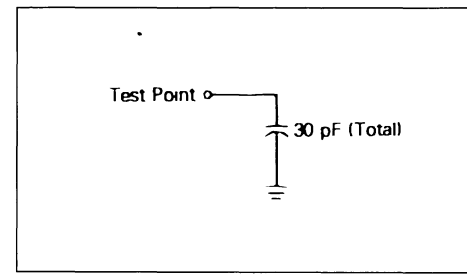
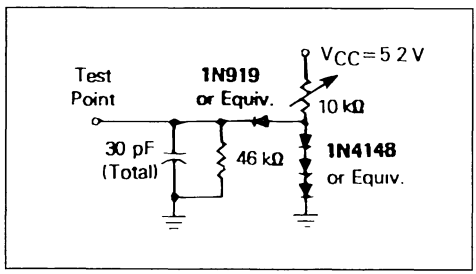


Figure 7.1 : LSTTL Equivalent Test Load (port B).



7.4. POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$T_J = T_A + (P_D \cdot \theta JA)$ (1)

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} << P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$P_D = K \div (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives :

$K = PD \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2$ (3)

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) interatively for any value of T_A.

7.5. ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.0 Vdc ± 0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
P_{INT}	Internal Power Dissipation–No Port Loading T_A = 0°C T_A = – 40°C		135	170 210	mV
V_{IH}	Input High Voltage	4.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
C_{in}	Input Capacitance		10		pF
I_{in}	Input Current($\overline{IRQ}$, $\overline{RESET}$)		2	20	µA

7.6. ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.0 Vdc ± 0.5 Vdc, V_{SS} = GND, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{OSC}	Oscillator Frequency	4.0		11.0	MHz
t_{bit}	Bit Time	0.364		1.0	µs
t_{byte}	Byte Cycle Time	4.36		12.0	µs
t_{WL}, t_{WH}	$\overline{IRQ}$ and $\overline{TIMER}$ Pulse Width	2 x t_{byte}			
t_{RWL}	$\overline{RESET}$ Pulse Width	2 x t_{byte}			
t_{RHL}	$\overline{RESET}$ Delay Time (external capacitance = 1.0µF)	100			ms

7.7. PORT DC ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.0 Vdc ± 0.5 Vdc, V_{SS} = GND, T_A = T_L to T_H unless otherwise noted)

TIMER AND PORTS A (standard)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA			0.5	V
V_{OH}	Output High Voltage, I_{Load} = – 50µA	2.3			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		4	40	µA

TIMER AND PORTS A (open drain)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA			0.5	V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		4	40	µA
I_{LOD}	Open Drain Leakage (V_{out} = V_{CC})		4	40	µA

TIMER AND PORTS A (CMOS drive)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA (sink)			0.5	V
V_{OH}	Output High Voltage, I_{Load} = – 10µA	V_{CC} – 1.0			V
V_{OH}	Output High Voltage, I_{Load} = – 50µA	2.3			V
V_{IH}	Input High Voltage, I_{Load} = – 300µA Max	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 300µA Max	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current (V_{in} = 0.4V to V_{CC})			– 300	µA

PORT B (standard)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.3			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{ISS}		0.8	V
I_{TSI}	Hi-Z State Input Current		8	80	µA

PORT B (open drain)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		8	80	µA
I_{LOD}	Open Drain Leakage (V_{out} = V_{CC})		8	80	µA

PORT B (CMOS drive)

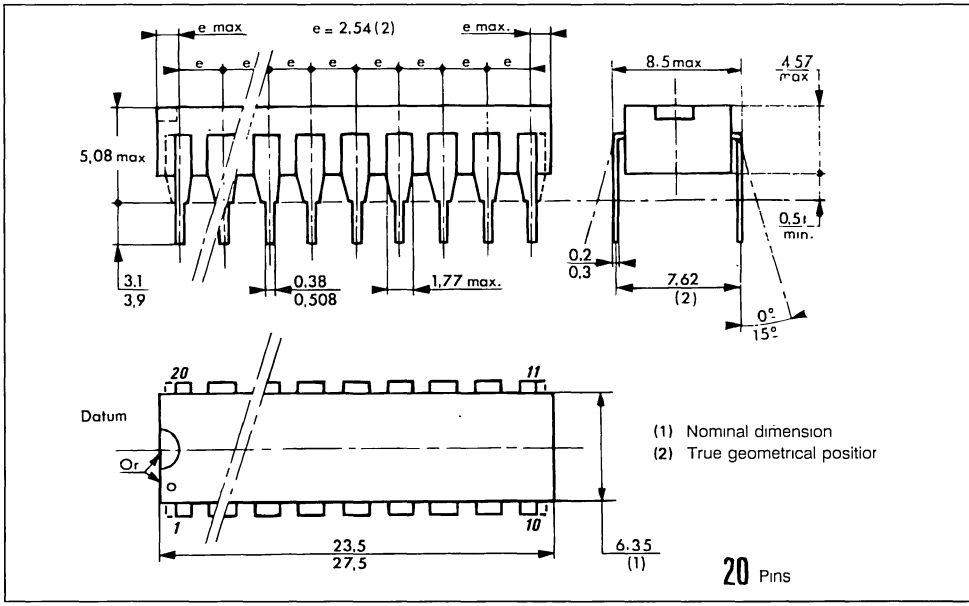
Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output High Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{OH}	Output High Voltage, I_{Load} = – 10µA	V_{CC} – 1.0			V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.3			V
V_{IH}	Input High Voltage, I_{Load} = – 300µA Max	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 300µA Max	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current (V_{in} = 0.4V to V_{CC})			– 300	µA

SECTION 8

MECHANICAL DATA

This section contains the pin assignment and package dimension diagrams for the EF6804J2 microcomputer.

MECHANICAL DATA



SECTION 9

ORDERING INFORMATION

9.1. INTRODUCTION

The following information is required when ordering a custom MCU. The information may be transmitted to SGS-THOMSON in the following media :

EPROM(s), 2716 or 2732

EFDOS/MDOS *, disk file

To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person, or your local SGS-THOMSON representative.

9.1.1. EPROMs. One 2716 or one 2732 type EPROM, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2716 or 2732 EPROM, the EPROM must be programmed as follows in order to emulate the EF6804J2 MCU. For a 2716, start the data space ROM at EPROM address $020 and start program space ROM at EPROM address $410 and continue to memory space $7FF. Memory spaces $7F8 through $7FB are reserved for SGS-THOMSON self-test vectors. For a 2732, the memory map shown in figure 2.1 can be used. All unused bytes, including the user's space, must be set to zero. For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packed securely. Do not use styrofoam.

9.1.2. EFDOS/MDOS * DISK FILE. An EFDOS/MDOS* disk, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. When using the EFDOS/MDOS* disk, include the entire memory image of both data and program space. All unused bytes, including the user's space, must be set to zero.

9.2. VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS-THOMSON. The signed verification form constitutes the contractural agreement for creation of the customer mask. If desired, SGS-THOMSON will program a blank 2716, 2732, or EFDOS/MDOS* disk (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

* Requires prior factory approval

9.3. ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually tested only at room temperature, five volts and may be unmarked and packaged in ceramic. These RVUs are included in the mask charge and are not production parts.

These RVUs are not backed nor guaranteed by SGS-THOMSON Quality Assurance.

9.4. FLEXIBLE DISKS

The disk media submitted must be single-sided single density, 8-inch, EFDOS/MDOS* compatible floppies. The customer must clearly label the disk with the ROM pattern file name. The minimum EFDOS/MDOS* system files as well as the absolute binary object file (file name. LO type of file) from the 6804 cross assembler must be on the disk. An object file made from a memory dump, using the ROLLOUT command is also admissable. Consider submitting a source listing as well as : file name, LX (DEVICE/EXORciser loadable format). This file will of course be kept confidential and is used 1) to speed up the process in house if any problems arise, and 2) to speed up our customer to factory interface if a user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

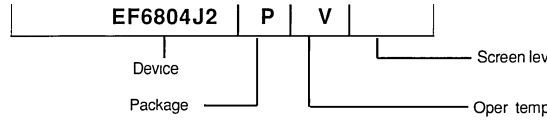
EFDOS is SGS-THOMSON Disk Operating System available on development systems such as DEVICE.

MDOS is MOTOROLA's Disk Operating System available on development systems such as EXORciser...

* Requires prior factory approval.

Whenever ordering a custom MCU is required, please contact your local SGS-THOMSON Microelectronics distributor and/or complete and send the attached "MCU customer ordering sheet SEMICONDUCTEURS representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device	Package					Oper. Temp			ScreeningLevel			
	C	J	P	E	FN	L*	V	T	Std	D		
EF6804J2			●			●	●		●	●		
Examples : EF6804J2P, EF6804J2PV, EF6804J2PLD, EF6804J2PVD												

Package : C : Ceramic DIL, **P :** Plastic DIL, **FN :** PLCC
Oper. temp. : L* : 0°C to + 70°C, **V :** – 40°C to + 85°C, **T :** – 40°C to + 105°C, *** :** may be omitted
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D

EXORciser is a registered trademark of MOTOROLA Inc

These specifications are subject to change without notice
Please inquire with our sales offices about the availability of the different packages

EF6804 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....

.....

ROM capacity required : | | | | bytes

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 40°C / + 85°C
- ❑ – 40°C / + 105°C

Package
- ❑ Plastic

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
- ❑ SGS-THOMSON Microelectronics reference
  .....
- ❑ Special customer data reference*
  .....

Quality level :
- ❑ STD
- ❑ D
- ❑ Other* (customer's quality specification ref.) :
  .....

Software developped by :
- ❑ SGS-THOMSON Microelectronics application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
- ❑ EPROM Reference :
- ❑ EFDOS/MDOS* disk file
  - ❑ 8" floppy
  - ❑ 5" 1/4 floppy
- ❑ Other *

OPTION LIST

-Oscillator input :
- ❑ Xtal
- ❑ RC

- Interrupt Trigger :
- ❑ Edge -sensitive
- ❑ Level-and edge-sensitive

- Port A output drive (4I/Os)
- ❑ Enabled
- ❑ Disabled

- Port B output drive :
- ❑ CMOS and TTL
- ❑ TTL only
- ❑ Open drain

* Requires prior factory approval

Yearly quantity forecast :
- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME : DATE : SIGNATURE :

SGS-THOMSON MICROELECTRONICS

EF6804P2

8 BIT MICROCOMPUTER

HARDWARE FEATURES

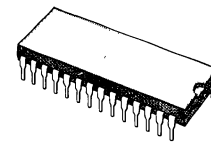
- 8-BIT ARCHITECTURE
- PIN COMPATIBLE WITH THE EF6805P2 AND EF68HC04P3
- 32 BYTES OF RAM
- MEMORY MAPPED I/O
- 1020 BYTES OF USER ROM
- 64 BYTES OF ROM FOR LOOK-UP TABLES
- 20 TTL/CMOS COMPATIBLE BIDIRECTIONAL I/O LINES (eight lines are led compatible)
- 8-BIT TIMER WITH 7-BIT SOFTWARE PROGRAMMABLE PRESCALER
- ON-CHIP CLOCK GENERATOR
- SELF-CHECK MODE AND ROM VERIFY MODE
- MASTER RESET
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE ®
- 5 VOLT SINGLE SUPPLY
- TIMER PIN IS PROGRAMMABLE AS INPUT OR OUTPUT

SOFTWARE FEATURES

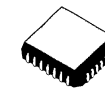
- SIMILAR TO EF6805 HMOS FAMILY
- BYTE EFFICIENT INSTRUCTION SET
- EASY TO PROGRAM
- TRUE BIT MANIPULATION
- BIT TEST AND BRANCH INSTRUCTION
- SEPARATE FLAGS FOR INTERRUPT AND NORMAL PROCESSING
- VERSATILE INDIRECT REGISTERS
- CONDITIONAL BRANCHES
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- TRUE LIFO STACK ELIMINATES STACK POINTER
- EIGHT POWERFUL ADDRESSING MODES
- ANY BIT IN DATA SPACE MEMORY MAY BE TESTED
- ANY BIT IN DATA SPACE MEMORY CAPABLE OF BEING WRITTEN TO MAY BE SET OR CLEARED

USER SELECTABLE OPTIONS

- 20 BIDIRECTIONAL I/O LINES WITH LSTTL, LSTTL/CMOS, OR OPEN-DRAIN INTERFACE
- CRYSTAL OR LOW-COST RESISTOR-CAPACITOR OSCILLATOR
- MASK SELECTABLE EDGE- OR LEVEL-SENSITIVE INTERRUPT PIN

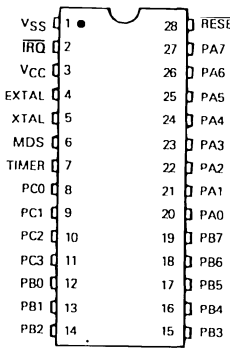


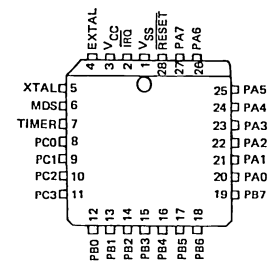
CB132
P
(Plastic Package)



CB520
FN
(PLCC28)

PIN CONNECTIONS



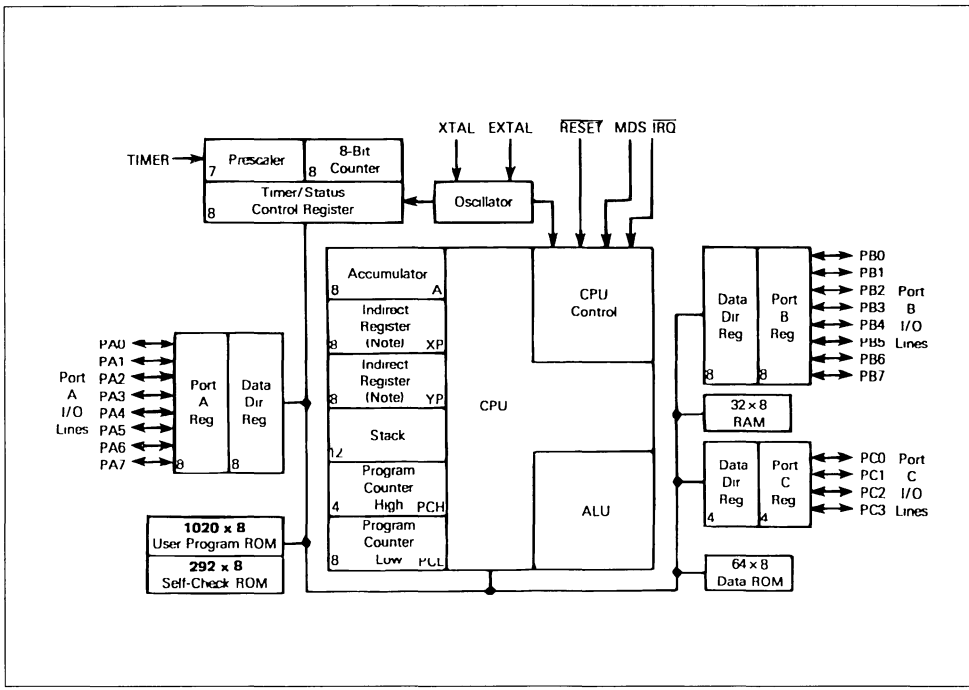


DESCRIPTION

The EF6804P2 Microcomputer Unit (MCU) is a member of the EF6804 Family of very low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the EF6800-based instruction set. The following are some of the hardware and software highlights of the EF6804P2 MCU.

INICE® is SGS THOMSON development/emulation tool

Figure 1.1 : EF6804P2 MCU Block Diagram.



Note : 8-Bit indirect registers XP and YP, although shown as part of the CPU, are actually located in the 32 x 8 RAM at locations $80 and $81.

SECTION 2

FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS

This section provides a description of the functional pins, memory spaces, the central processing unit (CPU), and the various registers and flags.

2.1. FUNCTIONAL PIN DESCRIPTION

2.1.1. V_{CC} and V_{SS}. Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

2.1.2. $\overline{IRQ}$. This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **4.1. INTERRUPT** for additional information.

2.1.3. XTAL and EXTAL. These pins provide connections to the on-chip clock oscillator circuit. A crystal,a resistor and capacitor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to **4.4 INTERNAL CLOCK GENERATOR OPTIONS** for recommendations concerning these inputs.

2.1.4. TIMER. In the input mode, the timer pin is connected to the prescaler input and serves as the timer clock. In the output mode, the timer pin signals that a time out of the timer has occurred. Refer to **SECTION 3 TIMER** for additional information.

2.1.5. $\overline{RESET}$. The $\overline{RESET}$ pin is used to restart the processor of the EF6804P2 to the beginning of a program. This pin, together with the MDS pin is also used to select the operating mode of the EF6804P2. If the MDS pin is at zero volts, the normal mode is selected and the program counter is loaded with the user restart vector. However, if the MDS pin is at + 5 volts, then pins PA6 and PA7 are decoded to allow selection of the operating mode. Refer to **4.3 RESET** for additional information.

2.1.6. MDS. The MDS (mode select) pin is used to place the MCU into special operating modes. If MDS is held at + 5 volts at the exit of the reset state, the decoded state of PA6 and PA7 is latched to determine the operating mode (single-chip, self-check, or ROM verify). However, if MDS is held at zero volts at the exit of the reset state, the single-chip operating mode is automatically selected (regardless of PA6 and PA7 state).

For those users familiar with the EF6801 microcomputer, mode selection is similar but much less complex in the EF6804P2. No special external diodes, switches, transistors, etc. are required in the EF6804P2.

2.1.7. INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC3). These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to **SECTION 5 INPUT/OUTPUT PORTS** for additional information.

2.2. MEMORY

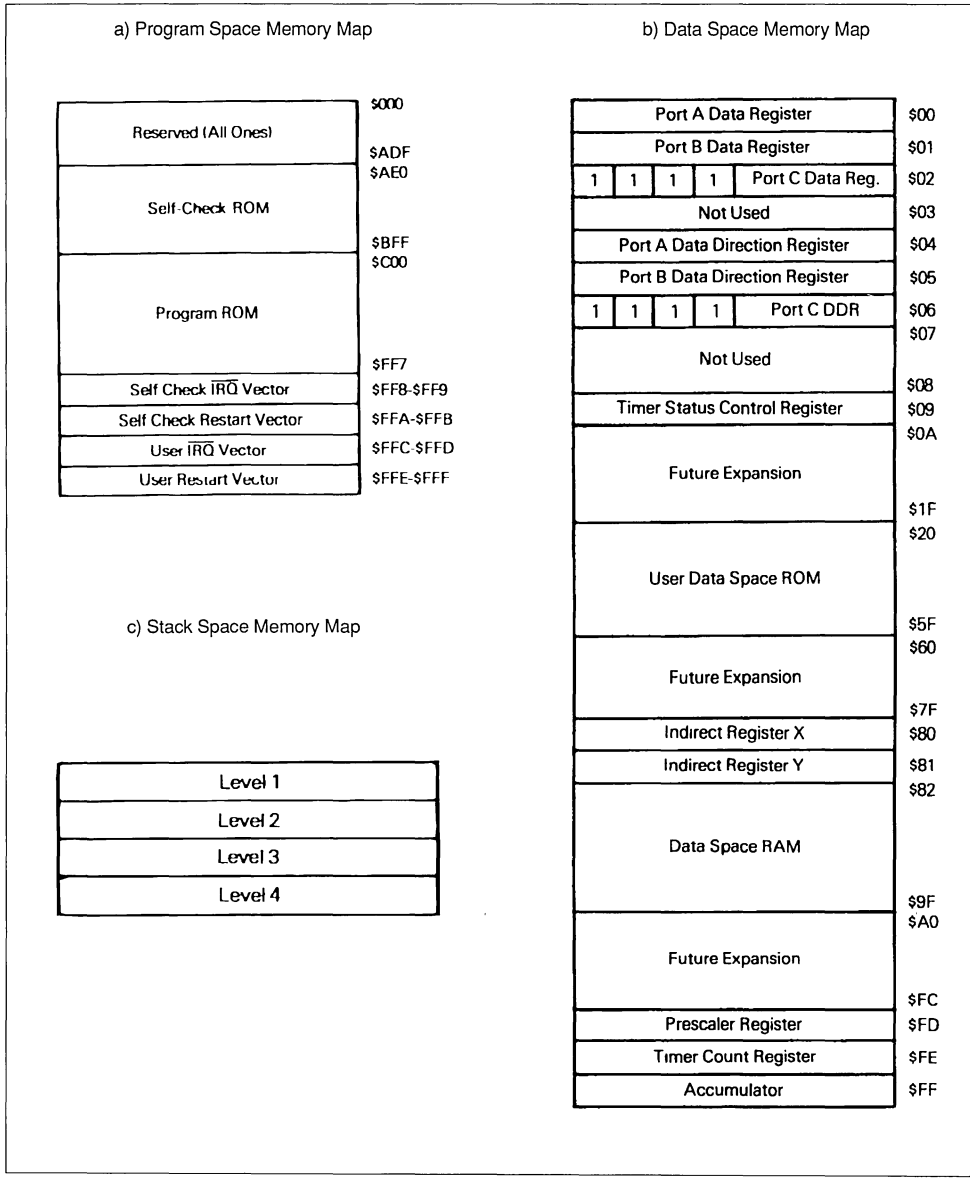
The MCU operates in three different memory spaces : program space, data space, and stack space. A representation of these memory spaces is shown in figure 2.1. The program space (figure 2.1a) contains all of the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, and the self-check and user vectors. The data space (figure 2.1b) contains all of the RAM locations, plus I/O locations and some ROM used for storage of tables and constants. The stack space (figure 2.1c) contains RAM which is used for stacking subroutine and interrupt return addresses.

The MCU is capable of addressing 4096 bytes of program space memory with its program counter and 256 bytes of data space memory with its instructions. The data space memory contains three bytes for port data registers, three bytes for port data direction registers, one byte for timer status/control, 64 bytes ROM, 32 bytes RAM (which includes two bytes for XP and YP indirect registers), two bytes for timer prescaler and count registers, and one byte for the accumulator. The program space section includes 288 bytes of self-check ROM, 1016 bytes program ROM, and eight bytes of vectors for self-check and user programs.

2.3. CENTRAL PROCESSING UNIT

The CPU of the EF6804 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control buses.

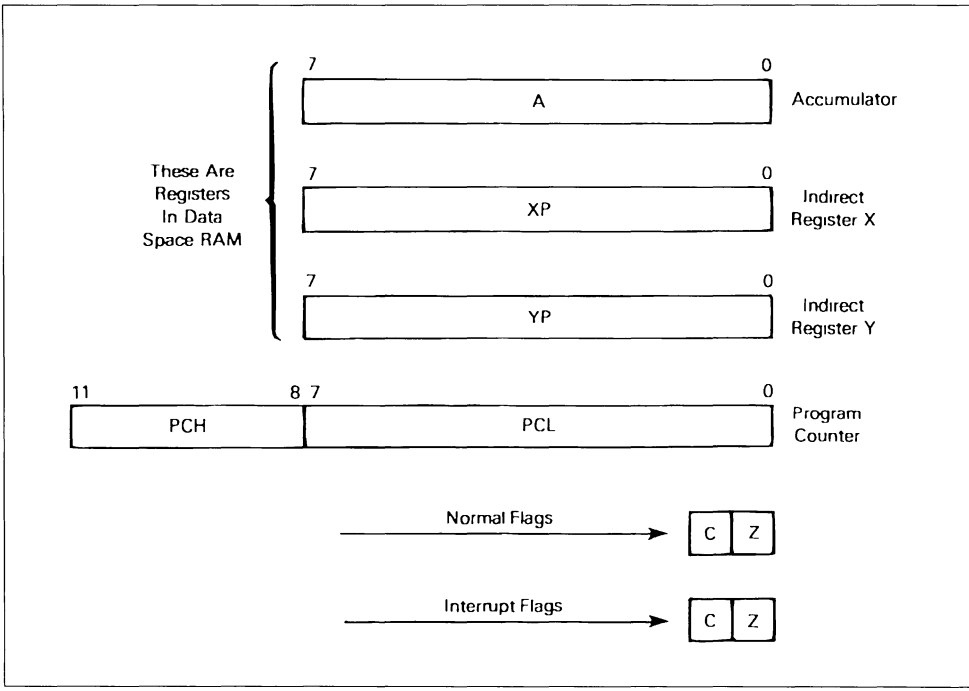
Figure 2.1 : EF6804P2 MCU Address Map.



2.4. REGISTERS

The EF6804 Family CPU has four registers and two flags available to the programmer. They are shown in figure 2.2 and are explained in the following paragraphs.

Figure 2.2 : Programming Model.



2.4.1. ACCUMULATOR (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is implemented as the highest RAM location ($FF) in data space and thus implies that several instructions exist which are not explicitly implemented. Refer to **6.3 IMPLIED INSTRUCTIONS** for additional information.

2.4.2. INDIRECT REGISTERS (XP, YP). These two indirect registers are used to maintain pointers to other memory locations in data space. They are used in the register-indirect addressing mode, and can be accessed with the direct, indirect, short direct, or bit set/clear addressing modes. These registers are implemented as two of the 32 RAM locations ($80, $81) and as such generate implied instructions and may be manipulated in a manner similar to any RAM memory location in data space. Refer to 6.3. IMPLIED INSTRUCTIONS for additional information.

2.4.3. PROGRAM COUNTER (PC). The program counter is a 12-bit register that contains the address of the next ROM word to be used (may be opcode, operand, or address of operand). The 12-bit program counter is contained in PCL (low byte) and PCH (high nibble).

2.4.4. FLAGS (C, Z). The carry (C) bit is set on a carry or a borrow out of the ALU. It is cleared if the result of an arithmetic operation does not result in a carry or a borrow. The (C) bit is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The zero (Z) bit is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

There are two sets of these flags, one set is for interrupt processing, the other for all other routines. When an interrupt occurs, a context switch is made from the program flags to the interrupt flags (inter-

rupt mode). An RTI forces the context switch back to the program flags (program mode). While in either mode, only the flags for that mode are available. Further, the interrupt flags will not be cleared upon entering the interrupt mode. Instead, the flags will be as they were at the exit of the last interrupt mode. Both sets of flags are cleared by reset.

2.4.5. STACK. There is a true LIFO stack incorporated in the EF6804P2 which eliminates the need for a stack pointer. Stack space is implemented in separate RAM (12-bits wide) shown in figure 2-1c. Whenever a subroutine call (or interrupt) occurs, the contents of the PC are shifted into the top register of the stack. At the same time (same cycle), the top register is shifted to the next level deeper. This happens to all registers with the bottom register falling out the bottom of the stack.

Whenever a subroutine or interrupt return occurs, the top register is shifted into the PC and all lower registers are shifted up one level higher. The stack RAM is four levels deep. If the stack is pulled more than four times without any pushes, the address that was stored in the bottom level will be shifted into the PC.

SECTION 3

TIMER

3.1. INTRODUCTION

A block diagram of the EF6804P2 timer circuitry is shown in figure 3.1. The timer logic in the MCU is comprised of a simple 8-bit counter (timer count register, TCR) with a 7-bit prescaler, and a timer status/control register (TSCR). The timer count register, which may be loaded under program control, is decremented towards zero by a clock input (prescaler output). The prescaler is used to extend the maximum interval of the overall timer. The prescaler tap is selected by bits 0-2 (PS0-PS2) of the timer status/control register. Bits PS0-PS2 control the actual division of the prescaler within the range of divide-by-1 (2^0) to divide-by-128 (2^7). The timer count register (TCR) and prescaler are decremented on rising clock edges. The coding of the TCSR PS0-PS2 bits produce a division in the prescaler as shown in table 3.1.

Table 3.1 : Prescaler Coding Table.

PS2	PS1	PS0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

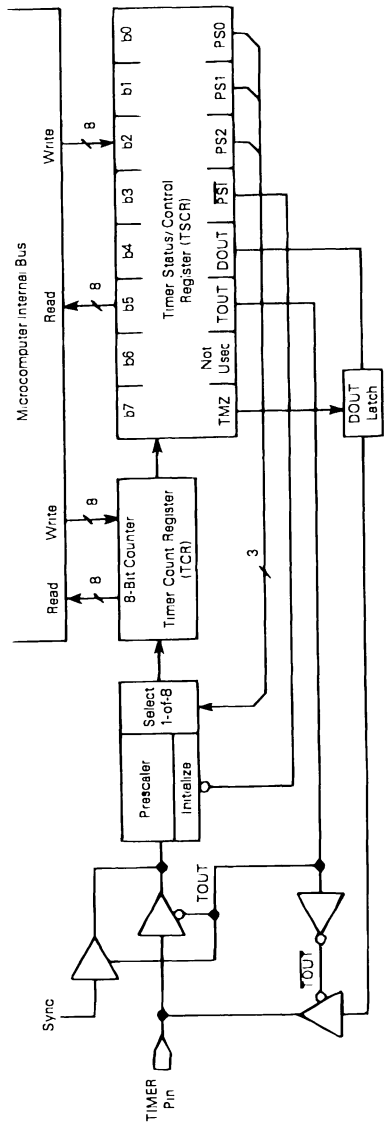
The TIMER pin may be programmed as either an input or an output depending on the status of TOUT (TSCR bit 5). Refer to figure 3.1. In the input mode, TOUT is a logic zero and the TIMER pin is connected directly to the prescaler input. Therefore, the timer prescaler is clocked by the signal applied from the TIMER pin. The prescaler then divides its clock input by a value determined by the coding of the TSCR bits PS0-PS2 as shown in table 3.1. The divided prescaler output then clocks the 8-bit timer count register (TCR). When the TCR is decremented to zero, it sets the TMZ bit in the timer status/control register (TSCR). The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The frequency of the external clock applied to the TIMER pin must be less than t_{byte} ($f_{osc}/48$).

In the output mode, TOUT is a logic one and the TIMER pin is connected to the DOUT latch. Therefore, the timer prescaler is clocked by the internal sync pulse (divide-by-48 of the internal oscillator). Operation is similar to that described above for the input mode. However, in the output mode, the low-to-high TMZ bit transition is used to latch the DOUT bit of the TSCR and provide it as output for the TIMER pin.

NOTE :

TMZ is normally set to logic one when the timer times out (TCR decrements to $00) ; however, it may be set by a write of $00 to the TCR or by a write to bit 7 of the TSCR.

Figure 3.1 : Timer Block Diagram.



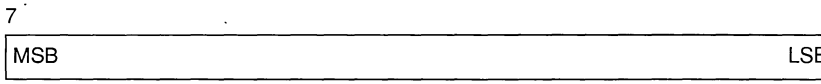
TIMER Pin Status

TOUT	Prescaler Clock	TIMER Pin
0	TIMER Pin	Input Mode
1	Sync	Output Mode

During reset, the timer count register and prescaler are set to $FF, while the timer status/control register is cleared to $00 and the DOUT LATCH (TIMER pin is in the high-impedance input mode) is forced to a logic high. The prescaler and timer count register are implemented in data space RAM locations ($FD, $FE) ; therefore, they are both readable and writeable. A write to either will predominate over the TCR decrement-to-$00 function ; i.e., if a write and a TCR decrement-to-$00 occur simultaneously, the write will take precedence, and the TMZ bit is not set until the next timer time out.

3.2. TIMER REGISTERS

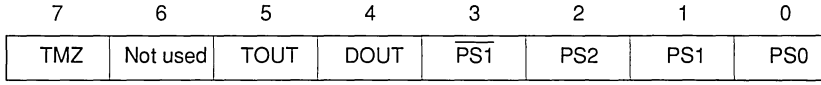
3.2.1. Timer Count Register (TCR)



TSCR Address = $FE

The timer count register indicates the state of the internal 8-bit counter.

3.2.2. TIMER STATUS/CONTROL REGISTER (TSCR)



TSCR Address = $09

b7, TMZ — Low-to-high transition indicates the timer count register has decremented to zero since the timer status/control register was last read. Cleared by a read of TSCR register if TMZ was read as a logic one.

b6 — Not used.

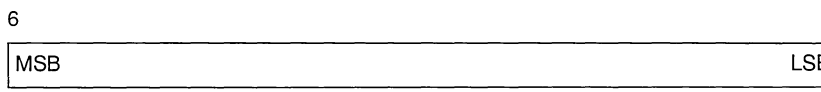
b5, TOUT — When low, this bit selects the input mode for the timer. When high, the output mode is selected.

b4, DOUT — Data sent to the timer output pin when TMZ is set high (output mode only).

b3, PSI — Used to initialize the prescaler and inhibit its counting while $\overline{PSI} = 0$. The initialized value is set to $FF. The timer count register will also be inhibited (contents unchanged). When $\overline{PSI} = 1$ the prescaler begins to count downward.

b0, b1, b2 PS0-PS1-PS2 — These bits are used to select the prescaler divide-by ratio ; therefore, effecting the clock input frequency to the timer count register.

3.2.3. TIMER PRESCALER REGISTER



TPR Address = $FD

The timer prescaler register indicates the state of the internal 7-bit prescaler. This 7-bit prescaler divide ratio is normally determined by bits PS0-PS2 of the timer status/control register (see table 3.1).

SECTION 4

INTERRUPT, SELF-CHECK, RESET, AND INTERNAL CLOCK GENERATOR

4.1. INTERRUPT

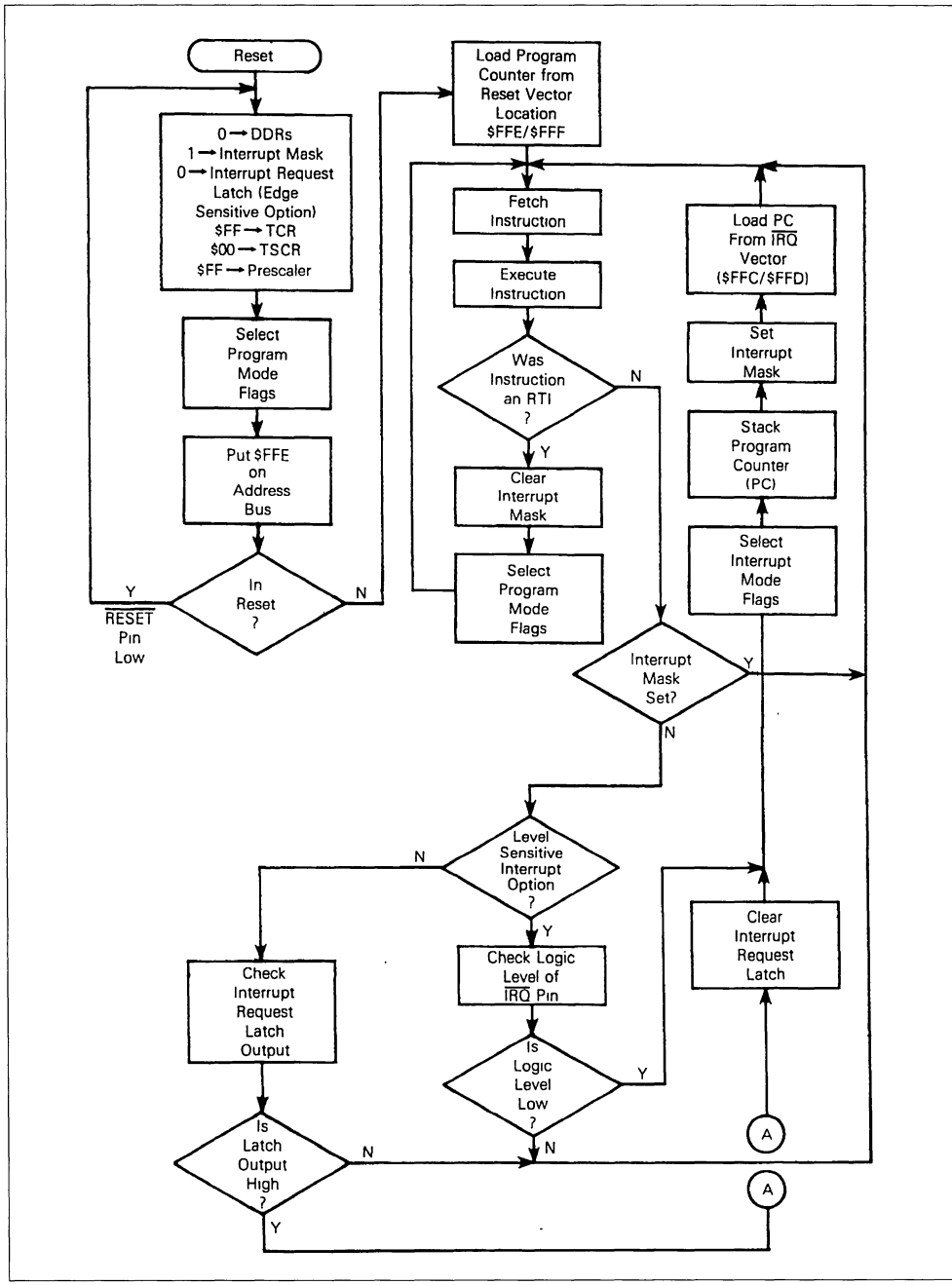
The EF6804P2 can be interrupted by applying a logic low signal to the $\overline{IRQ}$ pin ; however, a mask option selected at the time of manufacture determines whether the negative-going edge or the actual low level is sensed to indicate an interrupt.

4.1.1. EDGE-SENSITIVE OPTION. When the $\overline{IRQ}$ pin is pulled low, the internal interrupt request latch is set. Prior to each instruction fetch, the interrupt request latch is tested and, if its output is high, an interrupt sequence is initiated at the end of the current instruction (provided the interrupt mask is cleared). Figure 4.1 contains a flowchart which illustrates both the reset and interrupt sequence. The interrupt sequence consists of one cycle during which : the interrupt request latch is cleared, the interrupt mode flags are selected, the PC is saved on the stack, the interrupt mask is set, and the $\overline{IRQ}$ vector (single chip mode = $FFC/$FFD, self-check mode = $FF8/$FF9) is loaded into the PC. Internal processing of the interrupt continues until an RTI (return from interrupt) instruction is processed. During the RTI instruction, the interrupt mask is cleared and the program mode flags are selected. The next instruction of the program is then fetched and executed. Once the interrupt was initially detected and the interrupt sequence started, the interrupt request latch is cleared so that the next (second) interrupt may be detected even while the previous (first) one is being serviced. However, even though the second interrupt sets the interrupt request latch during processing of the first interrupt, the second interrupt sequence will not be initiated until completion of the interrupt service routine for the first interrupt. Completion of an interrupt service routine is always accomplished using an RTI instruction to return to the main program. The interrupt mask (which is not directly available to the programmer) is cleared during the last cycle of the RTI instruction.

4.1.2. LEVEL–SENSITIVE OPTION. The actual operation of the level-sensitive and edge-sensitive options are similar except that the level-sensitive option does not have an interrupt request latch. With no interrupt request latch, the logic level of the $\overline{IRQ}$ pin is checked for detection of the interrupt. Also, in the interrupt sequence, there is no need to clear the interrupt request latch. These differences are illustrated in the flowchart of figure 4.1.

4.1.3. POWER UP AND TIMING. During the power-up sequence the interrupt mask is set to preclude any false or "ghost" interrupts from occurring. To clear the interrupt mask, the programmer should write a JSR (instead of a JMP) instruction to an initialization routine as the first instruction in a program. The initialization routine should end with an RTI (instead of RTS). Maximum interrupt response time is eight machine (tbyte) cycles (see **4.4 INTERNAL CLOCK GENERATOR OPTIONS**). This includes five machine cycles for the longest instruction, plus one machine cycle for stacking the PC and switching flags, plus two machine cycles for synchronization of the IRQ input with the internal clock. Minimum response time is one machine cycle for stacking PC and switching flags (see **2.4.4 Flags** (C, Z)).

Figure 4.1 : Reset and Interrupt Processing Flowchart.



4.2. SELF-CHECK

The self check capability of the EF6804P2 MCU provides an internal check to determine if the part is functional. A schematic diagram of the self-check connections is shown in figure 4.2. To perform a functional check of the MCU, connect it as shown in figure 4.2a and monitor the LEDs for a 00100 ($04) pattern on port A. To initiate a ROM self-check of the memory simply connect the circuit as shown in figure 4.2b and check that the "good" LED turns on to indicate a good memory. The ROM verify uses a cyclical redundancy check (CRC) to conduct a ROM check by means of signature analysis circuit.This circuit consists of two 8-bit shift registers configured to perform the check using the CCITT polynominal.

4.3. RESET

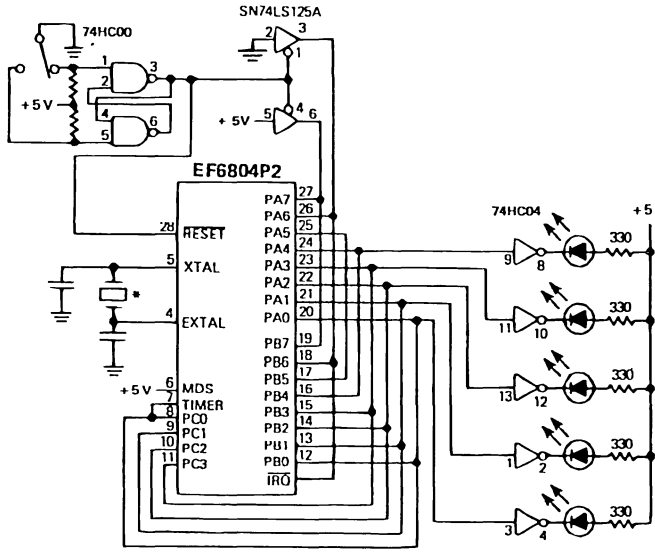
The MCU can be reset in two ways : by initial power up (see figure 4.1) and by the external reset input (RESET). During power up, a delay of tRHL is needed before allowing the RESET input to go high. This time delay allows the internal clock generator to stabilize. Connecting a capacitor and resistor to the RESET input, as shown in figure 4.3, typically provides sufficient delay.

4.4. INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor-capacitor, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different clock generator option connection methods are shown in figure 4.4, crystal specifications and suggested PC board layouts are given in figure 4.5, resistor-capacitor selection graph is given in figure 4.6, and a timing diagram is illustrated in figure 4.7. The crystal oscillator startup time is a function of many variables : crystal parameters (especially R_S), oscillator load capacitance (C_L), IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

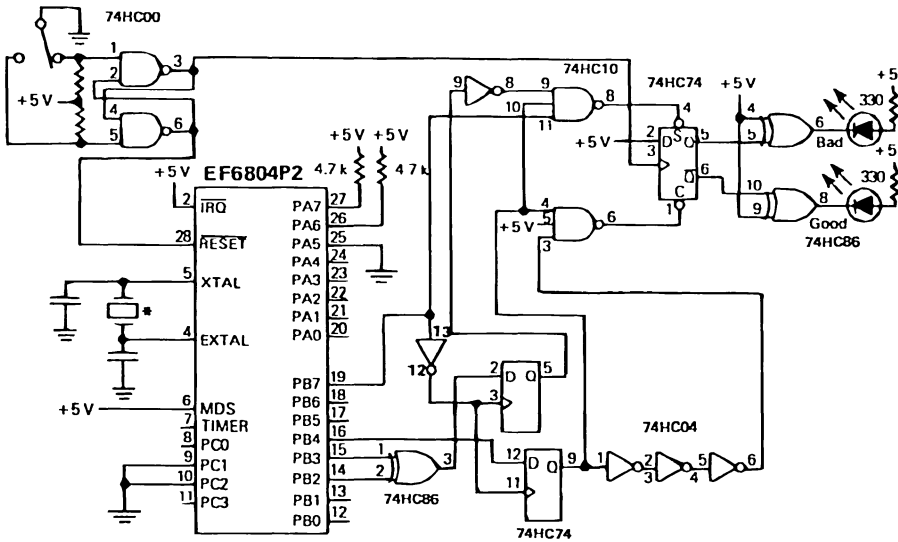
The oscillator output frequency is internally divided by four to produce the internal $\phi 1$ and $\phi 2$ clocks. The $\phi 1$ clock is divided by twelve to produce a machine byte (cycle) clock. A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to execute.

Figure 4.2 : Self-Check Circuit.



* Used with crystal option only

(a) functional check



* Used with crystal option only.

(b) Simple ROM Verify Check

Figure 4.3 : Power-Up Reset Delay Circuit.

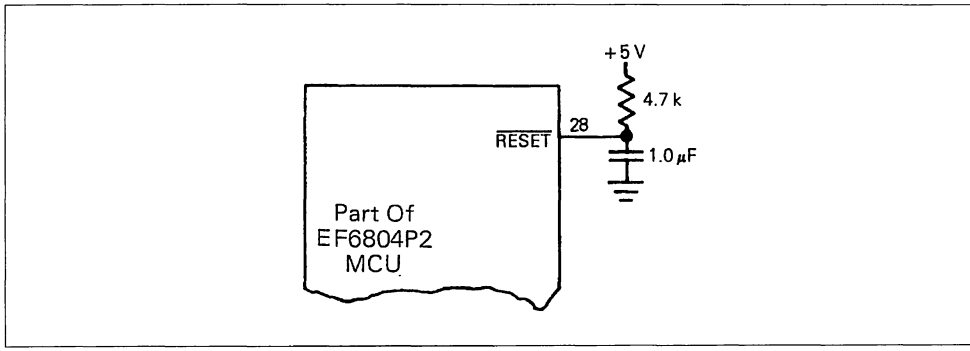


Figure 4.4 : Clock Generator Options.

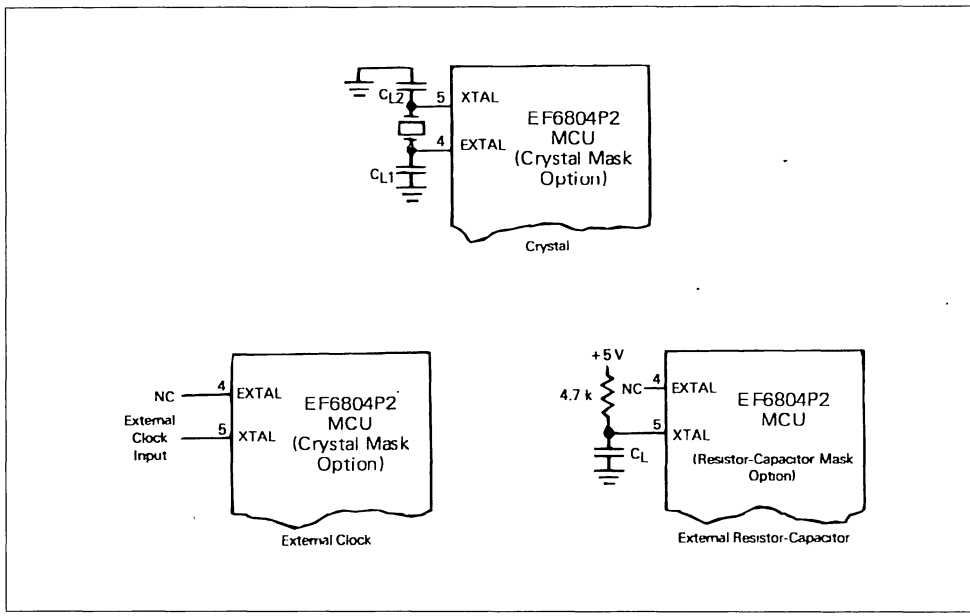
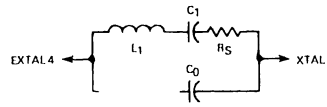
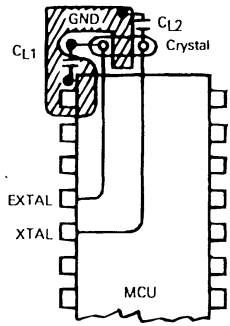


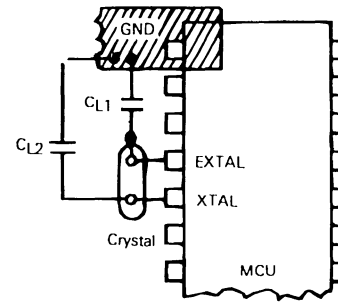
Figure 4.5 : Crystal Motional Arm Parameters and Suggested PC Board Layout.

(a) Crystal Parameters



Crystal Parameters
AT - Cut Parallel Resonance Crystal
C_0 = 7pF Maximum
Freq = 11MHz
R_S = 50Ω Maximum
Piezoelectric ceramic resonators which have the equivalent specifications may be used instead of crystal oscillators. Follow ceramic resonator manufacturer's suggestions for C_0, C_1, and R_S values.





Note : Keep crystal leads and circuit connections as short as possible.

Figure 4.6 : Typical Frequency Selection For Resistor-Capacitor Oscillator Option (C_L = 17pF).

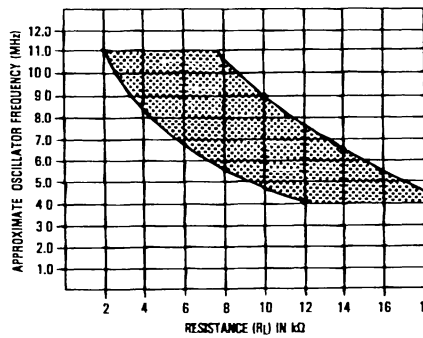
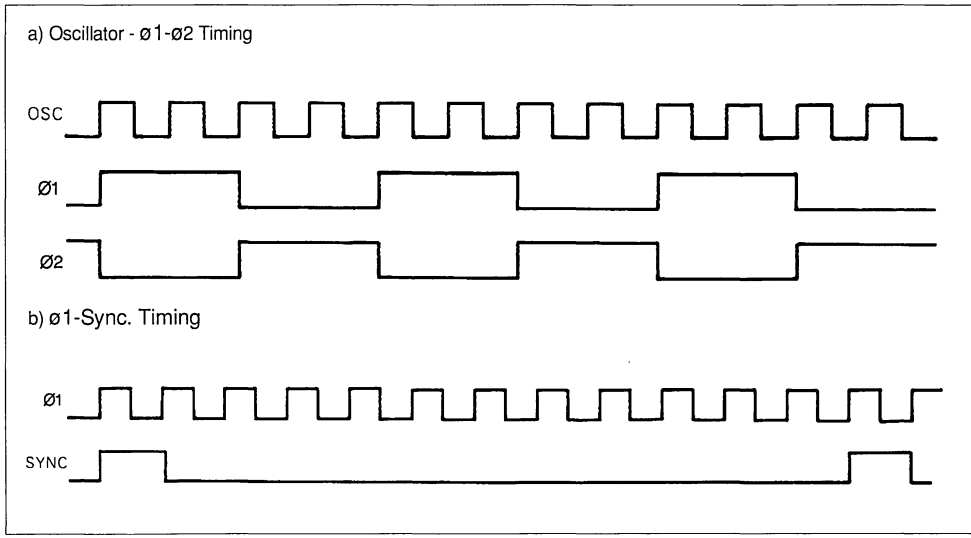


Figure 4.7 : Clock Generator Timing Diagram.



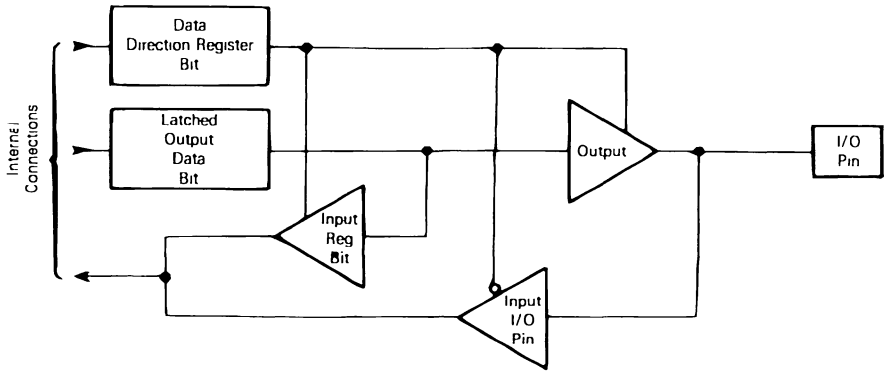
SECTION 5

INPUT/OUTPUT PORTS

5.1. INPUT/OUTPUT

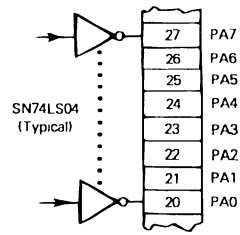
There are 20 input/output pins. All pins (port A, B, and C) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset, all the DDRs are initialized to a logic zero state to put the ports in the input mode. The port output registers are not initialized on reset but should be initialized before changing the DDR bits to avoid undefined levels. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading ; see figure 5-1. All input/output pins are LSTTL compatible as both inputs and outputs. In addition, all three ports may have one of two mask options : 1) internal pullup resistor for CMOS output compatibility, or 2) open drain output. The address map in figure 2-1 gives the address of data registers and DDRs. The register configuration is discussed under the registers paragraph below and figure 5.2 provides some examples of port connections.

Figure 5.1 : Typical I/O Port Circuitry.

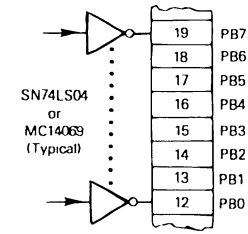


Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z	Pin

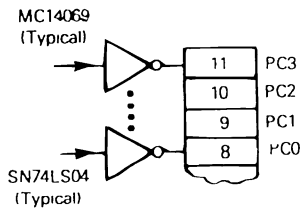
Figure 5.2 : Typical Port Connections.



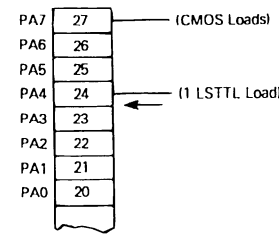
LSTTL Driving Port A Directly



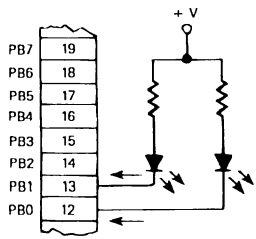
CMOS or LSTTL Driving Port B Directly



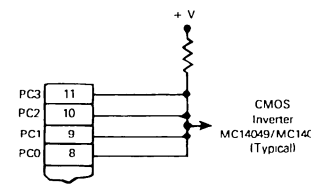
CMOS and LSTTL Driving Port C Directly



Port A, bit 7 programmed as output, driving CMOS loads and bit 4 driving one LSTTL load directly (using CMOS output option)



Port B, bit 0, and bit 1 programmed as output, driving LEDs directly.



Port C open drain option, with bits 0-3 programmed as output, driving CMOS load via wired-ORed configuration.

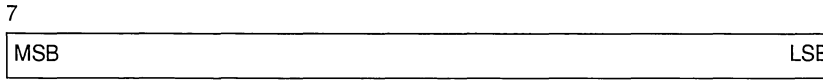
The latched output data bit (see figure 5.1) may always be written. Therefore, any write to a port writes to all of its data bits even though the port DDR is set to input. This may be used to initialize the data registers and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and corresponds to the latched output data when the DDR is an output (1). The 20 bidirectional lines may be configured by port to be LSTTL (standard configuration), LSTTL/CMOS (mask option), or open drain (mask option). Port B outputs are LED compatible.

NOTE :

The mask option only allows changes by port. For example, if the customer wishes PA7 to be open drain, then PA0-PA7 must all be open drain.

5.2. REGISTERS

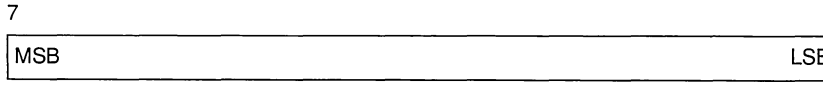
The registers described below are implemented as RAM locations and thus may be read or written.



Port A Address = $00
Port B Address = $01
Port C Address = $02 (bits 0-3)

5.2.1. PORT DATA REGISTER

The source of data read from the port data register will be the port I/O pin or previously latched output data depending upon the contents of the corresponding data direction register (DDR). The destination of data written to the port data register will be an output data latch. If the corresponding data direction register (DDR) for the port I/O pin is programmed as an output, the data will then appear on the port pin.



Port A Address = $04
Port B Address = $05
Port C Address = $06 (bits 0-3)

5.2.2. PORT DATA DIRECTION REGISTER

The port DDRs configure the port pins as either inputs or outputs. Each port pin can be programmed individually to act as an input or an output. A zero in the pins corresponding bit position will program that pin as an input while a one in the pins corresponding bit position will program that pin as an output.

SECTION 6

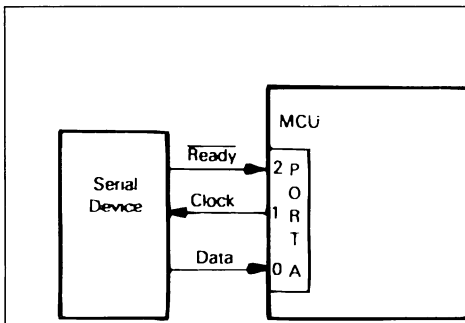
SOFTWARE AND INSTRUCTION SET

6.1. SOFTWARE

6.1.1. BIT MANIPULATION. The EF6804P2 MCU has the ability to set or clear any register or single random access memory (RAM) writable bit with a single instruction (BSET, BCLR). Any bit in data space, including ROM, can be tested, using the BRSET and BRCLR instructions, and the program may branch as a result of its state. The carry bit equals the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in figure 6.1 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line (to clock data one bit at a time, MSB first, out of the device). The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in the accumulator.

Figure 6.1 : Bit Manipulation Example.




```
                .
                .
                .
SELF    BRSET   2, PORTA, SELF
                .
                .
                .
        BSET    1, PORTA
        BRCLR   0, PORTA, CONT
CONT    BCLR    1, PORTA
        ROLA    .
                .
                .
                .
```

6.1.2. ADDRESSING MODES. The EF6804P2 MCU has nine addressing modes which are explained briefly in the following paragraphs. The EF6804P2 deals with objects in three different address spaces : program space, data space, and stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains all of the RAM locations, XP and YP registers, accumulator, timer, I/O locations, and some ROM (for storage of tables and constants). Stack space contains RAM for use in stacking the return addresses for subroutines and interrupts.

The term "Effective Address" (EA) is used in describing the address modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

6.1.2.1. Immediate.

In the immediate addressing mode, the operand is located in program ROM and is contained in a byte following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

6.1.2.2. Direct.

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the 256 bytes in data space memory with a single two-byte instruction.

6.1.2.3. Short Direct.

The MCU also has four locations in data space RAM ($80, $81, $82, $83) which may be used in a short-direct addressing mode. In this mode the opcode determines the data space RAM location, and the instruction is only one byte. Short direct addressing is a subset of the direct addressing mode. (The XP and YP registers are at locations $80 and $81 respectively).

6.1.2.4. Extended.

In the extended addressing mode, the effective address is obtained by concatenating the four least significant bits of the opcode (12-bit address). Instructions using the extended addressing mode (JMP, JSR) are capable of branching anywhere in program space. An extended addressing mode instruction is two bytes long.

6.1.2.5. Relative.

The relative addressing mode is only used in conditional branch instructions. In relative addressing, that address is formed by adding the sign extended lower five bits of the opcode (the offset) to the program counter if and only if the condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from - 15 to + 16 from the opcode address. The programmer need not worry about calculating the correct offset when using the assembler since it calculates the proper offset and checks to see if it is within the span of the branch.

6.1.2.6. Bit Set/clear.

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any bit in the 256 locations of data space memory, which can be written to, can be set or cleared.

6.1.2.7. Bit Test And Branch.

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit and condition (set or clear) which is to be tested is included in the opcode, and the data space address of the byte to be tested is in the single byte immediately following the opcode byte. The third byte is sign extended to twelve bits and becomes the offset added to the program counter if the condition is true. The single three-byte instruction allows the program to branch based on the condition of any bit in data space memory. The span of branching is from - 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry flag.

6.1.2.8. Register-indirect.

In the register-indirect addressing mode, the operand is at the address (in data space) pointed to by the contents of one of the indirect registers (X or Y). The particular X or Y register is selected by bit 4 of the opcode. Bit 4 of the opcode is then decoded into an address which selects the desired X or Y register ($80 or $81). A register-indirect instruction is one byte long.

6.1.2.9. Inherent.

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

6.2. INSTRUCTION SET

The EF6804P2 MCU has a set of 42 basic instructions which when combined with nine addressing modes produce 242 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

6.2.1. REGISTER/MEMORY INSTRUCTIONS. Most of these instructions use two operands. One operand is the accumulator and the other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operands. Refer to table 6.1.

6.2.2. READ-MODIFY-WRITE INSTRUCTIONS. These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. There are ten instructions which utilize read-modify-write cycles. All INC and DEC forms along with all bit manipulation instructions use this method. Refer to table 6.2.

6.2.3. BRANCH INSTRUCTIONS. The branch instructions cause a branch from the program when a certain condition is met. Refer to table 6.3.

6.2.4. BIT MANIPULATION INSTRUCTIONS. These instructions are used on any bit in data space memory. One group either sets or clears. The other group performs the bit test branch operations. Refer to table 6.4.

Table 6.1 : Register/Memory/Instructions.

		Addressing Modes																			
		Indirect				Immediate			Direct			Inherent			Extended			Short-Direct			
Function	Mnem	Opcode XP	Opcode YP	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Load A from Memory	LDA	E0	F0	1	4	E8	2	4	F8	2	4	–	–	–	–	–	–	AC-AF	1	4	1
Load XP from Memory	LDXI	–	–	–	–	B0	3	4	–	–	–	–	–	–	–	–	–	–	–	–	4
Load YP from Memory	LDYI	–	–	–	–	B0	3	4	–	–	–	–	–	–	–	–	–	–	–	–	4
Store A in Memory	STA	E1	F1	1	4	–	–	–	F9	2	4	–	–	–	–	–	–	BC-BF	1	4	2
Add to A	ADD	E2	F2	1	4	EA	2	4	FA	2	4	–	–	–	–	–	–	–	–	–	–
Subtract from A	SUB	E3	F3	1	4	EB	2	4	FB	2	4	–	–	–	–	–	–	–	–	–	–
Arithmetic Compare with Memory	CMP	E4	F4	1	4	EC	2	4	FC	2	4	–	–	–	–	–	–	–	–	–	–
AND Memory to A	AND	E5	F5	1	4	ED	2	4	FD	2	4	–	–	–	–	–	–	–	–	–	–
Jump to Subroutine	JSR	–	–	–	–	–	–	–	–	–	–	–	–	–	8 (TAR)	2	4	–	–	–	3
Jump Unconditional	JMP	–	–	–	–	–	–	–	–	–	–	–	–	–	9 (TAR)	2	4	–	–	–	3
Clear A	CLRA	–	–	–	–	–	–	–	FB	2	4	–	–	–	–	–	–	–	–	–	–
Clear XP	CLRX	–	–	–	–	–	–	–	FB	2	4	–	–	–	–	–	–	–	–	–	–
Clear YP	CLRY	–	–	–	–	–	–	–	FB	2	4	–			–	–	–	–	–	–	–
Complement A	COMA	–	–	–	–	–	–	–	–	–	–	B4	1	4	–	–	–	–	–	–	–
Move Immediate Value to Memory	MVI	–	–	–	–	B0	3	4	B0	3	4	–	–	–	–	–	–	–	–	–	5
Rotate A Left and Carry	ROLA	–	–	–	–	–	–	–	–	–	–	B5	1	4	–	–	–	–	–	–	–
Arithmetic Left Shift of A	ASLA	–	–	–	–	–	–	–	FA	2	4	–	–	–	–	–	–	–	–	–	–

SPECIAL NOTES

1. In Short-Direct addressing, the LDA mnemonic represents opcode AC, AD, AE, and AF. This is equivalent to RAM locations $80 (AC), $81 (AD), $82 (AE), and $83 (AF)
2. In Short-Direct addressing, the STA mnemonic represents opcode BC, BD, BE, and BF This is equivalent to RAM locations $80 (BC), $81 (BD), $82 (BE), and $83 (BF).
3. In Extended addressing, the four LSBs of the opcode (Mnemonic JSR and JMP) are formed by the four MSBs of the target address. (TAR)
4. In Immediate addressing, the LDXI and LDYI are mnemonics which are recognized as follows:
   LDXI = MVI $80,data
   LDYI = MVI $81,data Where data is a one-byte hexadecimal number.
5. The MVI instruction refers to both Immediate and Direct addressing.

Table 6.2 : Read-modify-write Instructions.

Function	Mnem	Addressing Modes										Special Notes
		Indirect				Direct			Short-direct			
		Opcode		#	#	Op	#	#	Op	#	#	
		XP	YP	Bytes	Cycles	code	Bytes	Cycles	code	Bytes	Cycles	
Increment Memory Location	INC	E6	F6	1	4	FE	2	4	A8-AB	1	4	1, 3
Increment A	INCA					FE	2	4				
Increment XP	INCX								A8	1	4	
Increment YP	INCY								A9	1	4	
Decrement Memory Location	DEC	E7	F7	1	4	FF	2	4	B8-BB	1	4	2, 4
Decrement A	DECA					FF	2	4				
Decrement XP	DECX								B8	1	4	
Decrement YP	DECY								B9	1	4	

SPECIAL NOTES :
1. In Short-direct addressing, the INC mnemonic represents opcode A8, A9, AA, and AB. These are equivalent to RAM locations $80 (A8), $81 (A9), $82 (AA), and $83 (AB).
2. In Short-direct addressing, the DEC mnemonic represents opcode B8, B9, BA, and BB. These are equivalent to RAM locations $80 (B8), $81 (B9), $82 (BA), and $83 (BB).
3. In Indirect addressing, the INC mnemonic represents opcode E6 or F6, and causes the location pointed to by XP (E6 opcode) or YP (F6 opcode) to be incremented.
4. In Indirect addressing, the INC mnemonic represents opcode E7 or F7, and causes the location pointed to by XP (E7 opcode) or YP (F7 opcode) to be incremented.

Table 6.3 : Branch Instructions.

Function	Mnem	Relative Addressing Mode			Special Notes
		Opcode	# Bytes	# Cycles	
Branch if Carry Clear	BCC	40-5F	1	2	1
Branch if Higher or Same	(BHS)	40-5F	1	2	1, 2
Branch if Carry Set	BCS	60-7F	1	2	1
Branch if Lower	(BLO)	60-7F	1	2	1, 3
Branch if Not Equal	BNE	00-1F	1	2	1
Branch if Equal	BEQ	20-3F	1	2	1

SPECIAL NOTES :
1. Each mnemonic of the Branch Instructions covers a range of 32 opcodes, e.g , BCC ranges from 40 through 5F. The actual memory location (target address) to which the branch is made is formed by adding the sign extended lower five bits of the opcode to the contents of the program counter.
2. The BHS instruction (shown in parentheses) is identical to the BCC instruction. The C bit is clear if the register was higher or the same as the location in the memory to which it was compared
3. The BLO instruction (shown in parentheses) is identical to the BCS instruction The C bit is set if the register was lower than the location in memory to which it was compared.

Table 6.4 : Bit Manipulation Instructions.

Function	Mnem	Addressing Modes						Special Notes
		Bit Set/clear			Bit Test and Branch			
		Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	
Branch IFF Bit n is set.	BRSET n (n = 0.....7)				C8 + n	3	5	1
Branch IFF Bit n is clear.	BRCLR n (n = 0.....7)				C0 + n	3	5	1
Set Bit n	BSET n (n = 0.....7)	D8 + n	2	4				1
Clear Bit n	BCLR n (n = 0.....7)	D0 + n	2	4				1

SPECIAL NOTE : 1 The opcode is formed by adding the bit number (0-7) to the basic opcode. For example . to clear bit six using the BSET6 instruction the opcode becomes DE (D8 + 6) ; BCLR5 becomes (C0 + 5), etc.

6.2.5. CONTROL INSTRUCTIONS. The control instructions control the MCU operations during program execution. Refer to table 6-5.

6.2.6. ALPHABETICAL LISTING. The complete instruction set is given in alphabetical order in table 6-6. There are certain mnemonics recognized by the assembler and converted to other instructions. The fact that all registers and accumulator are in RAM allows many implied instructions to exist. The implied instructions recognized by the assembler are identified in table 6-6.

6.2.7. OPCODE MAP SUMMARY. Table 6-7 contains an opcode map for the instructions used on the MCU.

6.3. IMPLIED INSTRUCTIONS

Since the accumulator and all other registers are located in RAM many implied instructions exist. The assembler-recognized implied instructions are given in table 6-6. Some examples not recognized by the assembler are shown below.

BCLR, 7 $FF	Ensures accumulator is plus
BSET, 7 $FF	Ensures accumulator is minus
BRCLR, 7 $FF	Branch iff accumulator is plus
BRSET, 7 $FF	Branch iff accumulator is minus
BRCLR, 7 $80	Branch iff X is plus (BXPL)
BRSET, 7 $80	Branch iff X is minus (BXMI)
BRCLR, 7 $81	Branch iff Y is plus (BYPL)
BRSET, 7 $81	Branch iff Y is minus (BYMI)

Table 6.5 : Control Instructions.

Function	Mnem	Addressing Modes									Special Notes
		Short-direct			Inherent			Relative			
		Op code	# Bytes	# Cycles	Op code	# Bytes	# Cycles	Op code	# Bytes	# Cycles	
Transfer A to XP	TAX	BC	1	4							
Transfer A to YP	TAY	BD	1	4							
Transfer XP to A	TXA	AC	1	4							
Transfer YP to A	TYA	AD	1	4							
Return from Subroutine	RTS				B3	1	2				
Return from Interrupt	RTI				B2	1	2				
No-operation	NOP										1

SPECIAL NOTE : 1.The NOP instruction is equivalent to a branch if equal (BEQ) to the location designated by PC + 1

Table 6.6 : Instruction Set.

Mnemonic	Addressing Modes									Flags	
	Inherent	Immediate	Direct	Short Direct	Bit/set Clear	Bit-test Branch	Register Indirect	Extended	Relative	Z	C
ADD		X	X				X			^	^
AND		X	X				X			^	•
ASLA			Assembler converts this to "ADD $FF"							•	•
BCC									X	^	^
BCLR					X					•	•
BCS									X	•	•
BEQ									X	•	•
BHS			Assembler converts this to "BCC"							•	•
BLO			Assembler converts this to "BCS"							•	•
BNE									X	•	•
BRCLR						X				•	^
BRSET						X				•	^
BSET					X					•	•
CLRA			Assembler converts this to "SUB $FF"							^	^
CLRX			Assembler converts this to "MVI # 0, $80".							•	•
CLRY			Assembler converts this to "MVI # 0, $81".							•	•
CMP		X	X				X			^	^
COMA	X									^	^
DEC			X	X			X			^	•
DECA			Assembler converts this to "DEC $FF".							^	•
DECX			Assembler converts this to "DEC $80"							^	•
DECY			Assembler converts this to "DEC $81"							^	•
INC			X	X			X			^	•
INCA			Assembler converts this to "INC $FF".							^	•
INCX			Assembler converts this to "INC $80"							^	•
INCY			Assembler converts this to "INC $81"							^	•
JMP								X		•	•
JSR								X		•	•
LDA		X	X	X			X			^	•
LDXI			Assembler converts this to "MVI DATA, $80"							•	•
LDYI			Assembler converts this to "MVI DATA, $81"							•	•
MVI		X	X							•	•
NOP			Assembler converts this to "BEQ (PC) + 1".							•	•
ROLA	X									^	^
RTI	X									^	^
RTS	X									•	•
STA			X	X			X			^	•
SUB		X	X				X			^	^
TAX	Assembler converts this to "STA $80"									^	•
TAY	Assembler converts this to "STA $81"									^	•
TXA	Assembler converts this to "LDA $80"									^	•
TYA	Assembler converts this to "LDA $81"									^	•

Flag Symbols Z = Zero. C = Carry/Borrow. ^ = Test and Set if True, Cleared Otherwise, • = Not Affected

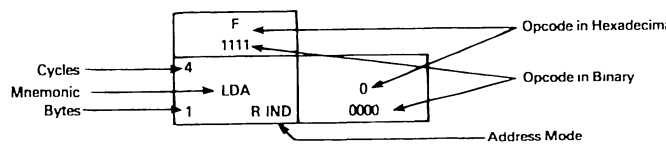
Table 6.7 : EF6804P2 Microcomputer Instruction Set Opcode Map.

	Branch Instructions							
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
1 0001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
2 0010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
3 0011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
4 0100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
5 0101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
6 0110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
7 0111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
8 1000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
9 1001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
A 1010	2 BNE 1 REL	2 BNE 2 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
B 1011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
C 1100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
D 1101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
E 1110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
F 1111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL

Abbreviations for Address Modes
INH Inherent
S-D Short Direct
B-T-B Bit Test and Branch
IMM Immediate
DIR Direct
EXT Extended
REL Relative
BSC Bit Set/Clear
R-IND Register Indirect
* Indicates Instruction Reserved for Future Use
Indicates Illegal Instruction

Table 6.7 : (continued).

Register/memory, Control, and Read/modify/write Instructions				Bit Manipulation Instructions		Register/memory and Read/modify/write		
8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi Low
4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 MVI 3 IMM	5 BRCLR0 3 B-T-B	4 BCLR0 2 BSC	4 LDA 1 R-IND	4 LDA 1 R-IND	0 0000
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR1 3 B-T-B	4 BCLR1 2 BSC	4 STA 1 R-IND	4 STA 1 R-IND	1 0001
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTI 1 INH	5 BRCLR2 3 B-T-B	4 BCLR2 2 BSC	4 ADD 1 R-IND	4 ADD 1 R-IND	2 0010
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTS 1 INH	5 BRCLR3 3 B-T-B	4 BCLR3 2 BSC	4 SUB 1 R-IND	4 SUB 1 R-IND	3 0011
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 COMA 1 INH	5 BRCLR4 3 B-T-B	4 BCLR4 2 BSC	4 CMP 1 R-IND	4 CMP 1 R-IND	4 0100
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 ROLA 1 INH	5 BRCLR5 3 B-T-B	4 BCLR5 2 BSC	4 AND 1 R-IND	4 AND 1 R-IND	5 0101
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR6 3 B-T-B	4 BCLR6 2 BSC	4 INC 1 R-IND	4 INC 1 R-IND	6 0110
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR7 3 B-T-B	4 BCLR7 2 BSC	4 DEC 1 R-IND	4 DEC 1 R-IND	7 0111
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET0 3 B-T-B	4 BSET0 2 BSC	4 LDA 1 IMM	4 LDA 2 DIR	8 1000
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET1 3 B-T-B	4 BSET1 2 BSC	#	4 STA 2 DIR	9 1001
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET2 3 B-T-B	4 BSET2 2 BSC	4 ADD 1 IMM	4 ADD 2 DIR	A 1010
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET3 3 B-T-B	4 BSET3 2 BSC	4 SUB 1 IMM	4 SUB 2 DIR	B 1011
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET4 3 B-T-B	4 BSET4 2 BSC	4 CMP 1 IMM	4 CMP 2 DIR	C 1100
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET5 3 B-T-B	4 BSET5 2 BSC	4 AND 1 IMM	4 AND 2 DIR	D 1101
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET6 3 B-T-B	4 BSET6 2 BSC	#	4 INC 2 DIR	E 1110
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET7 3 B-T-B	4 BSET7 2 BSC	#	4 DEC 2 DIR	F 1111



SECTION 7

ELECTRICAL SPECIFICATIONS

7.1. INTRODUCTION

This section contains the electrical specifications and associated timing for the EF6804P2.

7.2. ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage	– 0.3 to + 7.0	V
T_A	Operating Temperature Range Standard or L Suffix V Suffix T Suffix	TL to TH 0 to 70 – 40 to 85 – 40 to 105	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_J	Junction Temperature Range Plastic PLCC	 150 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages of electric fields , however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$ Reliability of operation is enhanced if unused inputs except EXTAL are connected to an appropriate logic voltage level (e g., either V_{SS} or V_{CC}).

7.3. THERMAL DATA

θ_{JA}	Thermal Resistance Plastic PLCC	 70 90	°C/W

Figure 7.1 : LSTTL Equivalent Test Load (port B).

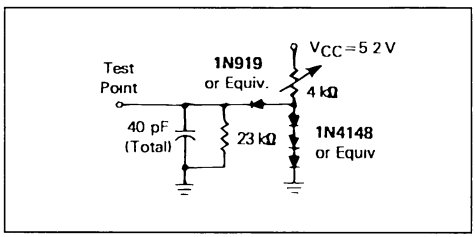


Figure 7.2 : CMOS Equivalent Test Load (port A, B, C).

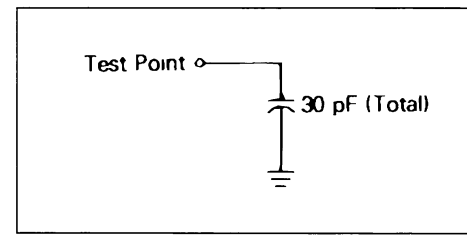
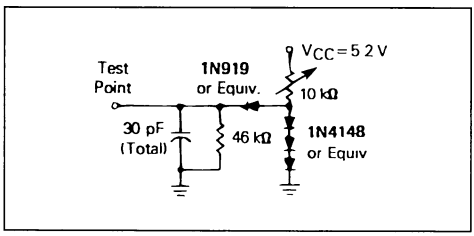


Figure 7.3 : LSTTL Equivalent Test Load (port A, C, and TIMER).



7.4. POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} < P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives :

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

7.5. ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.0Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
P_{INT}	Internal Power Dissipation-No Port Loading T_A = 0°C T_A = – 40°C		135	170 210	mV
V_{IH}	Input High Voltage	4.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
C_{in}	Input Capacitance		10		pF
I_{in}	Input Current ($\overline{IRQ}$, $\overline{RESET}$)		2	20	µA

7.6. SWITCHING CHARACTERISTICS

(V_{CC} = + 5.0Vdc ± 0.5Vdc, V_{SS} = GND, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{OSC}	Oscillator Frequency	4.0		11.0	MHz
t_{bit}	Bit Time	0.364		1.0	µs
t_{byte}	Byte Cycle Time	4.36		12.0	µs
t_{WL}, t_{WH}	$\overline{IRQ}$ and TIMER Pulse Width	2 x t_{byte}			
t_{RWL}	$\overline{RESET}$ Pulse Width	2 x t_{byte}			
t_{RHL}	$\overline{RESET}$ Delay Time (external capacitance = 1.0µF)	100			ms

7.7. PORT DC ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.0Vdc ± 0.5Vdc, V_{SS} = GND, T_A = T_L to T_H unless otherwise noted)

TIMER AND PORTS A AND C (standard)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA			0.5	V
V_{OH}	Output High Voltage, I_{Load} = – 50µA	2.3			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		4	40	µA

PORTS A AND C (open drain)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA			0.5	V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		4	40	µA
I_{LOD}	Open Drain Leakage (V_{out} = V_{CC})		4	40	µA

PORTS A AND C (CMOS drive)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 0.4mA (sink)			0.5	V
V_{OH}	Output High Voltage, I_{Load} = – 10µA	V_{CC}–1.0			V
V_{OH}	Output High Voltage, I_{Load} = – 50µA	2.3			V
V_{IH}	Input High Voltage, I_{Load} = – 300µA Max.	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 300µA Max.	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current (V_{IN} = 0.4V to V_{CC})			– 300	µA

PORT B (standard)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.3			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		8	80	µA

7.7 PORT DC ELECTRICAL CHARACTERISTICS (continued)

PORT B (open drain)

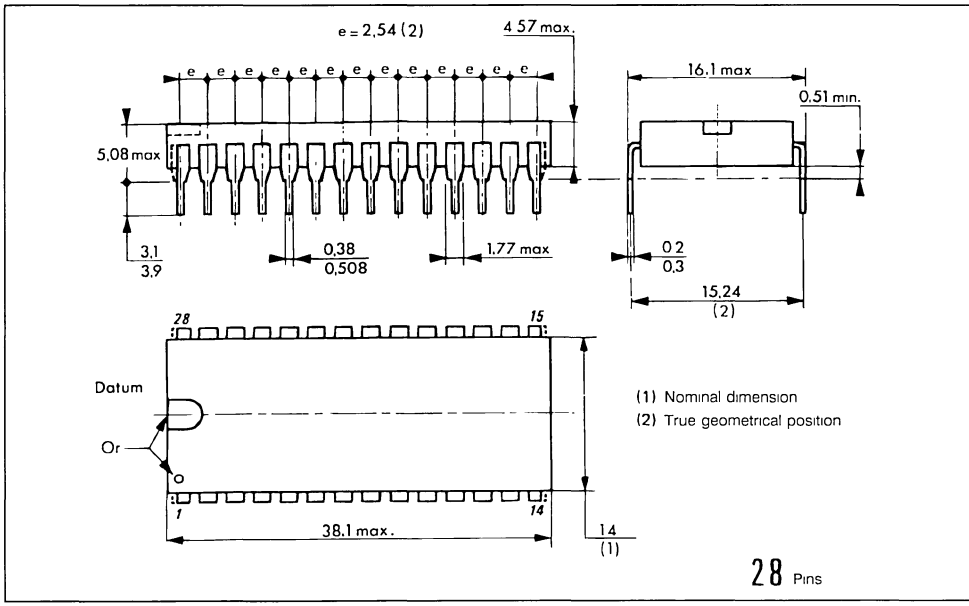
Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		8	80	µA
I_{LOD}	Open Drain Leakage (V_{out} = V_{CC})		8	80	µA

PORTS B (CMOS drive)

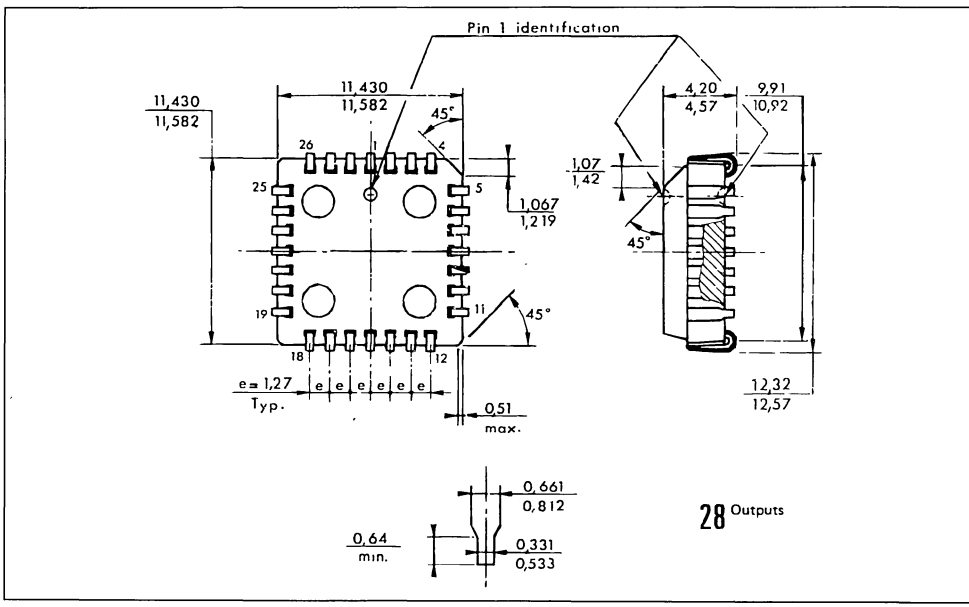
Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.0mA			0.5	V
V_{OL}	Output High Voltage, I_{Load} = 10mA (sink)			1.5	V
V_{OH}	Output High Voltage, I_{Load} = – 10µA	V_{CC}–1.0			V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.3			V
V_{IH}	Input High Voltage, I_{Load} = – 300µA Max.	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 300µA Max.	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current (V_{IN} = 0.4V to V_{CC})			– 300	µA

PACKAGE MECHANICAL DATA

28–PINS – PLASTIC PACKAGE



28–PIN – FN SUFFIX PLCC 28



SECTION 8

ORDERING INFORMATION

8.1. INTRODUCTION

The following information is required when ordering a custom MCU. The information may be transmitted to SGS-THOMSON in the following media :

EPROM(s), 2716 or 2732
EFDOS/MDOS* disk file

To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person, or your local SGS-THOMSON representative.

8.1.1. EPROMs. One 2716 or one 2732 type EPROM, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2716 EPROM, the EPROM must be programmed as follows in order to emulate the EF6804P2 MCU : **start the data space ROM at EPROM address $020 and start program space ROM at EPROM address $400 and continue to memory space $7FF. All unused bytes, including the user's space, must be set to zero, memory space $7F8 to $7FB is reserved for self-check vectors.** When using one 2732 EPROM, the memory map shown in figure 2-1 can be used. For shipment to SGS-THOMSON the EPROMs should be placed in a conductive IC carrier and packed securely. Do not use styrofoam.

8.1.2. EFDOS/MDOS* DISK FILE. An EFDOS/MDOS* disk, programmed with the customer program (positive logic sense for address and data) may be submitted for pattern generation. **When using the EFDOS/MDOS* disk, include the entire memory image of both data and program space. All unused bytes, including the user's space, must be set to zero.**

8.2. VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS-THOMSON. The signed verification form constitutes the contractural agreement for creation of the customer mask. If desired, SGS-THOMSON will program a blank 2716, 2732, or EFDOS/MDOS* disk (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

* Requires prior factory approval.

8.3. ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and five volts. These RVUs are included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by SGS-THOMSON Quality Assurance.

8.4. FLEXIBLE DISKS

The disk media submitted must be single-sided, single density, 8-inch, EFDOS/MDOS* compatible floppies. The customer must clearly label the disk with the ROM pattern file name. The minimum EFDOS/MDOS* system files as well as the absolute binary object file (file name. LO type of file) from the 6804 cross-assembler must be on the disk. An object file made from a memory dump, using the ROLLOUT command is also admissable. Consider submitting a source listing as well as : file name,. LX (DEVICE/EXORciser loadable format). This file will of course be kept confidential and is used 1) to speed up the process in house if any problems arises, and 2) to speed up our customer to factory interface if an user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

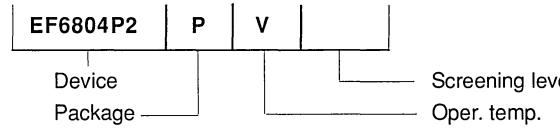
MDOS* (fully compatible with EFDOS) is Motorola's Disk Operating System available on development systems Such as EXORcisers, EXORsets, etc.

EFODS is SGS-THOMSON Disk Operating System available on development systems such as DEVICE/EXOR...

Whenever ordering a custom MCU is required, please contact your local SGS-THOMSON representative or SGS-THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS-THOMSON representative.

* Requires prior factory approval.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device	Package					Oper. Temp			Screening Level			
	C	J	P	E	FN	L*	V	T	Std	D		
EF6804P2			●		●	●	●		●	●		
Examples : EF6804P2P, EF6804P2FN, EF6804P2PV, EF6804P2FNV.												

Package : C : Ceramic DIL, **P :** Plastic DIL, **FN :** PLCC.
Oper. temp. : L* : 0°C to + 70°C, **V :** – 40°C to + 85°C, **T :** 40°C to + 105°C, * . may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D

EXORciser is a registered trademark to Motorola Inc

These specifications are subject to change without notice.
Please inquire with our sales offices about the availability of the different packages.

EF6804 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....

.....

ROM capacity required : | | | | bytes

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 40°C / + 85°C
- ❑ – 40°C / + 105°C

Package
- ❑ Plastic
- ❑ PLCC

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
- ❑ SGS-THOMSON Microelectronics reference

  .....
- ❑ Special customer data reference*

  .....

Number of interrupt vector : | |

Quality level :
- ❑ STD
- ❑ D
- ❑ Other* (customer's quality specification ref.) :

  .....

Software developped by :
- ❑ SGS-THOMSON Microelectronics application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
- ❑ EPROM Reference :
- ❑ EFDOS/MDOS* disk file
  - ❑ 8" floppy
  - ❑ 5" 1/4 floppy
- ❑ Other *

OPTION LIST

-Oscillator input
- ❑ Xtal
- ❑ RC

- Interrupt Trigger
- ❑ Edge-sensitive
- ❑ Level-and edge-sensitive

- Port A output drive (4 I/Os)
- ❑ CMOS and TTL
- ❑ TTL only
- ❑ Open Drain

- Port B output drive
- ❑ CMOS and TTL
- ❑ TTL only
- ❑ Open drain

- Port C output drive :
- ❑ CMOS and TTL
- ❑ TTL only
- ❑ Open drain

* Requires prior factory approval

Yearly quantity forecast :

- start of production date :
- for a shipment period of :

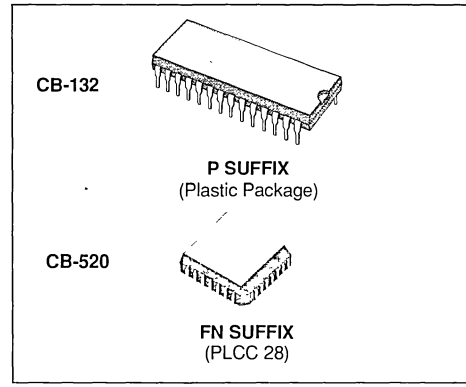
CUSTOMER CONTACT NAME DATE : SIGNATURE :

HCMOS 8–BIT MICROCOMPUTER

PRELIMINARY INFORMATION

SECTION 1 – INTRODUCTION

The EF68HC04P3 microcomputer unit (MCU) is a member of the EF68HC04 family of very low cost and low power single chip microcomputers. This 8 bit microcomputer contains a CPU, on-chip clock, ROM, RAM, I/O, and timer. It is designed for the user who needs an economical microcomputer with the proven capabilities of the EF6800 based instruction set. The following are some of the hardware and software highlights of the EF68HC04P3 MCU.



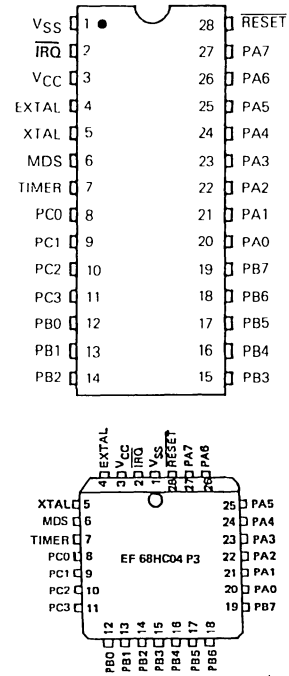
HARDWARE FEATURES

- Low power HCMOS
- Power saving stop and wait modes
- Single 2.0 to 6.0 volt power supply
- 8-bit architecture
- Fully static operation
- Pin compatible with the EF6805P2 and EF6804P2
- 124 bytes of on-chip RAM with standby mode
- 2 Kbytes of program ROM including 356 bytes for self-check program
- 72 bytes of user data ROM for look-up tables
- 20 CMOS compatible bidirectional I/O lines
- On-chip clock generator
- Extensive self-check capability allowing complete functional test of the chip (including ROM content)
- Master reset and power-on-reset
- 8-bit timer with 7-bit software programmable prescaler
- Timer pin programmable as input or output
- Complete development system support on device®

SOFTWARE FEATURES

- Similar to EF6800 family
- Byte efficient instruction set
- Easy to program
- True bit manipulation
- Stop, wait and bit manipulation instructions
- Bit test and branch instructions
- Versatile interrupt handling
- Separate flags for normal and interrupt processing
- True LIFO 4-level stack eliminating stack pointer
- Maskable timer interrupt

PIN CONNECTIONS



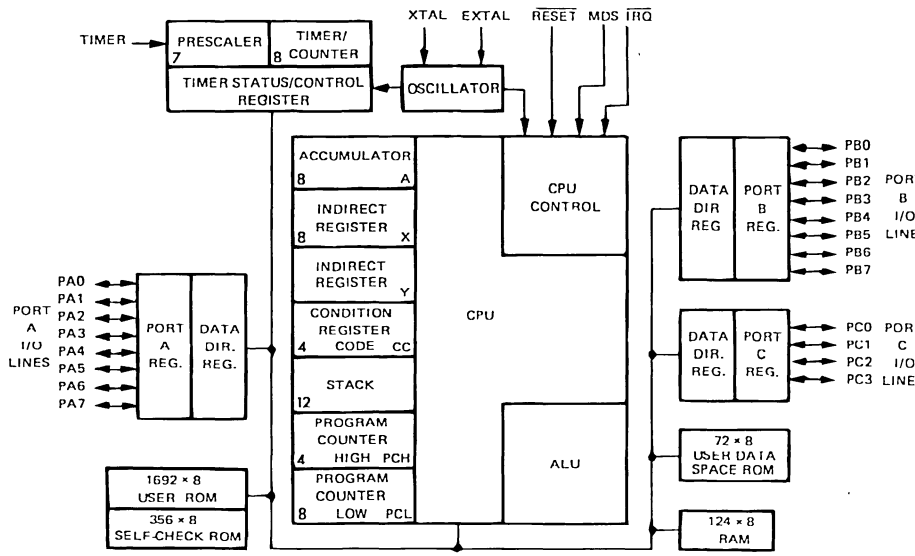
- Versatile indirect registers
- Conditional branches
- Single instruction memory examine/change
- 9 powerful addressing modes

USER SELECTABLE OPTIONS

- Crystal or low-cost resistor oscillator option
- Mask selectable internal clock generator options
- Mask selectable edge or level sensitive interrupt pin
- Program rom protection option
- Optional pull-down devices on I/O lines
- Optional pull-up devices on INT and RESET pins

DEVICE® is SGS THOMSON' Microelectronics development/emulation tool.

Figure 1.1 : EF68HC04P3 MCU Block Diagram.



Note : 8–Bit Indirect Registers X and Y, although shown as part of the CPU are actually located in the 124 x 8 RAM at locations $80 and $81

SECTION 2

FUNCTIONAL PIN DESCRIPTION, MEMORY, CPU, AND REGISTERS

This section provides a description of the functional pins, memory spaces, the central processing unit (CPU), and the various registers and flags.

2.1. FUNCTIONAL PIN DESCRIPTION

2.1.1. V_{CC} AND V_{SS}. Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

2.1.2 $\overline{IRQ}$. This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **4.1. INTERRUPT** for additional information.

2.1.3. XTAL AND EXTAL. These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor and capacitor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to **4.4. INTERNAL CLOCK GENERATOR OPTIONS** for recommendations concerning these inputs.

2.1.4. TIMER. In the input mode, the timer pin is connected to the prescaler input and serves as the timer clock or as an enable input for the internal clock. In the output mode, the timer pin signals that a time out of the timer has occurred. Refer to **SECTION 3 TIMER** for additional information.

2.1.5. RESET. The RESET pin is used to restart the processor of the EF68HC04P3 to the beginning of a program. This pin, together with the MDS pin is also used to select the operating mode of the EF68HC04P3. If the MDS pin is at zero volts, the normal mode is selected and the program counter is loaded with the user restart vector. However, if the MDS pin is at + 5 volts, then pins PA6 and PA7 are decoded to allow selection of the operating mode. Refer to **4.3. RESET** for additional information.

2.1.6. MDS. The MDS (mode select) pin is used to place the MCU into special operating modes. If MDS is held at + 5 volts at the exit of the reset state, the decoded state of PA6 and PA7 is latched to determine the operating mode (single-chip, self-check, or ROM verify). However, if MDS is held at zero volts at the exit of the reset state, the single-chip operating mode is automatically selected (regardless of PA6 and PA7 state).

For those users familiar with the EF6801 microcomputer, mode selection is similar but much less complex in the EF68HC04P3. No special external diodes, switches, transistors, etc. are required in the EF68HC04P3.

2.1.7. PORT INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC3). These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to **SECTION 5 INPUT/OUTPUT PORTS** for additional information.

2.2. MEMORY

The MCU operates in three different memory spaces : program space, data space, and stack space. A representation of these memory spaces is shown in figure 2.1. The program space (figure 2.1a) contains all of the instructions that are to be executed, as well as the data required for the immediate addressing mode instructions, and the self-check and user vectors. The data space (figure 2.1b) contains all of the RAM locations, plus I/O locations and some ROM used for storage of tables and constants. The stack space (figure 2.1c) contains RAM which is used for stacking subroutine and interrupt return addresses.

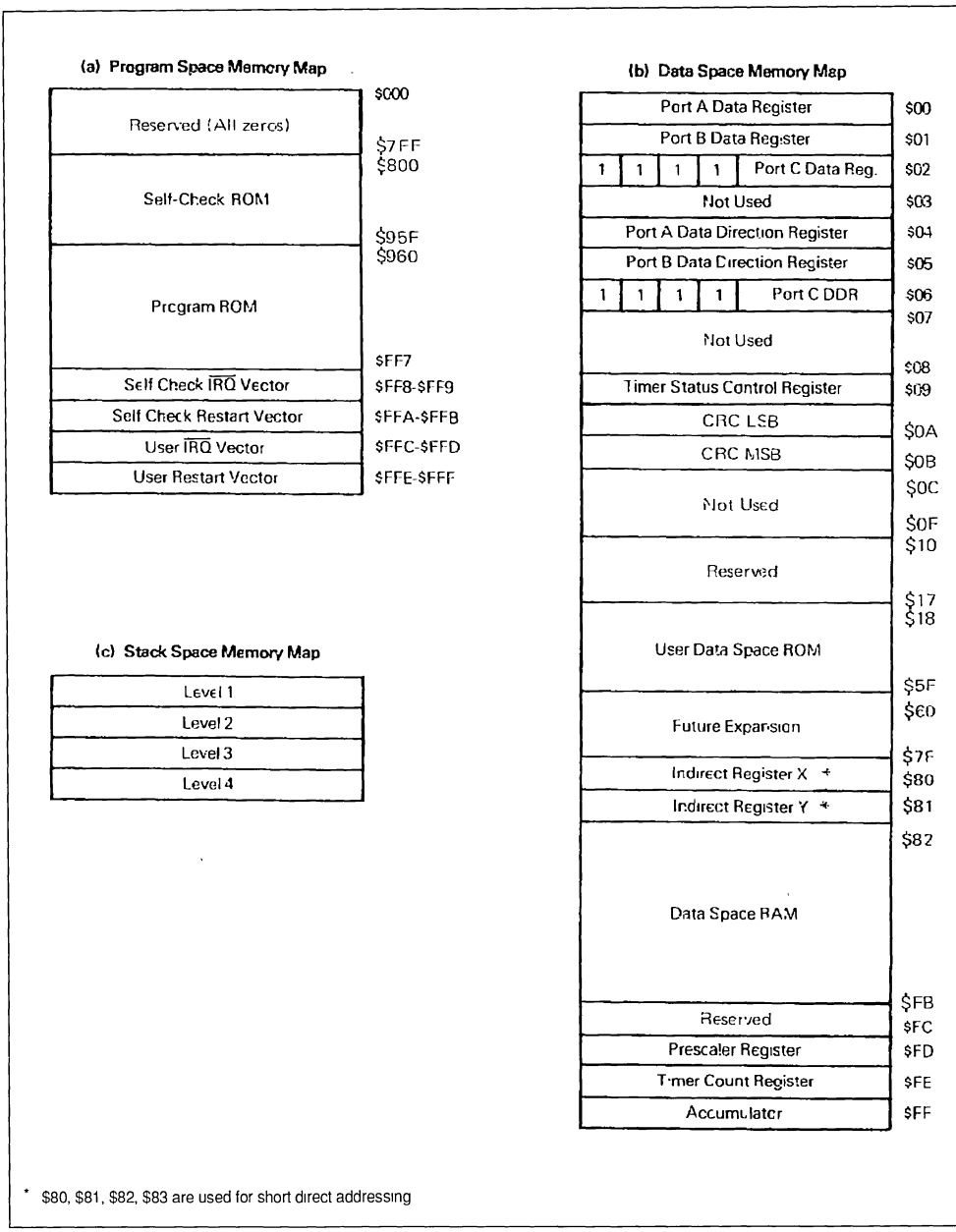
The MCU is capable of addressing 4096 bytes of program space memory with its program counter and 256 bytes of data space memory with its instructions. The data space memory contains three bytes for port data registers, three bytes for port data direction registers, one byte for timer status/control, 72 bytes ROM, 124 bytes RAM (which includes two bytes for XP and YP indirect registers), two bytes for timer prescaler and count registers, and one byte for the accumulator. The program space section contains 2048 bytes of ROM including 356 bytes of self-check ROM and 8 bytes of vectors for self-check and user programs.

2.2.1. PROGRAM ROM PROTECTION. A manufacturing mask option is available to the user to enable program ROM protection. If enabled, this security feature prevents the ROM contents being output during any operating mode.

2.3. CENTRAL PROCESSING UNIT

The CPU of the EF68HC04 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal addresses, data, and control buses.

Figure 2.1 : EF68HC04P3 MCU Address Map.

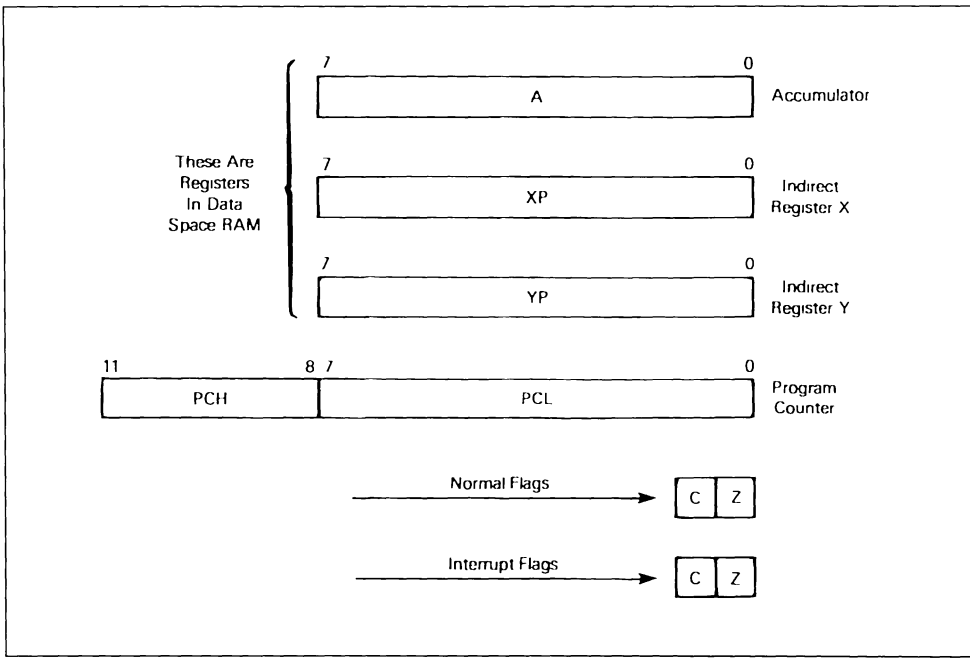


* $80, $81, $82, $83 are used for short direct addressing

2.4. REGISTERS

The EF68HC04 Family CPU has four registers and two flags available to the programmer. They are shown in figure 2.2 and are explained in the following paragraphs.

Figure 2.2 : Programming Model.



2.4.1. ACCUMULATOR (A). The accumulator is an 8-bit general purpose register used in all arithmetic calculations, logical operations, and data manipulations. The accumulator is implemented as the highest RAM location ($FF) in data space and thus implies that several instructions exist which are not explicitly implemented. Refer to **6.3. IMPLIED INSTRUCTIONS** for additional information.

2.4.2. INDIRECT REGISTERS (XP, YP). These two indirect registers are used to maintain pointers to other memory locations in data space. They are used in the register-indirect addressing mode, and can be accessed with the direct, indirect, short direct, or bit set/clear addressing modes. These registers are implemented as two of the 124 RAM locations ($80, $81) and as such generate implied instructions and may be manipulated in a manner similar to any RAM memory location in data space. Refer to 6.3. IMPLIED INSTRUCTIONS for additional information.

2.4.3. PROGRAM COUNTER (PC). The program counter is a 12-bit register that contains the address of the next ROM word to be used (may be opcode, operand, or address of operand). The 12-bit program counter is contained in PCL (low byte) and PCH (high nibble).

2.4.4. FLAGS (C, Z). The carry (C) bit is set on a carry or a borrow out of the ALU. It is cleared if the result of an arithmetic operation does not result in a carry or a borrow. The (C) bit is also set to the value of the bit tested in a bit test instruction, and participates in the rotate left instruction.

The zero (Z) bit is set if the result of the last arithmetic or logical operation was equal to zero, otherwise it is cleared.

There are two sets of these flags, one set is for interrupt processing, the other for all other routines. When an interrupt occurs, a context switch is made from the program flags to the interrupt flags (inter-

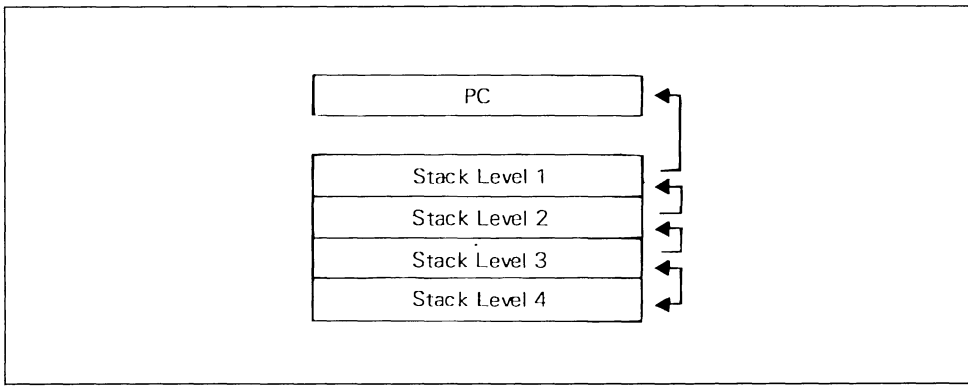
rupt mode). An RTI forces the context switch back to the program flags (program mode). While in either mode, only the flags for that mode are available. Further, the interrupt flags will not be cleared upon entering the interrupt mode. Instead, the flags will be as they were at the exit of the last interrupt mode. Both sets of flags are cleared by reset.

2.4.5. STACK. There is a true LIFO stack incorporated in the EF68HC04P3 which eliminates the need for a stack pointer. Stack space is implemented in separate RAM (12-bits wide) shown in figure 2.1c.

Whenever a subroutine call (or interrupt) occurs, the contents of the PC are shifted into the top register of the stack. At the same time (same cycle), the top register is shifted to the next level deeper. This happens to all registers with the bottom register falling out the bottom of the stack.

Whenever a subroutine or interrupt return occurs, the top register is shifted into the PC and all lower registers are shifted up one level higher. Stack level 4 is loaded with the previous content of stack level 3. The stack RAM is four levels deep.

Figure 2.3 : Stack Operation when RTS or RTI occurs.



2.4.6. CRC REGISTERS. Two eight bit registers are implemented in RAM primarily for use in self-check and ROM verify modes. These two registers are memory mapped in data space at addresses $0A (CRC low) and $0B (CRC high).

Provided no write or read/modify/write operation is performed to change the contents of these two locations, the registers are configured to perform CRC calculations. Hence by simply reading a register, a pseudo random number may be generated. If a write or read/modify/write is performed on addresses $0A or $0B then the CRC circuitry is disabled and both registers can be used as a RAM location until such time as a RESET enables the CRC circuitry again.

SECTION 3

3.1. INTRODUCTION

A block diagram of the EF68HC04P3 timer circuitry is shown in figure 3.1. The timer logic in the MCU is comprised of a simple 8-bit counter (timer count register, TCR) with a 7-bit prescaler, and a timer status/control register (TSCR). The timer count register, which may be loaded under program control is decremented towards zero by a clock input (prescaler output). The prescaler is used to extend the maximum interval of the overall timer. The prescaler tap is selected by bits 0-2 (PS0-PS2) of the timer status/control register. Bits PS0-PS2 control the actual division of the prescaler within the range of divide-by-1 (2^0) to divide-by-128 (2^7). The timer count register (TCR) and prescaler are decremented on rising clock edges. The coding of the TCSR PS0-PS2 bits produces a division in the prescaler as shown in table 3.1.

Table 3.1 : Prescaler Coding Table.

PS2	PS1	PS0	Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

TIMER pin may be programmed as either an output or an input depending on the status of DOUT and TOUT bits. Three modes are available.

Output mode (TOUT = 1)

The TIMER pin is connected to the DOUT latch. Therefore, the timer prescaler is clocked by the internal SYNC pulse. (Divide-by-12, -24 or -48 of the internal oscillator according to selected mask option, refer to 4.4. INTERNAL CLOCK GENERATOR OPTIONS). The prescaler then divides its clock input by a value determined by the coding of the TSCR bits PS0-PS2 as shown in table 3.1. The divided prescaler output then clocks the 8-bit timer count register (TCR). When the TCR count reaches zero, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high. The low-to-high TMZ bit transition is used to latch the DOUT bit of the TSCR and provides it for the TIMER pin.

Controlled mode (TOUT = 0, DOUT = 1)

The TIMER pin is an input which controls the counting by the prescaler-timer. When high, it enables counting Counting is disabled as long as this input remains low. Operation is similar to that described for the output mode.

Clock input mode (TOUT = 0, DOUT = 0)

The TIMER pin is connected directly to the prescaler input. Therefore the timer prescaler is clocked by the signal applied from the TIMER pin. Operation is similar to that described for the output mode. The frequency of the signal applied to the TIMER pin must be less than $1/t_{byte}$ ($f_{osc} \div 12$, $\div 24$ or $\div 48$ according to selected mask option) because of internal synchronization.

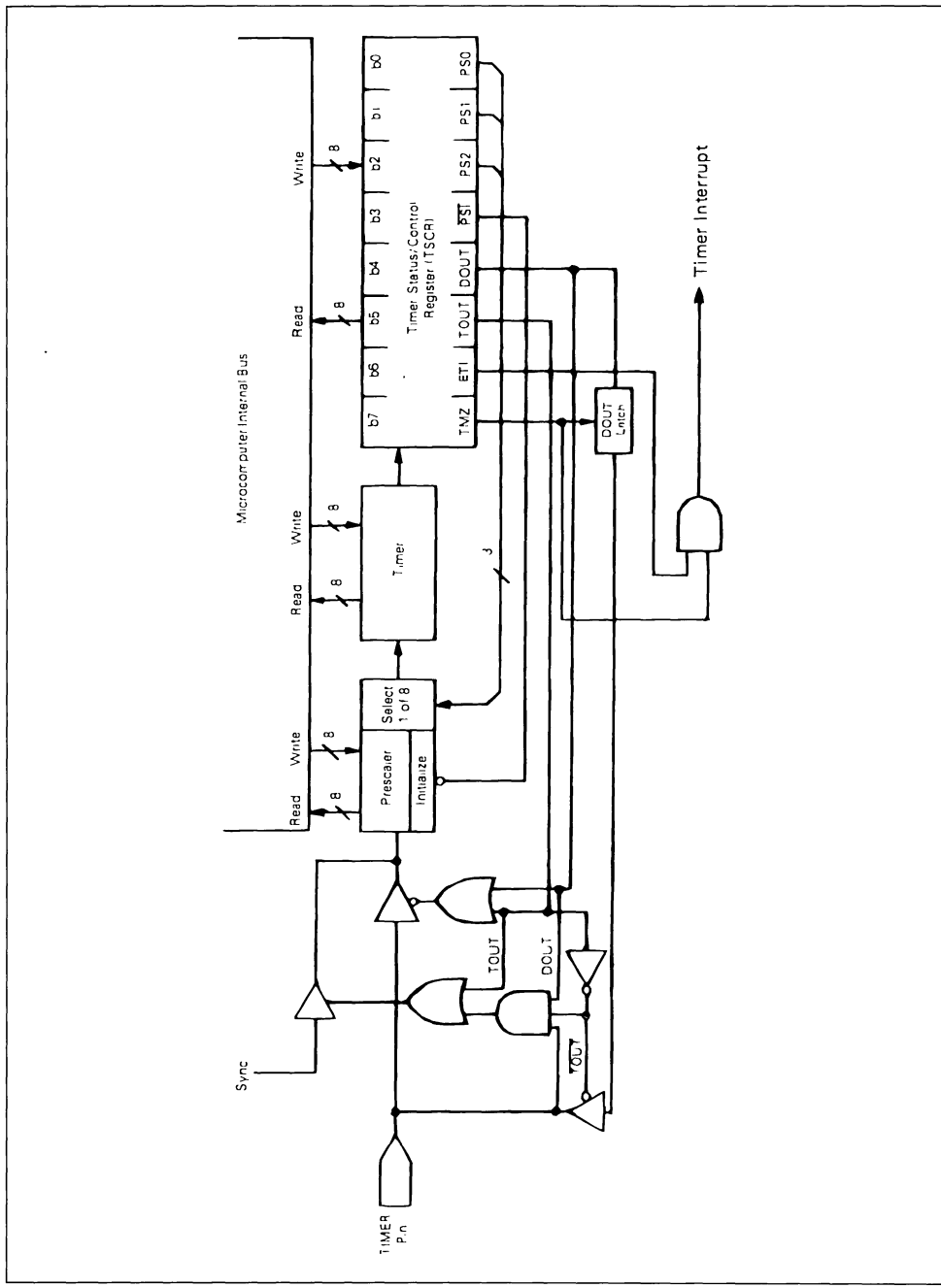
NOTE

TMZ is normally set to logic one when the timer times out (TCR count reaches $00) ; however, it may be set by a write of $00 to the TCR or by a write to bit 7 of the TSCR.

TMZ bit is cleared by a read-only of the TSCR even if TMZ bit is not concerned by this read.

Timer Status			
Timer Pin	TOUT	DOUT	Timer Mode
Input	0	0	Clock Input Mode
	0	1	Controlled Mode
Output	1	0	Output
	1	1	

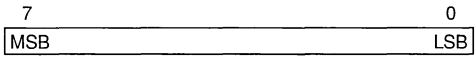
Figure 3.1 : Timer Block Diagram.



During reset, the timer count register and prescaler are set to $FF, while the timer status/control register is cleared to $00 and the DOUT LATCH (TIMER pin is in the high-impedance input mode) is forced to a logic high. The prescaler and timer count register are implemented in data space RAM locations ($FD, $FE) ; therefore, they are both readable and writeable. A write to either will predominate over the TCR decrement-to-$00 function ; i.e., if a write and a TCR decrement-to-$00 occur simultaneously, the write will take precedence, and the TMZ bit is not set until the next timer time out.

3.2. TIMER REGISTERS

3.2.1. TIMER COUNT REGISTER (TCR). The timer count register indicates the state of the internal 8-bit counter.



TCR Address = $FE

3.2.2. TIMER STATUS/CONTROL REGISTER (TSCR).

7	6	5	4	3	2	1	0
TMZ	ETI	TOUT	DOUT	PSI	PS2	PS1	PS0

TSCR Address = $09

b7, TMZ. Low-to-high transition indicates the timer count register has decremented to zero since the timer status/control register was last read. Cleared by a read of TSCR register if TMZ was read as a logic one.

b6, ETI. This bit, when set, enables the timer interrupt.

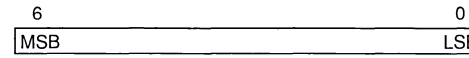
b5, TOUT. When low, this bit selects the input modes for the timer. When high, the output mode is selected.

b4, DOUT. Data sent to the timer output pin when TMZ is set high (output mode only). Choice of input mode (input mode only).

b3, PSI. Used to initialize the prescaler and inhibit its counting while PSI = 0. The initialized value is set to $FF. The timer count register will also be inhibited (contents unchanged). When PSI = 1 the prescaler begins to count downward.

b0, b1, b2, PS0-PS1-PS2. These bits are used to select the prescaler divide-by ratio ; therefore, effecting the clock input frequency to the timer count register.

3.2.3. TIMER PRESCALER REGISTER. The timer prescaler register indicates the state of the internal 7-bit prescaler. This 7-bit prescaler divide ratio is normally determined by bits PS0-PS2 of the timer status/control register (see table 3.1).



TPR Address = $FD

SECTION 4

INTERRUPT, POWER SAVING MODES, SELF-CHECK, RESET, AND INTERNAL CLOCK GENERATOR

4.1. INTERRUPT

There are two ways in which the MC68HC04P3 can be interrupted. Firstly by an external interrupt and secondly by a timer interrupt provided the ETI bit in TSCR is set. Note that both types of interrupt share the same vector ($FFC). The only way to differentiate a timer interrupt from an external interrupt is to test the TMZ bit (Timer Interrupt Request bit). The interrupt mask bit (IMASK), which controls interrupt processing, is not directly available to the programmer. It is set during reset and power up sequences to preclude any false or ghost interrupts from occurring, there after it is set during interrupt processing and cleared by the execution of an RTI instruction. To clear interrupt mask bit and so enable interrupts, an RTI instruction must be executed. A simple way to do this is to call a subroutine at the start of the program to do any required initialisation and return from the subroutine using an RTI instruction instead of RTS. See Figure 4.1 for an example of this technique.

Figure 4.4a illustrates the instruction processing sequence.

The external and timer interrupt sequences are detailed in Figure 4.4b. The interrupt sequence consists of one cycle during which :

- interrupt request latch is cleared
- interrupt condition code flags are selected
- PC is saved on the stack
- interrupt mask is set
- address of the IRQ vector location is loaded onto the PC

The vector locations $FFC/$FFD must contain the appropriate two-byte JMP instruction.

Figure 4.1 : Clearing the Interrupt Mask.

```
START    JSR INIT        Clear INT Mode Bit
           |
         etc.            Rest of Program
INIT  MVI # $FF, DDRA    Do any Required Initialisation
           |
         RTI             Use RTI to Clear INT MASK
$FFE     JMP START       Reset Vector
```

Internal processing of the interrupt continues until an RTI (return from interrupt) instruction is processed. During the RTI instruction, the interrupt mask is cleared, the program condition code flags are selected and the PC is restored. Providing no interrupt is pending the next instruction is then fetched and executed. An interrupt service routine must always finish with an RTI.

4.1.1. EXTERNAL INTERRUPTS. An external interrupt is requested by pulling the $\overline{IRQ}$ pin low. The $\overline{IRQ}$ pin can be pulled low by external circuitry.

The maximum response time to an external interrupt is 8 machine (tbyte) cycles. This includes five machine cycles for the longest instruction, plus one machine cycle for stacking PC and switching flags and a max of 2 cycles to synchronise $\overline{IRQ}$ input with the internal machine cycle frequency.

There are 2 external interrupt options available on the 68HC04P3, selectable by a manufacturing mask option.

4.1.1.1. IRQ Pin Pull-up Option

This is a manufacturing mask option for an internal high impedance pull-up device on the $\overline{IRQ}$ pin designed to reduce external component count. In a noisy environment however, it is recommended that a lower impedance external pull-up be used instead.

4.1.1.2. External Interrupt Edge Sensitive Option

A 1 to 0 (negative) transition on the $\overline{IRQ}$ pin will set the internal interrupt request latch. Prior to each instruction fetch, the interrupt request latch is tested and, if valid, an interrupt service sequence will be initiated at the end of the current instruction (providing the interrupt mask is clear). When the interrupt service routine is entered the interrupt request latch is cleared so that if a second edge occurs on the $\overline{IRQ}$ pin while the first interrupt is being serviced, it will be latched so that it can be serviced after the first interrupt sequence is complete. On return from an interrupt the interrupt request latch is tested before the next instruction is executed.

4.1.1.3. External Interrupt Level Sensitive Option

With this option there is no interrupt request latch, instead a check is made on the level of the $\overline{IRQ}$ pin after completion of each instruction and if low, an interrupt service sequence will be initiated. If on completion of this interrupt sequence the $\overline{IRQ}$ pin is low, a further interrupt will be recognised and the interrupt service sequence will be re-entered. As there is no interrupt request latch, the state of the $\overline{IRQ}$ pin during the interrupt service sequence will have no effect on program flow. The absence of the latch also means that a glitch on the interrupt pin of less than 60 interval clock cycles (max. instruction execution time) may not be recognised by the level sensitive option.

4.1.2. TIMER INTERRUPT. A timer interrupt is requested when the TMZ bit of the timer status/control register (TSCR) is set. The TMZ bit can be set either by the timer count register (TCR) reaching the zero state or by any program instruction that writes a one to the TMZ bit. Timer interrupt request is maskable by clearing ETI, bit 6 of TSCR (ETI is cleared on Reset). See Section 3.2.2 for more details.

4.2. STOP MODE

The STOP instruction places the EF68HC04P3 in its lowest power consumption mode. After a STOP instruction has been executed the interrupt mask is cleared and the internal oscillator is turned off causing all internal processing to be halted and the current consumption to drop to leakage levels. (see Section **7 ELECTRICAL SPECIFICATION**).

The contents of the timer status/control register, the accumulator and all data space RAM are unchanged by STOP providing that the supply voltage, V_{DD}, remains within data sheet limits. The processor can only be brought out of STOP mode by pulling the IRQ or RESET pins low. The timer is used to provide a delay, of 1920 external clock cycles, for the oscillator to stabilize during exit from STOP before processing is continued. Hence, the contents of timer count register (TCR) and the prescaler must be considered to be corrupted.

4.3. WAIT MODE

The WAIT instruction places EF68HC04P3 in a low power consumption mode. In WAIT mode, the clock is disabled from all internal circuitry except the timer circuit, halting all internal processing. The timer may continue to count down if PSI bit of TSCR is set. External interrupts are enabled. All other registers, memory and I/O lines remain in their last state. ETI bit of TSCR may be enabled by software prior to entering WAIT. This allows an exit from WAIT via a timer interrupt in addition to an external interrupt (IRQ) or RESET.

4.4. OPERATING MODES AND SELF-TEST

There are four operating modes on the 68HC04P3, one is the normal program execution mode (single chip mode), two are self test modes and the last is for Motorola internal use only (non user mode or NUM). The operating mode entered is determined during RESET by the state of the Mode Select pin (MDS) and the port pins PA6 and PA7. If MDS is held low at the exit from RESET then single chip mode will be selected, if however, MDS is high then the state of PB6 and PB7 determine the operating mode according to Figure 4.3.

4.4.1. SELF-CHECK MODE. Self-Check is one of the two self test modes of the 68HC04P3. It uses the on board cyclic redundancy checker (CRC) circuit to perform a very extensive functional check of the MCU by signature analysis. The simple external circuitry required to implement this test is shown in Figure 4.5a. Figure 4.6 illustrates the self check program flow, the RESET and Interrupts being supplied by the user. The status of the program is shown on the LEDs as shown beside the flow diagram, if the LEDs become "stuck" at one of these values then it indicates a fail in the previous section of the test.

4.4.2. ROM VERIFY MODE. ROM verify is the second of the two self test modes of the 68HC04P3. It uses the CRC to perform a signature analysis test of the Program ROM contents (Data ROM is tested in Self Check). The simple circuitry required to implement this test is shown in Figure 4.5b. The test is started by a RESET and successful completion will result in data output of $AA from CRCHI and $55 from CRCLO, thus illuminating the good LED, otherwise the bad LED will be lit. The ROM verify mode also gives access to the internal processor clocks sync and PHI1 via pins 18 and 19.

See Section **4.6. INTERNAL CLOCK GENERATOR OPTIONS** for more details.

Figure 4.3 : Mode Selection.

MDS			Mode
0	X	X	Single Chip
1	0	0	Single Chip
1	0	1	Self Check
1	1	0	NUM
1	1	1	ROM Verify

4.5. SELF–CHECK

The self-check capability of the EF68HC04P3 MCU provides an internal check to determine if the part a functional check of the MCU, connect it as shown in Figure 4.2a and monitor the LEDs for a 00100 ($04) pattern on port A. The MCU is left in the WAIT mode. A logical low signal applied to the IRQ pin places the MCU in the STOP mode. A 00101 ($05) pattern appears on port A. Another logical low signal applied on the IRQ pin enables exit from the STOP mode. The "final good" pattern (00110 –$06) appears on port A). To initiate a ROM self-check of the memory simply connect the circuit as shown in Figure 4.2b and check that the "good" LED turns on to indicate a good memory. The ROM verify uses a cyclical redundancy check (CRC) to conduct a ROM check by means of signature analysis circuit. This circuit consists of two 8-bit shift registers configured to perform the check using the CCITT polynominal. A manufacturing mask option inhibits the outputs of the CRC data and the ROM data until the final result is available in order to protect the program ROM when the option is selected.

Self–Check Flowchart

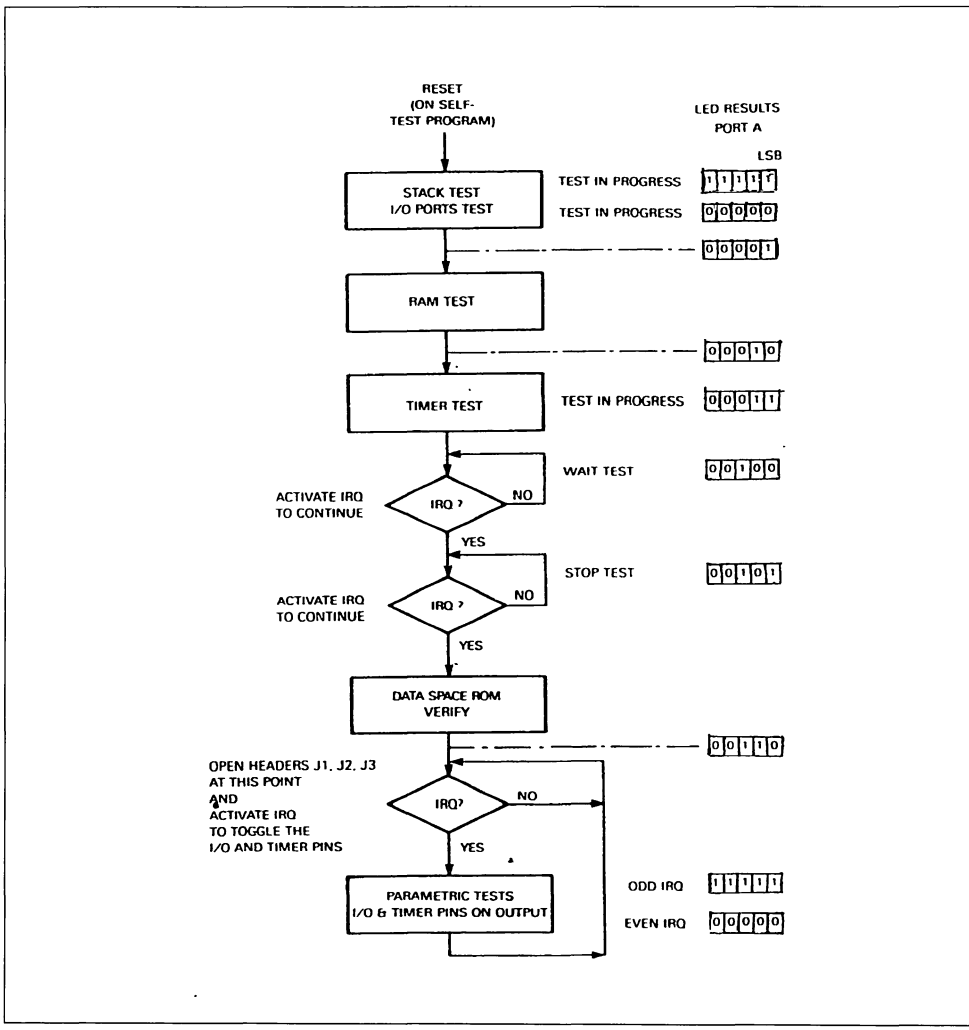
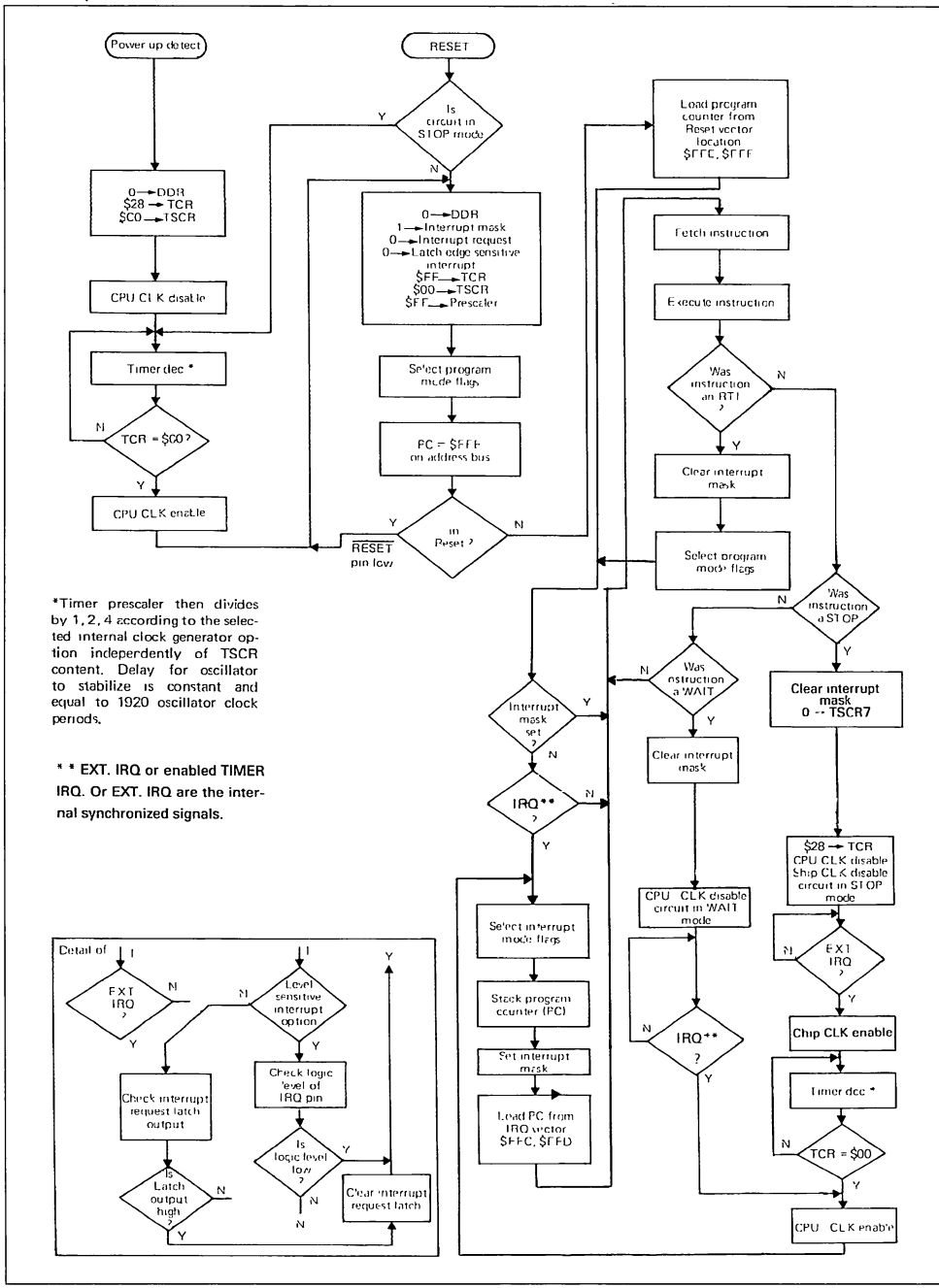


Figure 4.1 : STOP, WAIT, INTERRUPT and RESET Processing Flowchart.



*Timer prescaler then divides by 1, 2, 4 according to the selected internal clock generator option independently of TSCR content. Delay for oscillator to stabilize is constant and equal to 1920 oscillator clock periods.

*** * EXT. IRQ or enabled TIMER IRQ. Or EXT. IRQ are the internal synchronized signals.**

Figure 4.2 : ROM Verify Circuit.

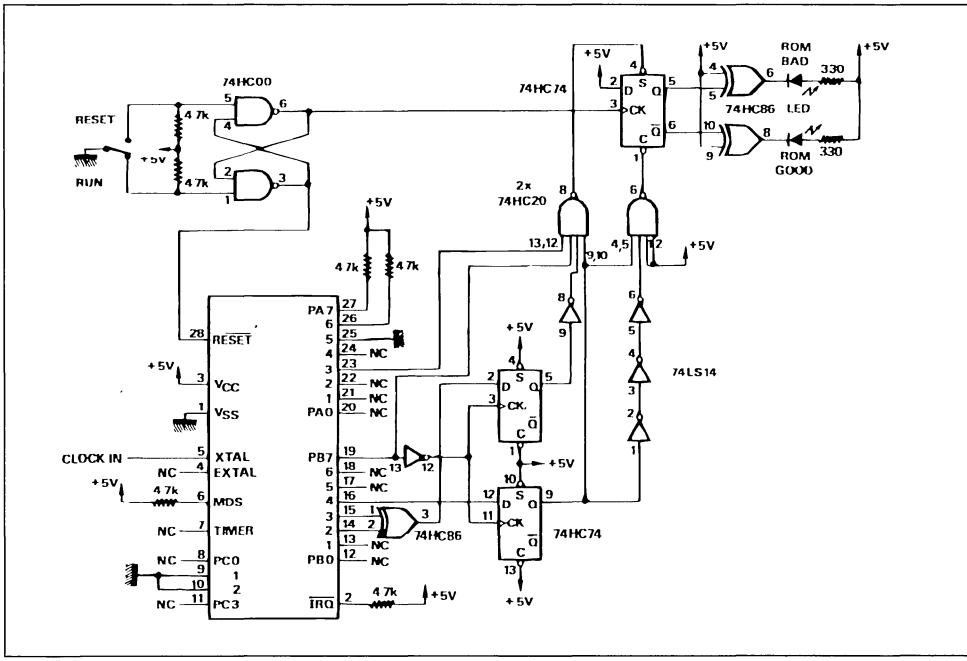
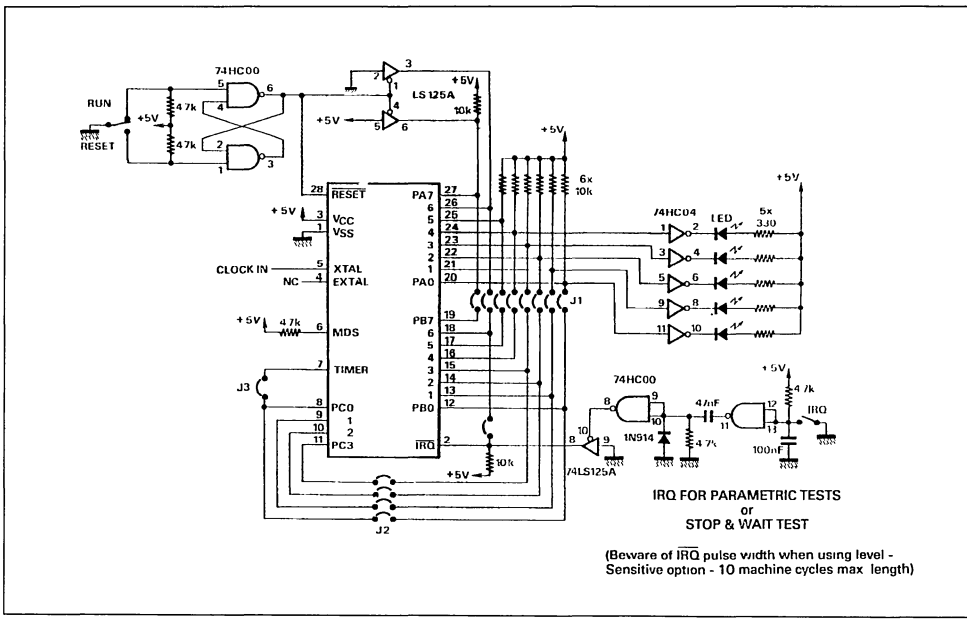


Figure 4.3 : Self-Check Circuit.



4.6. RESET

The MCU can be reset in two ways : by the external reset input (RESET) and by power-up detect (PUD).

4.6.1. RESET INPUT. This input can be used to reset the MCU internal state and provides an orderly software start-up procedure.

An external reset is achieved by pulling the RESET pin low for a minimum of two clock cycles. The oscillator start up delay is not implemented in this case.

After reset the MCU will be in the following state :

- All DDRs ($04, $05, $06) set to $00
- TCR ($FE) and prescaler register ($FD) set to $FF
- TSCR ($09) set to $00, DOUT latch set
- STOP and WAIT latches cleared
- All condition code flags cleared
- CRC ($0A, $0B) registers set to $FF
- All other registers are not affected
- Interrupt request latch cleared
- Restart vector loaded into PC
- Interrupt mask bit set

4.6.2. PUD. It occurs when a positive transition is detected on V_{CC} on initial power-up. No external RC network is needed. PUD is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision in this block for power-down detect. When the MCU is reset by means of PUD, an internal delay of 1920 oscillator clock periods is generated for the oscillator to stabilize. The MCU emerges from the reset condition at the end of this temporization.

4.6.3. RESET PIN PULL-UP OPTION. This is a manufacturing mask option for an internal high impedance pull-up device on the reset pin, designed to reduce external component count. In a noisy environment however, it is recommended that a lower impedance external pull-up be used instead.

4.7. INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor-capacitor, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different clock generator option connection methods are shown in Figure 4.4, crystal specifications and suggested PC board layouts are given in Figure 4.5, resistor-capacitor selection graph is given in Figure 4.6, and a timing diagram is illustrated in Figure 4.7. The crystal oscillator startup time is a function of many variables : crystal parameters (especially R_S), oscillator load capacitance (C_L), IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

The oscillator output frequency is internally divided by four, two or one depending upon a manufacturing mask option selection to produce the internal Ø1 and Ø2 clocks. The Ø1 clock is divided by twelve to produce a machine byte (cycle) clock (internal SYNC pulse). A byte cycle is the smallest unit needed to execute any operation (i.e., increment the program counter). An instruction may need two, four, or five byte cycles to be executed.

If the application is to be driven by an external signal, then the crystal mask option should be selected.

Figure 4.4 : Clock Generator Options.

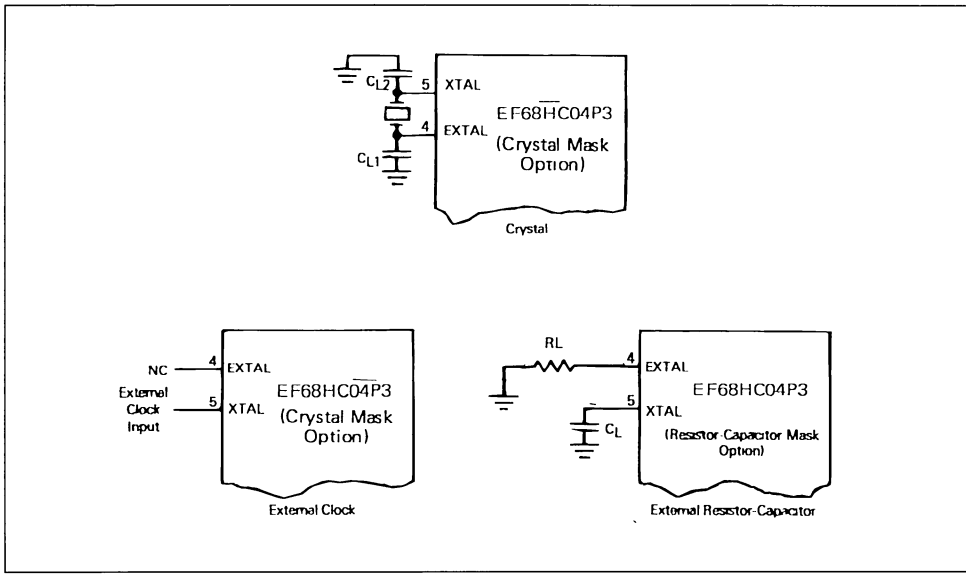


Figure 4.5 : Crystal Motional Arm Parameters and Suggested PC Board Layout.

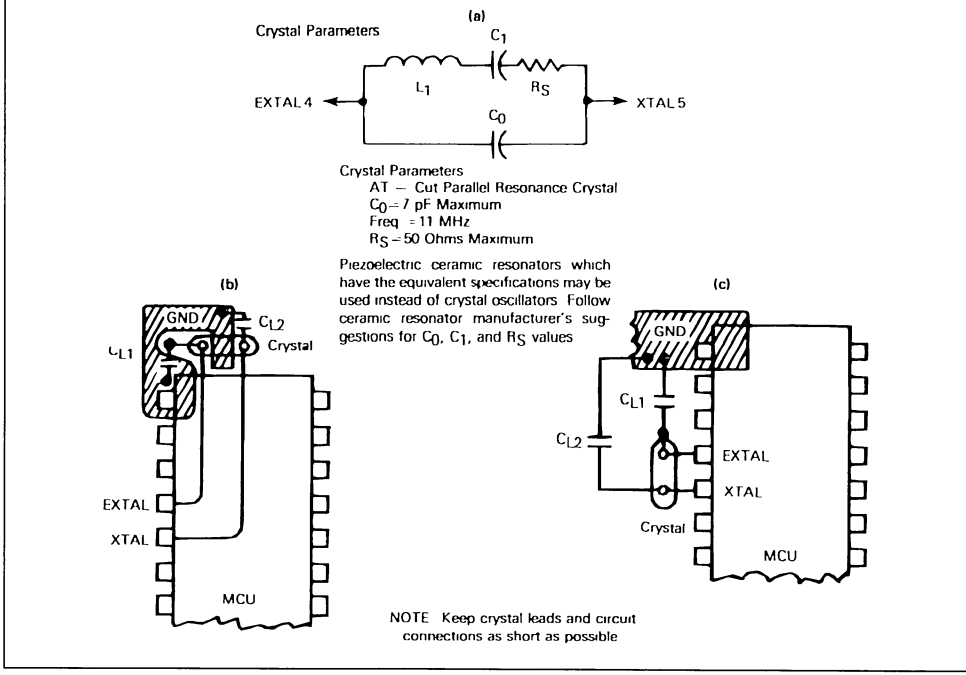


Figure 4.6 : Typical Frequency Selection for Resistor-capacitor Oscillator Option.

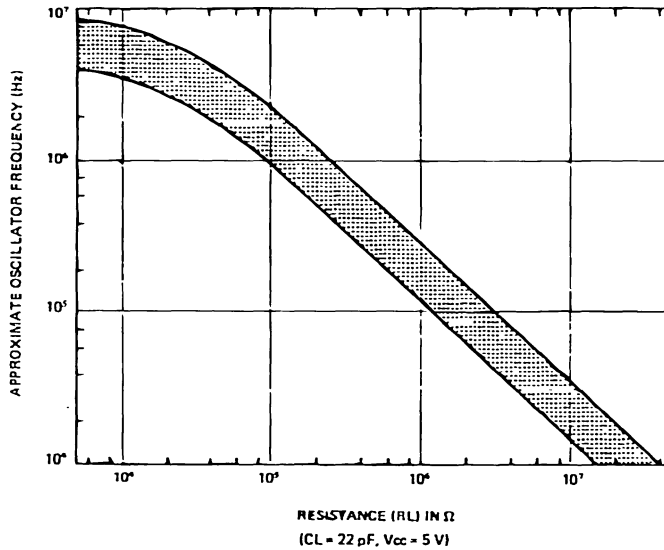
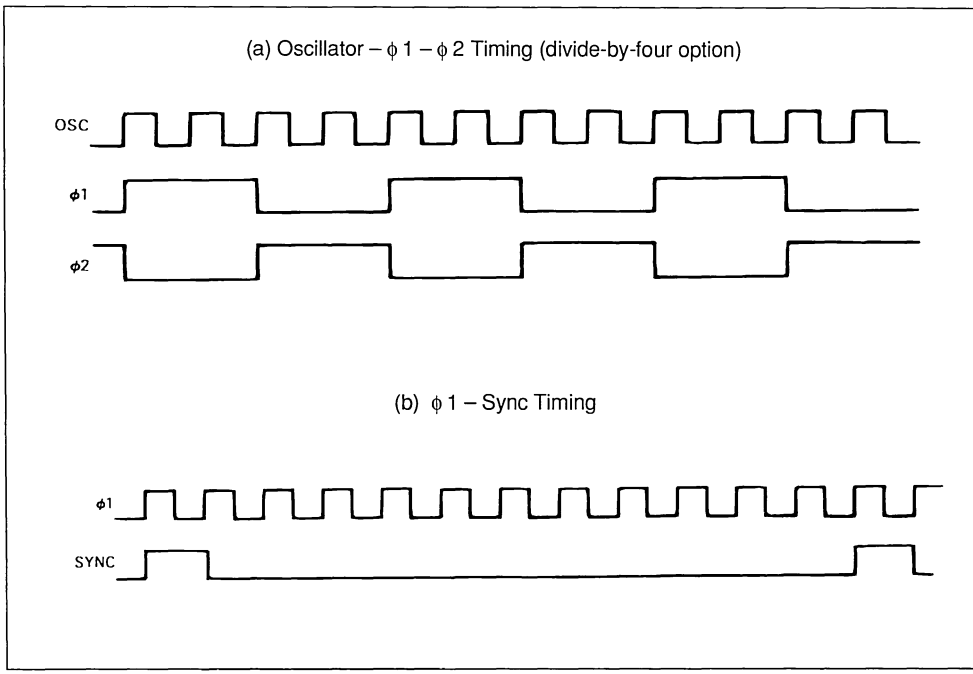


Figure 4.7 : Clock Generator Timing Diagram.



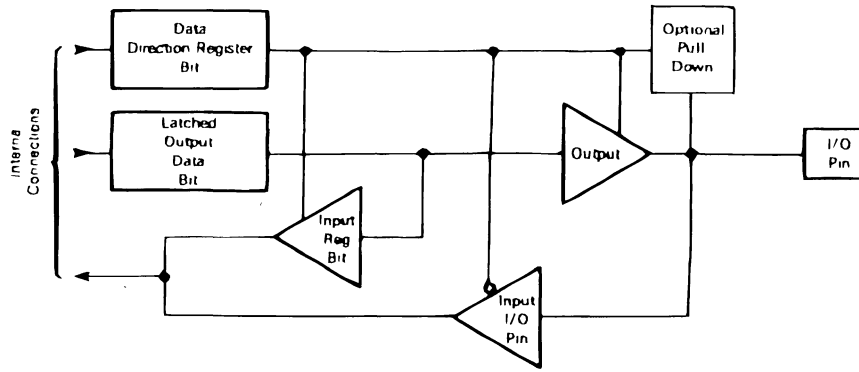
SECTION 5

5.1. INPUT/OUTPUT

There are 20 input/output pins. All pins (port A, B, and C) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset, all the DDRs are initialized to a logic zero state to put the ports in the input mode. The port output registers are not initialized on reset but should be initialized before changing the DDR bits to avoid undefined levels. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading ; see figure 5.1. A manufacturing mask option enables the choice of additional pull-down devices on all I/O pins (Selection in 5 groups : PA7, PA(5:6), PA(1:4), Port B, PA0 + Port C).

The address map in figure 2.1 gives the address of data registers and DDRs. The register configuration is discussed under the registers paragraph below.

Figure 5.1 : Typical I/O Circuitry.



Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	HI-Z	Pin

The latched output data bit (see figure 5.1) may always be written. Therefore, any write to a port writes to all of its data bits even though the port DDR is set to input. This may be used to initialize the data registers and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and corresponds to the latched output data when the DDR is an output (1).

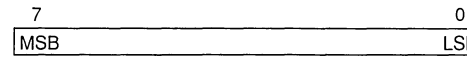
5.2. REGISTERS

The registers described below are implemented as RAM locations and thus may be read or written.

5.2.1. PORT DATA REGISTER. The source of data read from the port data register will be the port I/O pin or previously latched output data depending upon the contents of the corresponding data direction register (DDR). The destination of data written to the port data register will be an output data latch. If the corresponding data direction register (DDR) for the port I/O pin is programmed as an output, the data will then appear on the port pin.

Port A Address = $00
Port B Address = $01
Port C Address = $02 (Bits 0-3)

5.2.2. PORT DATA DIRECTION REGISTER. The port DDRs configure the port pins as either inputs or outputs. Each port pin can be programmed individually to act as an input or an output. A zero in the pins corresponding bit position will program that pin as an input while a one in the pins corresponding bit position will program that pin as an output. During reset the DDRs are initialized to $00.



Port A Address = $04
Port B Address = $05
Port C Address = $06 (Bits 0-3)

5.3. PULL DOWN DEVICE OPTION

The implementation of pull down devices on particular groupings of I/O port pins is a manufacturing mask option available to the user. This is typically of benefit in applications where keyboards are interfaced directly to the MCU and similar situations.

The permitted groupings are as follows :

- Port C
- Port C and Pin PA0
- Pins PA1, PA2, PA3, PA4
- Pins PA5, PA6
- Pin PA7
- Pins PA4, PA5, PA6, PA7
- Pins PA0, PA1, PA2, PA3
- Pins PB3, PB4, PB5, PB6, PB7
- Pins PB1, PB2
- Pin PB0

Note : That all the pull down device is disabled when the port pin is programmed as an output.

SECTION 6

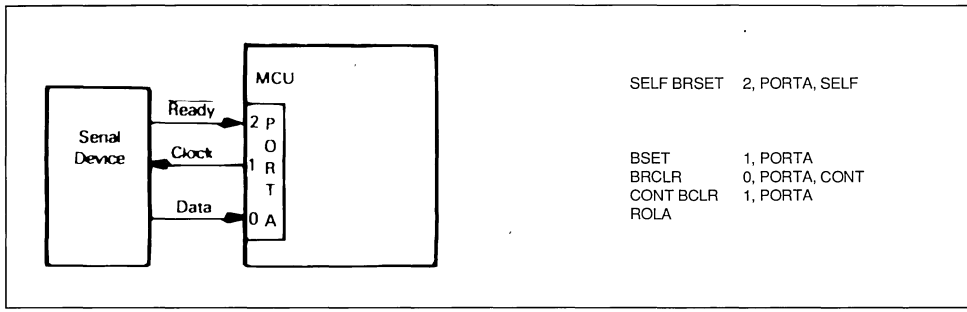
SOFTWARE AND INSTRUCTION SET

6.1. SOFTWARE

6.1.1. BIT MANIPULATION. The EF68HC04P3 MCU has the ability to set or clear any register or single random access memory (RAM) writable bit with a single instruction (BSET, BCLR). Any bit in data space, including ROM, can be tested, using the BRSET and BRCLR instructions, and the program may branch as a result of its state. The carry bit equals the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in figure 6.1 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line (to clock data one bit at a time, MSB first, out of the device). The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in the accumulator.

Figure 6.1 : Bit Manipulation Example.



6.1.2. ADDRESSING MODES. The EF68HC04P3 MCU has nine addressing modes which are explained briefly in the following paragraphs. The EF68HC04P3 deals with objects in three different address spaces : program space, data space, and stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains all of the RAM locations, XP and YP registers, accumulator, timer, I/O locations, and some ROM (for storage of tables and constants). Stack space contains RAM for use in stacking the return addresses for subroutines and interrupts.

The term "Effective Address" (EA) is used in describing the address modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

6.1.2.1. Immediate

In the immediate addressing mode, the operand is located in program ROM and is contained in a byte following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

6.1.2.2. Direct

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the 256 bytes in data space memory with a single two-byte instruction.

6.1.2.3. Short Direct

The MCU also has four locations in data space RAM ($80, $81, $82, $83) which may be used in a short-direct addressing mode. In this mode the opcode determines the data space RAM location, and the instruction is only one byte. Short direct addressing is a subset of the direct addressing mode. (Note : $80 and $81 are the X and Y register locations).

6.1.2.4. Extended

In the extended addressing mode, the effective address is obtained by concatenating the four least significant bits of the opcode with the byte following the opcode (12-bit address). Instructions using the extended addressing mode (JMP, JSR) are capable of branching anywhere in program space. An extended addressing mode instruction is two bytes long.

6.1.2.5. Relative

The relative addressing mode is only used in conditional branch instructions. In relative addressing, that address is formed by adding the sign extended lower five bits of the opcode (the offset) to the program counter if and only if the condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from - 15 to + 16 from the opcode address. The programmer need not worry about calculating the correct offset when using the assembler since it calculates the proper offset and checks to see if it is within the span of the branch.

6.1.2.6. Bit Set/Clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any bit in the 256 locations of data space memory, which can be written to, can be set or cleared.

6.1.2.7. Bit Test and Branch

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit and condition (set or clear) which is to be tested is included in the opcode, and the data space address of the byte to be tested is in the single byte immediately following the opcode byte. The third byte is sign extended to twelve bits and becomes the offset added to the program counter if the condition is true. The single three-byte instruction allows the program to branch based on the condition of any bit in data space memory. The span of branching is from – 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry flag.

6.1.2.8. Register-Indirect

In the register-indirect addressing mode, the operand is at the address (in data space) pointed to by the contents of one of the indirect registers (X or Y). The particular X or Y register is selected by bit 4 of the opcode. Bit 4 of the opcode is then decoded into an address which selects the desired X or Y register ($80 or $81). A register-indirect instruction is one byte long.

6.1.2.9. Inherent

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. These instructions are one byte long.

6.2. INSTRUCTION SET

The EF68HC04P3 MCU has a set of 44 basic instructions, which when combined with nine addressing modes produce 244 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

6.2.1. REGISTER/MEMORY INSTRUCTIONS. Most of these instructions use two operands. One operand is the accumulator and the other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operands. Refer to table 6-1.

6.2.2. READ-MODIFY-WRITE INSTRUCTIONS. These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. There are ten instructions which utilize read-modify-write cycles. All INC and DEC forms along with all bit manipulation instructions use this method. Refer to table 6-2.

6.2.3. BRANCH INSTRUCTIONS. The branch instructions cause a branch from the program when a certain condition is met. Refer to table 6-3.

6.2.4. BIT MANIPULATION INSTRUCTIONS. These instructions are used on any bit in data space memory. One group either sets or clears. The other group performs the bit test branch operations. Refer to table 6-4.

Figure 6.1 : Register/memory Instructions.

		Addressing Modes																			
		Indirect				Immediate			Direct			Inherent			Extended			Short-Direct			
Function	Mnem	Opcode XP	Opcode YP	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Load A from Memory	LDA	E0	F0	1	4	E8	2	4	F8	2	4	–	–	–	–	–	–	AC-AF	1	4	1
Load XP from Memory	LDXI	–	–	–	–	B0	3	4	–	–	–	–	–	–	–	–	–	–	–	–	4
Load YP from Memory	LDYI	–	–	–	–	B0	3	4	–	–	–	–	–	–	–	–	–	–	–	–	4
Store A in Memory	STA	E1	F1	1	4	–	–	–	F9	2	4	–	–	–	–	–	–	BC-BF	1	4	2
Add to A	ADD	E2	F2	1	4	EA	2	4	FA	2	4	–	–	–	–	–	–	–	–	–	–
Subtract from A	SUB	E3	F3	1	4	EB	2	4	FB	2	4	–	–	–	–	–	–	–	–	–	–
Arithmetic Compare with Memory	CMP	E4	F4	1	4	EC	2	4	FC	2	4	–	–	–	–	–	–	–	–	–	–
AND Memory to A	AND	E5	F5	1	4	ED	2	4	FD	2	4	–	–	–	–	–	–	–	–	–	–
Jump to Subroutine	JSR	–	–	–	–	–	–	–	–	–	–	–	–	–	8 (TAR)	2	4	–	–	–	3
Jump Unconditional	JMP	–	–	–	–	–	–	–	–	–	–	–	–	–	9 (TAR)	2	4	–	–	–	3
Clear A	CLRA	–	–	–	–	–	–	–	FB	2	4	–	–	–	–	–	–	–	–	–	–
Clear XP	CLRX	–	–	–	–	–	–	–	FB	2	4	–	–	–	–	–	–	–	–	–	–
Clear YP	CLRY	–	–	–	–	–	–	–	FB	2	4	–			–	–	–	–	–	–	–
Complement A	COMA	–	–	–	–	–	–	–	–	–	–	B4	1	4	–	–	–	–	–	–	–
Move Immediate Value to Memory	MVI	–	–	–	–	B0	3	4	B0	3	4	–	–	–	–	–	–	–	–	–	5
Rotate A Left and Carry	ROLA	–	–	–	–	–	–	–	–	–	–	B5	1	4	–	–	–	–	–	–	–
Arithmetic Left Shift of A	ASLA	–	–	–	–	–	–	–	FA	2	4	–	–	–	–	–	–	–	–	–	–

SPECIAL NOTES

1 In Short-Direct addressing, the LDA mnemonic represents opcode AC, AD, AE, and AF This is equivalent to RAM locations $80 (AC), $81 (AD), $82 (AE), and $83 (AF)

2. In Short-Direct addressing, the STA mnemonic represents opcode BC, BD, BE, and BF This is equivalent to RAM locations $80 (BC), $81 (BD), $82 (BE), and $83 (BF).

3. In Extended addressing, the four LSBs of the opcode (Mnemonic JSR and JMP) are formed by the four MSBs of the target address (TAR).

4. In Immediate addressing, the LDXI and LDYI are mnemonics which are recognized as follows

   LDXI = MVI $80,data

   LDYI = MVI $81,data Where data is a one-byte hexadecimal number

5 The MVI instruction refers to both Immediate and Direct addressing.

Figure 6.2 : Read-Modify-Write Instructions.

Function	Mnem	Addressing Modes										Special Notes
		Indirect				Direct			Short-Direct			
		Opcode		# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	
		XP	YP									
Increment Memory Location	INC	E6	F6	1	4	FE	2	4	A8-AB	1	4	1, 3
Increment A	INCA					FE	2	4				
Increment XP	INCX								A8	1	4	
Increment YP	INCY								A9	1	4	
Decrement Memory Location	DEC	E7	F7	1	4	FF	2	4	B8-BB	1	4	2, 4
Decrement A	DECA					FF	2	4				
Decrement XP	DECX								B8	1	4	
Decrement YP	DECY								B9	1	4	

SPECIAL NOTES

1. In short-direct addressing, the INC mnemonic represents opcode A8, A9, AA, and AB. These are equivalent to RAM locations $80 (A8), $81 (A9), $82 (AA), and $83 (AB).
2. In short-direct addressing, the DEC mnemonic represents opcode B8, B9, BA, and BB. These are equivalent to RAM locations $80 (B8), $81 (B9), $82 (BA), and $83 (BB).
3. In indirect addressing, the INC mnemonic represents opcode E6 or F6, and causes the location pointed to by XP (E6 opcode) or YP (F6 opcode) to be incremented.
4. In indirect addressing, the INC mnemonic represents opcode E7 or F7, and causes the location pointed to by XP (E7 opcode) or YP (F7 opcode) to be incremented.

Table 6.3 : Branch Instructions.

		Relative Addressing Mode			
Function	**Mnem**	**Opcode**	**# Bytes**	**# Cycles**	**Special Notes**
Branch if Carry Clear	BCC	40-5F	1	2	1
Branch if Higher or Same	(BHS)	40-5F	1	2	1, 2
Branch if Carry Set	BCS	60-7F	1	2	1
Branch if Lower	(BLO)	60-7F	1	2	1, 3
Branch if Not Equal	BNE	00-1F	1	2	1
Branch if Equal	BEQ	20-3F	1	2	1

SPECIAL NOTES

1 Each mnemonic of the Branch Instructions covers a range of 32 opcodes , e g , BCC ranges from 40 through 5F. The actual memory location (target address) to which the branch is made is formed by adding the sign extended lower five bits of the opcode to the contents of the program counter

2 The BHS instruction (shown in parentheses) is identical to the BCC instruction The C bit is clear if the register was higher or the same as the location in the memory to which it was compared

3 The BLO instruction (shown in parentheses) is identical to the BCS instruction The C bit is set if the register was lower than the location in memory to which it was compared

Table 6.4 : Bit Manipulation Instructions.

		Addressing Modes						
		Bit Set/Clear			Bit Test and Branch			
Function	**Mnem**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Special Note**
Branch IFF bit n is set	BRSET n (n = 0... 7)				C8 + n	3	5	1
Branch IFF bit n is clear.	BRCLR n (n = 0....7)				C0 + n	3	5	1
Set Bit n	BSET n (n = 0....7)	D8 + n	2	4				1
Clear Bit n	BCLR n (n = 0....7)	D0 + n	2	4				1

SPECIAL NOTE

1 The opcode is formed by adding the bit number (0-7) to the basic opcode. For example : to clear bit six using the BSET6 instruction the opcode becomes DE (D8 + 6) , BCLR5 becomes (C0 + 5) , etc

6.2.5. CONTROL INSTRUCTIONS. The control instructions control the MCU operations during program execution. Refer to table 6-5.

6.2.5.1. Stop Instruction

The STOP instruction places the EF68HC04P3 in its lowest power consumption mode. In STOP mode the internal oscillator is turned off causing all internal processing and the timer to be halted. In STOP mode, timer STATUS/CONTROL register bits 6 (ETI) and 7 (TMZ) are cleared to remove any pending timer interrupt requests and to disable any further timer interrupts. External interrupts are enabled. All I/O lines remain unchanged. The processor can only be brought out of the STOP mode by pulling low IRQ or RESET input pins. During the exit from the STOP mode, the timer is used to provide a delay of 1920 oscillator clock periods for the oscillator to stabilize. If an external clock is used, it should be kept high during all the time the MCU is in STOP mode.

6.2.5.2. Wait Instruction

The WAIT instruction places EF68HC04P3 in a low-power consumption mode, but WAIT mode consumes more power than the STOP mode. In WAIT mode the clock is disabled from all internal circuitry except the timer circuit. Thus all internal processing is halted. The timer may, if desired, continue to count down (PSI bit of TCSR).

During the WAIT mode, external interrupts are enabled. All other registers memory, and I/O lines remain in their last state. Timer interrupt (ETI bit) may be enabled by software prior to entering the WAIT mode to allow an exit from the WAIT mode via a Timer Interrupt.

6.2.6. ALPHABETICAL LISTING. The complete instruction set is given in alphabetical order in table 6-6. There are certain mnemonics recognized by the assembler and converted to other instructions. The fact that all registers and accumulator are in RAM al-

lows many implied instructions to exist. The implied instructions recognized by the assembler are identified in table 6-6.

6.2.7. OPCODE MAP SUMMARY. Table 6-7 contains an opcode map for the instructions used on the MCU.

6.3. IMPLIED INSTRUCTIONS

Since the accumulator and all other registers are located in RAM many implied instructions exist. The assembler-recognized implied instructions are given in table 6-6. Some examples not recognized by the assembler are shown below.

BCLR, 7 $FF	Ensures accumulator is plus
BSET, 7 $FF	Ensures accumulator is minus
BRCLR, 7 $FF	Branch iff accumulator is plus
BRSET, 7 $FF	Branch iff accumulator is minus
BRCLR, 7 $80	Branch iff X is plus (BXPL)
BRSET, 7 $80	Branch iff X is minus (BXMI)
BRCLR, 7 $81	Branch iff Y is plus (BYPL)
BRSET, 7 $81	Branch iff Y is minus (BYMI)

Table 6.5 : Control Instructions.

		Addressing Modes									
		Short-Direct			Inherent			Relative			
Function	Mnem	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Special Notes
Transfer A to XP	TAX	BC	1	4							
Transfer A to YP	TAY	BD	1	4							
Transfer XP to A	TXA	AC	1	4							
Transfer YP to A	TYA	AD	1	4							
Return from Subroutine	RTS				B3	1	2				
Return from Interrupt	RTI				B2	1	2				
No-operation	NOP										1
Stop	STOP				B6	1	2				
Wait	WAIT				B7	1	2				

SPECIAL NOTE

1. The NOP instruction is equivalent to a branch if equal (BEQ) to the location designated by PC + 1.

Table 6.6 : Instruction Set.

Mnemonic	Addressing Modes									Flags	
	Inherent	Immediate	Direct	Short Direct	Bit/Set Clear	Bit-Test-Branch	Register Indirect	Extended	Relative	Z	C
ADD		X	X				X			^	^
AND		X	X				X			^	•
ASLA	Assembler converts this to "ADD $FF"									^	^
BCC									X	•	•
BCLR					X					•	•
BCS									X	•	•
BEQ									X	•	•
BHS	Assembler converts this to "BCC"									•	•
BLO	Assembler converts this to "BCS"									•	•
BNE									X	•	•
BRCLR						X				•	^
BRSET						X				•	^
BSET					X					•	•
CLRA	Assembler converts this to "SUB $FF"									^	^
CLRX	Assembler converts this to "MVI # 0, $80"									•	•
CLRY	Assembler converts this to "MVI # 0, $81"									•	•
CMP		X	X				X			^	^
COMA	X									^	^
DEC			X	X			X			^	•
DECA	Assembler converts this to "DEC $FF"									^	•
DECX	Assembler converts this to "DEC $80"									^	•
DECY	Assembler converts this to "DEC $81"									^	•
INC			X	X			X			^	•
INCA	Assembler converts this to "INC $FF"									^	•
INCX	Assembler converts this to "INC $80"									^	•
INCY	Assembler converts this to "INC $81"									^	•
JMP								X		•	•
JSR								X		•	•
LDA		X	X	X			X			^	•
LDXI	Assembler converts this to "MVI DATA, $80"									•	•
LDYI	Assembler converts this to "MVI DATA, $81"									•	•
MVI		X	X							•	•
NOP	Assembler converts this to "BEQ (PC) + 1"									•	•
ROLA	X									^	^
RTI	X									^	^
RTS	X									•	•
STA			X	X			X			^	•
STOP	X									•	•
SUB		X	X				X			^	^
TAX	Assembler converts this to "STA $80"									^	•
TAY	Assembler converts this to "STA $81"									^	•
TXA	Assembler converts this to "LDA $80"									^	•
TYA	Assembler converts this to "LDA $81"									^	•
WAIT	X									•	•

Flag Symbols Z = Zero, C = Carry/borrow, ^ = Test and Set if True, Cleared Otherwise, • = Not affected

Table 6.7. : EF68HC04P3 Microcomputer Instruction Set Opcode Map.

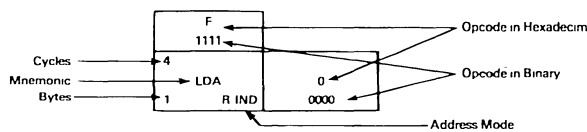
	Branch Instructions							
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
1 0001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
2 0010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
3 0011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
4 0100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
5 0101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
6 0110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
7 0111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
8 1000	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
9 1001	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
A 1010	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
B 1011	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
C 1100	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
D 1101	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
E 1110	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL
F 1111	2 BNE 1 REL	2 BNE 1 REL	2 BEQ 1 REL	2 BEQ 1 REL	2 BCC 1 REL	2 BCC 1 REL	2 BCS 1 REL	2 BCS 1 REL

Abbreviations for Address Modes

INH	Inherent
S-D	Short Direct
B-T-B	Bit Test and Branch
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
R-IND	Register Indirect
*	Indicates Instruction Reserved for Future Use
#	Indicates Illegal Instruction

Table 6.7. : EF68HC04P3 Microcomputer Instruction Set Opcode Map (continued).

Register/Memory,Control,and Read/Modify/Write Instructions				Bit Manipulation Instructions		Register/Memory and Read/Modify/Write		
8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi Low
4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 MVI 3 IMM	5 BRCLR0 3 B-T-B	4 BCLR0 2 BSC	4 LDA 1 R-IND	4 LDA 1 R-IND	0 0000
4 JSRn 2 EXT	4 JMPn 2 EXT	*	*	5 BRCLR1 3 B-T-B	4 BCLR1 2 BSC	4 STA 1 R-IND	4 STA 1 R-IND	0 0001
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTI 1 INH	5 BRCLR2 3 B-T-B	4 BCLR2 2 BSC	4 ADD 1 R-IND	4 ADD 1 R-IND	2 0010
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 RTS 1 INH	5 BRCLR3 3 B-T-B	4 BCLR3 2 BSC	4 SUB 1 R-IND	4 SUB 1 R-IND	3 0011
4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 COMA 1 INH	5 BRCLR4 3 B-T-B	4 BCLR4 2 BSC	4 CMP 1 R-IND	4 CMP 1 R-IND	4 0100
4 JSRn 2 EXT	4 JMPn 2 EXT	*	4 ROLA 1 INH	5 BRCLR5 3 B-T-B	4 BCLR5 2 BSC	4 AND 1 R-IND	4 AND 1 R-IND	5 0101
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 STOP 1 INH	5 BRCLR6 3 B-T-B	4 BCLR6 2 BSC	4 INC 1 R-IND	4 INC 1 R-IND	6 0110
4 JSRn 2 EXT	4 JMPn 2 EXT	*	2 WAIT 1 INH	5 BRCLR7 3 B-T-B	4 BCLR7 2 BSC	4 DEC 1 R-IND	4 DEC 1 R-IND	7 0111
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET0 3 B-T-B	4 BSET0 2 BSC	4 LDA 2 IMM	4 LDA 2 DIR	8 1000
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET1 3 B-T-B	4 BSET1 2 BSC	#	4 STA 2 DIR	9 1001
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET2 3 B-T-B	4 BSET2 2 BSC	4 ADD 2 IMM	4 ADD 2 DIR	A 1010
4 JSRn 2 EXT	4 JMPn 2 EXT	4 INC 1 S-D	4 DEC 1 S-D	5 BRSET3 3 B-T-B	4 BSET3 2 BSC	4 SUB 2 IMM	4 SUB 2 DIR	B 1011
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET4 3 B-T-B	4 BSET4 2 BSC	4 CMP 2 IMM	4 CMP 2 DIR	C 1100
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET5 3 B-T-B	4 BSET5 2 BSC	4 AND 2 IMM	4 AND 2 DIR	D 1101
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET6 3 B-T-B	4 BSET6 2 BSC	#	4 INC 2 DIR	E 1110
4 JSRn 2 EXT	4 JMPn 2 EXT	4 LDA 1 S-D	4 STA 1 S-D	5 BRSET7 3 B-T-B	4 BSET7 2 BSC	#	4 DEC 2 DIR	F 1111



SECTION 7

ELECTRICAL SPECIFICATIONS

7.1. INTRODUCTION

This section contains the electrical specifications and associated timing for the EF68HC04P3.

7.2. ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage	V_{SS} – 0.3 to V_{SS} + 0.3	V
I	Current Drain per Pin	10	mA
	Total Current for Ports A, B, C and XTAL, TIMER Pins		mA
I_I	Sink	30	
I_O	Source	– 15	
T_A	Operating Temperature Range	T_L to T_H	°C
	L Range	0 to + 70	
	D Range	– 25 to 70	
	V Range	– 40 to 85	
T_{stg}	Storage Temperature Range	– 55 to 150	°C
T_J	Junction Resistance Plastic	150	°C
	PLCC	150	

This device contains circuitry to protect the inputs against damage due to high static voltages of electric fields , however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range V_{SS} (V_{in} or V_{out}) V_{CC} Reliability of operation is enhanced if unused inputs except EXTAL are connected to an appropriate logic voltage level (e g , either V_{SS} or V_{CC})

7.3. EQUIVALENT TEST LOAD

Figure 7.1 : Open Collector LSTTL Compatible Equivalent Test Load (Ports A, B, C,).

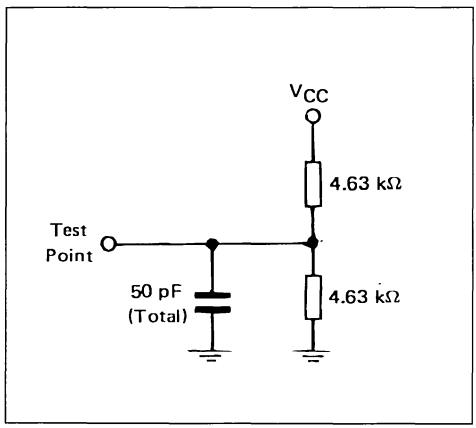
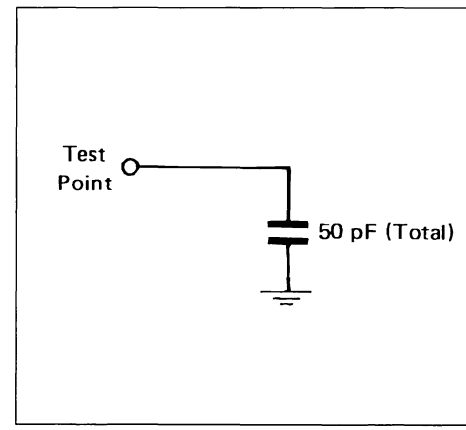


Figure 7.2 : CMOS Equivalent Test Load (Ports A, B, C,).



7.5. POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$T_J = T_A + (P_D \theta_{JA})$ (1)

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications PPORT ≤ P_{INT} and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$P_D = K \div (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives :

$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2$ (3)

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ELECTRICAL CHARACTERISTICS + 5.0V

(V_{CC} = + 5.0 Vdc ± 10% ; 0Vdc, $T_A = T_L$ to T_H unless otherwise noted).

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL} V_{OH}	Output Voltage (I_{Load} = 10A) (I_{Load} = 10A)	 V_{CC} – 0.1		0.1	V
V_{OL}	Output Low Voltage (I_{Load} = 0.8mA) - Timer Ports			0.4	V
V_{OH}	Output High Voltage (I_{Load} = 0.8mA) - Timer Ports	V_{CC} – 0.4			V
 V_{IL} V_{IL}	Input Low Voltage Ports, Timer XTAL, MDS, IRQ, RESET	 V_{SS} V_{SS}		 0 3 V_{CC} 0.2 V_{CC}	 V
 V_{IH} V_{IH}	Input High Voltage Ports, Timer XTAL, MDS, IRQ, RESET	 0.7 V_{CC} 0.8 V_{CC}		 V_{CC} V_{CC}	 V
 I_{DD} I_{DD} I_{DD}	Total Supply Current (no dc loads, fosc = 6Mhz) (no dc loads, V_{IL} = 0.2V ; V_{CC} – 0.2V) RUN Mode WAIT Mode (*) STOP Mode (*)		 1.5 0 5 0 5	 3 1 1	 mA mA µA
I_{TSI}	I/O Hi-Z Leakage Current (V_{in} = 0.4V to V_{CC} – 0.4V) Timer, Ports (with no pull-down)	 – 1		 1	 µA
I_{TSI}	Hi-Z State Input Current (V_{in} = V_{CC} – 0.4V) Timer, Ports (with pull-downs)	 10		 250	 µA
I_{in}	Input Current IRQ, RESET, XTAL, MDS (with no pull up)	– 1		+ 1	µA
I_{in}	Input Current IRQ, RESET, (with pull up) (V_{IN} = V_{CC} – 0.4V)	 – 100		 – 4	 µA
C_{out}	I/O Output Capacitance Timer, Ports			12	pF
C_{in}	Input Capacitance IRQ, RESET, XTAL, MDS			8	pF

Note : Test conditions for I_{DD} as follows
XTAL input is a square wave from 0.2V to V_{CC} – 0 2V
EXTAL output load = 10pF
Circuit in self check-mode
In WAIT and STOP Modes, Port A is programmed as output, Port B and C are programmed as inputs
In STOP Mode . all inputs are tied to V_{IL} excepted IRQn RST, MDS, XTAL, EXTAL which are tied to V_{IH} (when IRQ, RST have no pull up) when IRQ, RST have pull up, these pins are tied to V_{CC}
Ports pull downs not enabled, if ports pull downs enabled connect to V_{SS}

CONTROL TIMING CHARACTERISTICS (V_{DD} = + 5.0Vdc ± 10% ; V_{SS} = 0Vdc ; T_A = 0°C to 70°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency	0		12	Mhz
f_{CL}	PHI 1 Clock Frequency	0		6	Mhz
f_{CYC}	Cycle Time (min)	2			s
t_{IWL}	IRQ Pulse Width	2 x t_{CYC}			s
t_{RWLC}	RESET Pulse Width	2 x t_{CYC}			s
t_{OL} t_{OH}	Oscillator Clock Pulse Width	45			ns

ELECTRICAL CHARACTERISTICS + 3.0V

(V_{CC} = + 3.0 Vdc ± 10% ; 0Vdc, T_A = T_L to T_H unless otherwise noted).

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL} V_{OH}	Output Voltage (I_{Load} = 10A) (I_{Load} = 10A)	 V_{CC} – 0.1		0.1	V
V_{OL}	Output Low Voltage (I_{Load} = 0.8mA) - Timer Ports			0.3	V
V_{OH}	Output High Voltage (I_{Load} = 0.8mA) - Timer Ports	V_{CC} – 0.3			V
 V_{IL} V_{IL}	Input Low Voltage Ports, Timer XTAL, MDS, IRQ, RESET	 V_{SS} V_{SS}		 0.3 V_{CC} 0.2 V_{CC}	V
 V_{IH} V_{IH}	Input High Voltage Ports, Timer XTAL, MDS, IRQ, RESET	 0.7 V_{CC} 0.8 V_{CC}		 V_{CC} V_{CC}	V
  I_{DD} I_{DD} I_{DD}	Total Supply Current (no dc loads, fosc = 4.5Mhz) (no dc loads, V_{IL} = 0.2V ; V_{CC} – 0.2V) RUN Mode WAIT Mode (*) STOP Mode (*)	 	 0.6 0 2 0.3	 2 0.75 0.75	 mA mA µA
 I_{TSI}	I/O Hi-Z Leakage Current (V_{in} = 0.4V to V_{CC} – 0.3V) Timer, Ports (with no pull-down)	 – 1		 1	 µA
 I_{TSI}	Hi-Z State Input Current (V_{in} = V_{CC} – 0.3V) Timer, Ports (with pull-downs)	 	 125	 	 µA
I_{in}	Input Current IRQ, RESET, XTAL, MDS (with no pull up)	– 1		+ 1	µA
 I_{in}	Input Current IRQ, RESET, (with pull up) (V_{IN} = V_{CC} – 0.3V)	 	 – 10	 	 µA
C_{out}	I/O Output Capacitance Timer, Ports			12	pF
C_{in}	Input Capacitance IRQ, RESET, XTAL, MDS			8	pF

Note : Test conditions for I_{DD} as follows
XTAL input is a square wave from 0 2V to V_{CC} – 0 2V.
EXTAL output load = 10pF
Circuit in self check-mode
In WAIT and STOP Modes, Port A is programmed as output, Port B and C are programmed as inputs
In STOP Mode · all inputs are tied to V_{IL} excepted IRQn RST, MDS, XTAL, EXTAL which are tied to V_{IH} (when IRQ, RST have no pull up) when IRQ, RST have pull up, these pins are tied to V_{CC}
Ports pull downs not enabled, if ports pull downs enabled connect to V_{SS}

CONTROL TIMING CHARACTERISTICS (V_{DD} = + 3.0Vdc ± 10% ; V_{SS} = 0Vdc ; T_A = 0°C to 70°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency	0		12	Mhz
f_{CL}	PHI 1 Clock Frequency	0		4.5	Mhz
f_{CYC}	Cycle Time (min)	2			s
t_{IWL}	IRQ Pulse Width	2 x t_{CYC}			s
t_{RWLC}	RESET Pulse Width	2 x t_{CYC}			s
t_{OL} t_{OH}	Oscillator Clock Pulse Width	45			ns

ELECTRICAL CHARACTERISTICS + 2.2V

(V_{CC} = + 2.2Vdc ± 10% ; 0Vdc, T_A = T_L to T_H unless otherwise noted).

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL} V_{OH}	Output Voltage (I_{Load} = 10A) (I_{Load} = 10A)	 V_{CC} – 0.1		0.1	V
V_{OL}	Output Low Voltage (I_{Load} = 0.8mA) - Timer Ports			0.3	V
V_{OH}	Output High Voltage (I_{Load} = 0.8mA) - Timer Ports	V_{CC} – 0.3			V
 V_{IL} V_{IL}	Input Low Voltage Ports, Timer XTAL, MDS, IRQ, RESET	 V_{SS} V_{SS}		 0.3 V_{CC} 0 2 V_{CC}	 V
 V_{IH} V_{IH}	Input High Voltage Ports, Timer XTAL, MDS, IRQ, RESET	 0.7 V_{CC} 0.8 V_{CC}		 V_{CC} V_{CC}	 V
 I_{DD} I_{DD} I_{DD}	Total Supply Current (no dc loads, fosc = 3Mhz) (no dc loads, V_{IL} = 0.2V ; V_{CC} – 0.2V) RUN Mode WAIT Mode (*) STOP Mode (*)		 0.3 0.1 0 2	 1 0.5 0.5	 mA mA µA
 I_{TSI}	I/O Hi-Z Leakage Current (V_{in} = 0.4V to V_{CC} – 0.3V) Timer, Ports (with no pull-down)	 – 1		 1	 µA
 I_{TSI}	Hi-Z State Input Current (V_{in} = V_{CC} – 0 3V) Timer, Ports (with pull-downs)		 75		 µA
I_{in}	Input Current IRQ, RESET, XTAL, MDS (with no pull up)	– 1		+ 1	µA
 I_{in}	Input Current IRQ, RESET, (with pull up) (V_{IN} = V_{CC} – 0.3V)		 – 10		 µA
C_{out}	I/O Output Capacitance Timer, Ports			12	pF
C_{in}	Input Capacitance IRQ, RESET, XTAL, MDS			8	pF

Note : Test conditions for I_{DD} as follows :
XTAL input is a square wave from 0.2V to V_{CC} – 0 2V
EXTAL output load = 10pF
Circuit in self check-mode
In WAIT and STOP Modes, Port A is programmed as output, Port B and C are programmed as inputs
In STOP Mode all inputs are tied to V_{IL} excepted IRQn RST, MDS, XTAL, EXTAL which are tied to V_{IH} (when IRQ, RST have no pull up) when IRQ, RST have pull up, these pins are tied to V_{CC}
Ports pull downs not enabled, if ports pull downs enabled connect to V_{SS}

CONTROL TIMING CHARACTERISTICS (V_{DD} = + 2.2Vdc ± 10% ; V_{SS} = 0Vdc ; T_A = 0°C to 70°C)

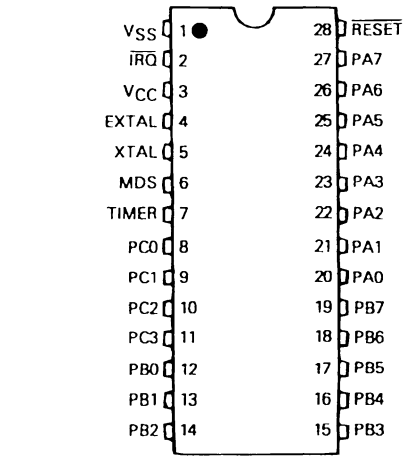
Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency	0		12	Mhz
f_{CL}	PHI 1 Clock Frequency	0		3	Mhz
f_{CYC}	Cycle Time (min)	2			s
t_{IWL}	IRQ Pulse Width	2 x t_{CYC}			s
t_{RWLC}	RESET Pulse Width	2 x t_{CYC}			s
t_{OL} t_{OH}	Oscillator Clock Pulse Width	45			ns

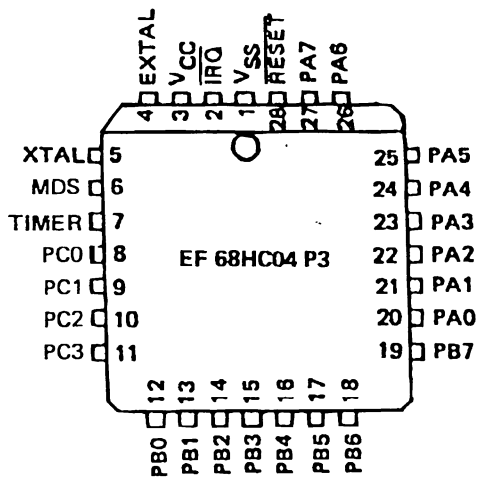
SECTION 8

MECHANICAL DATA

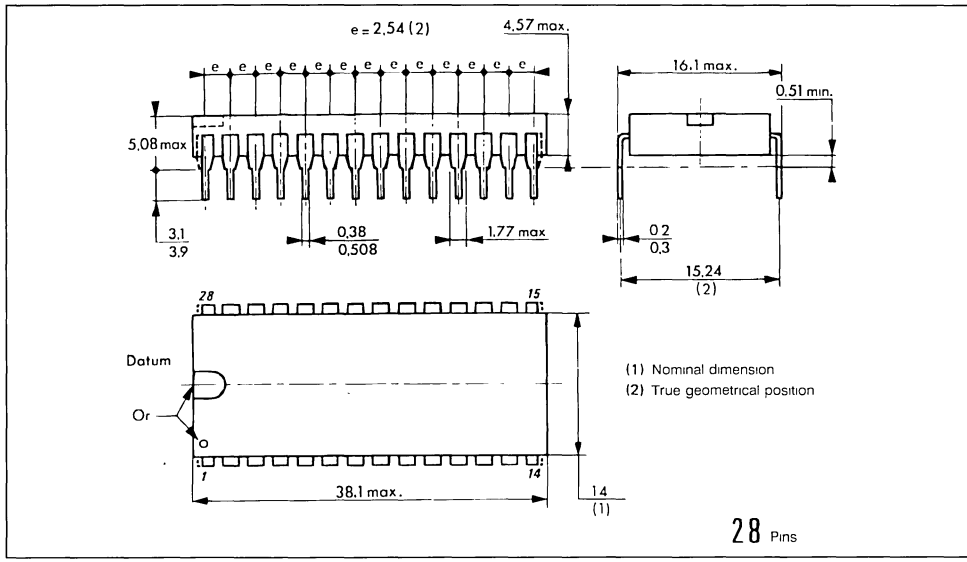
This section contains the pin assignment and package dimension diagrams for the EF68HC04P3 microcomputer.

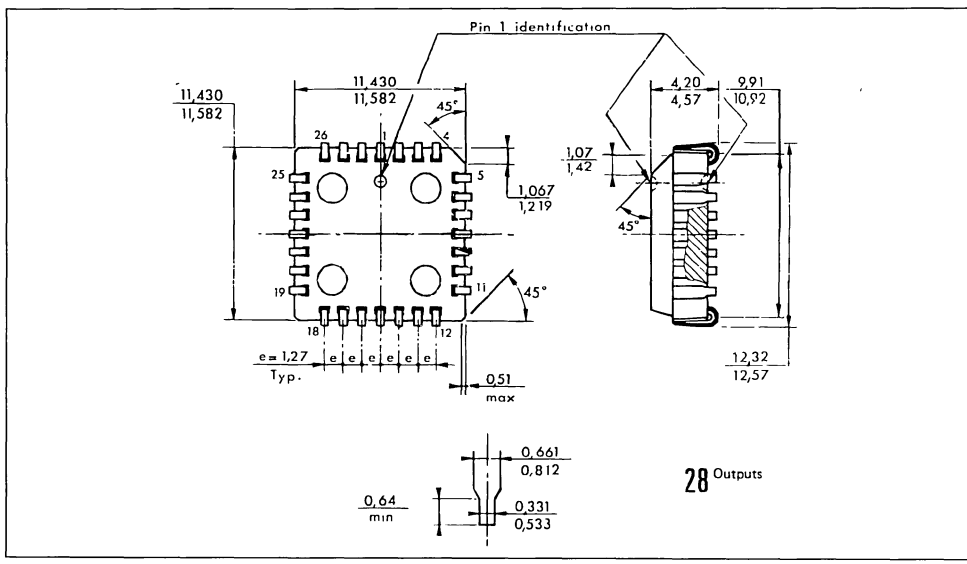
8.1 PIN ASSIGNMENTS



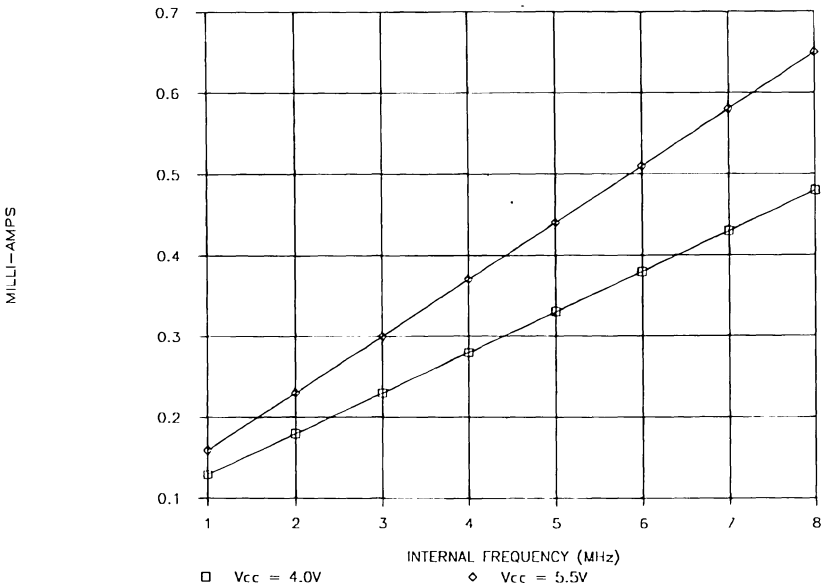


8.2 PHYSICAL DIMENSIONS

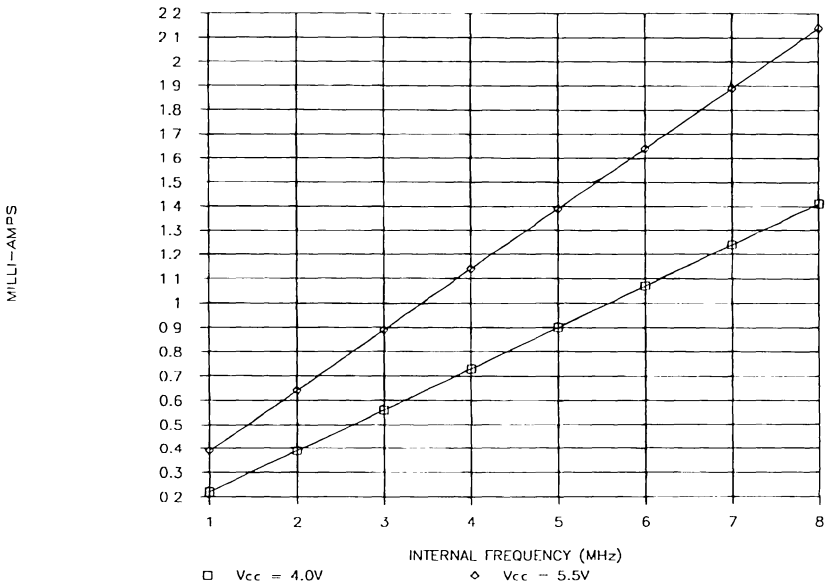




68HC04P3·TYPICAl WAIT--IDD/FREQUENCY

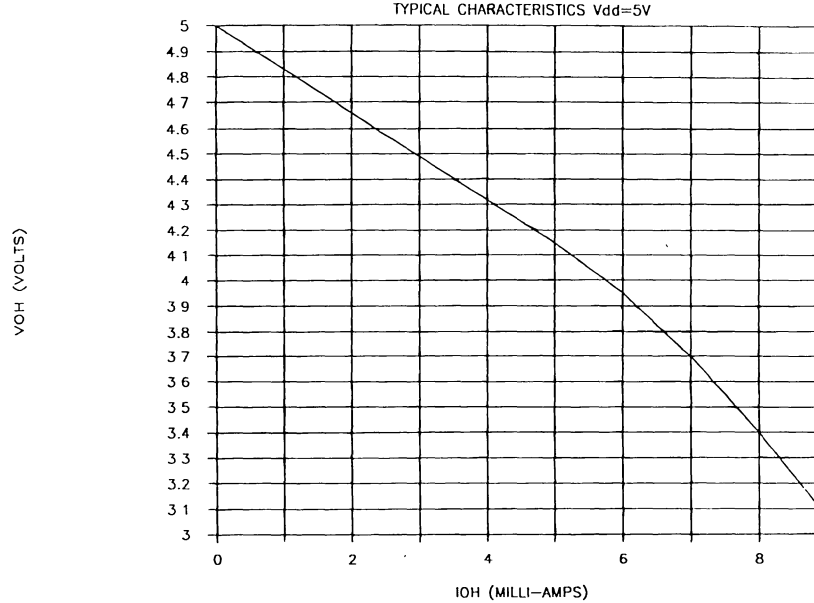


68HC04P3:TYPICAL RUN-IDD/FREQUENCY



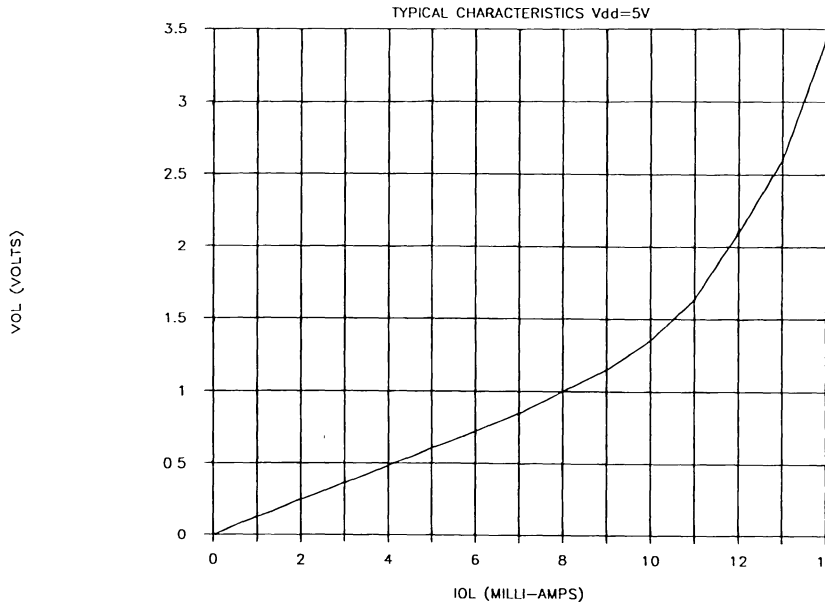
68HC04P3 : VOH–IOH

TYPICAL CHARACTERISTICS Vdd=5V

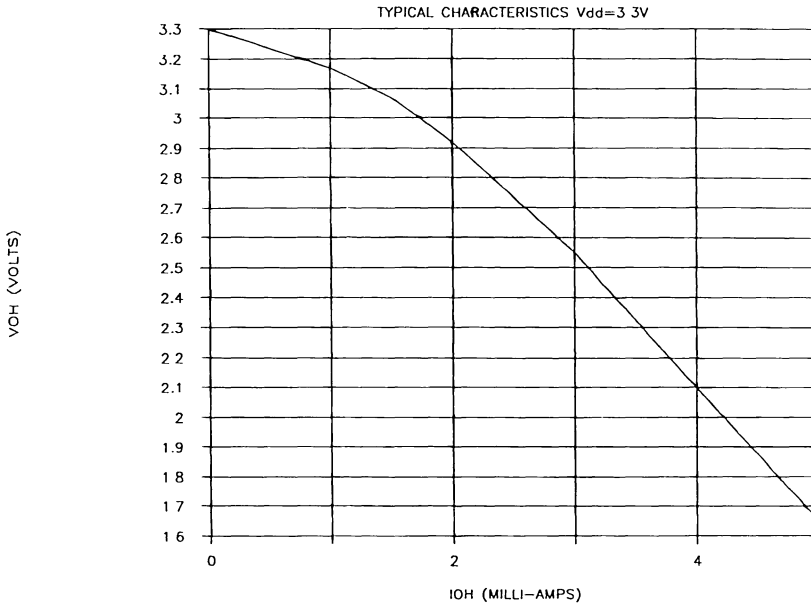


68HC04P3 : VOL–IOL

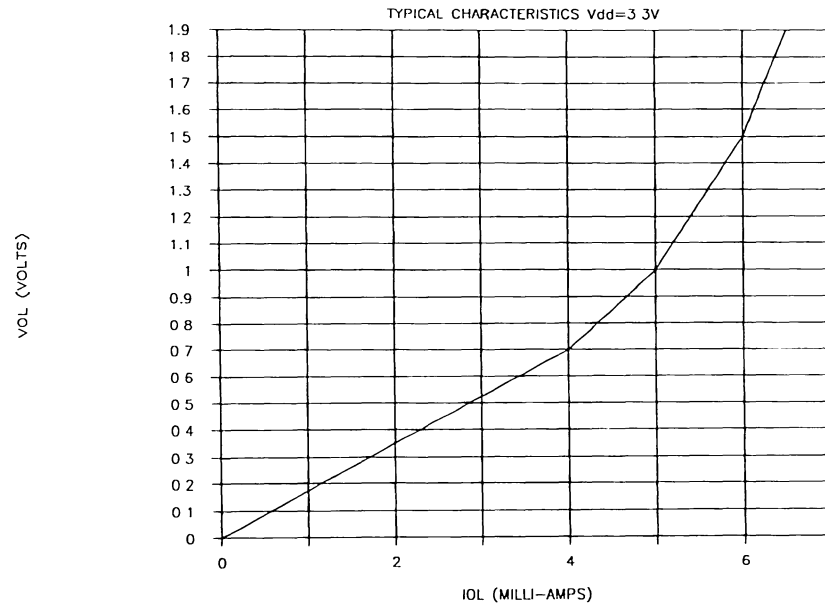
TYPICAL CHARACTERISTICS Vdd=5V



68HC04P3 : VOH–IOH



68HC04P3 : VOL–IOL



SECTION 9

ORDERING INFORMATION

9.1. INTRODUCTION

The following information is required when ordering a custom MCU. The information may be transmitted to SGS-THOMSON Microelectronics in the following media :

EPROM(s), ET2732

To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person, or your local SGS-THOMSON Microelectronics representative.

9.1.1. EPROMs. A 2716 or 2732 type EPROM, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one 2716 or 2732 EPROM, the EPROM must be programmed as follows in order to emulate the EF68HC04P3 MCU. Start the data space ROM at EPROM address $018 and start program space ROM at EPROM address $960 and continue to memory space $FFF. All unused bytes, including the user's space, must be set to zero. For shipment to SGS-THOMSON Microelectronics the EPROMs should be placed in a conductive IC carrier and packed securely. Do not use styrofoam.

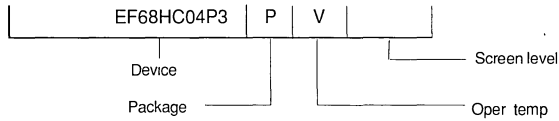
9.2. VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS-THOMSON Microelectronics. The signed verification form constitutes the contractural agreement for creation of the customer mask. If desired, SGS-THOMSON Microelectronics will program a blank 2716, 2732, or EFDOS disk (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

9.3. ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and five volts. These RVUs are included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by SGS-THOMSON Microelectronics Quality Assurance.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating and screening level. Other possibilities on request.

Device	Package					Oper. Temp			Screening Level			
	C		P		FN	L*	D	V	Std	D		
EF68HC04P3			●		●	●	●	●	●			
Examples : EF68HC04P3P, EF68HC04P3FN.												

Package : C : Ceramic DIL, DIL, **P :** Plastic DIL, **FN :** PLCC
Oper. temp. : L* : 0°C to + 70°C, **D :** – 25 °C to + 70°C, **V :** – 40°C to + 85°C, **T :** – 40°C to + 105°C
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
*** May be omitted**

A reduced-packaged version of the EF68HC04P3 (28pins) will be available in a 20-pin package EF68HC04J3

EF6804 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....
.....

ROM capacity required : | | | | bytes

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 25°C / + 70°C
- ❑ – 40°C / + 85°C

Package :
- ❑ Plastic
- ❑ PLCC

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference :
- ❑ SGS-THOMSON Microelectronics reference
  .....
- ❑ Special customer data reference*
  .....

Quality level :
- ❑ STD
- ❑ D
- ❑ Other* (customer's quality specification ref.) :
  .....

Software developed by :
- ❑ SGS-THOMSON Microelectronics application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
- ❑ EPROM Reference :
- ❑ Other *

TECHNICAL OPTIONS
- ❑ Frequency = : MHz
- ❑ Voltage Supply : (± 10%)
  - ❑ 2.2V
  - ❑ 3.0V
  - ❑ 5.0V
  - ❑ 2.2V – 5.0V
  - ❑ 3.0V – 5.0V

* Requires prior factory approval

OPTION LIST

– Oscillator input :
- ❑ Xtal
- ❑ RC

– I/O pull-down selection
- ❑ PA7 = on PA7 only
- ❑ PA6 = on PA5 and PA6
- ❑ PA4 = on PA1 to PA4
- ❑ PA5 = on PA4 to PA7
- ❑ PA3 = on PA0 to PA3
- ❑ PB7 = on PB3 to PB7
- ❑ PB2 = on PB1 and PB2
- ❑ PB0 = on PB0 only
- ❑ PB = on Port B
- ❑ PC = on Port C
- ❑ PC0 = on PA0 and Port C
- ❑ PT = on Timer Pin

– Interrupt trigger
- ❑ EM = Edge-sensitive
- ❑ LM = Level & edge-sensitive

– Oscillator divider :
- ❑ D1 = 2**0
- ❑ D2 = 2**1
- ❑ D4 = 2**2

– I/O pull-up selection :
- ❑ RE = on reset pin
- ❑ IN = on interrupt pin

- ❑ NPRO Program ROM protection disable

Yearly Quantity Forecast :
RVU Approval Waived – Yes, No
Waived Quantity :

- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME : DATE : SIGNATURE :

EF6805 FAMILY DATASHEETS

SGS-THOMSON MICROELECTRONICS

EF6805P2

8-BIT MICROCOMPUTER UNIT

ADVANCE DATA

HARDWARE FEATURES

- 8-BIT ARCHITECTURE
- 64 BYTES OF RAM
- MEMORY MAPPED I/O
- 1100 BYTES OF USER ROM
- 20 TTL/CMOS COMPATIBLE BIDIRECTIONAL I/O LINES (8 Lines are LED Compatible)
- ON-CHIP CLOCK GENERATOR
- SELF-CHECK MODE
- ZERO CROSSING DETECTION
- MASTER RESET
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE®
- 5V SINGLE SUPPLY

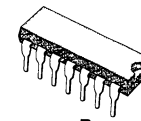
SOFTWARE FEATURES

- SIMILAR TO 6800 FAMILY
- BYTE EFFICIENT INSTRUCTION SET
- EASY TO PROGRAM
- TRUE BIT MANIPULATION
- BIT TEST AND BRANCH INSTRUCTION
- VERSATILE INTERRUPT HANDLING
- VERSATILE INDEX REGISTER
- POWERFUL INDEXED ADDRESSING FOR TABLES
- FULL SET OF CONDITIONAL BRANCHES
- MEMORY USABLE AS REGISTER/FLAGS
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- 10 POWERFUL ADDRESSING MODES
- ALL ADDRESSING MODES APPLY TO ROM, RAM, AND I/O

USER SELECTABLE OPTIONS

- INTERNAL 8-BIT TIMER WITH SELECTABLE CLOCK SOURCE (external timer input or internal machine clock)
- TIMER PRESCALER OPTION (7 Bits, 2^n)
- 8 BIDIRECTIONAL I/O LINES WITH ttl OR TTL/CMOS INTERFACE OPTION
- CRYSTAL OR LOW-COST RESISTOR OSCILLATOR OPTION
- LOW VOLTAGE INHIBIT OPTION
- VECTORED INTERRUPTS TIMER, SOFTWARE, AND EXTERNAL

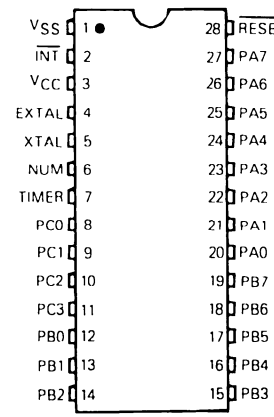
Inice ® SGS THOMSON development/emulation tool

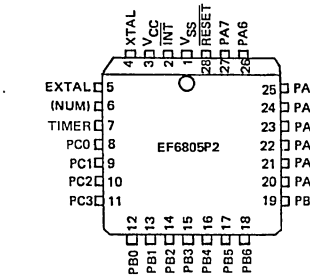


P
(Plastic Package)

FN
(PLCC 28)

PIN CONNECTIONS

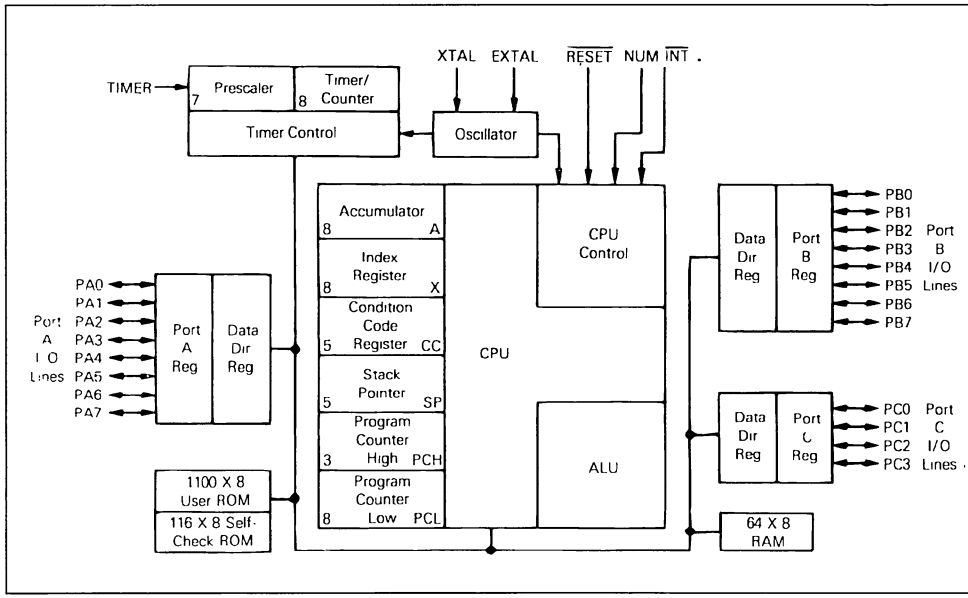




DESCRIPTION

The EF6805P2 Microcomputer Unit (MCU) is a member of the EF6805 Family of low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the 6800-based instruction set. A comparison of the key features of several members of the 6805 Family is shown at the end of this data sheet. The following are some of the hardware and software highlights of the EF6805P2 MCU.

Figure 1 : EF6805P2 HMOS Microcomputer Block Diagram.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage (except pin 6)	– 0.3 to + 7.0	V
T_A	Operating Temperature Range (T_L to T_H) V Suffix	0 to 70 – 40 to + 85	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_J	Junction Temperature Plastic PLCC	 150 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$ Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e g , either V_{SS} or V_{CC}).

THERMAL DATA

θ_{JA}	Thermal Resistance Plastic PLCC	70 110	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$T_J = T_A + (P_D \cdot \theta_{JA})$ (1)

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} << P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$P_D = K \div (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives :

$K = PD \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2$ (3)

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage				V
	$\overline{\text{RESET}}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0		V_{CC}	
	(4.75 < V_{CC})	V_{CC} – 0.5		V_{CC}	
	$\overline{\text{INT}}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0	*	V_{CC}	
	(4.75 < V_{CC})	V_{CC} – 0.5	*	V_{CC}	
	All Other (except TIMER)	2.0		V_{CC}	
V_{IH}	Input High Voltage Timer				V
	Timer Mode	2.0		V_{CC} + 1	
	Self-check Mode	9.0	10.0	15.0	
V_{IL}	Input Low Voltage				V
	$\overline{\text{INT}}$	V_{SS}	*	1.5	
	All Other	V_{SS}		0.8	
V_{IRES+} / V_{IRES-}	$\overline{\text{RESET}}$ Hystereris Voltage (see figures 10, 11 and 12)				V
	"Out of Reset"	2.1		4.0	
	"Into Reset"	0.8		2.0	
V_{INT}	$\overline{\text{INT}}$ Zero Crossing Input Voltage, through a Capacitor	2.0		4.0	$V_{ac\ p\text{-}p}$
P_{INT}	Internal Power Dissipation - No Port Loading V_{CC} = 5.75V, T_A = 0°C		400	690	mW
C_{in}	Input Capacitance				pF
	EXTAL		25		
	All Other		10		
V_{LVR}	Low Voltage Recover			4.75	V
V_{LVI}	Low Voltage Inhibit 0 to + 70°C	2.75	3.5		V
	– 40 to + 85°C	3.1	3.5		
I_{in}	Input Current				µA
	$\overline{\text{TIMER}}$ (V_{in} = 0.4V)			20	
	$\overline{\text{INT}}$ (V_{in} = 2.4V to V_{CC})		20	50	
	EXTAL (V_{in} = 2.4V to V_{CC}, crystal option)			10	
	(V_{in} = 0.4V, crystal option)			– 1600	
	$\overline{\text{RESET}}$ (V_{in} = 0.8V)	– 4.0		– 40	
	(external capacitor charging current)				

* Due to internal biasing, this input (when unused) floats to approximately 2 0 Vdc

PORT DC ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

PORT A WITH CMOS DRIVE ENABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.4			V
V_{OH}	Output High Voltage, I_{Load} = – 10µA	V_{CC} – 1.0			V
V_{IH}	Input High Voltage, I_{Load} = – 300µA (max.)	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 500µA (max.)	V_{SS}		0.8	V
I_{IH}	Hi-Z State Input Current (V_{IN} = 2.0V to V_{CC})			– 300	µA
I_{IL}	Hi-Z State Input Current (V_{IN} = 0.4V)			– 500	µA

PORT B

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 3.2mA			0.4	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.0	V
V_{OH}	Output High Voltage, I_{Load} = – 200µA	2.4			V
I_{OH}	Darlington Current Drive (source), V_O = 1.5V	– 1.0		– 10	mA
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		2	10	µA

PORT C AND PORT A WITH CMOS DRIVE DISABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.4			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	Hi-Z State Input Current		2	10	µA

SWITCHING CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency EF6805P2 EF68A05P2 EF68B05P2	0.4 0.4 0.4		4.2 6.0 8.0	MHz
t_{cyc}	Cycle Time ($4/f_{osc}$)	0.95		10	µs
t_{WL}, t_{WH}	$\overline{INT}$ and TIMER Pulse Width (see interrupt section)	t_{cyc} + 250			ns
t_{RWL}	$\overline{RESET}$ Pulse Width	t_{cyc} + 250			ns
t_{RHL}	$\overline{RESET}$ Delay Time (external capacitance = 1.0µF)		100		ms
f_{INT}	$\overline{INT}$ Zero Crossing Detection Input Frequency (± 5° Accuracy)	0.03		1.0	kHz
	External Clock Input Duty Cycle (EXTAL)	40	50	60	%

Figure 2 : TTL Equivalent Test Load (port B).

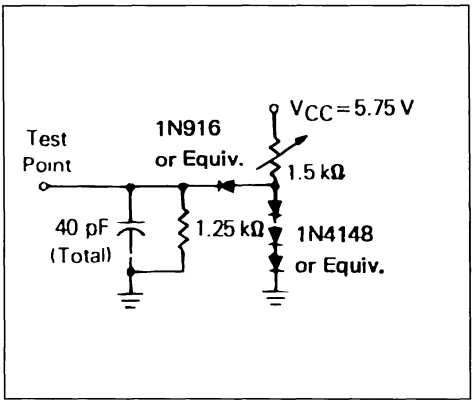


Figure 3 : CMOS Equivalent Test Load (port A).

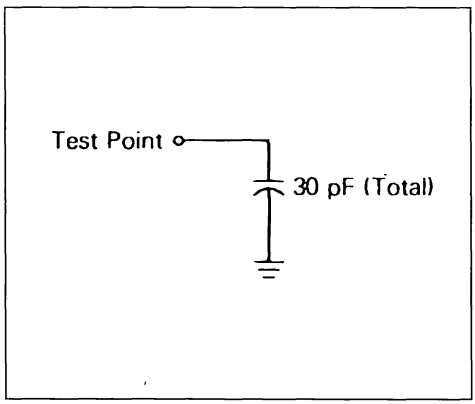
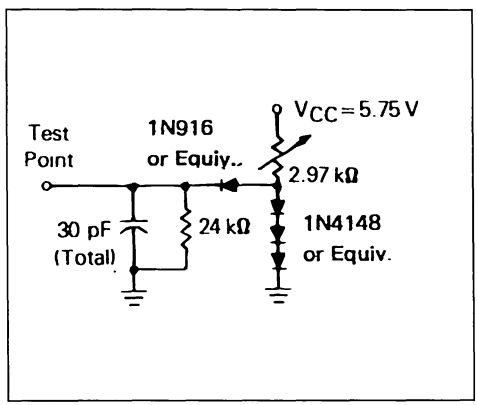


Figure 4 : TTL Equivalent Test Load (port A and C).



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in figure 1, are described in the following paragraphs.

V_{CC} AND V_{SS}. Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

$\overline{INT}$. This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts section for additional information.

XTAL AND EXTAL. These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to Internal Clock Generator Options section for recommendations about these inputs.

TIMER. This pin allow an external input to be used to decrement the internal timer circuitry. Refer to Timer section for additional information about the timer circuitry.

RESET. This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to Resets section for additional information.

NUM. This pin is not for user application and must be connected to V_{SS}.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC3)

These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to Inputs/Outputs section for additional information.

MEMORY

As shown in figure 5, the MCU is capable of addressing 2048 bytes of memory and I/O registers with is program counter. The EF6805P2 MCU has implemented 1288 of these locations. This consists of : 1100 bytes of user ROM, (from $080 to $OFF and from $3CO to $783) 116 bytes of self-check ROM, 64 bytes of user RAM, 6 bytes of port I/O, and 2 timers registers. The ROM division allows 128 bytes of ROM to be addressed with direct instructions.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in figure 6. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first ; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly, during pulls from the stack, since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack. The remaining CPU registers are not pushed.

CENTRAL PROCESSING UNIT

The CPU of the EF6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

REGISTERS

The 6805 Family CPU has five registers available to the programmer. They are shown in figure 7 and are explained on the following paragraphs.

Figure 5 : MCU Address Map.

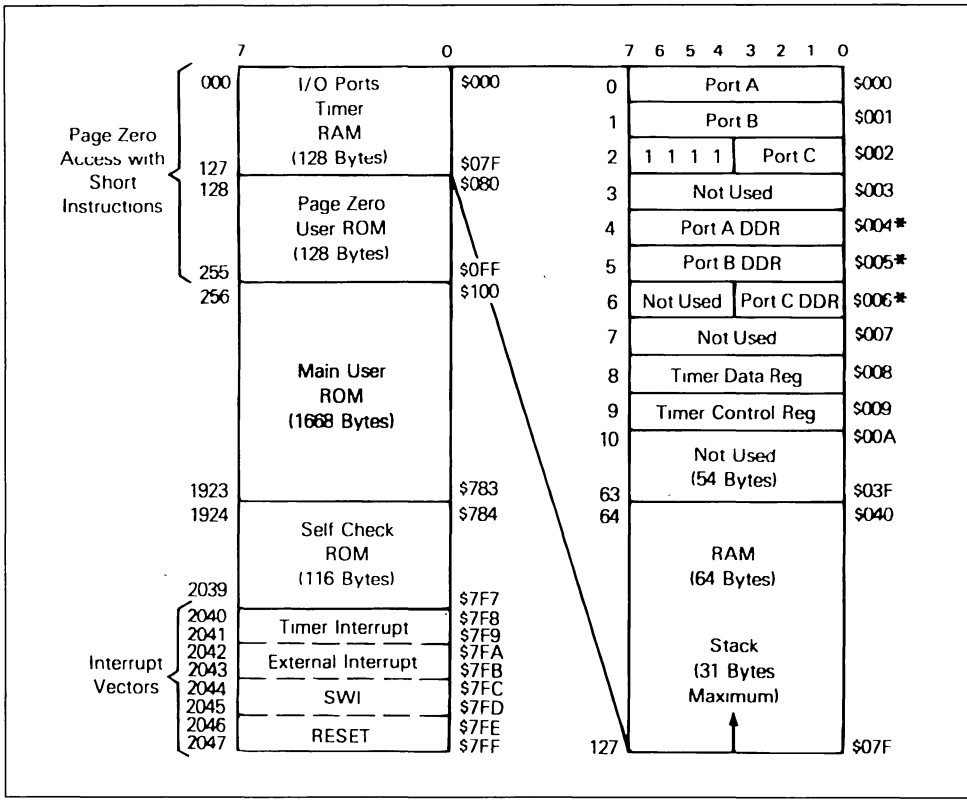


Figure 6 : Interrupt Stacking Order.

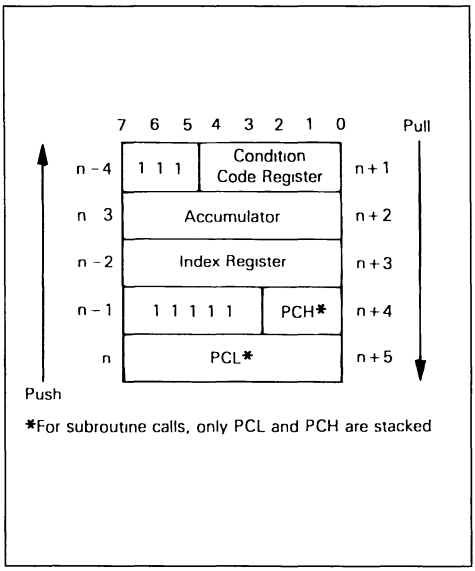
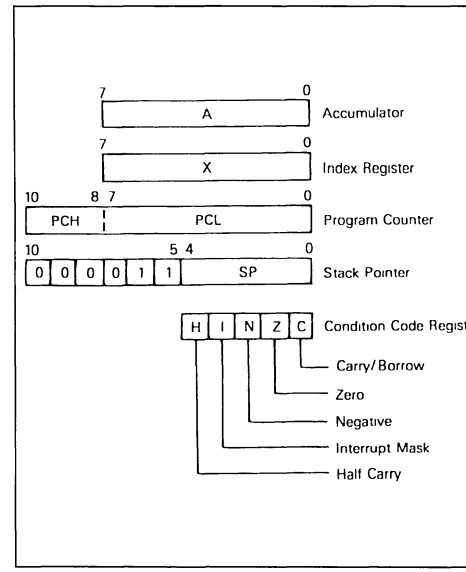


Figure 7 : Programming Model.



ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X). The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read modify-write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC). The program counter is an 11-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP). The stack pointer is an 11-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is pushed onto the stack and incremented as data is pulled from the stack. The six most significant bits of the stack pointer are permanently configured to 000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls.

CONDITION CODE REGISTER (CC). The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

HALF CARRY (H) - Set during ADD and ADC instructions to indicate that a carry occurred between bits 3 and 4.

INTERRUPT (I) - This bit is set to mask (disable) the timer and external interrupt (INT). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt is cleared.

NEGATIVE (N) - Used to indicate that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in result equal to a logical one).

ZERO (Z) - Used to indicate that the result of the last arithmetic, logical, or data manipulation was zero.

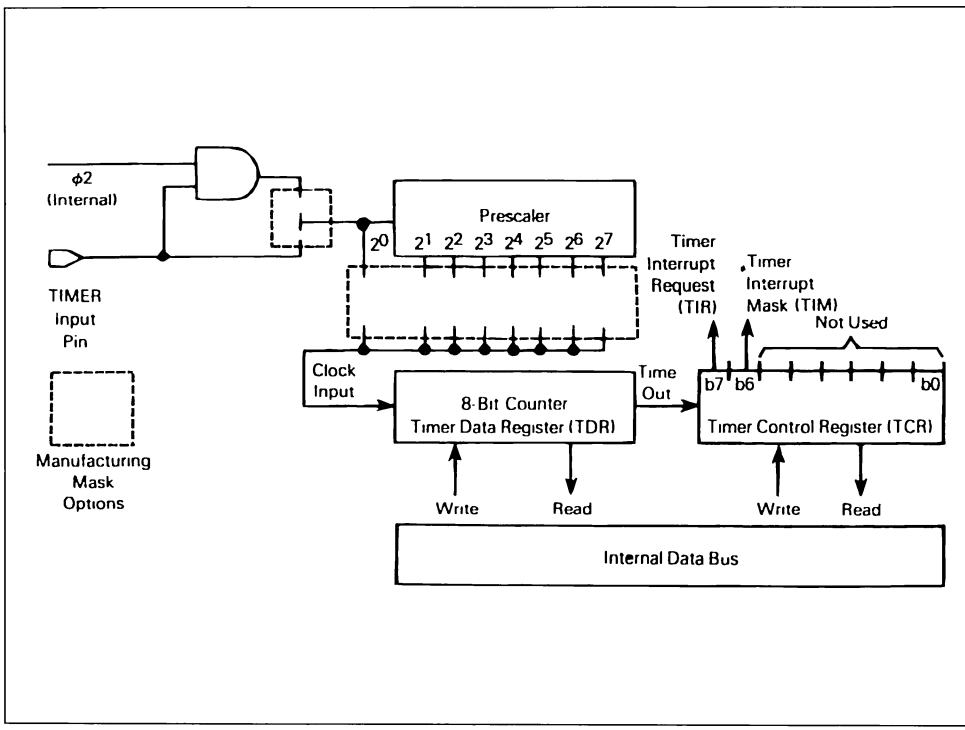
CARRY/BORROW (C) - Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic logic unit (ALU) occurred during the last airhtmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The EF6805P2 MCU timer circuitry is shown in figure 8. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9, and exexuting the interrupt routine ; see the Interrupts section. THE TIMER INTERRUPT REQUEST BIT MUST BE CLEARED BY SOFTWARE.

The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin or it can be the internal 2 signal. Three machine cycles are required for a change in state of the TIMER pin to decrement the timer prescaler. The maximum frequency of a signal that can be recognized by the TIMER or INT pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on

Figure 8 : Timer Block Diagram.



the pin must also recognize the low state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows : (assumes 50/50 duty cycle for a given period).

$$t_{cyc} \times 2 + 250ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. (NOTE : For ungated $\phi 2$ clock inputs to the timer prescaler, the TIMER pin should be tied to V_{CC}). The source of the clock input is one of the mask options that is specified before manufacture of the MCU.

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture. To avoid truncation errors, the prescaler is cleared when bit 3 of the timer counter register is written to a logic one. (This bit always needs a logic 0).

The timer continues to count past zero, falling from $00 to $FF and then continuing the countdown. Thus, the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones, the timer interupt request bit (bit 7) is cleared and the timer interrupt mask bit (bit 6) is set.

SELF-CHECK

The self-check capability of the EF6805P2 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in figure 9 and monitor the output of port C bit 3 for an oscillation of approximately 7Hz. A 10 volt level on the TIMER input, pin 7, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, interrupts, and I/O ports.

RESETS

The MCU can be reset three ways : by initial power-up, by the external reset input (RESET) and by optional, internal, low-voltage detect circuits. The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level. A typical reset Schmitt trigger hysteresis curve is shown in figure 11. The Schmitt trigger provides an internal reset voltage if it senses a logical zero on the RESET pin.

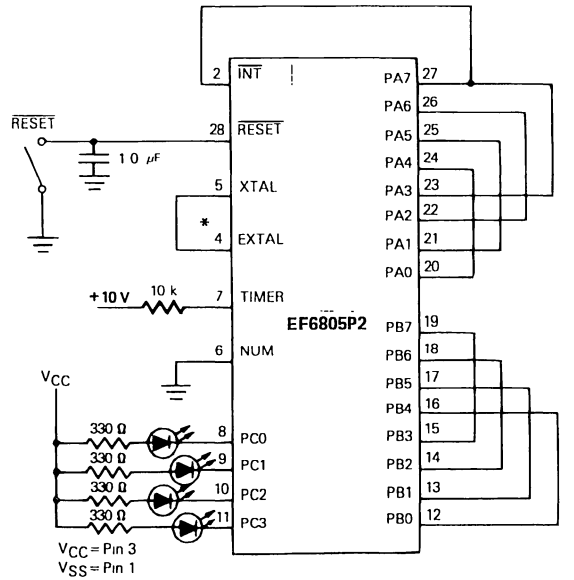
POWER-ON RESET (POR)

An internal reset is generated upon power-up that allows the internal clock generator to stabilize. A delay of t_{RHL} milliseconds is required before allowings the RESET input to go high. See the power and reset timing diagram (see figure 10). Connecting a capacitor to the RESET input (see figure 12) typically provides sufficient delay. During power-up, the Schmitt trigger switches on (removes reset) when RESET rise to V_{IRES}^{+}.

EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the RESET input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off are V_{IRES}- to provide an internal reset voltage.

Figure 9 : Self-check Connections.



SELF-CHECK ERROR PATTERNS

PC1	PC0	Problem
0	0	Interrupt Failure
0	1	Bad Port A or Port B
1	0	Bad RAM
1	1	Bad RAM
All 4 LEDs Flashing		Good Device

Note : When PC1 or PC0 is 0, the LED is on.

LOW VOLTAGE INHIBIT (LVI)

The optional low-voltage detection circuit causes a reset of the MCU if the power supply voltage falls below a certain level (V_{LVI}). The only requirement is that the V_{CC} must remain at or below the V_{LVI} threshold for one t_{cyc} minimum.

In this applications, the V_{CC} bus filter capacitor will eliminate negative-going voltage glitches of less than one t_{cyc}. The output from the low-voltage detector is connected directly to the internal reset circuitry. It also forces the RESET pin low via a strong discharge device through a resistor. The internal reset is removed once the power supply voltage rises above a recovery level (V_{LVR}) at which time a normal power-one reset occurs.

INTERNAL CLOCK GENERATORS OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs.

A manufacturing mask option is used to select crystal or resistor operation.

The different connection methods are shown in figure 13. Crystal specifications and suggested PC board layouts are given in figure 14. A resistor selection graph is given in figure 15.

The crystal oscillator start-up time is a function of many variables : crystal parameters (especially R_S, oscillator load capacitances, IC parameters, ambient temperature, supply voltage and supply voltage turn-on time). To ensure rapid oscillator start-up, neitherthe crystal characteristics nor the load capacitances should exceed recommendations.

When utilizing the on-board oscillator, the MCU should remain in the reset condition ($\overline{RESET}$ pin voltage below V_{IRES}^+) until the oscillator has stabilized at its operating frequency. Several factors are involved in calculating current specifications.

One V_{CC} minimum is reached, the external $\overline{RESET}$ capacitor will begin to charge at a rate dependent on the capacitor value. The charging current is supplied from V_{CC} through a large resistor, so its functions almost like a constant current source until the reset voltages rises above V_{IRES}^+. Therefore, the RESET pin will charge a approximately

$(V_{IRES}^+) \cdot C_{ext} = I_{RES} \cdot t_{RHL}$

Figure 10 : Power and Reset Timing.

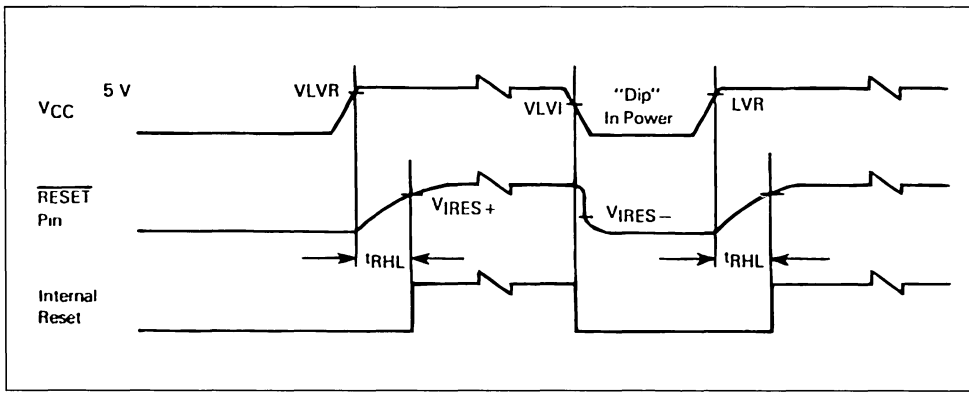


Figure 11 : Typical Reset Schmitt Trigger Hysteresis.

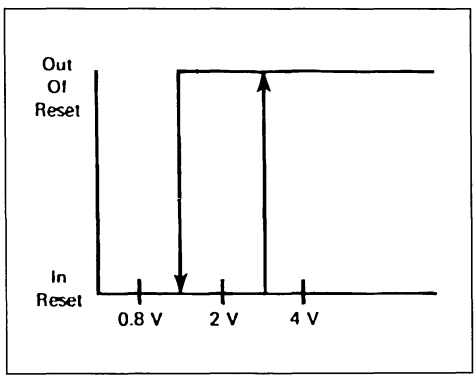


Figure 12 : Power-up Reset Delay Circuit.

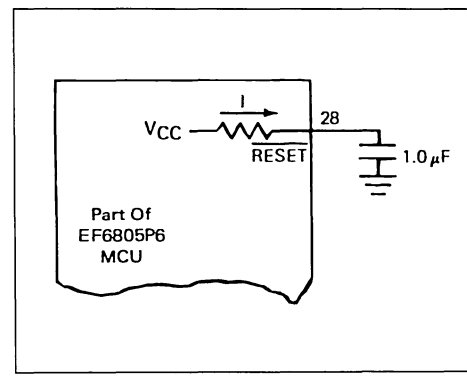
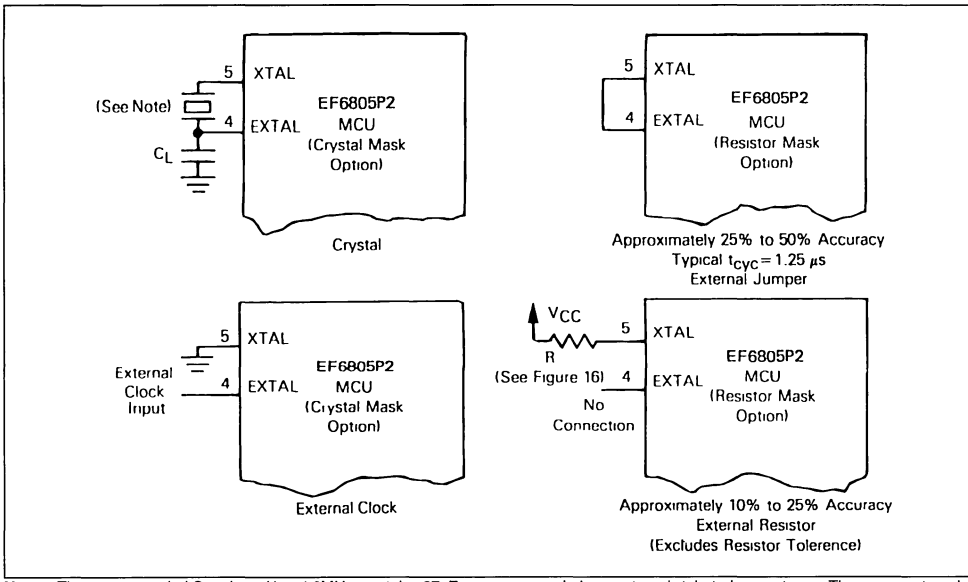
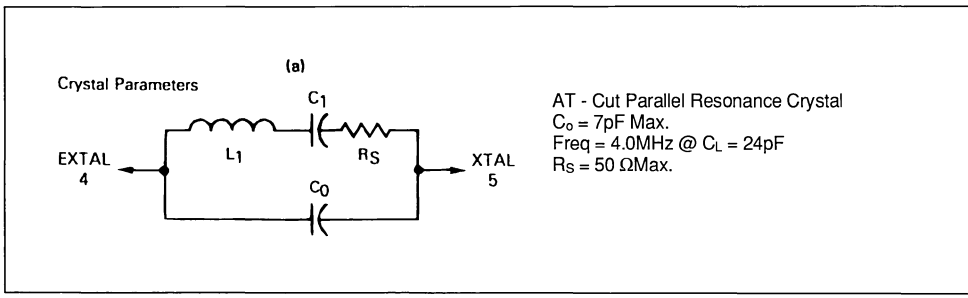


Figure 13 : Clock Generator Options.



Note : The recommended C_L value with a 4.0MHz crystal is 27pF, maximum, including system distributed capacitance. There is an internal capacitance of approximately 25pF on the XTAL pin For crystal frequencies other than 4MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2MHz crystal, use approximately 50pF on EXTAL and approximately 25pF on XTAL The exact value depends on the Motional-Arm parameters of the crystal used

Figure 14 : Crystal Motional ARM Parameters and Suggested PC Board Layout.



Note : Piezoelectric ceramic resonators which have the equivalent specifications may be use instead of crystal oscillator. Follow ceramic resonator manufacturer's suggestions for C_0, C_1 and R_S values..

Figure 14 : Crystal Motional Arm Parameters and Suggested PC Board Layout (continued).

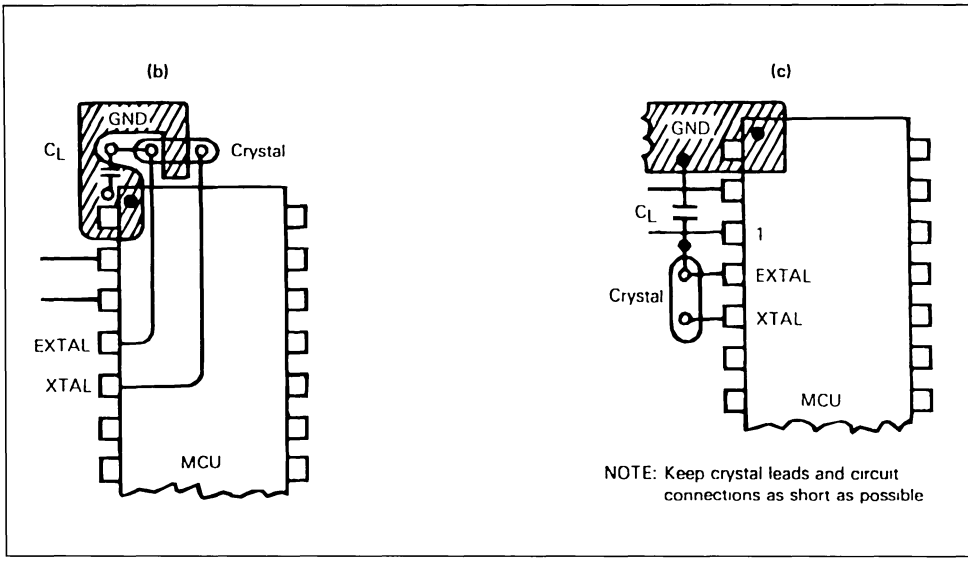
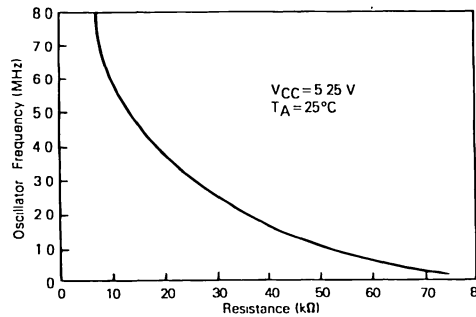


Figure 15 : Typical Frequency Selection for Resistor Oscillator Option.



INTERRUPTS

The EF6805P2 MCU can be interrupted three different ways through the external interrupt (INT) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs, the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires a total of 11 t_{cyc} periods for completion.

A flowchart of the interrupt sequence is shown in figure 16. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing ; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt device.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{INT}$. A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt, as shown in figure 17(a), for use as a zero-crossing detector (for negative transitions of the ac sinusoid). This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full-wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock.

For digital applications, the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "rearm" the internal logic. Therefore, the period can be calculated as follows : (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See figure 17(b).

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. Note that if the I bit is zero, SWI executes after the other interrupts. SWIs are usually used as break-points for debugging or as system calls.

Figure 16 : $\overline{RESET}$ and Interrupt Processing Flowchart.

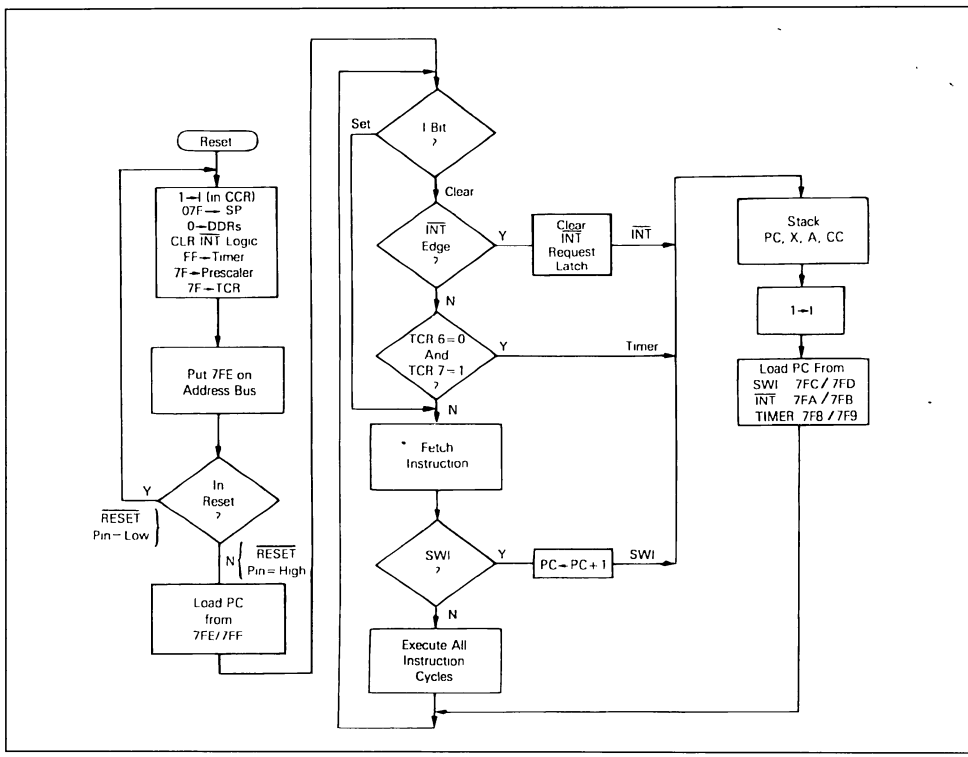
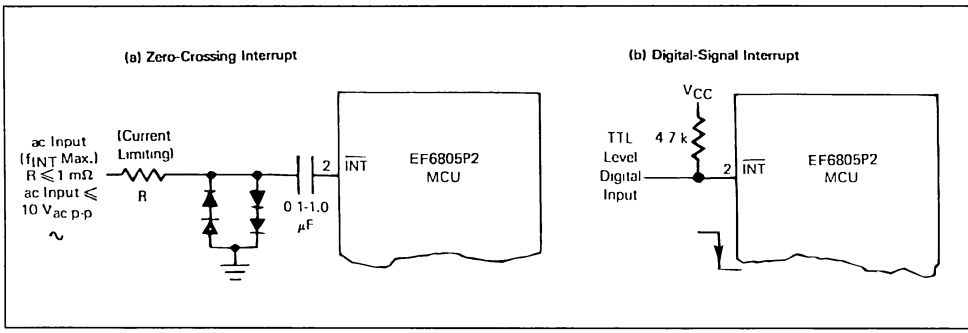


Figure 17 : Typical Interrupt Circuits.



INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may also be polled with branch instructions to provide an additional input pin. All pins (port A, B, and C) are programmable as either inputs or outputs under software control of the corresponding write-only data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic "1" for output or a logic "0" state to put the ports in the input mode. To avoid undefined levels, the port output registers are not initialized on reset, but may be written before setting the DDR bits. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading ; see figure 18. When port B is programmed for outputs, it is capable of sinking 10mA and sourcing 1mA on each pin.

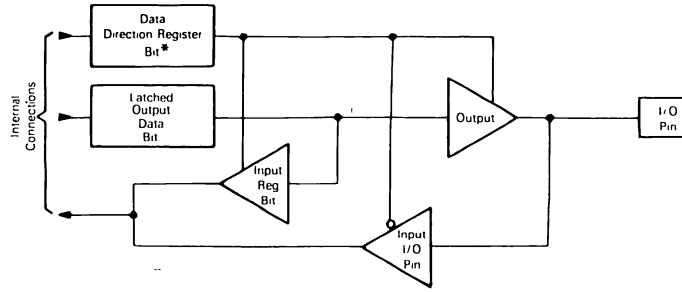
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in figure 5 gives the address of data registers and DDRs. The register configuration is provided in figure 19 and figure 20 provides some examples of port connections.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see figure 18) may always be written. Therefore, any write to a port writes all of its data bits even though the port DDR is set to input. This may be used to initialize the data registers and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input ("0") and corresponds to the latched output data when the DDR is an output ("1").

Figure 18 : Typical Port I/O Circuitry.



Data Direction Register Bit	Output Data Bit	Latched Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

* DDR is a write-only register and reads as all "1s"

** Ports A (with CMOS drive disabled), B, and C are three state ports Port A has optional internal pullup devices to provide CMOS drive capability See Electrical Characteristics tables for complete information

Figure 19 : MCU Register Configuration.

PORT DATA REGISTER

7 ... 0

Port A Addr = $000
Port B Addr = $001
Port C Addr = $002 (Bits 0→3)

PORT DATA DIRECTION REGISTER (DDR)

7 ... 0

(1) Write Only; reads as all "1s"
(2) 1 = Output; 0 = Input Cleared to 0 by reset
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006 (Bits 0→3)

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0	
		1	1	1	1	1	1	$009

TCR7 — Timer Interrupt Status Bit Set when TDR goes to zero, must be cleared by software Cleared to 0 by reset

TCR6 Bit 6 — Timer Interrupt Mask Bit 1 = timer interrupt masked (disabled) Set to 1 by reset

TCR Bits 5, 4, 3, 2, 1, 0 read as "1s" — unused bits

TIMER DATA REGISTER (TDR)

7 ... 0

MSB ... LSB $008

Figure 20 (a) : Typical Output Mode Port Connections.

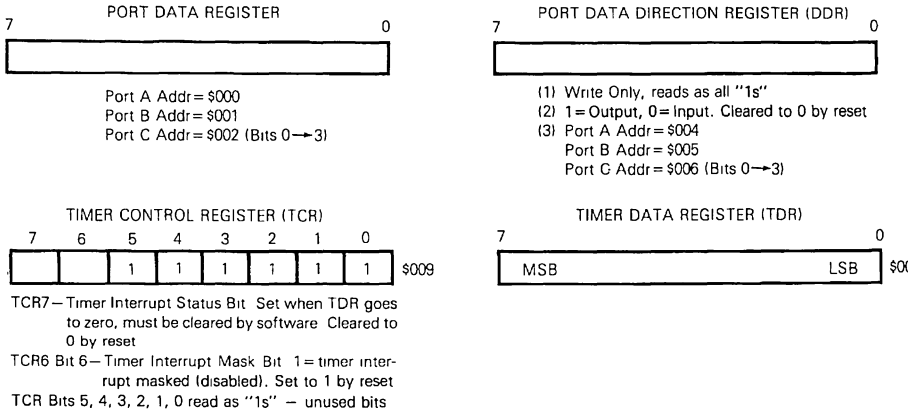
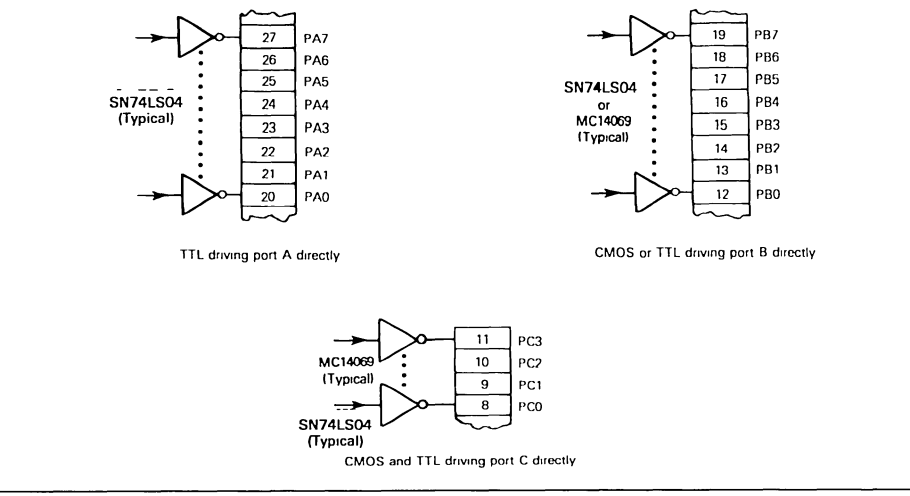


Figure 20 (b) : Typical Output Mode Port Connections.



SOFTWARE

BIT MANIPULATION

The EF6805P2 MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction register, see Caution below), with a single instruction (BSET, BCLR). Any bit in page zero including ROM, except the DDRs, can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. The carry bit equals the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

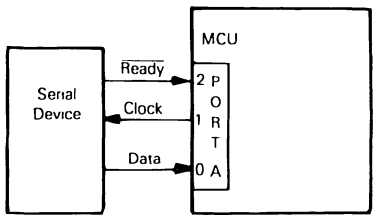
The coding example in figure 21 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device.

The external device has a data ready signal, a data output line, and a clock line to clock data one bit at a time. LSB first, out of the device. The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in a RAM location.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

Figure 21 : Bit Manipulation Example.




```
                .
                .
                .
SELF    BRSET   2, PORTA, SELF
                .
                .
                .
        BSET    1, PORTA
        BRCLR   0, PORTA, CONT
CONT    BCLR    1, PORTA
        ASR     RAMLOC
                .
                .
                .
```

ADDRESSING MODES

The EF6805P2 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the 6805 Family User's Manual.

The term "effective address" (EA) is used in describing the address modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE - In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g;, a constant used to initialize a loop counter).

DIRECT - In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addresing allows the user to directly address the lowest 256 bytes in memory with a single 2-byte instruction. This includes the on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED - In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The as-

sembler automatically selects the shortest for of the instruction.

RELATIVE - The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from - 126 to + 129 from the opcode address. The programmer need not worry about calculating the correct offset when using the Motorola assembler since it calculates the proper offset and checks to see if it is within the span of the branch.

INDEXED, NO OFFSET - In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET - In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET - In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset, except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR - In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

BIT TEST AND BRANCH - The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit and condition (set or clear) which is to be tested is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset is in the third byte and is added to the value of the PC if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from - 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

INHERENT - In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The EF6805P2 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS - Most of these instructions use two operands. One operand is either the accumulator or the index register. The

other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operands. Refer to table 1.

READ-MODIFY-WRITE MODIFICATIONS - These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is included in read-modify-write instructions through it does not perform the write. Rfer to table 2.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

BRANCH INSTRUCTIONS - The branch instructions cause a branch from the program when a certain condition is met. Refer to table 3.

BIT MANIPULATION INSTRUCTIONS - These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test branch operations. Refer to table 4.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

CONTROL INSTRUCTIONS - The control instructions control the MCU operations during program execution. Refer to table 5.

ALPHABETICAL LISTING - The complete instruction set is given in alphabetical order in table 6.

OPCODE MAP SUMMARY - Table 7 is an opcode map for the instructions used on the MCU.

Table 1 : Register/Memory Instructions.

Function	Mnemonic	Addressing Modes: Immediate Op Code	Immediate # Bytes	Immediate # Cycles	Direct Op Code	Direct # Bytes	Direct # Cycles	Extended Op Code	Extended # Bytes	Extended # Cycles	Indexed (No Offset) Op Code	Indexed (No Offset) # Bytes	Indexed (No Offset) # Cycles	Indexed (8 Bit Offset) Op Code	Indexed (8 Bit Offset) # Bytes	Indexed (8 Bit Offset) # Cycles	Indexed (16 Bit Offset) OP Code	Indexed (16 Bit Offset) # Bytes	Indexed (16 Bit Offset) # Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DF	3	6
Store A in Memory	STA	–	–		B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	-		BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP				BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–			BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 2 : Read-Modify-Write Instructions.

Function	Mnemonic	Addressing Modes: Inherent (A) Op Code	Inherent (A) # Bytes	Inherent (A) # Cycles	Inherent (X) Op Code	Inherent (X) # Bytes	Inherent (X) # Cycles	Direct Op Code	Direct # Bytes	Direct # Cycles	Indexed (No Offset) Op Code	Indexed (No Offset) # Bytes	Indexed (No Offset) # Cycles	Indexed (8 B t Offset) Op Code	Indexed (8 B t Offset) # Bytes	Indexed (8 B t Offset) # Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 3 : Branch Instructions.

		Relative Addressing Mode		
Function	**Mnemonic**	**Op Code**	**# Bytes**	**# Cycles**
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(branch IFF higher or same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(branch IFF lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF interrupt mask bit is clear.	BMC	2C	2	4
Branch IFF interrupt mask bit is set.	BMS	2D	2	4
Branch IFF interrupt line is low.	BIL	2E	2	4
Branch IFF interrupt line is high.	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 4 : Bit Manipulation Instructions.

		Addressing Modes					
		Bit Set/clear			Bit Test and Branch		
Function	**Mnemonic**	**Op Code**	**# Bytes**	**# Cycles**	**Op Code**	**# Bytes**	**# Cycles**
Branch IFF Bit n is set	BRSET n (n = 0... 7)				2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0... 7)				01 + 2 • n	3	10
Set Bit n	BSET n (n = 0... 7)	10 + 2 • n	2	7			
Clear Bit n	BCLR n (n = 0... 7)	11 + 2 • n	2	7			

Table 5 : Control Instructions.

		Inherent		
Function	**Mnemonic**	**Op Code**	**# Bytes**	**# Cycles**
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-operation	NOP	9D	1	2

Table 6 : Instruction Set.

	Addressing Modes										Condition Code				
Mnem	**Inherent**	**Immediate**	**Direct**	**Extended**	**Relative**	**Indexed (no offset)**	**Indexed (8 Bits)**	**Indexed (16 Bits)**	**Bit Set/Clear**	**Bit Test & Branch**	**H**	**I**	**N**	**Z**	**C**
ADC		X	X	X		X	X	X			↕	•	↕	↕	↕
ADD		X	X	X		X	X	X			↕	•	↕	↕	↕
AND		X	X	X		X	X	X			•	•	↕	↕	•
ASL	X		X			X	X				•	•	↕	↕	↕
ASR	X		X			X	X				•	•	↕	↕	↕
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	↕	↕	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	↕
BRSET										X	•	•	•	•	↕
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLL	X										•	•	•	•	0

Table 6 : Instruction Set (continued).

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/clear	Bit Test & Branch	H	I	N	Z	C
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	^	^	^
COM	X		X			X	X				●	●	^	^	1
CPX		X	X	X		X	X	X			●	●	^	^	^
DEC	X		X			X	X				●	●	^	^	●
EOR		X	X	X		X	X	X			●	●	^	^	●
INC	X		X			X	X				●	●	^	^	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	^	^	●
LDX		X	X	X		X	X	X			●	●	^	^	●
LSL	X		X			X	X				●	●	^	^	^
LSR	X		X			X	X				●	●	0	^	^
NEQ	X		X			X	X				●	●	^	^	^
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	^	^	●
ROL	X		X			X	X				●	●	^	^	^
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols :

H	Half Carry (from bit 3)	Z	Zero
I	Interrupt Mask	C	Carry/borrow
N	Negative (sign bit)	^	Test and Set if True, Cleared Otherwise
		•	Not Affected

HMOS 6805 FAMILY

Features	EF6805P2	EF6805P6	EF6805R2	EF6805R3	EF6805U2	EF6805U3
Technology	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	40	40	40
On-chip RAM (bytes)	64	64	64	112	64	112
On-chip User ROM (bytes)	1100	1796	2048	3776	2048	3776
External Bus	None	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	24	24	24
Unidirectional I/O Lines	None	None	6 Inputs	6 Inputs	8 Inputs	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, A/D	Timer	Timer
External Interrupt Inputs	1	1	2	2	2	2
STOP and WAIT	No	No	No	No	No	No

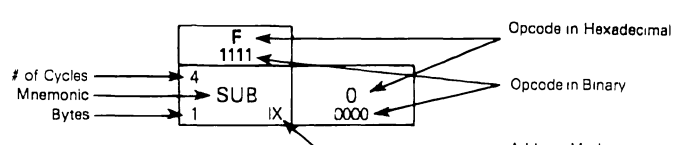
Table 7 : 6805 HMOS Family Opcode MAP.

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DIR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	5 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 FSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

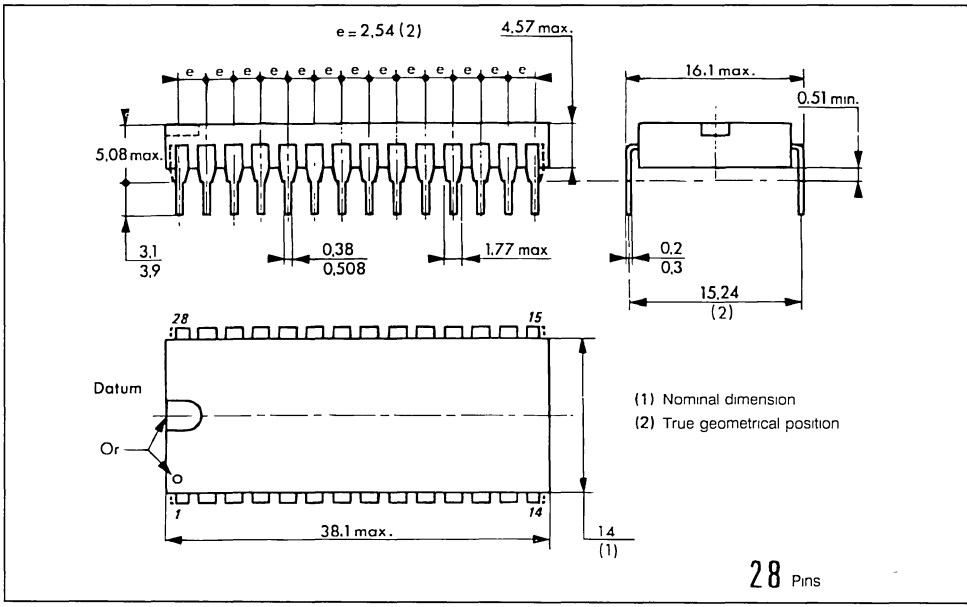
INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

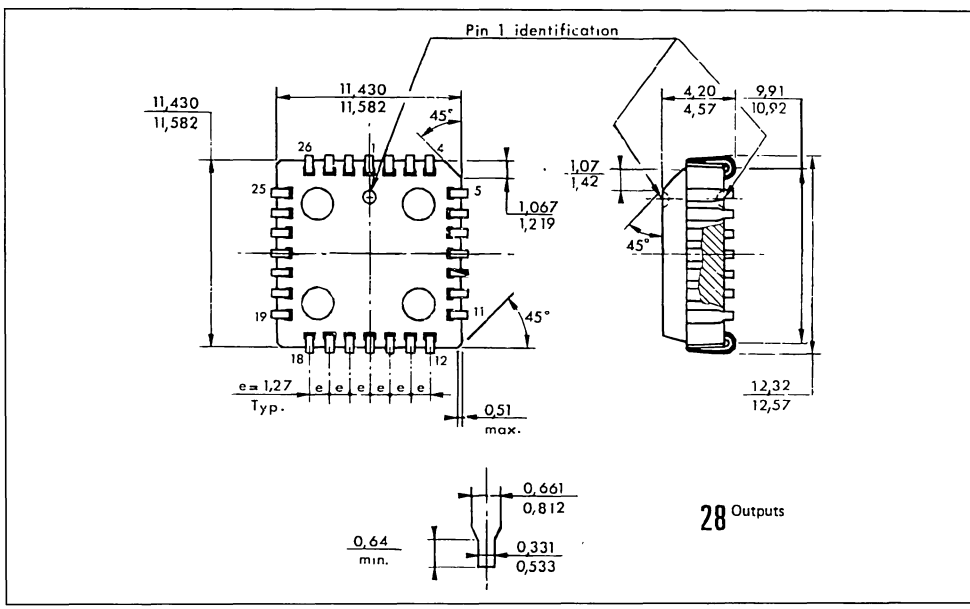


PACKAGE MECHANICAL DATA

CB–132 PLASTIC PACKAGE



CB–520 PLCC28



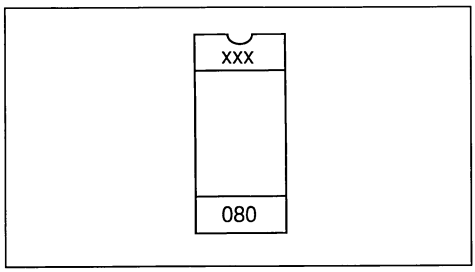
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS-THOMSON on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local SGS-THOMSON representative or distributor.

EPROMs

One 2716 or 2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation.



XXX = Customer ID)

After the EPROM is marked, it should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS-THOMSON. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, SGS-THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask change and are not production parts. The RVUs are thus not guaranteed by SGS THOMSON. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6805 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

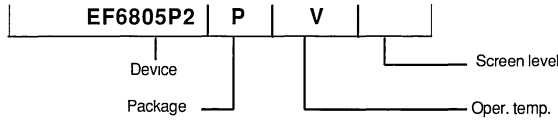
EFDOS is SGS-THOMSON Disk Operating System available on development systems such as DEVICE...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser...

* Requires prior factory approval.

Whenever ordering a custom MCU is required, please contact your local SGS-THOMSON representative or SGS-THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS-THOMSON Microelectronics representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device	Package					Oper. Temp			Screening Level			
	C		P		FN	L*	V	T	Std	D		
EF6805P2 (1.0MHz)			●		●	●	●		●	●		
Examples : EF6805P2P, EF6805P2FN, EF6805P2PLD, EF6805P2FNLD.												

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC
Oper. temp. : L* : 0°C to + 70°C, **V :** – 40 °C to + 85°C, **T :** – 40°C to + 105°C, *** :** may be omitted
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
EXORciser is a registered trademark of MOTOROLA Inc

EF6805 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....

.....

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
❑ SGS-THOMSON Microelectronic reference
.....
❑ Special customer data reference*
.....

ROM capacity required : | | | | bytes

Temperature range :
❑ 0°C / + 70°C
❑ – 40°C / + 85°C
❑ – 40°C / + 105°C

Quality level :
❑ STD
❑ D
❑ Other* (customer's quality specification ref.) :
.....

Package
❑ Plastic
❑ PLCC

Software developped by :
❑ SGS-THOMSON Microelectronic application lab.
❑ External lab.
❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
❑ EPROM Reference :
❑ EFDOS/MDOS* disk file
  ❑ 8" floppy
  ❑ 5" 1/4 floppy
❑ Other *

OPTION LIST

-Oscillator input
❑ Xtal
❑ RC

- Low voltage inhibit :
❑ Enabled
❑ Disabled

- Port A output drive :
❑ CMOS and TTL
❑ TTL only

- Timer clock source :
❑ Internal φ2 clock
❑ TIMER input pin

- Timer Prescaler
❑ 2^0 (divide by 1)
❑ 2^1 (divide by 2)
❑ 2^2 (divide by 4)
❑ 2^3 (divide by 8)
❑ 2^4 (divide by 16)
❑ 2^5 (divide by 32)
❑ 2^6 (divide by 64)
❑ 2^7 (divide by 128)

- Internal max. clock frequency :
❑ 1.0 MHz

Yearly quantity forecast

- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME : DATE : SIGNATURE :

SGS-THOMSON MICROELECTRONICS

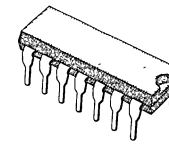
EF6805P6

8-BIT MICROCOMPUTER UNIT

ADVANCE DATA

HARDWARE FEATURES

- 8-BIT ARCHITECTURE
- 64 BYTES OF RAM
- MEMORY MAPPED I/O
- 1796 BYTES OF USER ROM
- 20 TTL/CMOS COMPATIBLE BIDIRECTIONAL I/O LINES (8 lines are LED compatible)
- ON-CHIP CLOCK GENERATOR
- SELF-CHECK MODE
- ZERO CROSSING DETECTION
- MASTER RESET
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE®
- 5V SINGLE SUPPLY



P
(Plastic Package)

FN
(PLCC 28)

SOFTWARE FEATURES

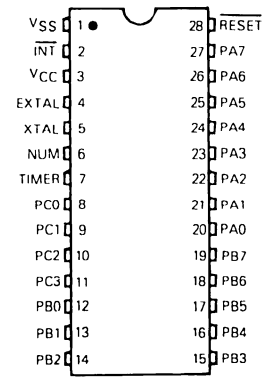
- SIMILAR TO 6800 FAMILY
- BYTE EFFICIENT INSTRUCTION SET
- EASY TO PROGRAM
- TRUE BIT MANIPULATION
- BIT TEST AND BRANCH INSTRUCTION
- VERSATILE INTERRUPT HANDLING
- VERSATILE INDEX REGISTER
- POWERFUL INDEXED ADDRESSING FOR TABLES
- FULL SET OF CONDITIONAL BRANCHES
- MEMORY USABLE AS REGISTER/FLAGS
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- 10 POWERFUL ADDRESSING MODES
- ALL ADDRESSING MODES APPLY TO ROM, RAM, AND I/O

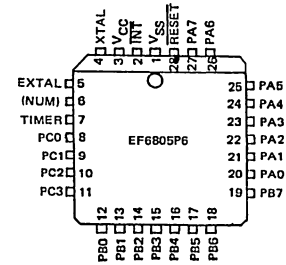
USER SELECTABLE OPTIONS

- INTERNAL 8-BIT TIMER WITH SELECTABLE CLOCK SOURCE (external timer input or internal machine clock)
- TIMER PRESCALER OPTION (7 Bits, 2^n)
- 8 BIDIRECTIONAL I/O LINES WITH TTL OR TTL/CMOS INTERFACE OPTION
- CRYSTAL OR LOW-COST RESISTOR OSCILLATOR OPTION
- LOW VOLTAGE INHIBIT OPTION
- VECTORED INTERRUPTS : TIMER, SOFTWARE, AND EXTERNAL
- PORT B OPEN DRAIN DRIVE OPTION

INICE® SGS-THOMSON development/emulation tool

PIN CONNECTIONS

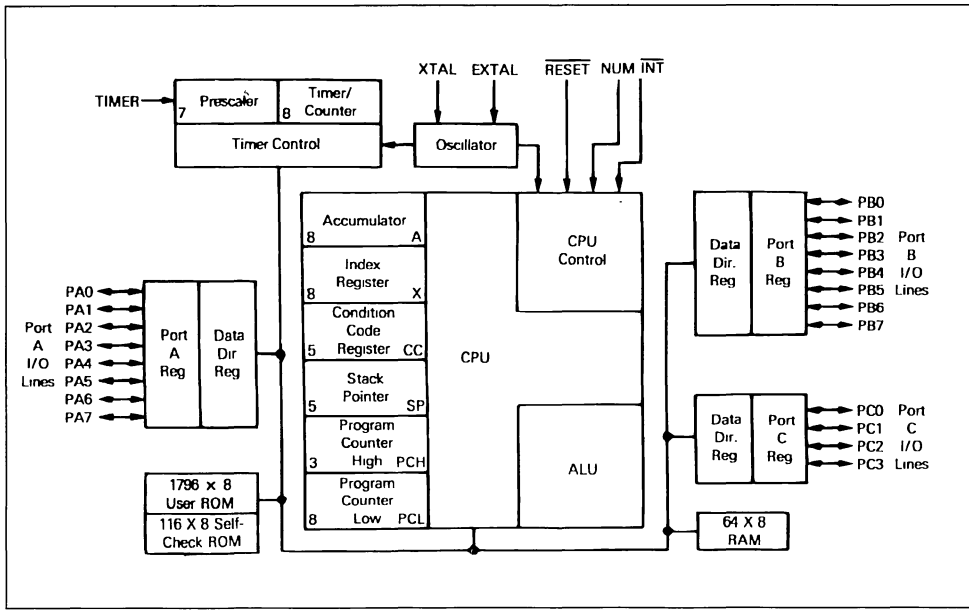




DESCRIPTION

The EF6805P6 Microcomputer Unit (MCU) is a member of the 6805 Family of low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the 6800-based instruction set. The following are some of the hardware and software high-lights of the EF6805P6 MCU.

Figure 1 : EF6805P6 HMOS Microcomputer Block Diagram.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage (except pin 6)	– 0.3 to + 7.0	V
T_A	Operating Temperature Range	0 to 70	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_J	Junction Temperature Plastic SURPICOP (Plastic chip-carrier)	150 150	°C

This device contains circuitry to protect time inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$ Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g. either V_{SS} or V_{CC}).

THERMAL DATA

θ_{JA}	Thermal Resistance Plastic SURPICOP (plastic chip-carrier)	120 120	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$T_J = T_A + (P_D \cdot \theta_{JA})$ (1)

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} << P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$P_D = K \div (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives :

$K = PD \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2$ (3)

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ELECTRICAL CHARACTERISTICS (V_{CC} = + 5.25 Vdc ± 0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0°C to 70°C unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage				V
	$\overline{\text{RESET}}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0		V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5		V_{CC}	
	$\overline{\text{INT}}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0	*	V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5	*	V_{CC}	
	All Other	2.0		V_{CC}	
V_{IH}	Input High Voltage Timer				V
	Timer Mode	2.0		V_{CC}	
	Self-check Mode		10.0	15.0	
V_{IL}	Input Low Voltage				V
	$\overline{\text{RESET}}$	– 0.3		0.8	
	$\overline{\text{INT}}$	– 0.3	*	1.5	
	All Other	– 0.3		0.8	
V_{IRES+} V_{IRES-}	$\overline{\text{RESET}}$ Hysteresis Voltage (See figures 11, 12, and 13)				V
	"Out of Reset"	2.1		4.0	
	"Into Reset"	0.8		2.0	
V_{INT}	$\overline{\text{INT}}$ Zero Crossing Input Voltage, Through a Capacitor	2.0		4.0	$V_{ac\ p\text{-}p}$
P_{INT}	Internal Power Dissipation – No Port Loading V_{CC} = 5.75V, T_A = 0°C		400	690	mW
C_{in}	Input Capacitance				pF
	XTAL		25		
	All Other		10		
V_{LVR}	Low Voltage Recover			4.75	V
V_{LVI}	Low Voltage Inhibit 0 to + 70°C	2.75	3.5		V
	– 40 to + 85°C	3.1	3.5		
I_{in}	Input Current				µA
	$\overline{\text{TIMER}}$ (V_{in} = 0.4V)			20	
	$\overline{\text{INT}}$ (V_{in} = 2.4V to V_{CC})		20	50	
	EXTAL (V_{in} = 2.4V to V_{CC}, crystal option)			10	
	_____ (V_{in} = 0.4V, crystal option)			– 1600	
	$\overline{\text{RESET}}$ (V_{in} = 0.8V) (external capacitor charging current)	– 4.0		– 50	

* Due to internal biasing, this input (when unused) floats to approximately 2 0 Vdc.

PORT DC ELECTRICAL CHARACTERISTICS (V_{CC} = + 5.25 Vdc ± 0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0°C to 70°C unless otherwise noted)

PORT A WITH CMOS DRIVE ENABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.4			V
V_{OH}	Output High Voltage, I_{Load} = – 10µA	3.5			V
V_{IH}	Input High Voltage, I_{Load} = – 300µA (max)	2.0		V_{CC}	V
V_{IL}	Input Low Voltage, I_{Load} = – 500µA (max)	0.3		0.8	V
I_{IH}	Hi-Z State Input Current (V_{in} = 2.0V to V_{CC})			– 300	µA
I_{IL}	Hi-Z State Input Current (V_{in} = 0.4V)			– 500	µA

PORT B

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 3.2mA			0.4	V
V_{OL}	Output Low Voltage, I_{Load} = 10mA (sink)			1.0	V
V_{OH}	Output High Voltage, I_{Load} = – 200µA	2.4			V
I_{OH}	Darlington Current Drive (source), V_O = 1.5V	– 1.0		– 10	mA
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	– 0.3		0.8	V
I_{TSI}	Hi-Z State Input Current		2	20	µA

PORT B WITH OPEN DRAIN OPTION

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OH}	Output High Voltage	2.4		13.0	V
I_{TSI}	Hi-Z State Input Current		2	20	µA

PORT C AND PORT A WITH CMOS DRIVE DISABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.4			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	– 0.3		0.8	V
I_{TSI}	Hi-Z State Input Current		2	20	µA

SWITCHING CHARACTERISTICS (V_{CC} = + 5.25 Vdc ± 0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0°C to 70°C unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency	0.4		4.2	MHz
t_{cyc}	Cycle Time ($4/f_{osc}$)	0.95		10	µs
t_{WL}, t_{WH}	$\overline{INT}$ and TIMER Pulse Width (see interrupt section)	t_{cyc} + 250			ns
t_{RWL}	$\overline{RESET}$ Pulse Width	t_{cyc} + 250			ns
t_{RHL}	$\overline{RESET}$ Delay Time (external capacitance = 10µF)		100		ms
f_{INT}	$\overline{INT}$ Zero Crossing Detection Input Frequency (± 5° accuracy)	0.03		1.0	kHz
	External Clock Input Duty Cycle (EXTAL)	40	50	60	%

Figure 3 : TTL Equivalent Test Load (port B).

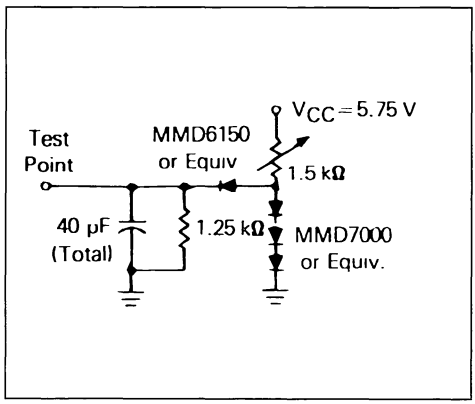


Figure 4 : CMOS Equivalent Test Load (port A).

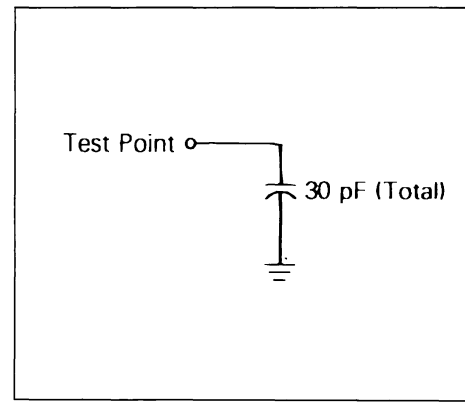
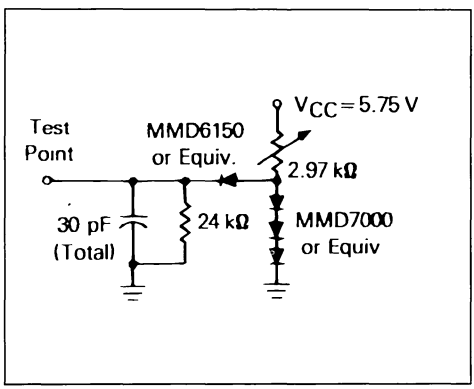


Figure 5 : TTL Equivalent Test Load (ports A and C).



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in figure 1, are described in the following paragraphs.

V_{CC} AND V_{SS}. Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

INT. This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts section for additional information.

XTAL AND EXTAL. These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on the user selectable manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead lengths and stray capacitance on these two pins should be minimized. Refer to Internal Clock Generator Options section for recommendations about these inputs.

TIMER. This pin allows an external input to be used to decrement the internal timer circuitry. Refer to Timer section for additional information about the timer circuitry.

RESET. This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to Resets section for additional information.

NUM. This pin is not for user application and must be connected to V_{SS}.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC3). These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to Inputs/outputs section for additional information.

MEMORY

As shown in figure 6, the MCU is capable of addressing 2048 bytes of memory and I/O registers with its program counter. The EF6805P6 MCU has implemented 1984 of these locations. This consists of : 1796 bytes of user ROM, 116 bytes of self-check ROM, 64 bytes of user RAM, 6 bytes of port I/O, and 2 timer registers.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in figure 7. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first ; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly, during pulls from the stack, since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack. The remaining CPU registers are not pushed.

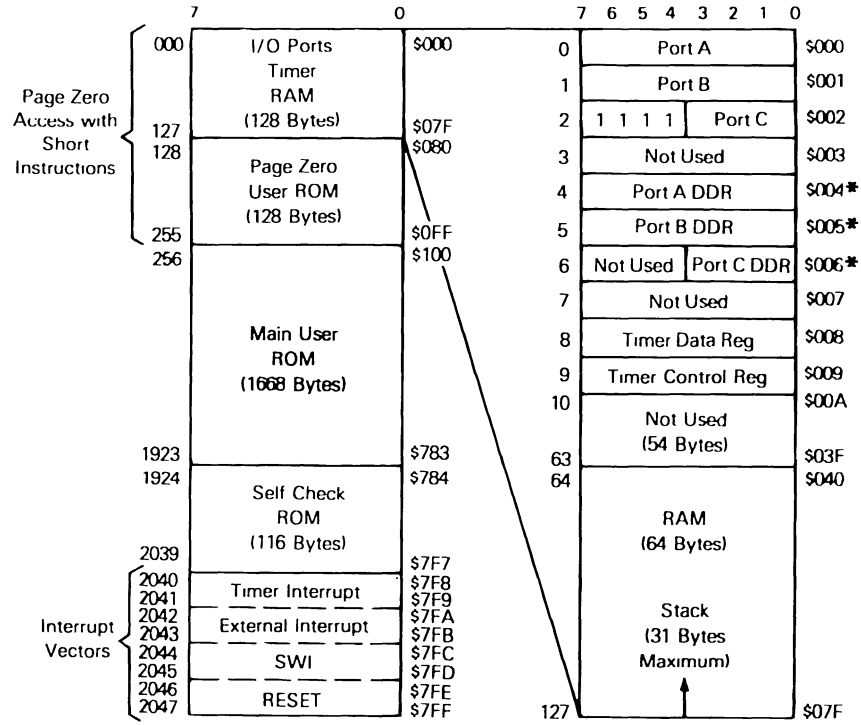
CENTRAL PROCESSING UNIT

The CPU of the EF6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

REGISTERS

The 6805 Family CPU has five registers available to the programmer. They are shown in figure 8 and are explained in the following paragraphs.

Figure 6 : MCU Address Map.



* Caution : Data direction registers (DDRs) are wnte-only ; they read as $FF.

Figure 7 : Interrupt Stacking Order.

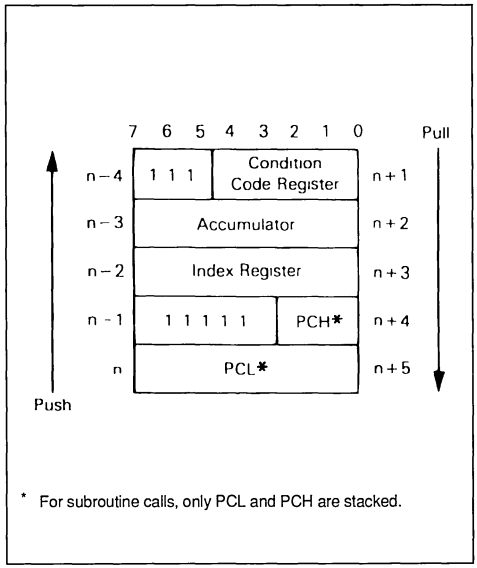
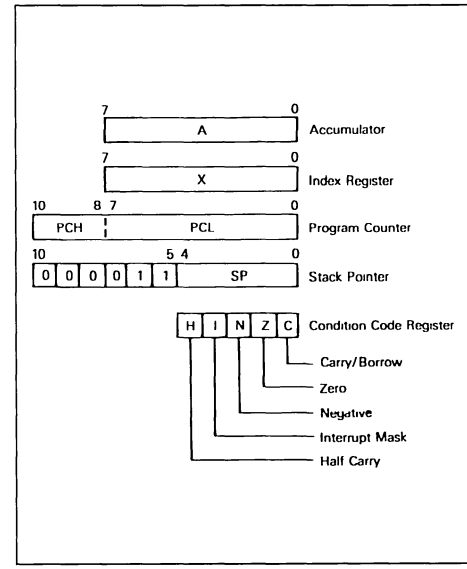


Figure 8 : Programming Model.



ACCUMULATOR (A). The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X). The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC). The program counter is an 11-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP). The stack pointer is an 11-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is pushed onto the stack and incremented as data is pulled from the stack. The six most significant bits of the stack pointer are permanently configured to 000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls.

CONDITION CODE REGISTER (CC). The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

HALF CARRY (H) - Set during ADD and ADC instructions to indicate that a carry occurred between bits 3 and 4.

INTERRUPT (I) - This bit is set to mask (disable) the timer and external interrupt (INT). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt is cleared.

NEGATIVE (IN) - Used to indicate that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in result equal to a logical one).

ZERO (Z) - Used to indicate that the result of the last arithmetic, logical, or data manipulation was zero.

CARRY/BORROW (C) - Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The EF6805P6 MCU timer circuitry is shown in figure 9. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9, and executing the interrupt routine ; see the Interrupt section. THE TIMER INTERRUPT REQUEST BIT MUST BE CLEARED BY SOFTWARE.

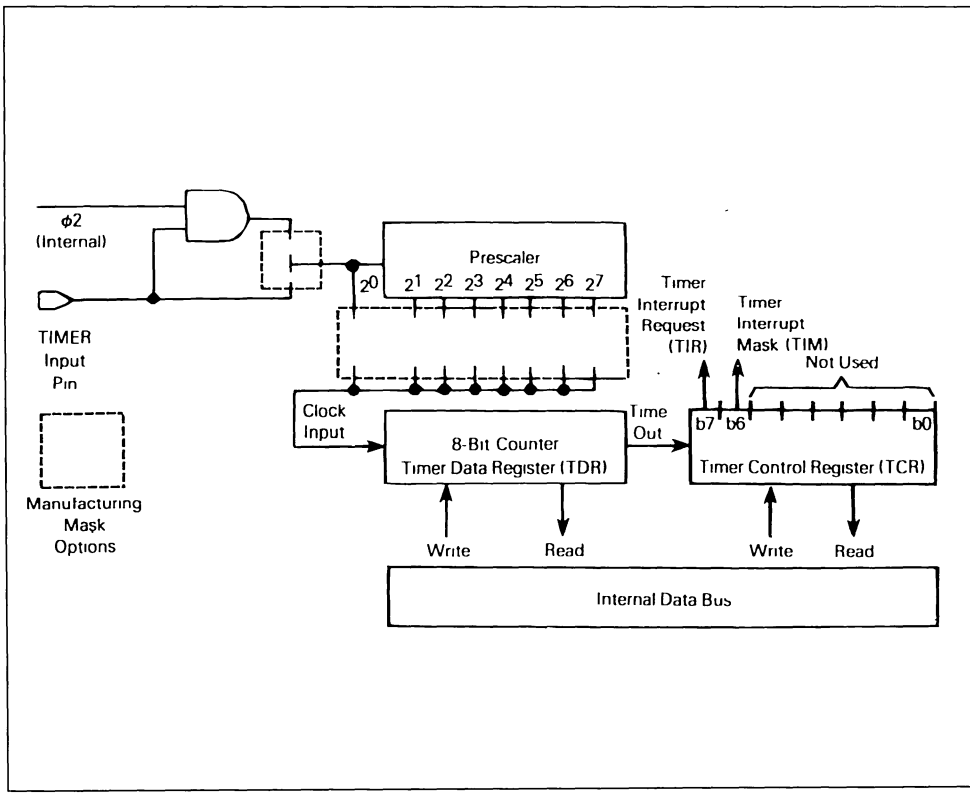
The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin or it can be the internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER or INT pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows : (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. (NOTE : For ungated $\phi 2$ clock inputs to the timer prescaler, the TIMER pin should be tied to V_{CC}). The source of the clock input is one of

Figure 9 : Timer Block Diagram.



the mask options that is specified before manufacture of the MCU.

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture.

The timer continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones, the timer interrupt request bit (bit 7) is cleared, and the timer interrupt mask bit (bit 6) is set.

SELF-CHECK

The self-check capability of the EF6805P6 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in figure 10 and monitor the output of port C bit 3 for an oscillation of approximately 7Hz. A 9-volt level on the TIMER input, pin 7, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, interrupts, and I/O ports.

RESETS

The MCU can be reset three ways : by initial power-up, by the external reset input (RESET), and by an optional internal low voltage detect circuit ; see figure 11. The internal circuit connected to the RESET pin consists of a Schmitt trigger which senses the RESET line logic level. The Schmitt trigger provides an internal reset voltage if it senses a logic "0" on the RESET pin. During power-up, the Schmitt trigger switches on (removes reset) when the RESET pin voltage rises to V_{IRES+}. When the RESET pin voltage falls to a logical "0" for a period longer than one t_{cyc}, the Schmitt trigger switches off to provide an internal reset voltage. The "switch off" voltage occurs at V_{IRES-}. A typical reset Schmitt trigger hysteresis curve is shown in figure 12.

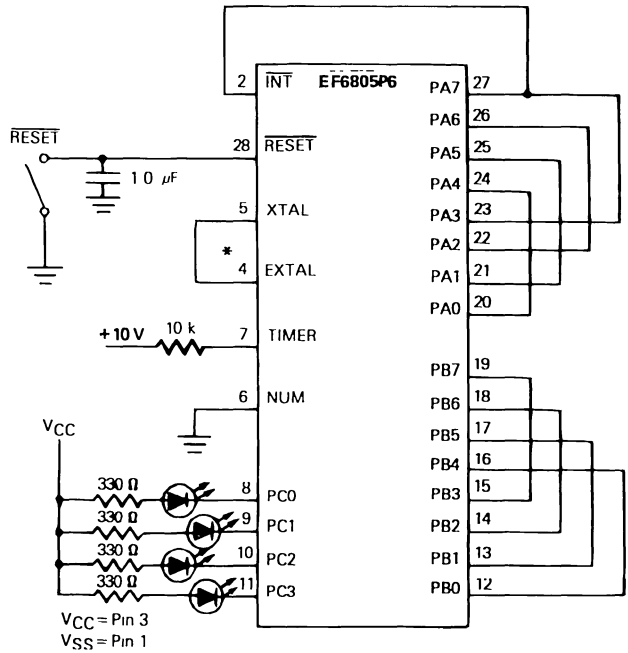
During power-up, a delay of t_{RHL} is needed before allowing the RESET input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the RESET input, as shown in figure 13, typically provides sufficient delay. See figure 17 under Interrupts section for the complete reset sequence.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The oscillator frequency is internally divided by four to produce the internal system clocks.

The different connection methods are shown in figure 14. The crystal specifications and suggested PC board layouts are given in figure 15. A resistor selection graph is given in figure 16.

Figure 10 : Self-check Connections.



* This connection depends on the clock oscillator user selectable mask option Use crystal if that option is selected

The crystal oscillator startup time is a function of many variables : crystal parameters (especially R_S), oscillator load capacitance, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

Figure 11 : Power and $\overline{\text{RESET}}$ Timing.

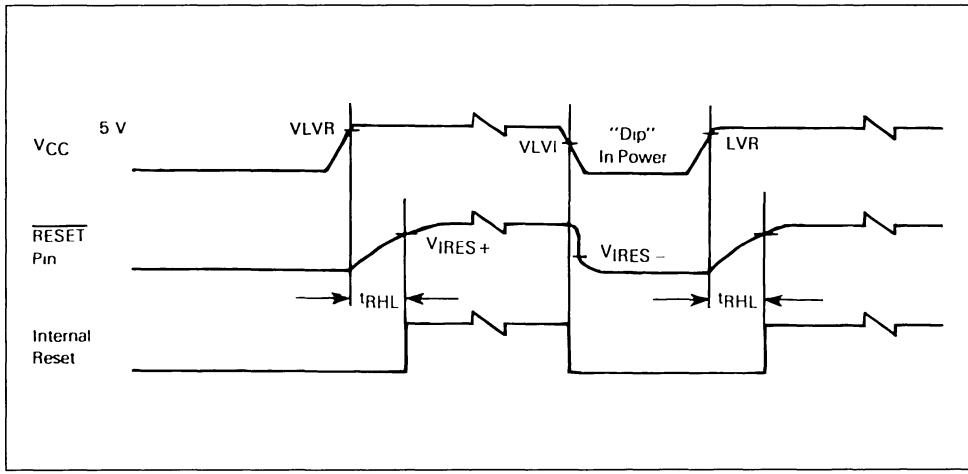


Figure 12 : Typical Reset Schmitt Trigger Hysteresis.

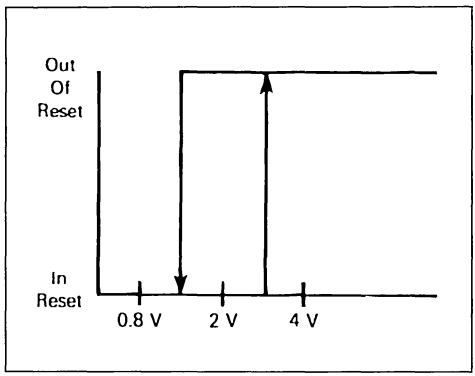


Figure 13 : Power-up Reset Delay Circuit.

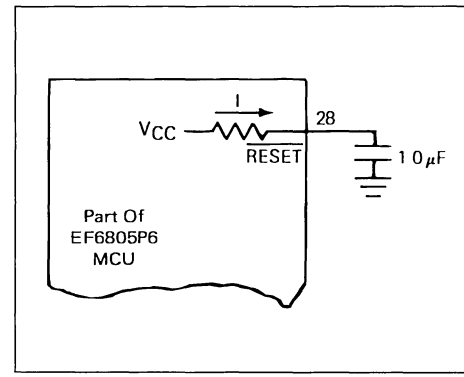
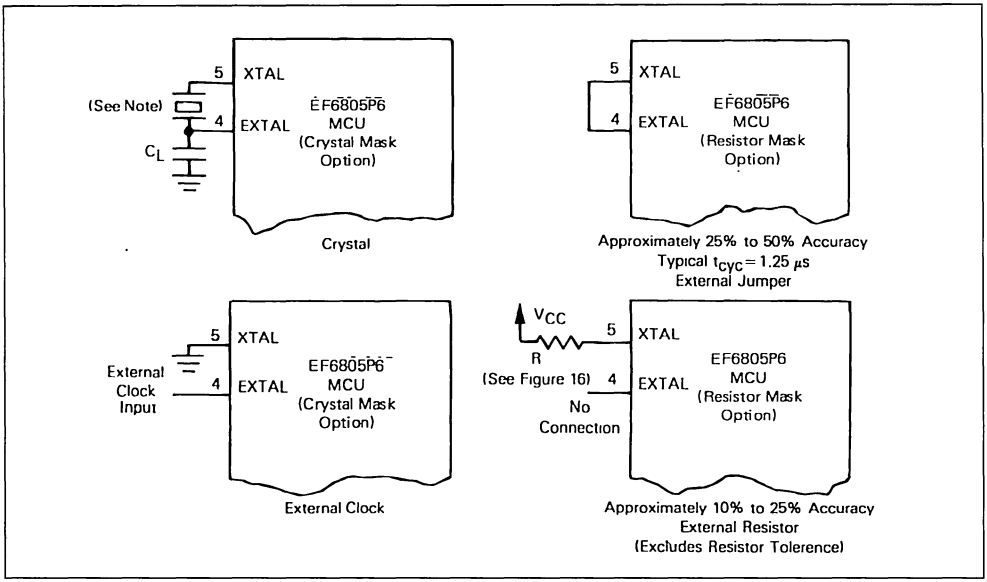
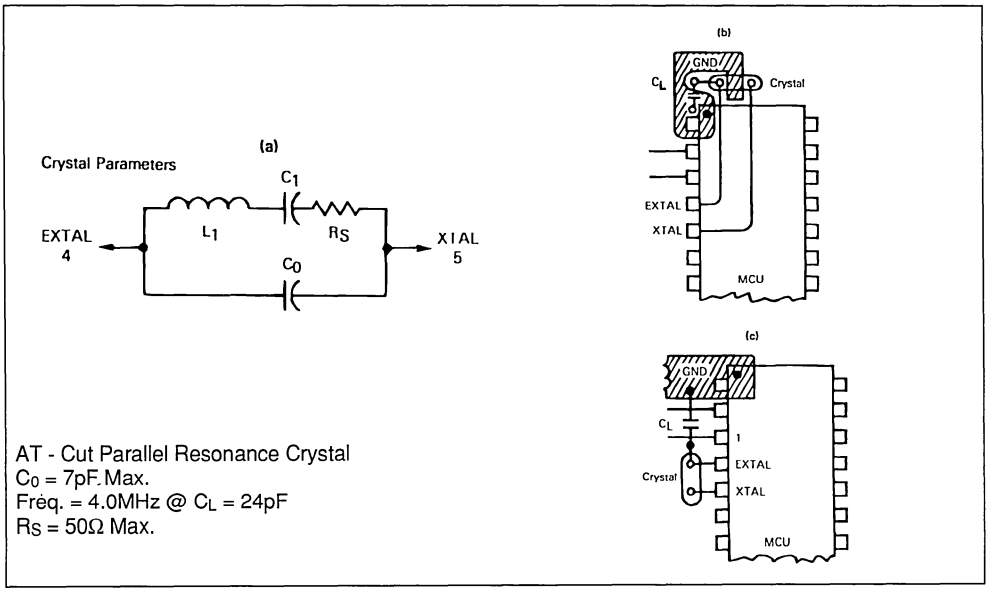


Figure 14 : Clock Generator Options.



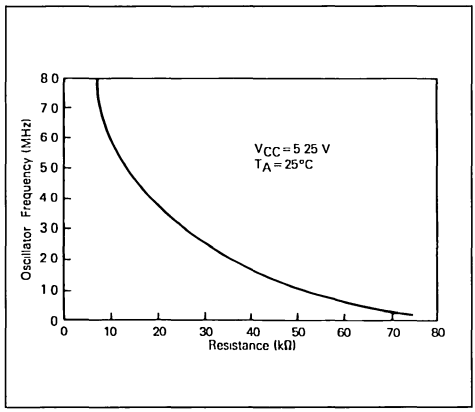
Note : The recommended C_L value with a 4.0MHz crystal is 27pF, maximum, including system distributed capacitance There is an internal capacitance of approximately 25pF on the XTAL pin. For crystal frequencies other than 4MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2MHz crystal, use approximately 50pF on EXTAL and approximately 25pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

Figure 15 : Crystal Motional Arm Parameters and Suggested PC Board Layout.



Note : Keep crystal leads and circuit connections as short as possible.

Figure 16 : Typical Frequency Selection for Resistor Oscillator Option.



INTERRUPTS

The EF6805P6 MCU can be interrupted three different ways : through the external interrupt (INT) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs : processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires a total of 11 t_{cyc} periods for completion.

A flowchart of the interrupt sequence is shown in figure 17. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing ; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of INT. A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt, as shown in figure 18(a), for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock.
For digital applications, the INT pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or INT pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on

Figure 17 : $\overline{\text{RESET}}$ and Interrupt Processing Flowchart.

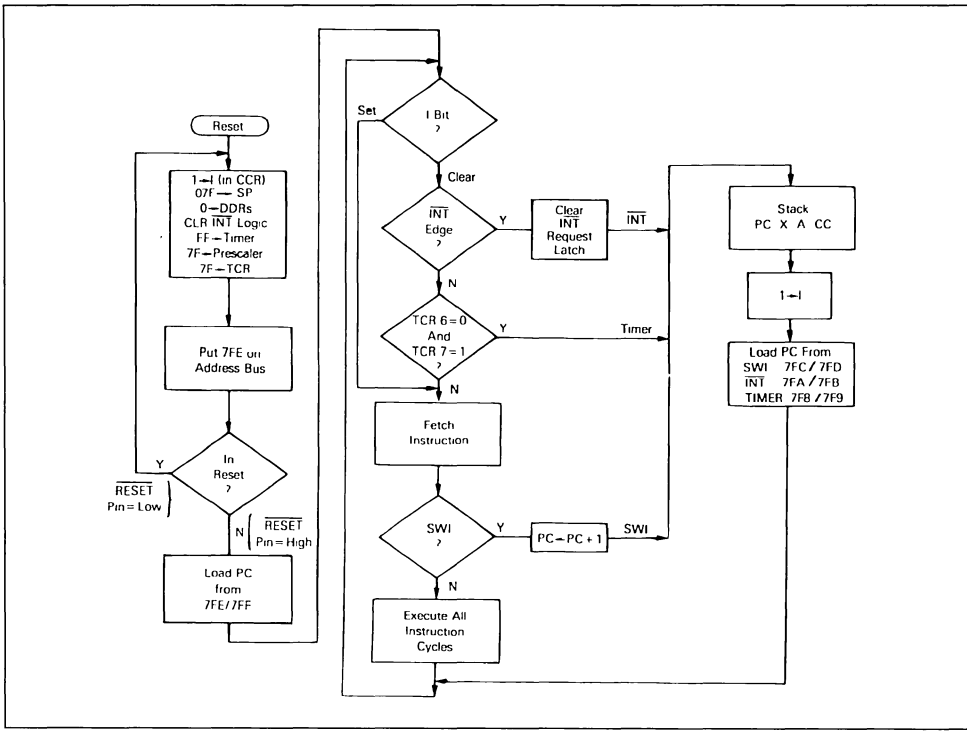
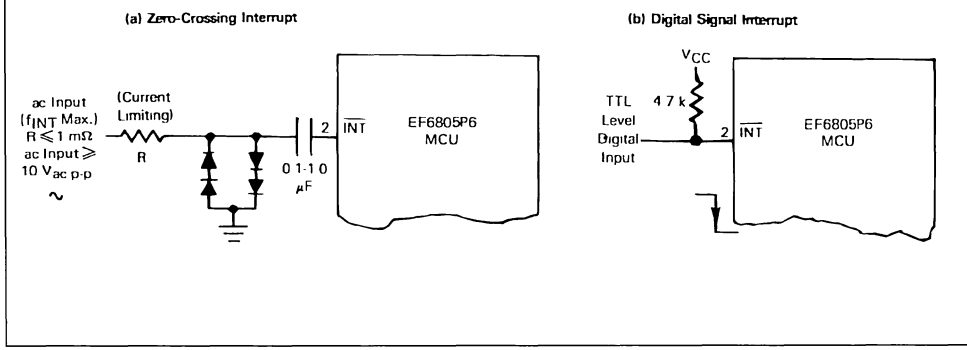


Figure 18 : Typical Interrupt Circuits.



the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows : (assumes 50/50 duty cycle for a given period).

$$t_{cyc} \times 2 + 250ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250ns twice). See figure 18(b).

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. SWIs are usually used as break-points for debugging or as system calls.

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may also be polled with branch instructions to provide an additional input pin. All pins (port A, B, and C) are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic "1" for output or a logic "0" for input. On reset, all the DDRs are initialized to a logic "0" state to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading ; see figure 19. When port B is programmed for outputs, it is capable of sinking 10mA and sourcing 1mA on each pin.

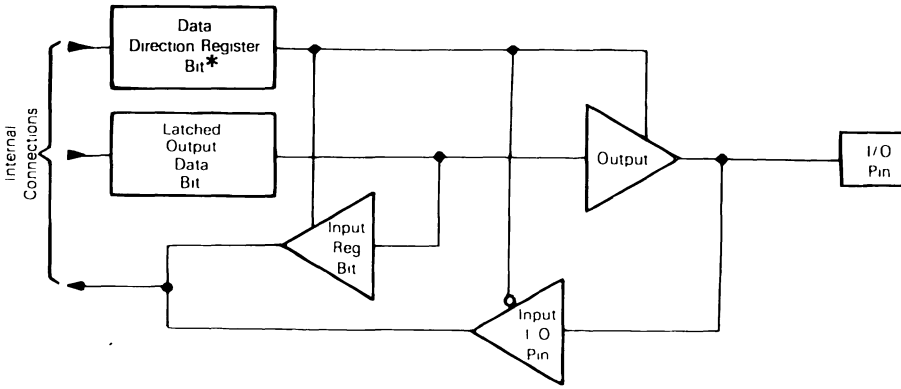
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in figure 6 gives the address of data registers and DDRs. The register configuration is provided in figure 20 and figure 21 provides some examples of port connections.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see figure 19) may always be written. Therefore, any write to a port writes all of its data bits even though the port DDR is set to input. This may be used to initialize the data registers and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input ("0") and corresponds to the latched output data when the DDR is an output ("1").

Figure 19 : Typical Port I/O Circuitry.



Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z**	Pin

* DDR is a write-only register and reads as all "1s"

** Ports A (with CMOS drive disabled), B and C are three state ports Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information

Figure 20 : MCU Register Configuration.

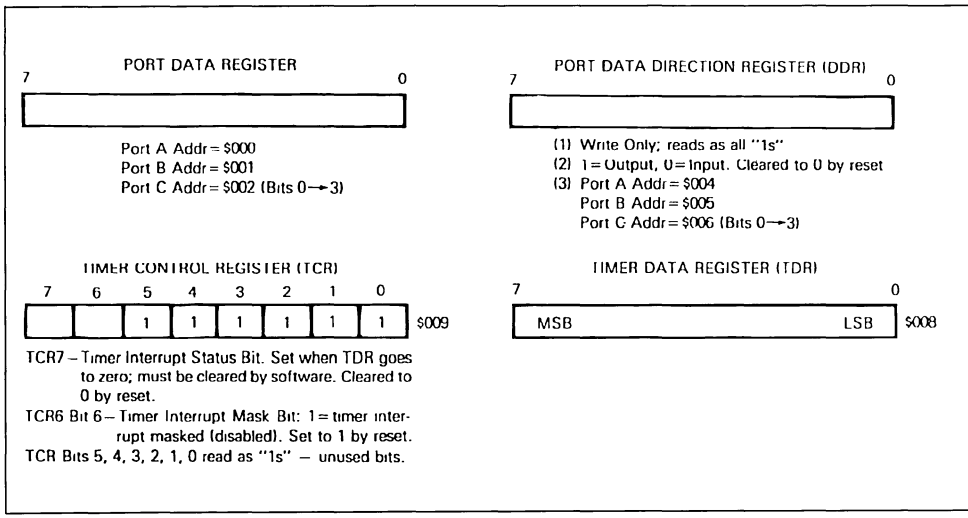
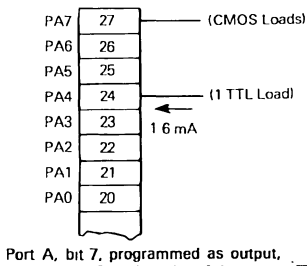
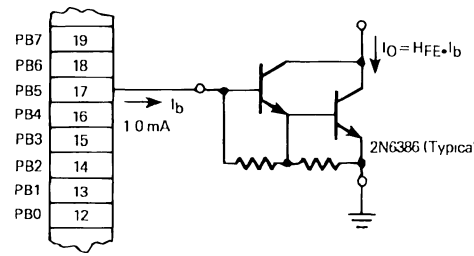


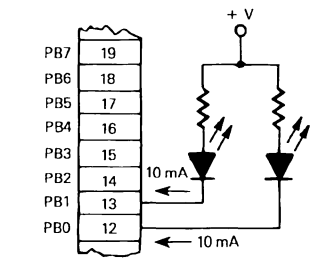
Figure 21 (a) **:** Typical Output Mode Port Connections.



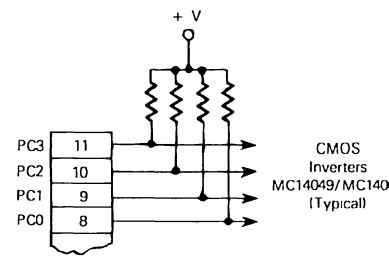
Port A, bit 7, programmed as output, driving CMOS loads and bit 4 driving one TTL load directly using CMOS output option.



Port B, bit 5 programmed as output, driving Darlington-base directly.

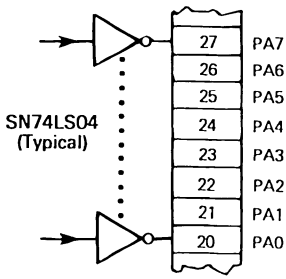


Port B, bit 0 and bit 1 programmed as output, driving LEDs directly.

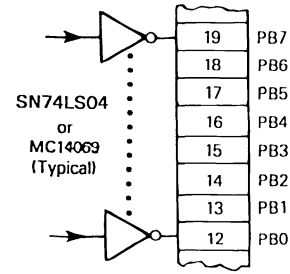


Port C, bits 0-3 programmed as output, driving CMOS loads, using external pullup resistors

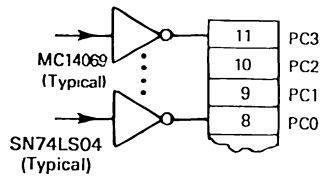
Figure 21(b) : Typical Input Mode Port Connections.



TTL driving port A directly.



CMOS or TTL driving port B directly.



CMOS and TTL driving port C directly.

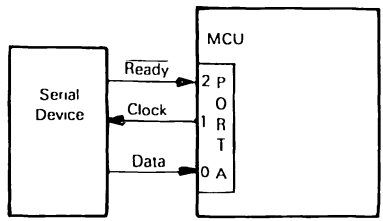
SOFTWARE

BIT MANIPULATION

The EF6805P6 MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction register, see Caution under Input/Output section), with a single instruction (BSET, BCLR). Any bit in page zero including ROM, except the DDRs, can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. The carry bit equals the value of the bit referenced by BRSET or BRCLR. A rotate instruction may then be used to accumulate serial input data in a RAM location or register. The capability to work with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in figure 22 illustrates the usefulness of the bit manipulation and test instructions. Assume that the MCU is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line to clock data one bit at a time, LSB first, out of the device. The MCU waits until the data is ready, clocks the external device, picks up the data in the carry flag (C bit), clears the clock line, and finally accumulates the data bit in a RAM location.

Figure 22 : Bit Manipulation Example.




```
                .
                .
SELF    BRSET   2, PORTA, SELF
                .
                .
        BSET    1, PORTA
        BRCLR   0, PORTA, CONT
CONT    BCLR    1, PORTA
        ROR     RAMLOC
                .
                .
                .
```

ADDRESSING MODES

The EF6805P6 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the 6805 Family User's Manual.

The term "effective address" (EA) is used in describing the address modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE - In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT - In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single 2-byte instruction. This includes the on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED - In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE - The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from - 126 to + 129 from the opcode address. The programmer need not worry about calculating the correct offset when using the assembler

since it calculates the proper offset and checks to see if it is within the span of the branch.

INDEXED, NO OFFSET - In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET - In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET - In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset, except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR - In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction. See Caution under the Input/Output section.

BIT TEST AND BRANCH - The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit and condition (set or clear) which is to be tested is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset is in the third byte and is added to the value of the PC if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from - 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register. See Caution under the Input/Output section.

INHERENT - In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The EF6805P6 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS - Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operands. Refer to table 1.

READ-MODIFY-WRITE INSTRUCTIONS - These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register (see caution under Input/output section). The test for negative or zero (TST) insruction is included in read-modify-write instructions though it does not perform the write. Refer to Table 2.

BRANCH INSTRUCTIONS - The branch instructions cause a branch from the program when a certain condition is met. Refer to table 3.

BIT MANIPULATION INSTRUCTIONS - These instructions are used on any bit in the first 256 bytes of the memory (see caution under Input/output section). One group either sets or clears. The other group performs the bit test branch operations. Refer to table 4.

CONTROL INSTRUCTIONS - The control instructions control the MCU operations during program execution. Refer to table 5.

ALPHABETICAL LISTING - The complete instruction set is given in alphabetical order in table 6.

OPCODE MAP SUMMARY - Table 7 is an opcode map for the instructions used on the MCU.

Table 1 : Register/memory Instructions.

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8 Bit Offset)			Indexed (16 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	—	—	—	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	—	—	—	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 2 : Read-Modify-Write Instructions.

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (no offset)			Indexed (8 bit offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 3 : Branch Instructions.

		Relative Addressing Mode		
Function	**Mnemonic**	**Op Code**	**# Bytes**	**# Cycles**
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(branch IFF higher or same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(branch IFF lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 4 : Bit Manipulation Instructions.

		Addressing Modes					
		Bit Set / Clear			Bit Test and Branch		
Function	**Mnemonic**	**Op Colde**	**# Bytes**	**# Cycles**	**Op Code**	**# Bytes**	**# Cycles**
Branch IFF Bit n is set	BRSET n (n = 0... 7)				2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0... 7)				01 + 2 • n	3	10
Set Bit n	BSET n (n = 0... 7)	10 + 2 • n	2	7			
Clear Bit n	BCLR n (n = 0... 7)	11 + 2 • n	2	7			

Table 5 : Control Instructions.

		Inherent		
Function	**Mnemonic**	**Op Code**	**# Bytes**	**# Cycles**
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-operation	NOP	9D	1	2

Table 6 : Instruction Set.

	Addressing Modes										Condition Code				
Mnem.	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 bits)	Indexed (16 bits)	Bit Set/Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			^	•	^	^	^
ADD		X	X	X		X	X	X			^	•	^	^	^
AND		X	X	X		X	X	X			•	•	^	^	•
ASL	X		X			X	X				•	•	^	^	^
ASR	X		X			X	X				•	•	^	^	^
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	^	^	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	^
BRSET										X	•	•	•	•	^
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLL	X										•	•	•	•	0

Table 6: Instruction Set (continued).

Mnem.	Addressing Modes										Condition Code				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 bits)	Indexed (16 bits)	Bit Set/Clear	Bit Test & Branch	H	I	N	Z	C
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	^	^	^
COM	X		X			X	X				•	•	^	^	1
CPX		X	X	X		X	X	X			•	•	^	^	^
DEC	X		X			X	X				•	•	^	^	•
EOR		X	X	X		X	X	X			•	•	^	^	•
INC	X		X			X	X				•	•	^	^	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	^	^	•
LDX		X	X	X		X	X	X			•	•	^	^	•
LSL	X		X			X	X				•	•	^	^	^
LSR	X		X			X	X				•	•	0	^	^
NEQ	X		X			X	X				•	•	^	^	^
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	^	^	•
ROL	X		X			X	X				•	•	^	^	^
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	^	^	^
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	^	^	•
STX			X	X		X	X	X			•	•	^	^	•
SUB		X	X	X		X	X	X			•	•	^	^	^
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	^	^	•
TXA	X										•	•	•	•	•

Condition Code Symbols :

H Half Carry (from bit 3)
I Interrupt Mask
N Negative (sign bit)
Z Zero
C Carry/Borrow
^ Test and Set if true, cleared otherwise
• Not affected
? Load CC Register from Stack

Table 7 : EF6805 HMOS Family OP Code Mape.

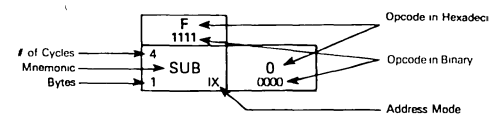
	Bit Manipulation		Branch	Read Modify Write				
	BTB	BSC	REL	DIR	INH	INH	IX1	IX
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL					
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL					
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DTR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL					
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX
9 1001	10 BRCLRA 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL< 2 IX1	6 ROL 1 IX
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL					
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL					
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX

Table 7 : (continued).

Control		Register/Memory						
INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi Low
9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
		2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
		2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
		2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
		2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
	2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
	2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
	2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	5 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
	2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
	2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
	2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
	2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1110
		2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
	2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

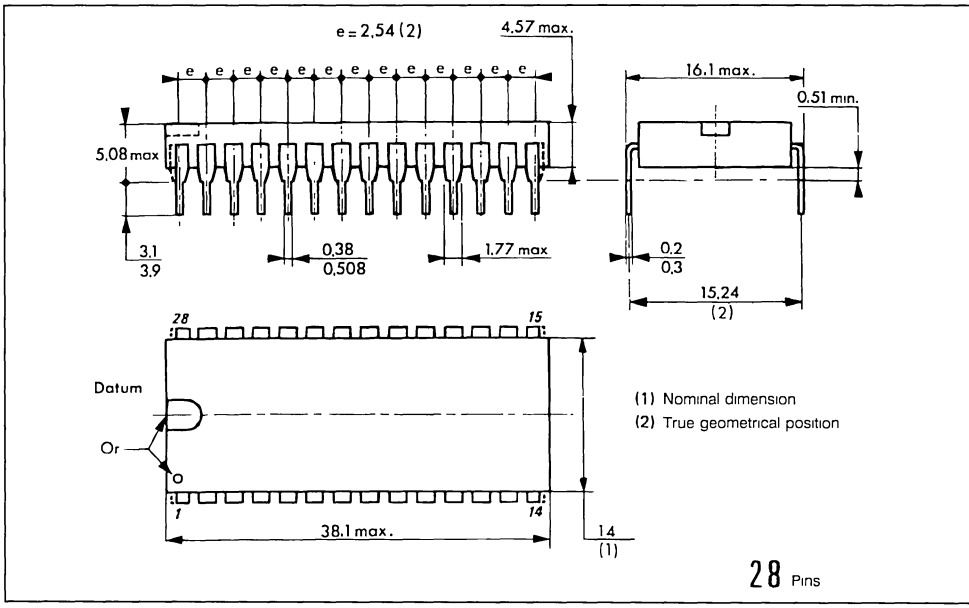
Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/clear
BTB	Bit Test and Branch
IX	Indexed (no offset)
IX1	Indexed, 1 Byte (8-bit) Offset
IX2	Indexed, 2 Byte (16-bit) Offset

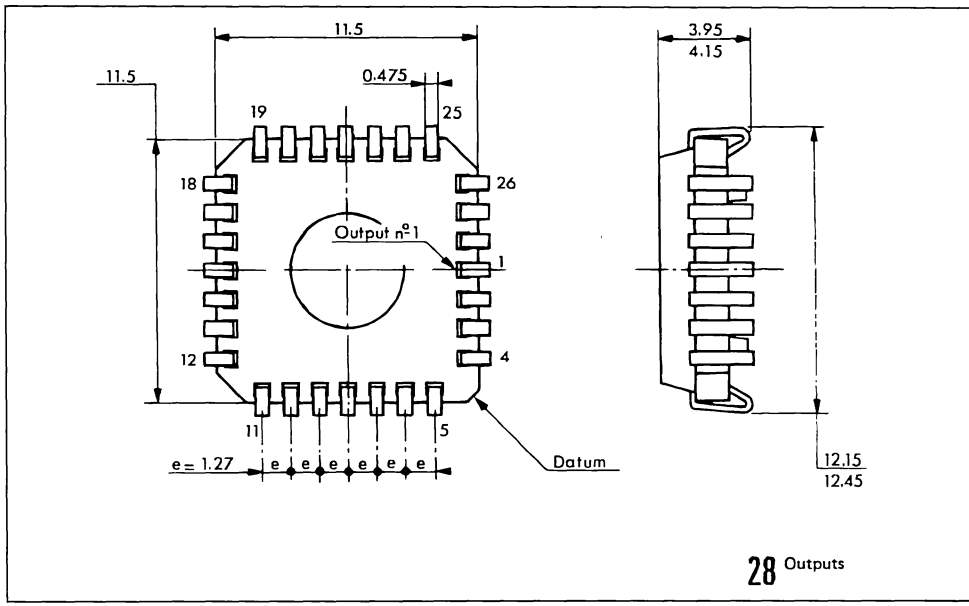


PACKAGE MECHANICAL DATA

DIL–132 PLASTIC PACKAGE



PLASTIC LEADED CHIP CARRIER



EF6805 HMOS FAMILY

Features	EF6805P2	EF6805P6	EF6805R2	EF6805R3	EF6805U2	EF6805U3
Technology	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	40	40	40
On-chip RAM (bytes)	64	64	64	112	64	112
On-chip User ROM (bytes)	1100	1796	2048	3776	2048	3776
External Bus	None	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	24	24	24
Unidirectional I/O Lines	None	None	6 Inputs	6 Inputs	8 Inputs	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, A/D	Timer	Timer
External Interrupt Inputs	1	1	2	2	2	2
STOP and WAIT	No	No	No	No	No	No

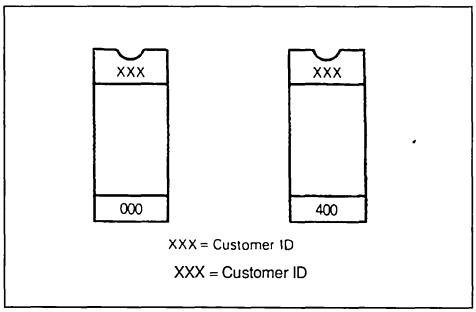
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS–THOMSON Microelectronics on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local SGS-THOMSON representative or distributor.

EPROMs

The ET2716 or ET2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS-THOMSON. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, SGS-THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by SGS THOMSON Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

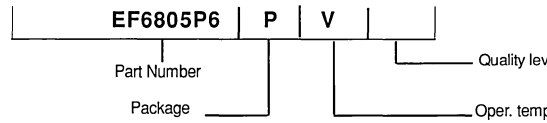
The disk media submitted must be single-sided, single-density, 8-inch, EFDOS/MDOS* compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6805 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

EFDOS is SGS-THOMSON Microelectronics Disk Operating System available on development systems such as DEVICE...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser...

* Requires prior factory approval.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and quality level. Other possibilities on request.

Part Number	Package		Oper. Temp.			Quality Level			
	P	FN	L*	V	A	Std**	D	G	B
EF6805P6	•	•	•	•		•			

Examples : EF6805P6P, EF6805P6FN, EF6805P6PV, EF6805P6FNV

P : Plastic DIL, **FN :** SURPICOP (plastic chip-carrier at JEDEC std 0 50" center leaded type A)
L* : 0°C to + 70°C, **V :** – 40 °C to + 85°C, **A :** – 40°C to + 105°C, * **:** may be omitted
Quality levels in accordance with NFC 96883 – ** : No end-suffix for standard level products.

EXORciser is a registered trademark of MOTOROLA Inc.

EF6805 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....

.....

ROM capacity required : | | | | bytes

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 40°C / + 85°C
- ❑ – 40°C / + 105°C

Package
- ❑ Plastic
- ❑ SURPICOP

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
- ❑ SGS-THOMSON Microelectronics reference
  .....
- ❑ Special customer data reference*
  .....

Number of interrupt vectors :
Quality level :
- ❑ STD
- ❑ D
- ❑ G
- ❑ B

.....

Software developped by :
- ❑ SGS-THOMSON Microelectronics application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA

- ❑ EPROM Reference :
- ❑ EFDOS/MDOS* disk file
  - ❑ 8" floppy
  - ❑ 5" 1/4 floppy
- ❑ Listing
- ❑ Other *

OPTION LIST

-Oscillator input
- ❑ Xtal
- ❑ RC

- Low voltage inhibit :
- ❑ Enabled
- ❑ Disabled

- Port A output drive :
- ❑ CMOS and TTL
- ❑ TTL only

- Port B output drive :
- ❑ TTL
- ❑ Open drain

- Timer clock source :
- ❑ Internal f2 clock
- ❑ Timer input pin

- Timer prescaler

❑ 1	❑ 4	❑ 16	❑ 64
❑ 2	❑ 8	❑ 32	❑ 128

* Requires prior factory approval

Yearly quantity forecast :

- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME : DATE : SIGNATURE :

SGS-THOMSON MICROELECTRONICS

EF6808R2

8–BIT MICROCOMPUTER WITH A/D

HARDWARE FEATURES

- A/D CONVERTER
  - 8-BIT CONVERSION, MONOTONIC
  - 4 MULTIPLEXED ANALOG INPUTS
  - RATIOMETRIC CONVERSION
- 32 TTL/CMOS COMPATIBLE I/O LINES
  - 24 BIDIRECTIONAL (8 Lines are LED Compatible)
  - 8 INPUT-ONLY
- 2048 BYTES OF USER ROM
- 64 BYTES OF RAM
- SELF-CHECK MODE
- ZERO-CROSSING DETECT/INTERRUPT
- INTERNAL 8-BIT TIMER WITH 7-BIT MASK PROGRAMMABLE PRESCALER AND CLOCK SOURCE
- 5V SINGLE SUPPLY

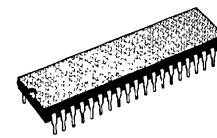
SOFTWARE FEATURES

- 10 POWERFUL ADDRESSING MODES
- BYTE EFFICIENT INSTRUCTION SET WITH TRUE BIT MANIPULATION, BIT TEST, AND BRANCH INSTRUCTIONS
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- POWERFUL INDEXED ADDRESSING FOR TABLES
- FULL SET OF CONDITIONAL BRANCHES
- MEMORY USABLE AS REGISTER/FLAGS
- COMPLETE DEVELOPMENT SYSTEM SUPPORT INICE®

USER SELECTABLE OPTIONS

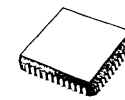
- INTERNAL 8-BIT TIMER WITH SELECTABLE CLOCK SOURCE (external timer input or internal machine clock)
- TIMER PRESCALER OPTION (7 Bits, 2^n)
- 8 BIDIRECTIONAL I/O LINES WITH TTL OR TTL/CMOS INTERFACE OPTION
- 8 BIDIRECTIONAL I/O LINES WITH TTL OR OPEN-DRAIN INTERFACE OPTION
- CRYSTAL OR LOW-COST RESISTOR OSCILLATOR OPTION
- LOW VOLTAGE INHIBIT OPTION
- VECTORED INTERRUPTS : TIMER, SOFTWARE, AND EXTERNAL
- USER CALLABLE SELF-CHECK SUBROUTINES

CB-182



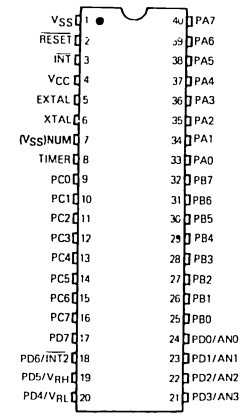
P SUFFIX
Plastic Package

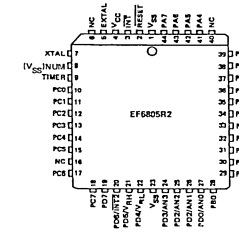
CB-521



FN SUFFIX
PLCC44

PIN CONNECTIONS

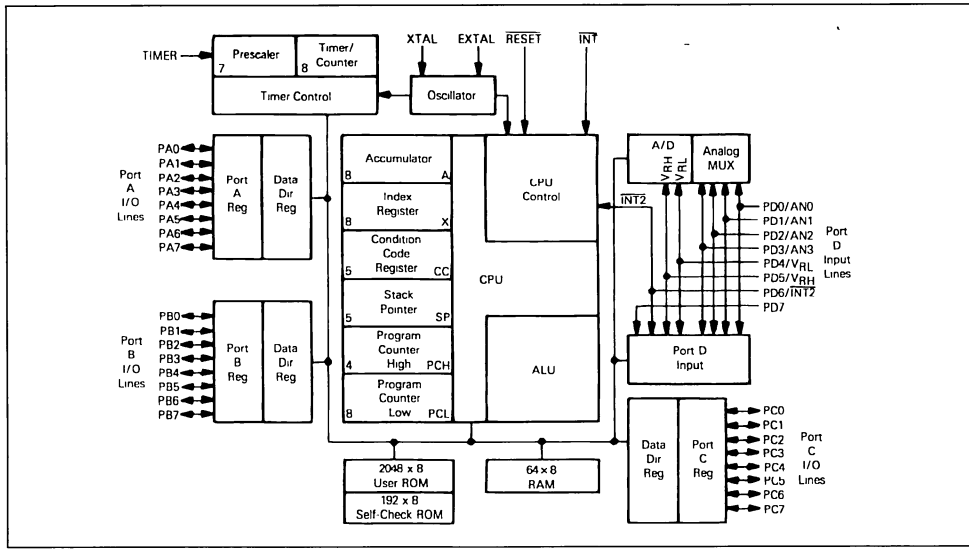




DESCRIPTION

The EF6805R2 Microcomputer Unit (MCU) is a member of the 6805 Family of low-cost single-chip Microcomputers. The 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, 4-channel 8-bit A/D, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the 6800-based instruction set. A comparison of the key features of several members of the 6805 Family of Microcomputers is shown **at the end of this data sheet. The following are some of the hardware and software highlights of the EF6805R2 MCU.**

Figure 1 : EF6805R2 HMOS Microcomputer Block Diagram.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage (except TIMER in self-check mode and open-drain inputs)	– 0.3 to + 7.0	V
V_{in}	Input Voltage (open-drain pins, TIMER pin in self-check mode)	– 0.3 to + 15.0	V
T_A	Operating Temperature Range V Suffix (T_L to T_H) T Suffix	0 to + 70 – 40 to + 85 – 40 to + 105	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_J	Junction Temperature Plastic Package PLCC	150 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electrical fields ; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$ Reliability of operation is enhanced if unused inputs except EXTAL are tied to an appropriate logic voltage level (e g., either V_{SS} or V_{CC}).

THERMAL DATA

θ_{JA}	Thermal Resistance	Plastic PLCC	50 80	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$T_J = T_A + (PD \cdot \theta_{JA})$ (1)

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} << P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$P_D = K \div (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives :

$K = PD \cdot (T_A + 273°C) + \theta JA \cdot P_D^2$ (3)

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage				V
	$\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0		V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5		V_{CC}	
	$\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0	•	V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5	•	V_{CC}	
	All Other (except timer)	2.0		V_{CC}	
V_{IH}	Input High Voltage Timer				V
	Timer Mode	2.0		V_{CC} + 1.0	
	Self-check Mode	9.0	10.0	15.0	
V_{IL}	Input Low Voltage				V
	$\overline{RESET}$	V_{SS}		0.8	
	$\overline{INT}$	V_{SS}	•	1.5	
	All Other (except A/D inputs)	V_{SS}		0.8	
	$\overline{RESET}$ Hystereris Voltages (see figures 10, 11 and 12)				V
V_{IRES+}	"Out of Reset"	2.1		4.0	
V_{IRES-}	"Into Reset"	0.8		2.0	
V_{INT}	$\overline{INT}$ Zero Crossing Input Voltage, Through a Capacitor	2		4	$V_{ac\ p-p}$
P_D	Power Dissipation - (no port loading, V_{CC} = 5.75V)				mW
	T_A = 0°C		520	740	
	T_A = – 40°C		580	800	
C_{in}	Input Capacitance				pF
	EXTAL		25		
	All Other Except Analog Inputs (see note)		10		
V_{LVR}	Low Voltage Recover			4.75	V
V_{LVI}	Low Voltage Inhibit	2.75	3.75	4.70	V
I_{in}	Input Current				µA
	$\overline{TIMER}$ (V_{in} = 0.4V)			20	
	$\overline{INT}$ (V_{in} = 2.4V to V_{CC})		20	50	
	EXTAL (V_{in} = 2.4V to V_{CC} - crystal option)			10	
	_____ (V_{in} = 0.4V - crystal option)			– 1600	
	$\overline{RESET}$ (V_{in} = 0.8V) - External Capacitor Charging Current	– 4.0		– 40	

Note : Port D Analog Inputs, when selected, C_{in} = 25pF for the first 5 out of 30 cycles
* Due to internal biasing this input (when unused) floats to approximately 2.2V.

SWITCHING CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency	0.4		4.2	MHz
t_{cyc}	Cycle Time ($4/f_{osc}$)	0.95		10	µs
t_{WL}, t_{WH}	$\overline{INT}$, $\overline{INT2}$, and TIMER Pulse Width (see interrupt section)	t_{cyc} + 250			ns
t_{RWL}	$\overline{RESET}$ Pulse Width	t_{cyc} + 250			ns
f_{INT}	$\overline{INT}$ Zero-crossing Detection Input Frequency	0.03		1	kHz
	External Clock Input Duty Cycle (EXTAL)	40	50	60	%
	Crystal Oscillator Start-up Time*			100	ms

* See Figure 16 for typical crystal parameters.

A/D CONVERTER CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Parameter	Comments	Min.	Typ.	Max.	Unit
Resolution		8	8	8	Bits
Non-linearity	For V_{RH} = 4.0 to 5.0V and V_{RL} = 0V			± 1/2	LSB
Quantizing Error				± 1/2	LSB
Conversion Range		V_{RL}		V_{RH}	V
V_{RH}	A/D accuracy may decrease proportionately as V_{RH} is reduced below 4.0V. The sum of V_{RH} and V_{RL} must not exceed V_{CC}.			V_{CC}	V
V_{RL}		V_{SS}		0.2	V
Conversion Time	Includes Sampling Time	30	30	30	t_{cyc}
Monotonicity	Inherent (within total error)				
Zero Input Reading	V_{in} = 0	00	00	01	Hexa-decimal
Ratiometric Reading	$V_{in} = V_{RH}$	FE	FF	FF	Hexa-decimal
Sample Time		5	5	5	t_{cyc}
Sample/hold Capacitance, Input				25	pF
Analog Input Voltage	Negative transients on any analog lines (pins 19-24) are not allowed at any time during conversion	V_{RL}		V_{RH}	V

PORT ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, T_A = T_L to T_H unless otherwise noted)

PORT A WITH CMOS DRIVE ENABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage (I_{Load} = 1.6mA)			0.4	V
V_{OH}	Output High Voltage I_{Load} = – 100µA I_{Load} = – 10µA	 2.4 V_{CC} – 1.0			V
V_{IH}	Input High Voltage (I_{Load} = – 300µA max.)	2.0		V_{CC}	V
V_{IL}	Input Low Voltage (I_{Load} = – 500µA max.)	V_{SS}		0.8	V
I_{IH}	High Z State Input Current (V_{in} = 2.0V to V_{CC})			– 300	µA
I_{IL}	High Z State Input Current (V_{in} = 0.4V)			– 500	µA

PORT B

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage I_{Load} = 3.2mA I_{Load} = 10mA (sink)			 0.4 1.0	V
V_{OH}	Output High Voltage I_{Load} = – 200µA	2.4			V
I_{OH}	Darlington Current Drive (source) V_O = 1.5V	– 1.0		– 10	mA
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	High Z State Input Current		< 2	10	µA

PORT C AND PORT A WITH CMOS DRIVE DISABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage I_{Load} = – 100µA	2.4			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	High Z State Input Current		< 2	10	µs

PORT C (open-drain option)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage	2.0		13.0	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{LOD}	Input Leakage Current		< 3	15	µA
V_{OL}	Output Low Voltage I_{Load} = 1.6mA			0.4	V

PORT D (digital inputs only)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{in}	Input Current*		< 1	5	µA

* PD4/VRL – PD5/VRH : The A/D conversion resistor (15kΩ typical) is connected internally between these two lines, impacting their use as digital inputs in some applications.

Figure 2 : TTL Equivalent Test Load (port B).

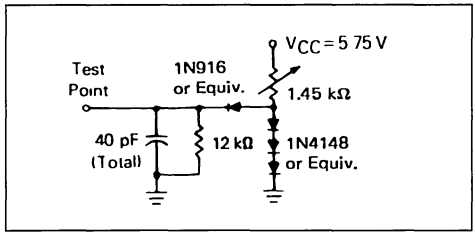


Figure 3 : CMOS Equivalent Test Load (port A).

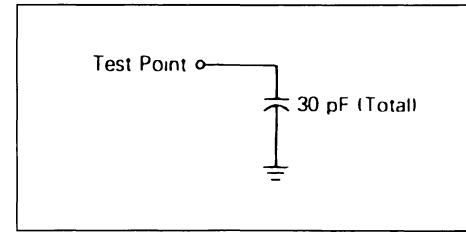


Figure 4 : TTL Equivalent Test Load (port A and C).

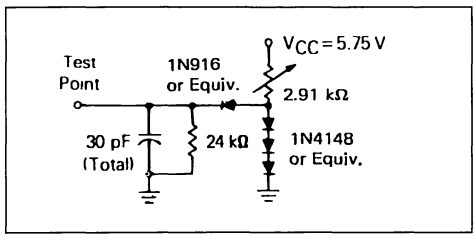
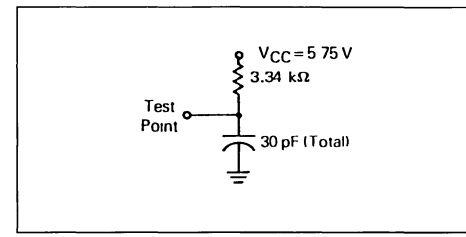


Figure 5 : Open–Drain Equivalent Test Load (port C).



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in figure 1, are described in the following paragraphs.

V_{CC} and V_{SS} - Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

INT - This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts Section for additional information.

NUM (NON USER MODE) - This pin is not for user application and must be connected to V_{SS}.

XTAL AND EXTAL - These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on user selectable manufacturing mak option, can be connected to these pins to provide a system clock with various degrees of stability/cost tradeoffs. Lead length and stray capacitance on these two pins should be minimized. Refer to Internal Clock Generator Options Section for recommendations about these inputs.

TIMER - The pin allows an external input to be used to control the internal timer circuitry and also to initiate the self test program. Refer to Timer Section for additional information about the timer circuitry.

RESET - This pins allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. The MCU can be reset by pulling $\overline{\text{RESET}}$ low. Refer to Resets Section for additional information.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) - These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction registers (DDRs). Port D has up to four analog inputs, plus two voltage reference inputs when the $\overline{\text{A/D}}$ converter is used (PD5/V_{RH}, PD4/V_{RL}), and an $\overline{\text{INT2}}$ input, and from one to eight digital inputs. If any analog input is used, then the voltage reference pins (PD5/V_{RH}, PD4/V_{RL}) must be used in the analog mode. Refer to Input/Output Section, A/D Converter Section, and Interrupts Section for additional information.

MEMORY - The MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The EF6805R2 MCU has implemented 2316 of these bytes. This consists of : 2048 user ROM bytes, 192 self-check ROM bytes, 64 user RAM bytes, 7 port I/O bytes, 2 timer registers. 2 A/D registers, and a miscellaneous register ; see figure 6 for the Address map. The user ROM has been split into three areas. The main user ROM area is from $080 to $OFF and from $7C0 to $F37. The last 8 user ROM locations at the bottom of memory are for the interrupt vectors.

The MCU reserves the first 16 memory locations for I/O features, of which 12 have been implemented.

These locations are used for the ports, the ports DDRs, the timer, the INT2 miscellaneous register, and the A/D. Of the 64 RAM bytes, 31 bytes are shared with the stack area. The stack must be used with care when data shares the stack area.

The shared stack area is used during the processing of an interrupt or subroutine calls to save the contents of the CPU state. The register contents are pushed onto the stack in the order shown in figure 7. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first, then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack since the stack pointer increments when it pulls data from the stack. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack ; the remaining CPU registers are not pushed.

Figure 6 : EF6805R2 MCU Address Map.

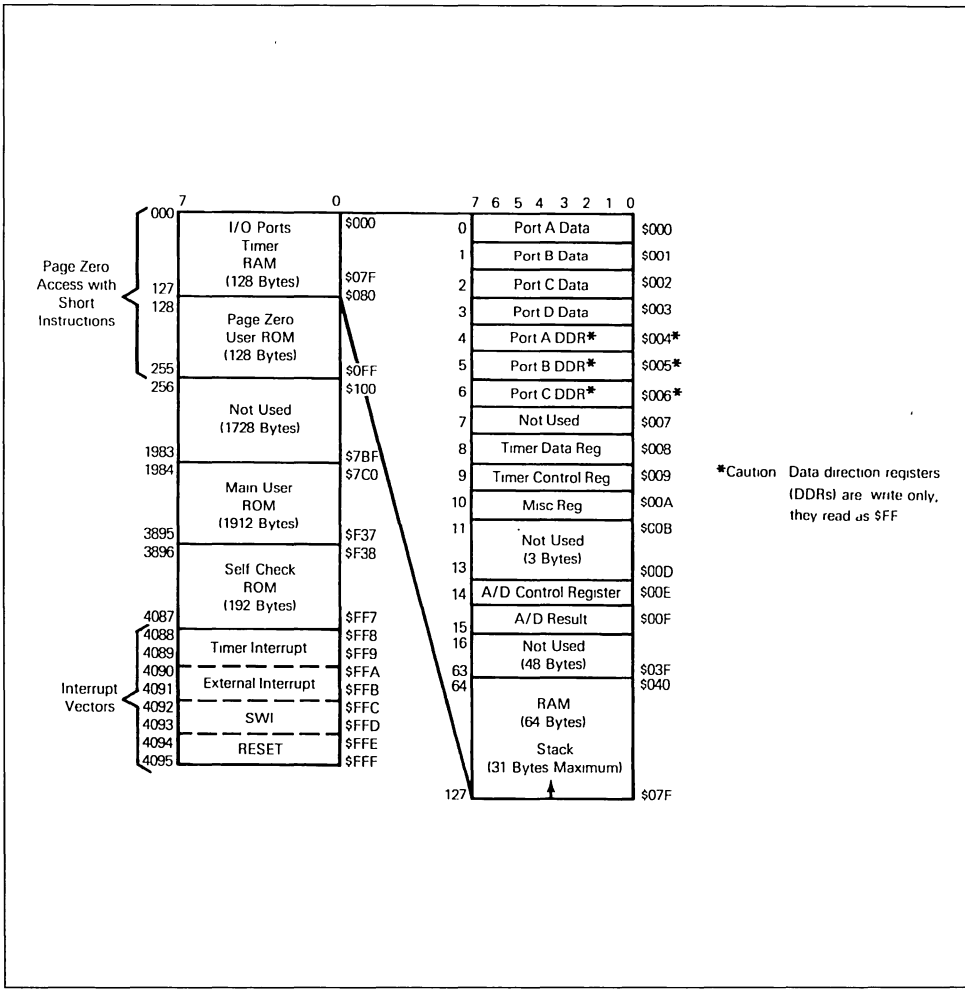
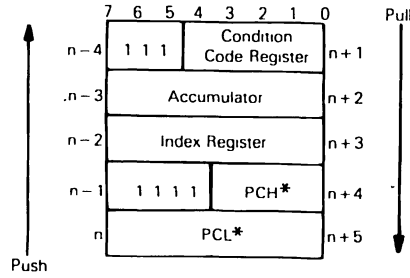


Figure 7 : Interrupt Stacking Order.



CENTRAL PROCESSING UNIT

The CPU of the EF6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

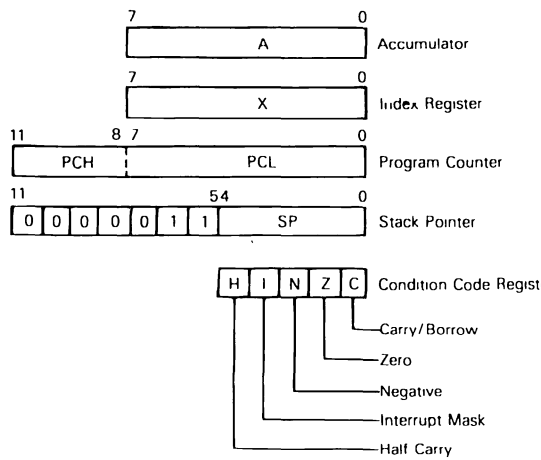
REGISTERS

The 6805 Family CPU has five registers available to the programmer. They are shown in figure 8 and are explained in the following paragraphs.

ACCUMULATOR (A) - The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X) - The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions. The Index Register may also be used as a temporary storage area.

Figure 8 : Programming Model.



PROGRAM COUNTER (PC) - The program counter is a 12-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP) - The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F.

The stack pointer is then decremented as data is pushed onto the stack and incremented as data is then pulled from the stack. The seven most significant bits of the stack pointer are permanently set to 0000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

CONDITION CODE REGISTER (CC) - The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

Half Carry (H) - Set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I) - When this bit is set, the timer and external interrupt ($\overline{INT}$ and $\overline{INT2}$) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical "1").

Zero (Z) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C) - When set, this bit indicates that a carry or borrow out of the Arithmetic Logic Unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The timer circuitry for the MC6805R2 is shown in figure 10. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (or prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine (see RESET, CLOCK, AND INTERRUPT STRUCTURE SECTIONS). *The timer interrupt request bit must be cleared by software. The TIMER and INT2 share the same interrupt vector. The interrupt routine must check the request bits to determine the source of the interrupt.*

The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal phase two signal. Three machine cycles are required for a change in state of the TIMER pin to decrement the timer prescaler. The maximum frequency of a signal that can be recognized by the TIMER pin logic is dependent on the parameter labeled t_{WH}. The pin logic that recognizes the high state on the pin must also recognize the low state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows (assumes 50/50 duty cycle for a given period) :

$$t_{cyc} \times 2 + 250ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 nanoseconds times two).

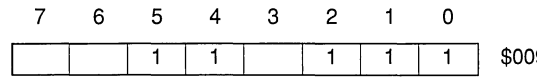
When the phase two signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. The source of the clock input is one of the mask options that is specified before manufacture of the MCU.

NOTE

For ungated phase two clock input to the timer prescaler, the TIMER pin should be tied to V_{CC}.

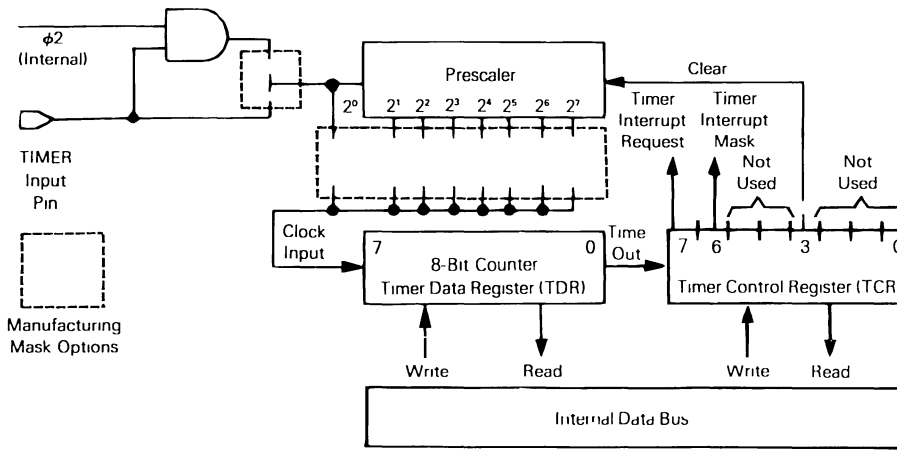
A prescaler option, divide by 2^n, can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture. To avoid truncation errors, the prescaler is cleared when bit 3 of the timer control register is written to a logic one (this bit always reads a logic zero). See figure 9.

Figure 9 : Timer Register (TCR).



TCR7 - Timer Interrupt Request Status Bit. Set when TDR goes to zero ; must be cleared by software. Cleared to 0 by Reset.
TCR6 - Timer Interrupt Mask Bit . 1 = timer interrupt masked (disabled). Set to 1 by Reset.
TCR3 - Clear prescaler always reads as a 0 , clears prescaler when written to a logic "1".
TCR Bits 5, 4, 2, 1, 0 reads "1s" - unused bits.

Figure 10 : Timer Block Diagram.



The timer continues to count past zero, falling through to $FF from $00 and then continuing the countdown. Thus, the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred, and not disturb the counting process.

At power up or reset, the prescaler and counter are initialized with all logic ones ; the timer interrupt request bit (bit 7) is cleared and the timer interrupt mask bit (bit 6) is set.

SELF-CHECK

The self-check capability of the EF6805R2 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in figure 11 and monitor the output of Port C bit 3 for an oscillation of approximately 7Hz. A 10-volt level (through a 10k resistor) on the timer input, pin 8 and pressing then releasing the RESET button, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, A/D, interrupts, and I/O ports.

Several of the self-check subroutines can be called by a user program with a JSR or BSR instruction. They are the RAM, ROM, and 4-channel A/D tests. The timer routine may also be called if the timer input is the internal φ2 clock.

To call those subroutines in customer applications, please contact your local SGS THOMSON sales office in order to obtain the complete description of the self-check program and the entrance/exit conditions.

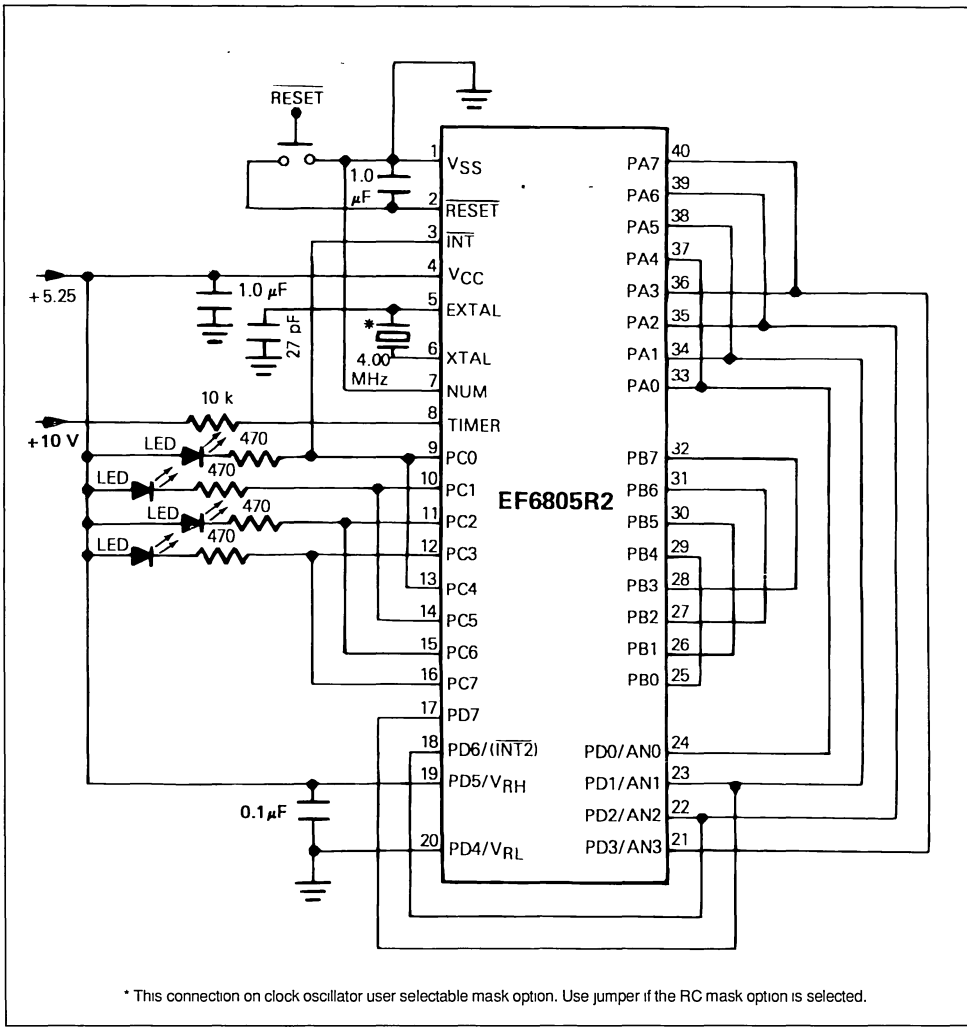
RAM SELF-CHECK SUBROUTINE - The RAM self-check is called at location $F6F and returns with the Z bit clear if any error is detected ; otherwise the Z bit is set. The walking diagnostic pattern method is used on the EF6805R2.

The RAM test must be called with the stack pointer at $07F. When run, the test checks every RAM cell except for $07F and $07E which are assumed to contain the return address.

The A and X registers and all RAM locations except $07F and $07E are modified.

ROM CHECKSUM SUBROUTINE - The ROM self-check is called at location $F8A. If any error is detected, it returns with the Z bit cleared ; otherwise Z = 1, X = 0 on return, and A is zero if the test passes. RAM location $040 to $043 is overwritten. The checksum is the complement of the execution OR of the contents of the user ROM.

Figure 11 : Self–Check Connections.



* This connection on clock oscillator user selectable mask option. Use jumper if the RC mask option is selected.

LED MEANINGS

PC0	PC1	PC2	PC3	Remarks [1 : LED ON ; 0 : LED OFF]
1	0	1	0	Bad I/O
0	0	1	0	Bad Timer
1	1	0	0	Bad RAM
0	1	0	0	Bad ROM
1	0	0	0	Bad A/D
0	0	0	0	Bad Interrupts or Request Flag
Any Flashing				Good Device

Anything else bad Device. Bad Port C, etc

ANALOG-TO-DIGITAL CONVERTER SELF-CHECK - The A/D self-check is called at location $FA4 and returns with the Z bit cleared if any error was found, otherwise Z = 1.

The A and X register contents are lost. The X register must be set to 4 before the call. On return, X = 8 and A/D channel 7 is selected. The A/D test uses the internal voltage references and confirms port connections.

TIMER SELF-CHECK SUBROUTINE - The timer self-check is called at location $FCF and returns with the Z bit cleared if any error was found ; otherwise Z = 1.

In order to work correctly as a user subroutine, the internal 2 clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock is running and the interrupt mask is not set so the caller must protect from interrupts if necessary.

The A and X register contents are lost. The timer self-check routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of 2 since the prescaler is a power of 2. If not, the timer is probably not counting correctly. The routine also detects a timer which is not running.

RESET

The MCU can be reset three ways : by initial power-up, by the external reset input (RESET) and by an optional internal low-voltage detect circuit. The RESET input consists mainly of a Schmitt trigger which senses the RESET line logic level. A typical reset Schmitt trigger hysteresis curve is shown in figure 12. The Schmitt trigger provides an internal reset voltage if it senses a logical zero on the RESET pin.

Power-On Reset (POR) - An internal reset is generated upon powerup that allows the internal clock generator to stabilize. A delay of t_{RHL} milliseconds is required before allowing the RESET input to go high. Refer to the power and reset timing diagram of figure 13. Connecting a capacitor to the RESET input (as illustrated in figure 14) typically provides sufficient delay. During powerup, the Schmitt trigger switches on (removes reset) when RESET rises to V_{IRES+}.

Figure 12 : Typical Reset Schmitt Trigger Hysteresis.

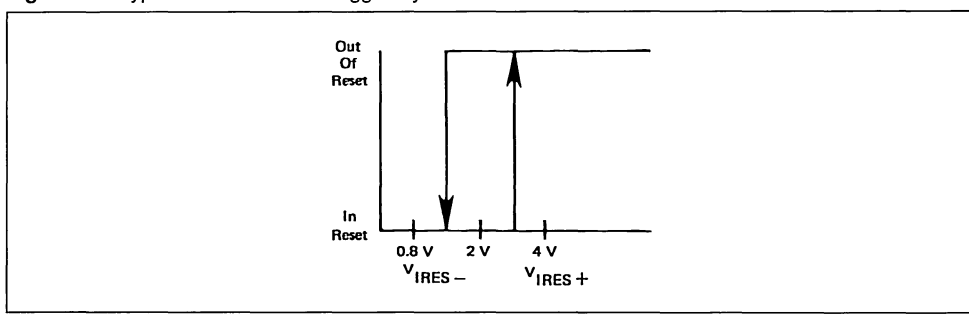


Figure 13 : Power and Reset Timing.

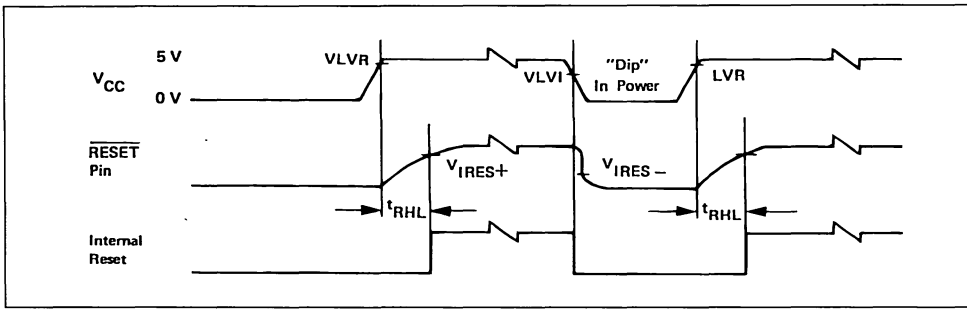
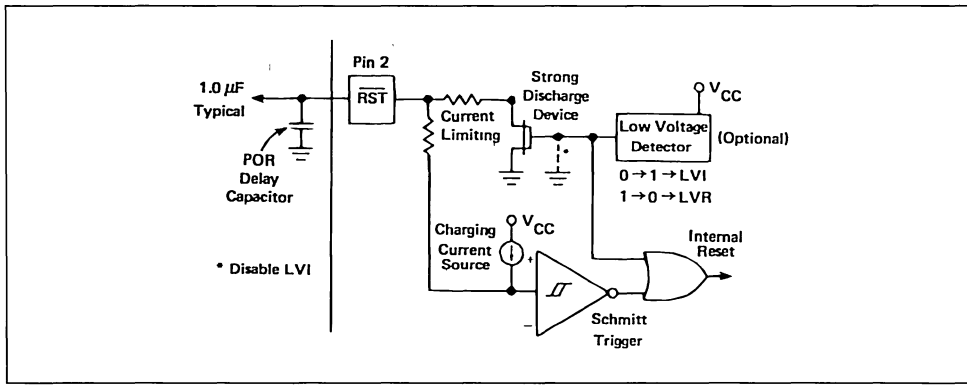


Figure 14 : RESET Configuration.



External Reset Input - The MCU will be reset if a logical zero is applied to the RESET input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

Low-Voltage Inhibit (LVI) - The optional low-voltage detection circuit causes a reset of the MCU if the power supply voltage falls below a certain level (V_{LVI}). The only requirement is that V_{CC} remains at or below the V_{LVI} threshold for one t_{cyc} minimum. In typical applications, the V_{CC} bus filter capacitor will eliminate negative-going voltage glitches of less than one t_{cyc}. The output from the low-voltage detector is connected directly to the internal reset circuitry. It also forces the RESET pin low via a strong discharge device through a resistor. The internal reset will be removed once the power supply voltage rises above a recovery level (V_{LVR}), at which time a normal power-on-reset occurs.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. The oscillator frequency is internally divided by four to produce the internal system clocks. A manufacturing mask option is used to select crystal or resistor operation.

The different connection methods are shown in figure 15. Crystal specifications and suggested PC board layouts are given in figure 16. A resistor selection graph is given in figure 17.

The crystal oscillator start-up time is a function of many variables : crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator start up, neither the crystal characteristics nor the load capacitances should exceed recommendations.

When utilizing the on-board oscillator, the MCU should remain in a reset condition (reset pin voltage below V_{IRES+}) until the oscillator has stabilized at its operating frequency. Several factors are involved in calculating the external reset capacitor required to satisfy this condition : the oscillator start-up voltage,

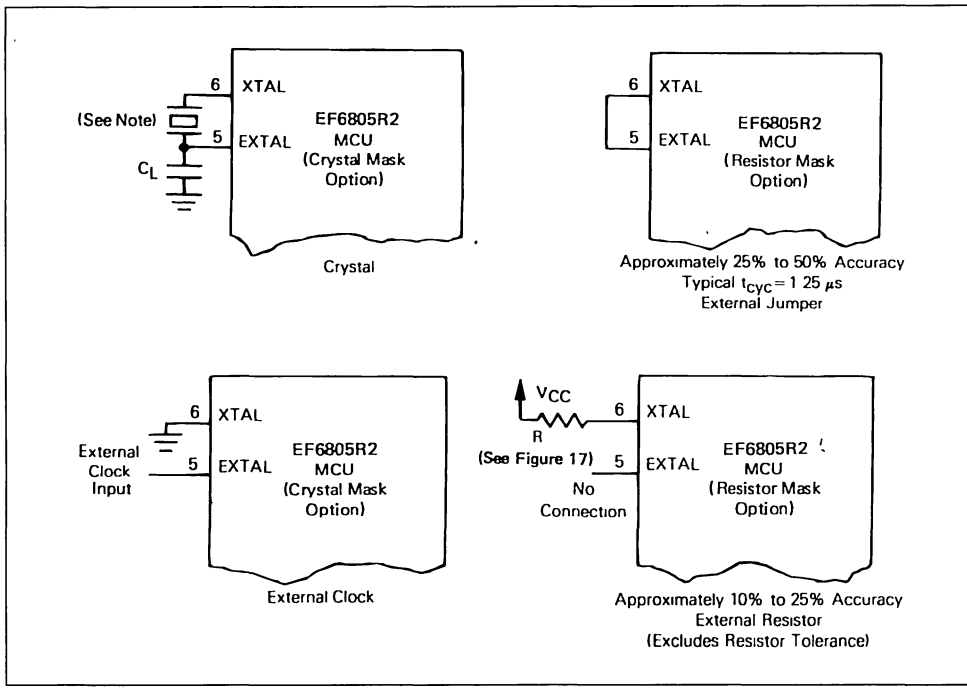
the oscillator stabilization time, the minimum V_{IRES+}, and the reset charging current specification.

Once V_{CC} minimum is reached, the external $\overline{RESET}$ capacitor will begin to charge at a rate dependent on the capacitor value. The charging current is supplied from V_{CC} through a large resistor, so it appears almost like a constant current source until the reset voltage rises above V_{IRES+}. Therefore, the $\overline{RESET}$ pin will charge at approximately :

$$(V_{IRES+}) \cdot C_{ext} = I_{RES} \cdot t_{RHL}$$

Assuming the external capacitor is initially discharged.

Figure 15 : Clock Generator Options.



Note : The recommended C_L value with a 4.0MHz crystal is 27pF, maximum, including system distributed capacitance There is an internal capacitance of approximately 25pF on the XTAL pin. For crystal frequencies other than 4MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio For example, with a 2MHz crystal, use approximately 50pF on EXTAL and approximately 25pF on XTAL The exact value depends on the Motional-Arm parameters of the crystal used.

Figure 16 : Crystal Monotial Arm Parameters and Suggested PC Board Layout.

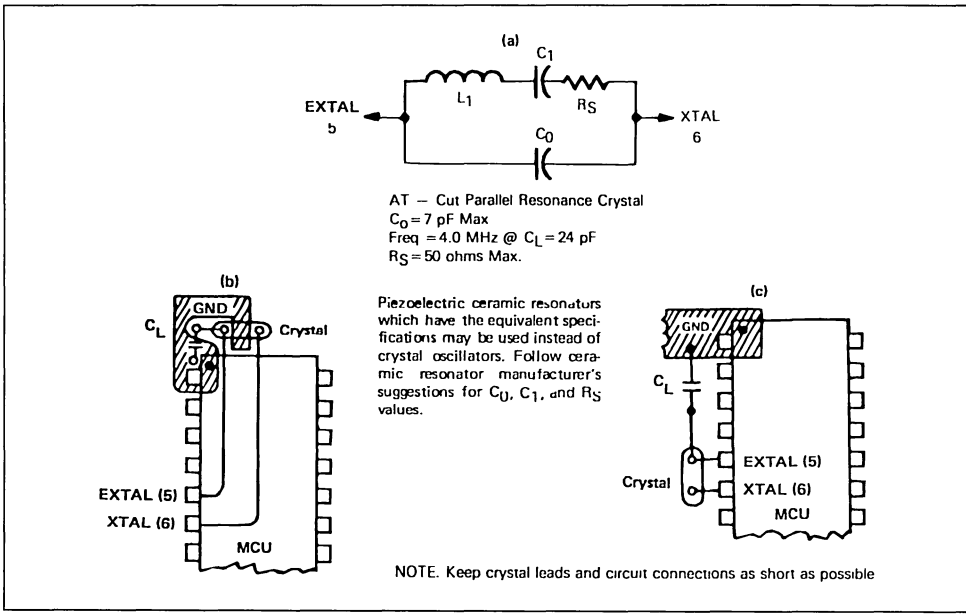
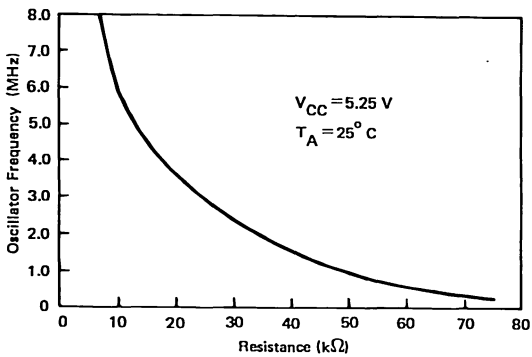


Figure 17 : Typical Frequency Selection for Resistor (oscillator option).



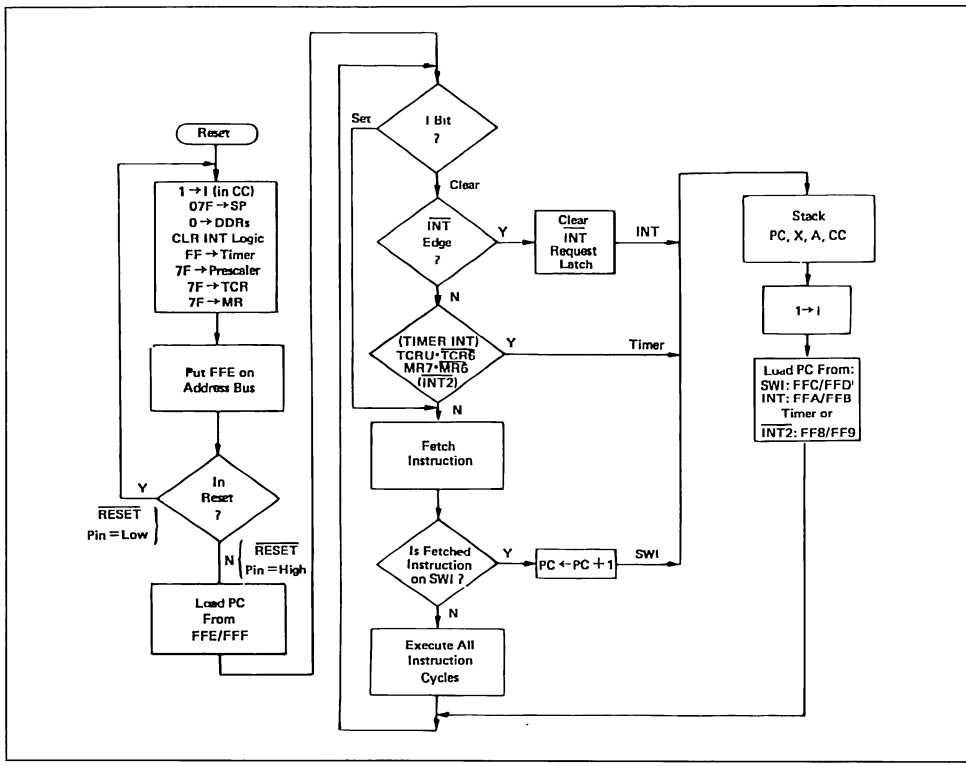
INTERRUPTS

The microcomputers can be interrupted four different ways : through the external interrupt (INT) input pin, the internal timer interrupt request, the external port D bit 6 (INT2) input pin, or the software interrupt instruction (SWI). When any interrupt occurs : the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU register, setting the I bit, and vector fetching require a total of 11 t_{cyc} periods for completion. A flowchart of the interrupt sequence is shown in figure 18. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceedswith interrupt processing ; otherwise the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

Figure 18 : Reset and Interrupt Processing Flowchard.



NOTE

The timer and $\overline{INT2}$ interrupts share the same vector address. The interrupt routine must determine the source by examining the interrupt request bits (TCR b7 and MR b7). Both TCR b7 and MR b7 can only be written to zero by software.

The external interrupt, $\overline{INT}$ and $\overline{INT2}$, are synchronized and then latched on the falling edge of the input signal. The $\overline{INT2}$ interrupt has an interrupt request bit (bit 7) and a mask bit (bit 6) located in the miscellaneous register (MR). The $\overline{INT2}$ interrupt is inhibited when the mask bit is set. The $\overline{INT2}$ is always read as a digital input on port D. The $\overline{INT2}$ and timer interrupt request bits, if set, cause the MCU to process an interrupt when the condition code I bit is clear.

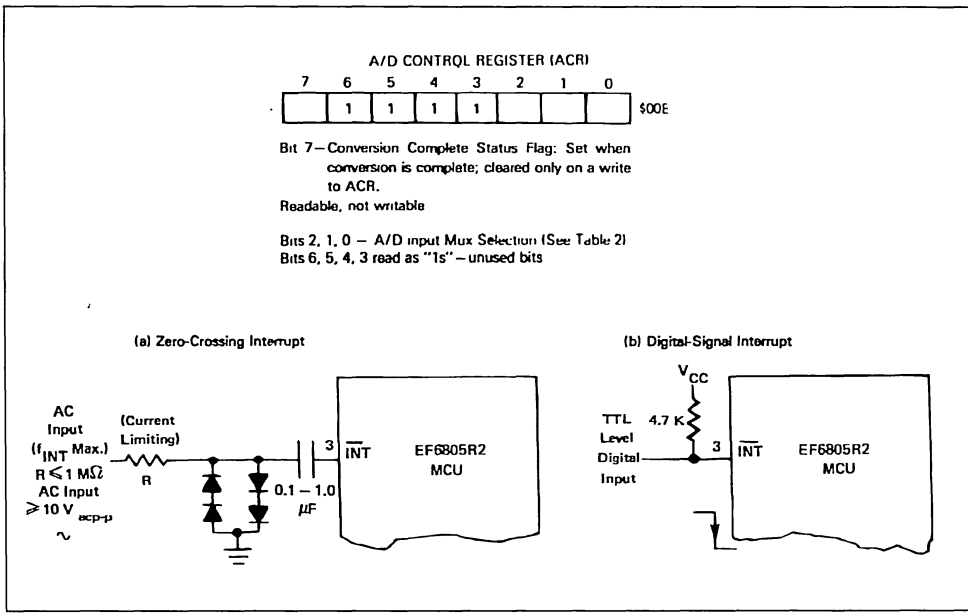
A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock. See figure 19.

NOTE

The $\overline{INT}$ (pin 3) is internally biased at approximately 2.2V due to the internal zero-crossing detection.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. SWIs are usually used as break-points for debugging or as systems calls.

Figure 19 : Typical Interrupt Circuits.



INPUT/OUTPUT CIRCUITRY

There are 32 input/output pins. The $\overline{INT}$ pin may be polled will branch instructions to provide an additional input pin. All pins on ports A, B, and C are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). See below I/O port control registers configuration. The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset all the DDRs are initialized to a logic zero state, placing the ports in the input mode. The port output registers are not initialized on reset and should be initialized by software before changing the DDRs from input to output. A read operation on a port programmed as an output will read the contents of the output latch regardless of the logic at the output pin, due to output loading. Refer to figure 20.

PORT DATA REGISTER

7 0

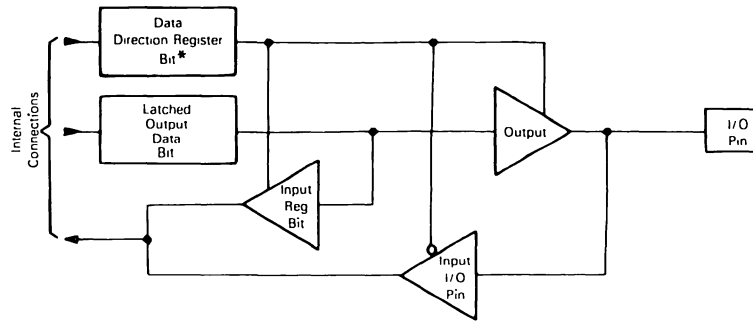
Port A Addr = $000
Port B Addr = $001
Port C Addr = $002
Port D Addr = $003

PORT DATA DIRECTION REGISTER (DDR)

7 0

(1) Write Only ; reads as all "1s"
(2) 1 = Output, 0 = Input Cleared to 0 by Reset
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006

Figure 20 : Typical Port I/O Circuitry.



Data Direction Register Bit	Latched Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

* DDR is a write-only register and reads as all "1s".
* Ports B, and C are three-state ports.
Port A has optional internal pullup devices to provide CMOS data drive capability See Electrical Characteristics tables for complete information

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs (mask option) while port B, C, and D lines are CMOS compatible as inputs. Port D lines are input only ; thus, there is no corresponding DDR. When programmed as outputs, port B is capable of sinking 10 milliamperes and sourcing 1 milliampere on each pin.

Port D provides the reference voltage, $\overline{INT2}$, and multiplexed analog inputs. All of these lines are shared with the port D digital inputs. Port D may always be used as digital inputs and may also be used as analog inputs providing V_{RH} and V_{RL} are connected to the appropriate reference voltages. The V_{RL} and V_{RH} lines (PD4 and PD5) are internally connected to the A/D resistor. Analog inputs may be prescaled to attain the V_{RL} and V_{RH} recommended input voltage range.

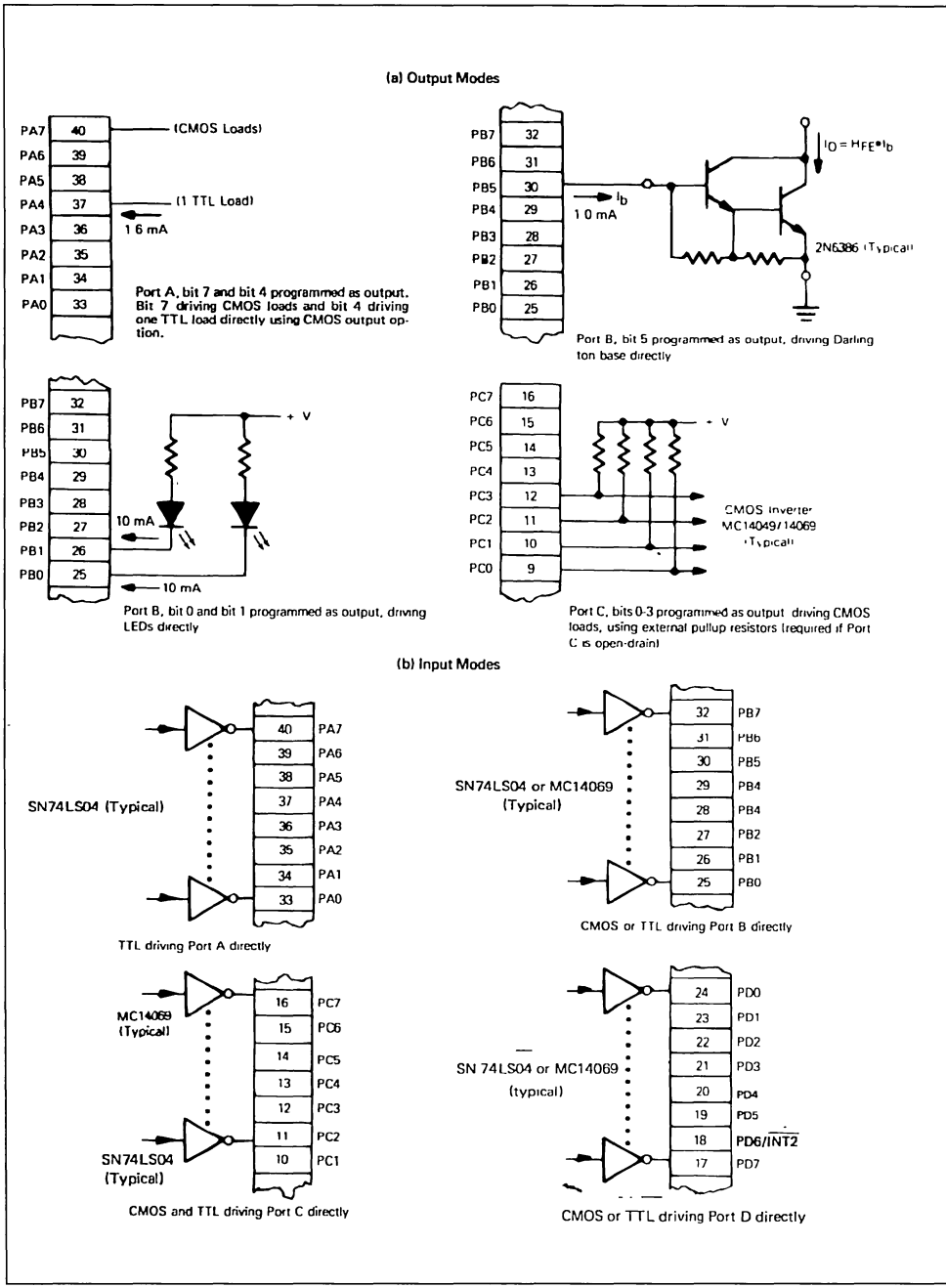
The address map (figure 6) gives the addresses of data registers and data direction registers. Figure 21 provides some examples of port connections.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see figure 20) must always be written. Therefore, any write to a port writes all of its data bits even though the port DDR is set to input. This may be used to initialize the data register and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions, since the data read corresponds to the pin level if the DDR is an input (zero) and corresponds to the latched output data when the DDR is an output (one).

Figure 21 : Typical Port Connections.



ANALOG-TO-DIGITAL CONVERTER

The EF6805R2 has an 8-bit analog-to-digital (A/D) converter implemented on the chip using a successive approximation technique, as shown in figure 22. Up to four external analog inputs, via port D, are connected to the A/D through a multiplexer. Four internal analog channels may be selected for calibration purposes (V_{RH}-V_{RL}, V_{RH}-$V_{RL}/2$, V_{RH}-$V_{RL}/4$, and V_{RL}). The accuracy of these internal channels will not necessarily meet the accuracy specifications of the external channels.

The multiplexer selection is controlled by the A/D control register (ACR) bits 0, 1, and 2 ; see table 1. This register is cleared during any reset condition.

Figure 22 : A/D Block Diagram.

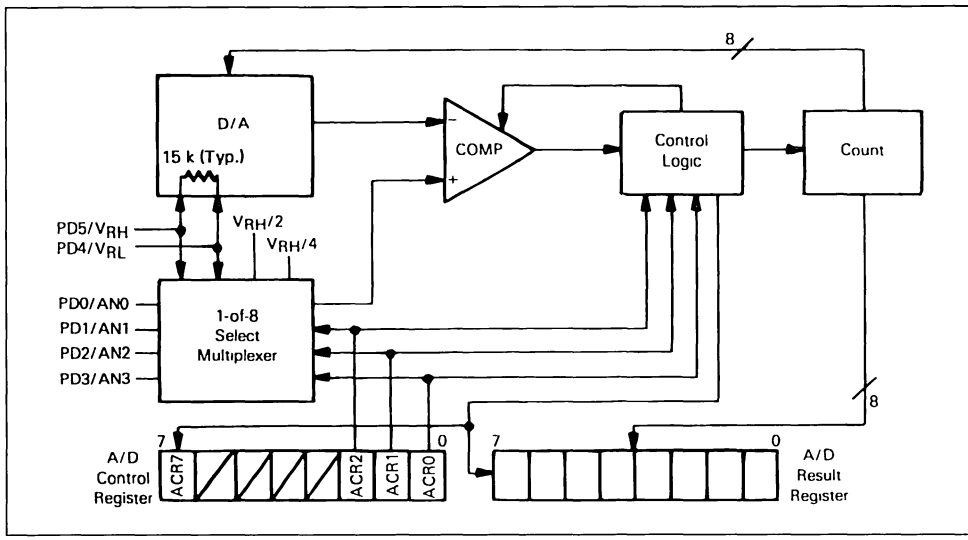


Table 1 : A/D Input Mux Selection.

A/D Control Register			Input Selected	A/D Output (hex)		
ACR2	ACR1	ACR0		Min.	Typ.	Max.
0	0	0	AN0			
0	0	1	AN1			
0	1	0	AN2			
0	1	1	AN3			
1	0	0	V_{RH}*	FE	FF	FF
1	0	1	V_{RL}*	00	00	01
1	1	0	$V_{RH/4}$*	3F	40	41
1	1	1	$V_{RH/2}$*	7F	80	81

* Internal (calibration) levels

MISCELLANEOUS REGISTER (MR)

7	6						0	
		1	1	1	1	1	1	$00A

MR7 Bit 7 – $\overline{INT2}$ Interrupt Request Bit : Set when falling edge detected on $\overline{INT2}$ pin, must be cleared by software. Cleared to 0 by Reset

MR6 Bit 6 – $\overline{INT2}$ Interrupt Mask Bit · 1 = $\overline{INT2}$ Interrupt masked (disabled). Set to 1 by Reset.

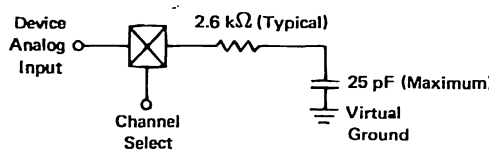
MR Bits 5, 4, 3, 2, 1, 0 – Read as "1s" – unused bits.

A/D RESULT REGISTER (ARR)

7						0	
MSB	1	1	1	1	1	LSB	$00F

Whenever the ACR is written, the conversion in progress is aborted, the conversion complete flag (ACR bit 7) is cleared, and the selected input is sampled for five machine cycles and held internally. During these five cycles, the analog input will appear approximately like a 25 picofarad (maximum) capacitor (plus approximately 10pF for packaging) charging through a 2.6 kiloohm resistor (typical). Refer to figure 23.

Figure 23 : Effective Analog Input Impedance (during sampling only).



The converter operates continuously using 30 machine cycles to complete a conversion of the sampled analog input. When the conversion is complete, the digitized sample of digital value is placed in the A/D result register (ARR), the conversion complete flag is set, the selected input is sampled again, and a new conversion is started.

The A/D is ratiometric. Two reference voltages (V_{RH} and V_{RL}) are supplied to the converter via port D pins. An input voltage equal to V_{RH} converts to $FF (full scale) and an input voltage equal to V_{RL} converts to $00. An input voltage greater than V_{RH} converts to SFF and no overflow indication is provided. Similarly, an input voltage less than V_{RL}, but greater than V_{SS} converts to $00. Maximum and minimum ratings must not be exceeded. For ratiometric conversion, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}. To maintain the full accuracy on the A/D, V_{RH} should be equal to or less than V_{DD}, V_{RL} should be equal to or greater than V_{SS} but less than the maximum specification and (V_{RH}-V_{RL}) should be equal to or greater than 4 volts.

The A/D has a built-in LSB offset intended to reduce the magnitude of the quantizing error to LSB, rather than + 0, - 1 LSB with no offset. This implies that, ignoring errors, the transition point from $00 to $01 occurs at LSB above V_{RL}. Similarly, the transition from $FE to $FF occurs 1 1/2 LSB below V_{RH}, ideally. Refer to figure 24 and 25.

On release of reset, the A/D control register (ACR) is cleared therefore after reset, channel zero will be selected and the conversion complete flag will be clear.

Figure 24 : Ideal Converter Transfer Characteristic.

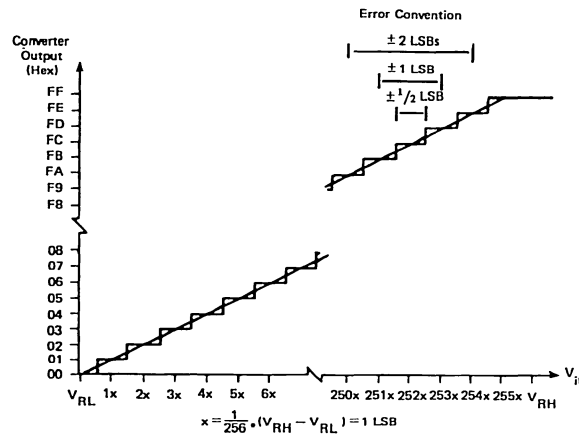
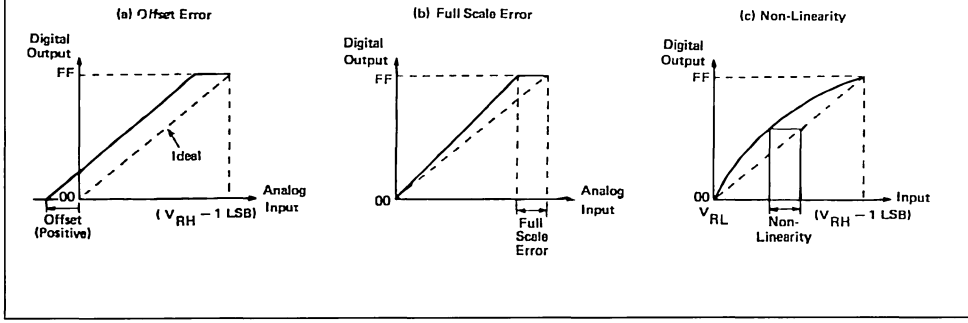


Figure 25 : Types of Conversion Errors.



BIT MANIPULATION

The EF6805R2 as the ability to set or clear any single RAM or I/O bit (except the data direction registers) with a single instruction (BSET, BCLR) (see Caution below). Any bit in page zero can be tested using the BRSET and BRCLR instructions and the program branches as a result of its state. The carry bit equals the value of the bit references by BRSET or BRCLR. The capability to working with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines.

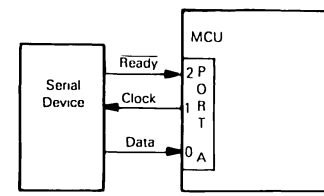
CAUTION

The corresponding data direction registers for ports A, B, and C are write-only registers locations $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set a data direction register bit (all "unaffected" bits would be set). It is recommended that all data direction register bits in a port be written using a single-store instruction.

The coding examples shown in figure 26 illustrate the usefulness of the bit manipulation and test instruction. Assume that the microcomputer is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line to clock data one bit at a time, least significant bit first out, of the device. The microcomputer waits until the data is ready, clocks the external device, picks up the data in the carry flag, clears the clock line, and finally accumulates the data bit in a random-access memory location.

Figure 26 : Bit Manipulation Example.

```
        BRSET    2, PORTA, *    WAIT FOR READY
        BSET     1, PORTA       CLOCK NEXT BIT IN
        BRCLR    0, PORTA, NEXT PICKUP BIT IN C-BIT
NEXT    BCLR     1, PORTA       RETURN CLOCK LINE HIGH
        ASR      RAMLOC         MOVE C-BIT INTO RAM
```



ADDRESSING MODES

The EF6805R2 MCU has ten addressing modes available for use by the programmer. They are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the EF6805 Family Users Manual.

The term "effective address" (EA) is used in describing the addressing modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE - In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT - In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single 2-byte instruction. This address area includes all on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED - In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE - The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if, and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from - 126 to + 129 from the opcode address. The programmer need not worry about calculating the correct offset if he uses the assembler, since it calculates the proper offset and checks to see if it is within the span of the branch.

INDEXED, NO OFFSET - In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET - In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET - In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended, the assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR - In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be must written using a single-store instruction.

BIT TEST AND BRANCH - The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit which is to be tested and condition (set or clear) is included in the opcode, and the address of the byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from - 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code registers.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be must written using a single-store instruction.

INHERENT - In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instructions with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The EF6805R2 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS - Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to table 1.

READ-MODIFY-WRITE INSTRUCTIONS - These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register ; see Caution under Input/Output section. The test for negative or zero (TST) instruction is included in the read-modify-write instruction though it does not perform the write. Refer to table 2.

BRANCH INSTRUCTIONS - The branch instructions cause a branch from the program when a certain condition is met. Refer to table 3.

BIT MANIPULATION INSTRUCTIONS - The instructions are used on any bit in the first 256 bytes of the memory ; One group either sets or clears. The other group performs the bit test and branch operations. Refer to table 4.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be must written using a single-store instruction.

CONTROL INSTRUCTION - The control instructions control the MCU operations during program execution. Refer to table 5.

ALPHABETICAL LISTING - The complete instruction set is given in alphabetical order in table 6.

OPCODE MAP - Table 8 is an opcode map for the instruction used on the MCU.

Table 1 : Register/memory Instructions.

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (no offset)			Indexed (8 bit offset)			Indexed (16 bit offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA				B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX				BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (logical compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP				BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR				BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 2 : Read- Modify-Write Instructions.

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (no offset)			Indexed (8 bit offset)		
Function	Mnem	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 3 : Branch Instructions.

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
Branch IFF Higher or Same	BHS	24	2	4
Branch IFF Carry Set	BCS	25	2	4
Branch IFF Lower	BLO	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF interrupt mask bit is clear.	BMC	2C	2	4
Branch IFF interrupt mask bit is set.	BMS	2D	2	4
Branch IFF interrupt line is low.	BIL	2E	2	4
Branch IFF interrupt line is high.	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 4 : Bit Manipulation Instructions.

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF bit n is set.	BRSET n (n = 0...7)				2 • n	3	10
Branch IFF bit n is clear.	BRCLR n (n = 0...7)				01 + 2 • n	3	10
Set Bit n	BSET n (n = 0...7)	11 + 2 • n	2	7			
Clear Bit n	BCLR n (n = 0...7)	11 + 2 • n	2	7			

Table 5 : Control Instructions.

		Inherent		
Function	**Mnemonic**	**Op Code**	**# Bytes**	**# Cycles**
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-operation	NOP	9D	1	2

Table 6 : Instruction Set.

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			^	●	^	^	^
ADD		X	X	X		X	X	X			^	●	^	^	^
AND		X	X	X		X	X	X			●	●	^	^	●
ASL	X		X			X	X				●	●	^	^	^
ASR	X		X			X	X				●	●	^	^	^
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	^	^	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	^
BRSET										X	●	●	●	●	^
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLL	X										●	●	●	●	0

Condition Code Symbols :

H Half Carry (from bit 3)
I Interrupt Mask
N Negative (sign bit)
Z Zero
C Carry/borrow
^ Test and Set if True, Cleared Otherwise
· Not Affected

Table 6 : Instruction Set (continued).

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/clear	Bit Test & Branch	H	I	N	Z	C
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	^	^	^
COM	X		X			X	X				◔	●	^	^	1
CPX		X	X	X		X	X	X			●	●	^	^	^
DEC	X		X			X	X				●	●	^	^	●
EOR		X	X	X		X	X	X			●	●	^	^	●
INC	X		X			X	X				●	●	^	^	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	^	^	●
LDX		X	X	X		X	X	X			●	●	^	^	●
LSL	X		X			X	X				●	●	^	^	^
LSR	X		X			X	X				●	●	0	^	^
NEQ	X		X			X	X				●	●	^	^	^
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	^	^	●
ROL	X		X			X	X				●	●	^	^	^
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols :

H Half Carry (from bit 3)
I Interrupt Mask
N Negative (sign bit)
Z Zero
C Carry/borrow
^ Test and Set if True, Cleared Otherwise
• Not Affected
? Load CC Register from Stack

EF6805 HMOS FAMILY

Features	EF6805P2	EF6805P6	EF6805R2	EF6805R3	EF6805U2	EF6805U3
Technology	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	40	40	40
On-chip RAM (bytes)	64	64	64	112	64	112
On-chip User ROM (bytes)	1100	1796	2048	3776	2048	3776
External Bus	None	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	24	24	24
Unidirectional I/O Lines	None	None	6 Inputs	6 Inputs	8 Inputs	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, A/D	Timer	Timer
External Interrupt Inputs	1	1	2	2	2	2
STOP and WAIT	No	No	No	No	No	No

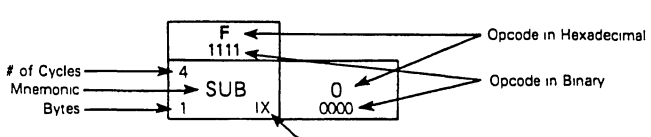
Table 7 : 6805 HMOS Family Instruction Set Opcode Map.

Hi / Low	Bit Manipulation BTB 0 0000	Bit Manipulation BSC 1 0001	Branch REL 2 0010	Read-Modify-Write DIR 3 0011	Read-Modify-Write INH 4 0100	Read-Modify-Write INH 5 0101	Read-Modify-Write IX1 6 0110	Read-Modify-Write IX 7 0111	Control INH 8 1000	Control INH 9 1001	Register/Memory IMM A 1010	Register/Memory DIR B 1011	Register/Memory EXT C 1100	Register/Memory IX2 D 1101	Register/Memory IX1 E 1110	Register/Memory IX F 1111	Hi / Low
0 0000	BRSET0 3 BTB	BSET0 2 BSC	BRA 2 REL	NEG 2 DIR	NEG 1 INH	NEG 1 INH	NEG 2 IX1	NEG 1 IX	RTI 1 INH		SUB 2 IMM	SUB 2 DIR	SUB 3 EXT	SUB 3 IX2	SUB 2 IX1	SUB 1 IX	0 0000
1 0001	BRCLR0 3 BTB	BCLR0 2 BSC	BRN 2 REL						RTS 1 INH		CMP 2 IMM	CMP 2 DIR	CMP 3 EXT	CMP 3 IX2	CMP 2 IX1	CMP 1 IX	1 0001
2 0010	BRSET1 3 BTB	BSET1 2 BSC	BHI 2 REL								SBC 2 IMM	SBC 2 DIR	SBC 3 EXT	SBC 3 IX2	SBC 2 IX1	SBC 1 IX	2 0010
3 0011	BRCLR1 3 BTB	BCLR1 2 BSC	BLS 2 REL	COM 2 DIR	COMA 1 INH	COMX 1 INH	COM 2 IX1	COM 1 IX	SWI 1 INH		CPX 2 IMM	CPX 2 DIR	CPX 3 EXT	CPX 3 IX2	CPX 2 IX1	CPX 1 IX	3 0011
4 0100	BRSET2 3 BTB	BSET2 2 BSC	BCC 2 REL	LSR 2 DIR	LSRA 1 INH	LSRX 1 INH	LSR 2 IX1	LSR 1 IX			AND 2 IMM	AND 2 DIR	AND 3 EXT	AND 3 IX2	AND 2 IX1	AND 1 IX	4 0100
5 0101	BRCLR2 3 BTB	BCLR2 2 BSC	BCS 2 REL								BIT 2 IMM	BIT 2 DIR	BIT 3 EXT	BIT 3 IX2	BIT 2 IX1	BIT 1 IX	5 0101
6 0110	BRSET3 3 BTB	BSET3 2 BSC	BNE 2 REL	ROR 2 DIR	RORA 1 INH	RORX 1 INH	ROR 2 IX1	ROR 1 IX			LDA 2 IMM	LDA 2 DIR	LDA 3 EXT	LDA 3 IX2	LDA 2 IX1	LDA 1 IX	6 0110
7 0111	BRCLR3 3 BTB	BCLR3 2 BSC	BEQ 2 REL	ASR 2 DIR	ASRA 1 INH	ASRX 1 INH	ASR 2 IX1	ASR 1 IX		TAX 1 INH		STA 2 DIR	STA 3 EXT	STA 3 IX2	STA 2 IX1	STA 1 IX	7 0111
8 1000	BRSET4 3 BTB	BSET4 2 BSC	BHCC 2 REL	LSL 2 DIR	LSLA 1 INH	LSLX 1 INH	LSL 2 IX1	LSL 1 IX		CLC 1 INH	EOR 2 IMM	EOR 2 DIR	EOR 3 EXT	EOR 3 IX2	EOR 2 IX1	EOR 1 IX	8 1000
9 1001	BRCLR4 3 BTB	BCLR4 2 BSC	BHCS 2 REL	ROL 2 DIR	ROLA 1 INH	ROLX 1 INH	ROL 2 IX1	ROL 1 IX		SEC 1 INH	ADC 2 IMM	ADC 2 DIR	ADC 3 EXT	ADC 3 IX2	ADC 2 IX1	ADC 1 IX	9 1001
A 1010	BRSET5 3 BTB	BSET5 2 BSC	BPL 2 REL	DEC 2 DIR	DECA 1 INH	DECX 1 INH	DEC 2 IX1	DEC 1 IX		CLI 1 INH	ORA 2 IMM	ORA 2 DIR	ORA 3 EXT	ORA 3 IX2	ORA 2 IX1	ORA 1 IX	A 1010
B 1011	BRCLR5 3 BTB	BCLR5 2 BSC	BMI 2 REL							SEI 1 INH	ADD 2 IMM	ADD 2 DIR	ADD 3 EXT	ADD 3 IX2	ADD 2 IX1	ADD 1 IX	B 1011
C 1100	BRSET6 3 BTB	BSET6 2 BSC	BMC 2 REL	INC 2 DIR	INCA 1 INH	INCX 1 INH	INC 2 IX1	INC 1 IX		RSP 1 INH		JMP 2 DIR	JMP 3 EXT	JMP 3 IX2	JMP 2 IX1	JMP 1 IX	C 1100
D 1101	BRCLR6 3 BTB	BCLR6 2 BSC	BMS 2 REL	TST 2 DIR	TSTA 1 INH	TSTX 1 INH	TST 2 IX1	TST 1 IX		NOP 1 INH	BSR 2 REL	JSR 2 DIR	JSR 3 EXT	JSR 3 IX2	JSR 2 IX1	JSR 1 IX	D 1101
E 1110	BRSET7 3 BTB	BSET7 2 BSC	BIL 2 REL								LDX 2 IMM	LDX 2 DIR	LDX 3 EXT	LDX 3 IX2	LDX 2 IX1	LDX 1 IX	E 1110
F 1111	BRCLR7 3 BTB	BCLR7 2 BSC	BIH 2 REL	CLR 2 DIR	CLRA 1 INH	CLRX 1 INH	CLR 2 IX1	CLR 1 IX		TXA 1 INH		STX 2 DIR	STX 3 EXT	STX 3 IX2	STX 2 IX1	STX 1 IX	F 1111

Abbreviations for Address Modes

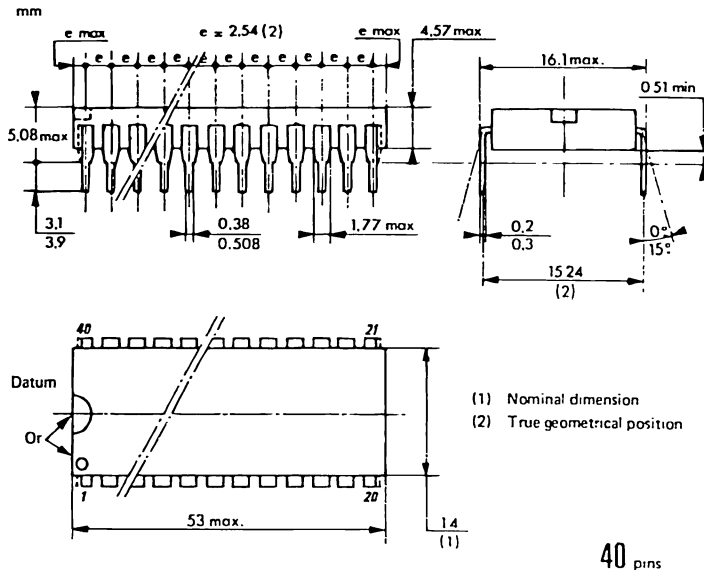
INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

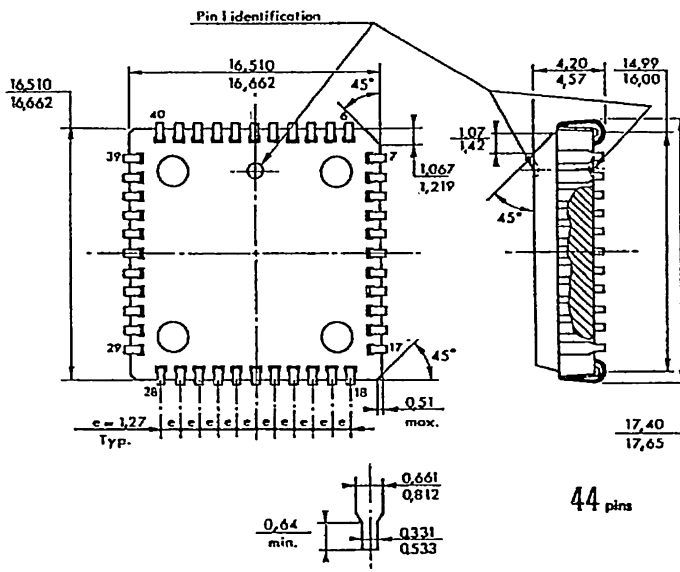


PACKAGE MECHANICAL DATA

P SUFFIX – PLASTIC PACKAGE



FN SUFFIX – PLCC44



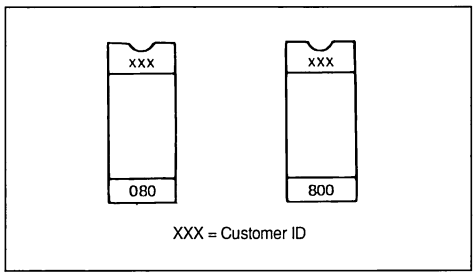
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS–THOMSON on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local SGS THOMSON representative or distributor.

EPROMs

Two 2716 or one 2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filled for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS THOMSON. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, SGS–THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by SGS–THOMSON. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6805 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS–THOMSON factory representatives.

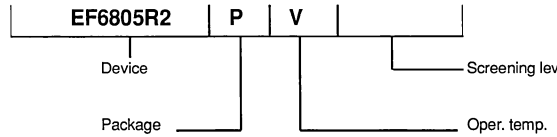
EFDOS is SGS THOMSON' Disk Operating System available on development systems such as DEVICE...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser, ...

* Requires prior factory approval.

Whenever ordering a custom MCU is required, please contact your local SGS THOMSON representative or SGS THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS THOMSON representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating and screening level. Other possibilities on request.

Device	Package					Oper. Temp			ScreeningLevel			
	C	J	P	E	FN	L*	V	T	Std	D	G/B	B/B
EF6805R2			●		●	●	●	●	●	●		
Examples: EF6805R2P, EF6805R2FN, EF6805R2PV, EF6805R2FNV.												

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to +70°C, **V :** –40°C to +85°C, **T :** –40°C to +105°C, *** :** may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.

EXORciser is a registered trademark of MOTOROLA Inc.

EF6805 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....

.....

ROM capacity required : | | | | bytes

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 25°C / + 70°C
- ❑ – 40°C / + 85°C

Package
- ❑ Plastic
- ❑ PLCC

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
- ❑ SGS-THOMSON Microelectronic reference
  .....
- ❑ Special customer data reference*
  .....

Quality level :
- ❑ STD
- ❑ D
- ❑ Other* (customer's quality specification ref.) :
  .....

Software developped by :
- ❑ SGS-THOMSON Microelectronic application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
- ❑ EPROM Reference :
- ❑ EFDOS/MDOS* disk file
  - ❑ 8" floppy
  - ❑ 5" 1/4 floppy
- ❑ Other *

OPTION LIST

-Oscillator input
- ❑ Xtal
- ❑ RC

- Port A output drive :
- ❑ CMOS and TTL
- ❑ TTL only

- Timer clock source :
- ❑ Internal f2 clock
- ❑ Timer input pin

- Low voltage inhibit :
- ❑ Enabled
- ❑ Disabled

- Port C output drive :
- ❑ TTL
- ❑ Open drain

* Requires prior factory approval

Yearly quantity forecast

- start of production date :
- for a shipment period of :

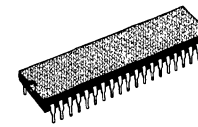
CUSTOMER CONTACT NAME : DATE : SIGNATURE :

SGS-THOMSON MICROELECTRONICS

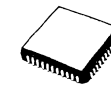
EF6805R3

8 BIT MICROCOMPUTER WITH A/D

- A/D CONVERTER
  8-BIT CONVERSION, MONOTONIC
  4 MULTIPLEXED ANALOG INPUTS
  RATIOMETRIC CONVERSION
- 32 TTL/CMOS COMPATIBLE I/O LINES
  24 BIDIRECTIONAL (8 lines are LED compatible)
  8 INPUT-ONLY
- 3776 BYTES OF USER ROM
- 112 BYTES OF RAM
- SELF-CHECK MODE
- ZERO-CROSSING DETECT/INTERRUPT
- INTERNAL 8-BIT TIMER WITH 7-BIT SOFTWARE PROGRAMMABLE PRESCALER AND CLOCK SOURCE
- 5V SINGLE SUPPLY



P SUFFIX
(Plastic Package)



FN SUFFIX
(PLCC44)

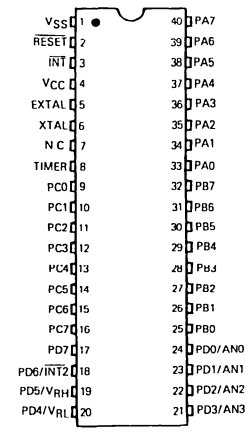
SOFTWARE FEATURES

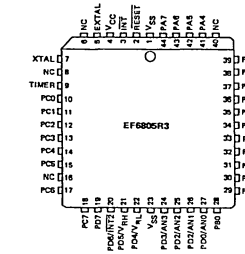
- 10 POWERFUL ADDRESSING MODES
- BYTE EFFICIENT INSTRUCTION SET WITH TRUE BIT MANIPULATION, BIT TEST, AND BRANCH INSTRUCTIONS
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- POWERFUL INDEXED ADDRESSING FOR TABLES
- FULL SET OF CONDITIONAL BRANCHES
- MEMORY USABLE AS REGISTER/FLAGS
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE®

USER SELECTABLE OPTIONS

- 8 BIDIRECTIONAL I/O LINES WITH TTL OR TTL/CMOS INTERFACE OPTION
- 8 BIDIRECTIONAL I/O LINES WITH TTL OR OPEN-DRAIN INTERFACE OPTION
- CRYSTAL OR LOW-COST RESISTOR OSCILLATOR OPTION
- LOW VOLTAGE INHIBIT OPTION
- VECTORED INTERRUPTS : TIMER, SOFTWARE, AND EXTERNAL
- USER CALLABLE SELF-CHECK SUBROUTINES

PIN CONNECTIONS



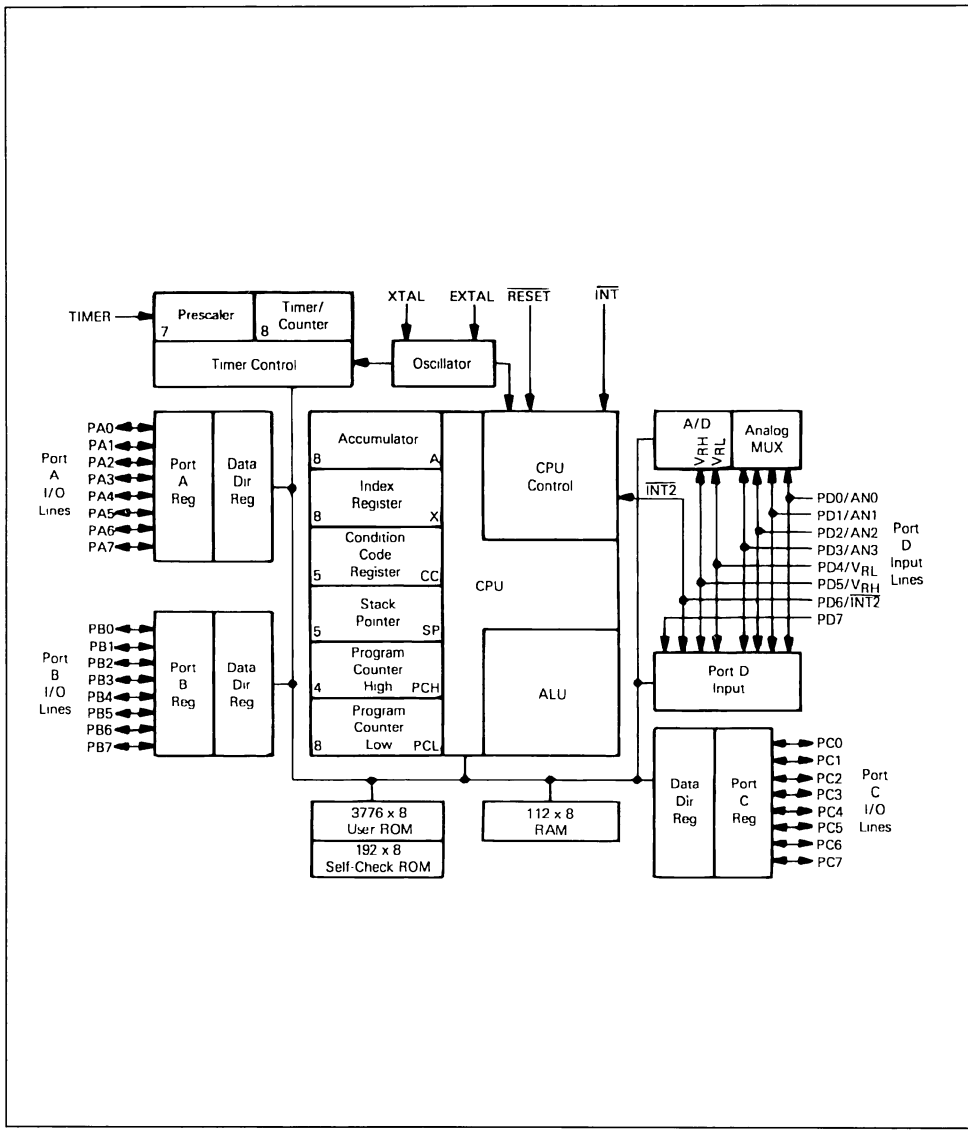


INICE® is SGS-THOMSON development/emulation tool.

DESCRIPTION

The EF6805R3 Microcomputer Unit (MCU) is a member of the 6805 Family of low-cost single-chip Microcomputers. The 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, 4-channel 8-bit A/D, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the 6800-based instruction set. A comparison of the key features of several members of the 6805 Family of Microcomputers is shown at the end of this data sheet. The following are some of the hardware and software highlights of the EF6805R3 MCU.

Figure 1 : EF6805R3 HMOS MICROCOMPUTER BLOCK DIAGRAM.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage (except TIMER in self-check mode and open-drain inputs)	– 0.3 to + 7.0	V
V_{in}	Input Voltage (open-drain pins, TIMER pin in self-check mode)	– 0.3 to + 15.0	V
T_A	Operating Temperature Range (T_L to T_H) V Suffix T Suffix	0 to + 70 – 40 to + 85 – 40 to + 105	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_J	Junction Temperature Plastic Package PLCC	150 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electrical fields ; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$ Reliability of operation is enhanced if unused inputs except EXTAL are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL DATA

θ_{JA}	Thermal Resistance Plastic PLCC	50 80	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} << P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives :

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage				V
	RESET (4.75 ≤ V_{CC} ≤ 5.75)	4.0		V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5		V_{CC}	
	INT (4.75 ≤ V_{CC} ≤ 5.75)	4.0	•	V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5	•	V_{CC}	
	All Other (except TIMER)	2.0		V_{CC}	
V_{IH}	Input High Voltage Timer				V
	Timer Mode	2.0		V_{CC} + 1	
	Self-check Mode	9.0	10.0	15.0	
V_{IL}	Input Low Voltage				V
	RESET	V_{SS}		0.8	
	INT	V_{SS}	•	1.5	
	All Other (except A/D inputs)	V_{SS}		0.8	
	RESET Hystereris Voltages (see figures 10, 11 and 12)				V
V_{IRES+}	"Out of Reset"	2.1		4.0	
V_{IRES-}	"Into Reset"	0.8		2.0	
V_{INT}	INT Zero Crossing Input Voltage, through a Capacitor	2		4	$V_{ac\ p\text{-}p}$
P_D	Power Dissipation - (no port loading, V_{CC} = 5.75V)				mW
	T_A = 0°C		520	740	
	T_A = – 40°C		580	800	
C_{in}	Input Capacitance				pF
	EXTAL		25		
	All Other Except Analog Inputs (see note)		10		
V_{LVR}	Low Voltage Recover			4.75	V
V_{LVI}	Low Voltage Inhibit	2.75	3.75	4.70	V
I_{in}	Input Current				µA
	TIMER (V_{in} = 0.4V)			20	
	INT (V_{in} = 2.4V to V_{CC})		20	50	
	EXTAL (V_{in} = 2.4V to V_{CC}, crystal option)			10	
	(V_{in} = 0.4V, crystal option)			– 1600	
	RESET (V_{in} = 0.8V)	– 4.0		– 40	
	(external capacitor charging current)				

Note : Port D Analog Inputs, when selected, C_{in} = 25pF for the first 5 out of 30 cycles.
* Due to internal biasing this input (when unused) floats to approximately 2.2V.

SWITCHING CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency	0.4		4.2	MHz
t_{cyc}	Cycle Time ($4/f_{osc}$)	0.95		10	µs
t_{WL}, t_{WH}	INT, INT2 and TIMER Pulse Width (see interrupt section)	t_{cyc} + 250			ns
t_{RWL}	RESET Pulse Width	t_{cyc} + 250			ns
f_{INT}	INT Zero-crossing Detection Input Frequency	0.03		1	kHz
	External Clock Input Duty Cycle (EXTAL)	40	50	60	%
	Crystal Oscillator Start-up Time*			100	ms

* See figure 16 for typical crystal parameters.

A/D CONVERTER CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Parameter	Comments	Min.	Typ.	Max.	Unit
Resolution		8	8	8	Bits
Non-linearity	For V_{RH} = 4.0 to 5.0V and V_{RL} = 0V			± 1/2	LSB
Quantizing Error				± 1/2	LSB
Conversion Range		V_{RL}		V_{RH}	V
V_{RH}	A/D accuracy may decrease proportionately as V_{RH} is reduced below 4.0V. The sum of V_{RH} and V_{RL} must not exceed V_{CC}.			V_{CC}	V
V_{RL}		V_{SS}		0.2	V
Conversion Time	Includes Sampling Time	30	30	30	t_{cyc}
Monotonicity	Inherent (within total error)				
Zero Input Reading	V_{in} = 0	00	00	01	Hexa-decimal
Ratiometric Reading	$V_{in} = V_{RH}$	FE	FF	FF	Hexa-decimal
Sample Time		5	5	5	t_{cyc}
Sample/hold Capacitance, Input				25	pF
Analog Input Voltage	Negative transients on any analog lines (pins 19-24) are not allowed at any time during conversion.	V_{RL}		V_{RH}	V

PORT ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

PORT A WITH CMOS DRIVE ENABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage (I_{Load} = 1.6mA)			0.4	V
V_{OH}	Output High Voltage I_{Load} = – 100µA I_{Load} = – 10µA	 2.4 V_{CC}–1.0			V
V_{IH}	Input High Voltage (I_{Load} = – 300µA max.)	2.0		V_{CC}	V
V_{IL}	Input Low Voltage (I_{Load} = – 500µs max.)	V_{SS}		0.8	V
I_{IH}	High-Z State Input Current (V_{IN} = 2.0V to V_{CC})			– 300	µA
I_{IL}	High-Z State Input Current (V_{IN} = 0.4V)			– 500	µA

PORT ELECTRICAL CHARACTERISTICS (continued)

PORT B

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage I_{Load} = 3.2mA I_{Load} = 10mA (sink)			 0.4 1.0	V
V_{OH}	Output High Voltage, I_{Load} = – 200µA	2.4			V
I_{OH}	Darlington Current Drive (source), V_O = 1.5V	– 1.0		– 10	mA
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	High-Z State Input Current		< 2	10	µA

PORT C AND PORT A WITH CMOS DRIVE DISABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage, I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage, I_{Load} = – 100µA	2.4			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	High-Z State Input Current		< 2	10	µA

PORT C (open-drain option)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage	2.0		13.0	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{LOD}	Input Leakage Current		< 3	15	µA
V_{OL}	Output Low Voltage I_{Load} = 1.6mA			0.4	V

PORT D (digital inputs only)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{in}	Input Current*		< 1	5	µA

* PD4/V_{RL} – PD5/V_{RH} : The A/D conversion resistor (15kΩ typical) is connected internally between these two lines, impacting their use as digital inputs in some applications.

Figure 2 : TTL Equivalent Test Load (port B).

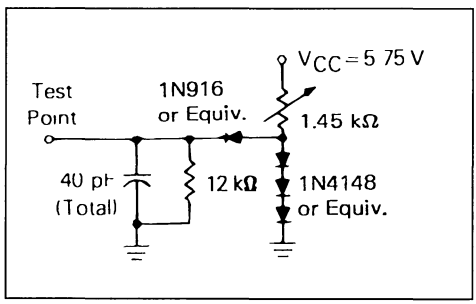


Figure 3 : CMOS Equivalent Test Load (port A).

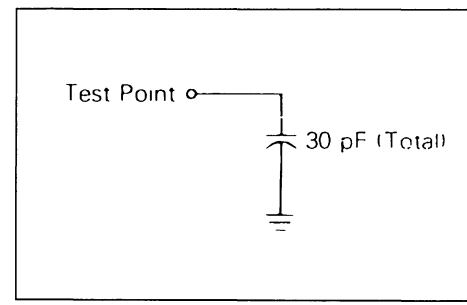


Figure 4 : TTL Equivalent Test Load (ports A and C).

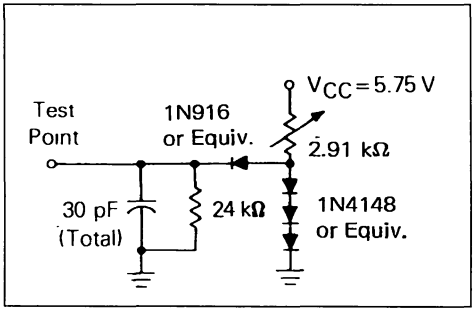
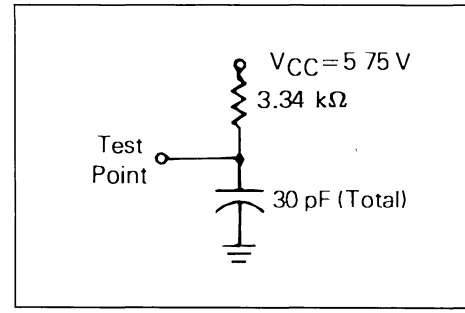


Figure 5 : Open - Drain Equivalent Test Load (port C).



SIGNAL DESCRIPTION

The input and output signals for the CMU, shown in figure 1, are described in the following paragraphs.

V_{CC} AND V_{SS} . Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

$\overline{INT}$. This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts Section for additional information.

XTAL AND EXTAL. These pins provide control input for the on-chip clock oscillator circuit. A crystal, aresistor, or an external signal, depending on user selectable manufacturing mask option, can be connected to these pins to provide a system clock with various degrees of stability/cost tradeoffs. Lead lenght and stray capacitance on these two pins should be minimized. Refer to Internal Clock Generator Options Section for recommendations about these inputs.

Note : Pin 7 in DIL package/Pin 8 in PLCC package is connected to internal protection.

TIMER - The pin allows an external input to be used to control the internal timer circuitry and also to initiate the self test program. Refer to Timer Section for additional information about the timer circuitry.

$\overline{RESET}$. This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. The MCU can be reset by pulling $\overline{RESET}$ low. Refer to Resets Section for additional information.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7). These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction registers (DDRs). Port D has up to four analog inputs, plus two voltage reference inputs when the A/D converter is used (PD5/V_{RH}, PD4/V_{RL}), and an INT2 input, and from one to eight digital inputs. If any analog input is used, then the voltage reference pins (PD5/V_{RH}, PD4/V_{RL}) must be used in the analog mode. The two analog reference inputs are tied together internally with a resistor, therefore, if they are both used as digital inputs problems may occur. Refer to Input/Output Section, A/D Converter Section, and Interrupts Section for additional information.

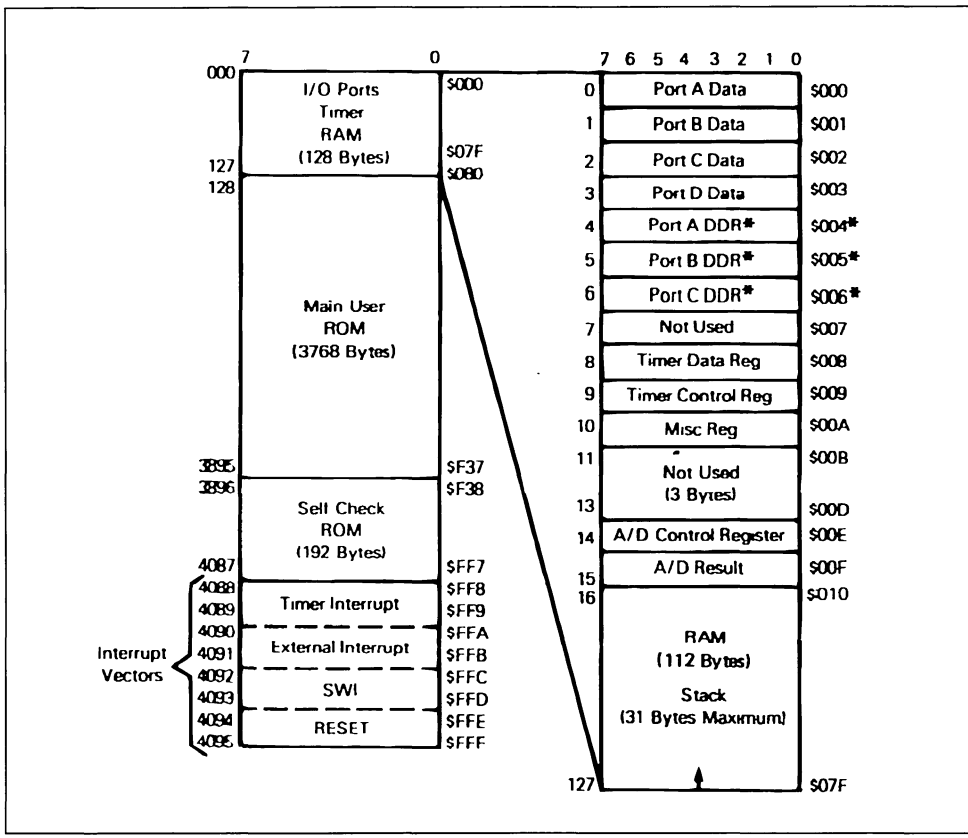
MEMORY. The MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The EF6805R3 MCU has implemented 4092 of these bytes. This consists of : 3776 user ROM bytes, 192 self-check ROM bytes, 112 user RAM bytes, 7 port I/O bytes, 2 timer registers, 2 A/D registers, and a miscellaneous register ; see figure 6 for the Address map. The user ROM has been split into two areas. The main user ROM area is from $080 to $F37. The last 8 user ROM locations at the bottom of memory are for the interrupt vectors.

The MCU reserves the first 16 memory locations for I/O features, of which 12 have been implemented. These locations are used for the ports, the port DDRs, the timer, the INT2 miscellaneous register, and the A/D. Of the 112 RAM bytes, 31 bytes are shared with the stack area. The stack must be used with care when data shares the stack area.

The shared stack area is used during the processing of an interrupt or subroutine calls to save the contents of the CPU state. The register contents are pushed onto the stack in the order shown in figure 7. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first, then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack since the

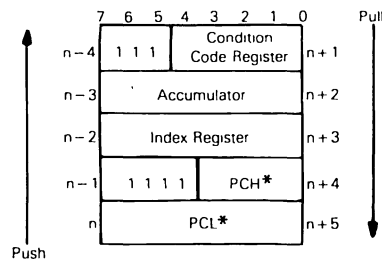
stack pointer increments when it pulls data from the stack. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack ; the remaining CPU registers are not pushed.

Figure 6 : EF6805R3 CMU Address Map.



* Caution : Data direction registers (DDRs) are write-only ; they read as $FF.

Figure 7 : Interrupt Stacking Order.



* For subroutine calls, only PCH and PCL are stacked.

CENTRAL PROCESSING UNIT

The CPU of the EF6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

REGISTERS

The 6805 Family CPU has five registers available to the programmer. They are shown in figure 8 and are explained in the following paragraphs.

ACCUMULATOR (A). The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X). The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions. The Index Register may also be used as a temporary storage area.

PROGRAM COUNTER (PC). The program counter is a 12-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP). The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F.

The stack pointer is then decremented as data is pushed onto the stack and incremented as data is then pulled from the stack. The seven most significant bits of the stack pointer are permanently set to 0000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

CONDITION CODE REGISTER (CC). The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

Half Carry (H). Set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

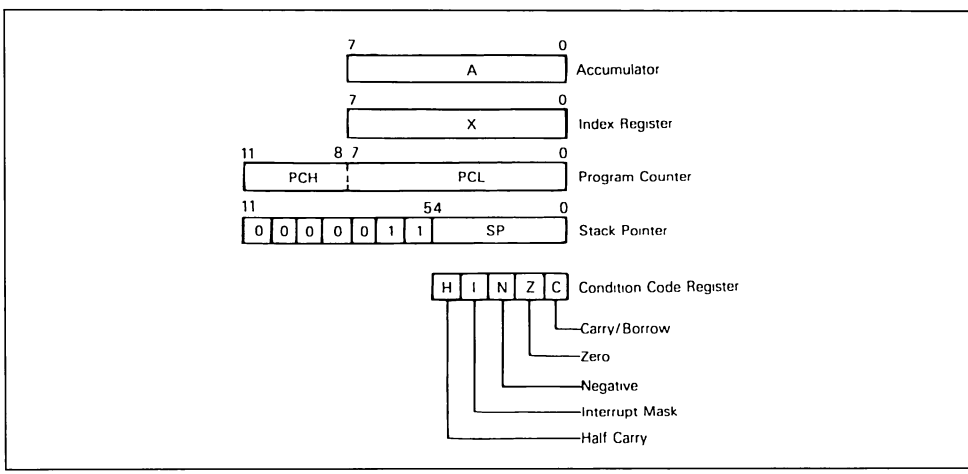
Interrupt (I). When this bit is set, the timer and external interrupts ($\overline{INT}$ and $\overline{INT2}$) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N).When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical "1").

Zero (Z).When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C).When set, this bit indicates that a carry or borrow out of the Arithmetic Logic Unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

Figure 8 : Programming Model.



TIMER

The timer circuitry for the EF6805R3 is shown in figure 10. The timer contains a single 8-bit software programmable counter with a 7-bit software selectable prescaler. The counter may be preset under program control and decrements toward zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the timer control register (TCR), is set. Then if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer interrupt vector from locations $FF8 and $FF9 in order to begin servicing the interrupt.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle and do not change during the read. The timer interrupt request bit remains set until cleared by the software. If a write occurs before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler ; however, its contents are cleared to all zeros by the write operation into TCR when bit 3 of the written data equals one, which allows for truncation-free counting.

The timer input can be configured for three different operating modes, plus a disable mode, depending on the value written to the TCR4 and TCR5 control bits. For further information see figure 9.

Timer Input Mode 1 - If TCR5 and TCR4 are both programmed to a zero, the input to the timer is from an internal clock and the external TIMER input is disabled. The internal clock mode can be used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock.

Timer Input Mode 2 - With TCR5 = 0 and TCR4 = 1, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse widths.

Timer Input Mode 3 - If TCR5 = 1 and TCR4 = 0, then all inputs to the timer are disabled.

Timer Input Mode 4 - If TCR5 = 1 and TCR4 = 1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The external TIMER pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

7	6	5	4	3	2	1	0	
TCR7	TCR6	TCR5	TCR4	TCR3*	TCR2	TCR1	TCR0	$009

* Write only (read as zero)

TCR7 - Timer Interrupt Request Bit :

1 - Set when TDR goes to zero, or under program control

0 - Cleared on external Reset, Power-On-Reset, or under Program Control.

TCR6 - Timer Interrupt Mask Bit :

1 - Timer Interrupt masked (disabled) Set on external Reset, Power-On-Reset, or under Program Control

0 - Cleared under Program Control.

TCR5 - External or Internal Clock Source Bit :

1 - External Clock Source. Set on external Reset, Power-On-Reset, or under Program Control

0 - Cleared under Program Control.

TCR4 - External Enable Bit :

1 - Enable external TIMER pin. Set on external Reset, Power-On-Reset, or under Program Control.

0 - Cleared under Program Control.

TCR5	TCR4	Result
0	0	Internal Clock to Timer
0	1	AND of Internal Clock and TIMER Pin to Timer
1	0	Input to Timer Disabled
1	1	TIMER Pin to Timer

TCR3 - Timer prescaler reset bit : A read of TCR3 always indicates a zero.

1 - Set on external Reset, Power-On-Reset or under Program Control.

0 - Cleared under Program Control.

TCR2, TCR1, and TCR0 - Prescaler address bits :

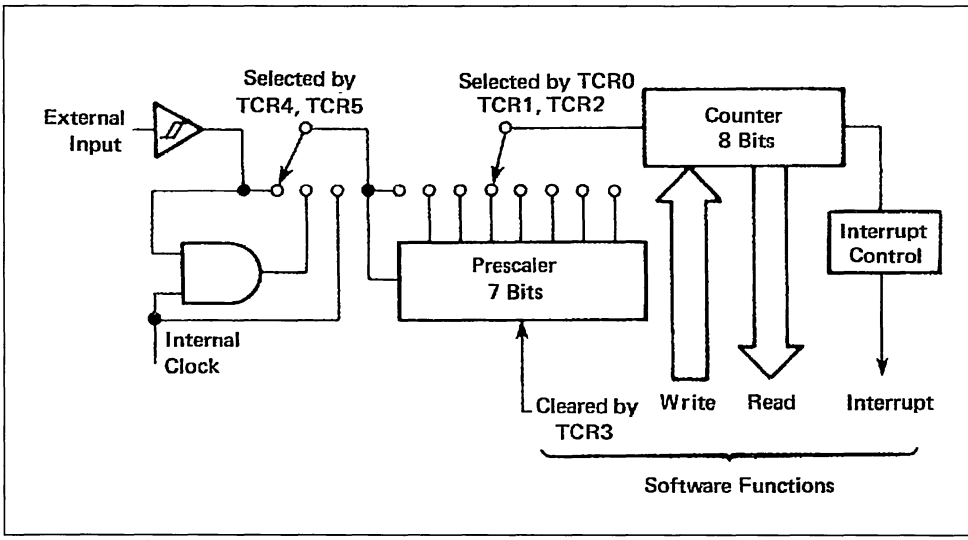
1 - All set on external Reset, Power-On-Reset or under Program Control.

0 - Cleared under Program Control.

Figure 9 : Timer Control Register (TCR).

TCR2	TCR1	TCR0	Result
0	0	0	+ 1
0	0	1	+ 2
0	1	0	+ 4
0	1	1	+ 8
1	0	0	+ 16
1	0	1	+ 32
1	1	0	+ 64
1	1	1	+ 128

Figure 10 : Timer Block Diagram.



Notes : 1. Prescaler and 8-bit counter are clocked on the failing edge of the internal clock (AS) or external input
2. Counter is written to during data strobe (DS) and counts down continuously.

SELF-CHECK

The self-check capability of the EF6805R3 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in figure 11 and monitor the output of Port C bit 3 for an oscillation of approximately 7Hz. A 10-volt level (through a 10k resistor) on the timer input, pin 8 and pressing then releasing the RESET button, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, A/D, interrupts, and I/O ports.

Several of the self-check subroutines can be called by a user program with a JSR or BSR instruction. They are the RAM, ROM, and 4-channel A/D tests. The timer routine may also be called if the timer input is the internal ø2 clock.

To call those subroutines in customer applications, please contact your local SGS THOMSON sales office in order to obtain the complete description of the self-check program and the entrance/exit conditions.

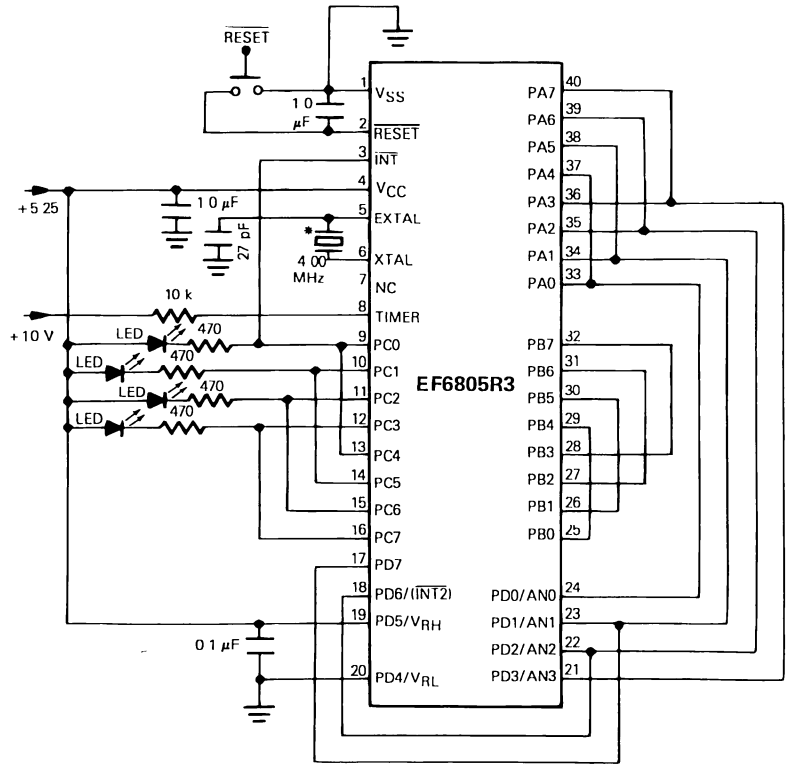
RAM SELF-CHECK SUBROUTINE. The RAM self-check is called at location $F84 and returns with the Z bit clear if any error is detected ; otherwise the Z bit is set. The RAM test causes each byte to count from 0 up to 0 again with a check after each count.

The RAM test must be called with the stack pointer at $07F and A = 0. When run, the test checks every RAM cell except for $07F and $07E which are assumed to contain the return address.

The A and X registers and all RAM locations except $07F and $07E are modified.

ROM CHECKSUM SUBROUTINE - The ROM self-check is called at location $F95. The A register should be cleared before calling the routine. If any error is detected, it returns with the Z bit cleared ; otherwise Z = 1, X = 0 on return, and A is zero if the test passes. RAM location $40 to $043 is overwritten. The checksum is the complement of the execution OR of the contents of the user ROM.

Figure 11 : Self-check Connections.



* This connection depends on clock oscillator user selectable mask option. Use jumper if the RC mask option is selected

LED Meanings

PC0	PC1	PC2	PC3	Remarks [1 : LED ON ; 0 : LED OFF]
1	0	1	0	Bad I/O
0	0	1	0	Bad Timer
1	1	0	0	Bad RAM
0	1	0	0	Bad ROM
1	0	0	0	Bad A/D
0	0	0	0	Bad Interrupts or Request Flag
All Flashing				Good Device

Anything else bad Device, Bad Port C, etc.

ANALOG-TO-DIGITAL CONVERTER SELF-CHECK. The A/D self-check is called at location $FAE. It returns with the Z bit cleared if any error was found ; otherwise the Z bit is set. The A and X register contents are lost. The X register must be set to four before the call. On return, X = 8 and A/D channel 7 is selected. The A/D test uses the internal voltage references and confirms port connections.

TIMER SELF-CHECK SUBROUTINE. The timer self-check is called at location $F6D and returns with the Z bit cleared if any error was found ; otherwise Z = 1.

In order to work correctly as a user subroutine, the internal ø2 clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock is running and the interrupt mask is not set so the caller must protect from interrupts if necessary.

The A and X register contents are lost. This routine sets the prescaler for divide-by-128 and the timer data register is cleared. The X register is configured to count down the same as the timer data register. The two registers are then compared every 128 cycles until they both count down to zero. Any mismatch during the count down is considered as an error. The A and X registers are cleared on exit from the routine.

RESET

The MCU can be reset three ways : by initial power-up, by the external reset input (RESET) and by an optional internal low-voltage detect circuit. The RESET input consists mainly of a Schmitt trigger which senses the RESET line logic level. A typical reset Schmitt trigger hysteresis curve is shown in figure 12. The Schmitt trigger provides an internal reset voltage if it senses a logical zero on the RESET pin.

Power-On Reset (POR).An internal reset is generated upon powerup that allows the internal clock generator to stabilize. A delay of t_{RHL} milliseconds is required before allowing the RESET input to go high. Refer to the power and reset timing diagram of figure 13. Connecting a capacitor to the RESET input (as illustrated in figure 14) typically provides sufficient delay. During powerup, the Schmitt trigger switches on (removes reset) when RESET rises to V_{IRES+}.

Figure 12 : Typical Reset Schmitt Trigger Hysteresis.

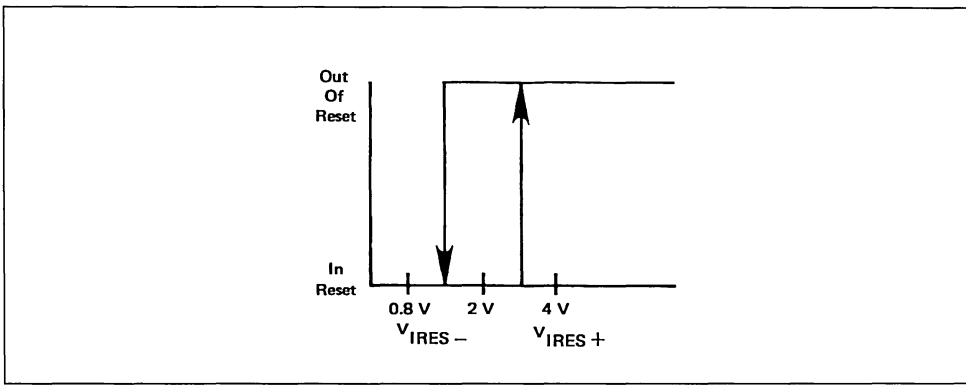


Figure 13 : Power and Reset Timing.

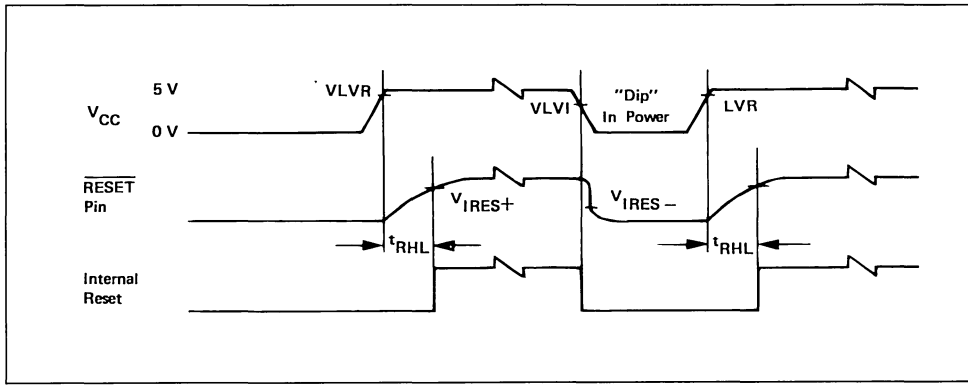
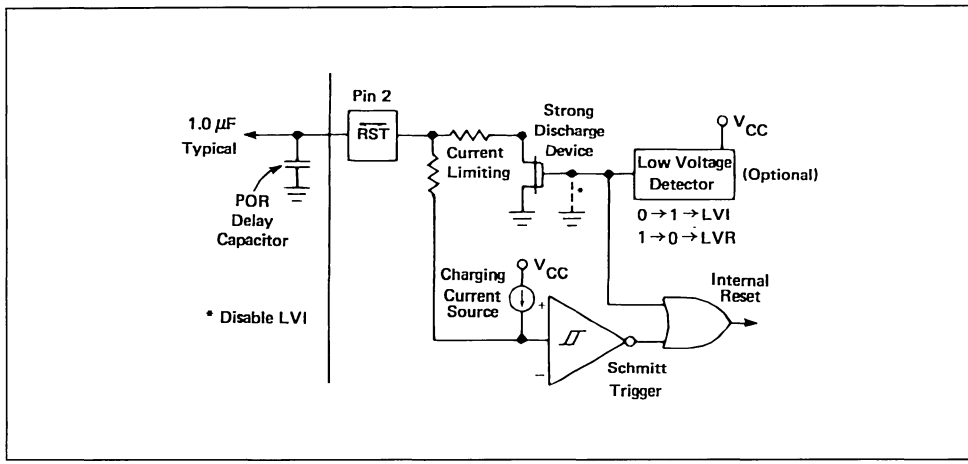


Figure 14 : $\overline{\text{RESET}}$ Configuration.



External Reset Input. The MCU will be reset if a logical zero is applied to the $\overline{\text{RESET}}$ input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

Low-Voltage Inhibit (LVI)**.** The optional low-voltage detection circuit causes a reset of the MCU if the power supply voltage falls below a certain level (V_{LVI}). The only requirement is that V_{CC} remains at or below the V_{LVI} threshold for one t_{cyc} minimum. In typical applications, the V_{CC} bus filter capacitor will eliminate negative-going voltage glitches of less than one t_{cyc}. The output from the low-voltage detector is connected directly to the internal reset circuitry. It also forces the $\overline{\text{RESET}}$ pin low via a strong discharge device through a resistor. The internal reset will be removed once the power supply voltage rises above a recovery level (V_{LVR}), at which time a normal power-on-reset occurs.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. The oscillator frequency is internally divided by four to produce the internal system clocks. A manufacturing mask option is used to select crystal or resistor operation.

The different connection methods are shown in figure 15. Crystal specifications and suggested PC board layouts are given in figure 16. A resistor selection graph is given in figure 17.

The crystal oscillator start-up time is a function of many variables : crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator start up, neither the crystal characteristics nor the load capacitances should exceed recommendations.

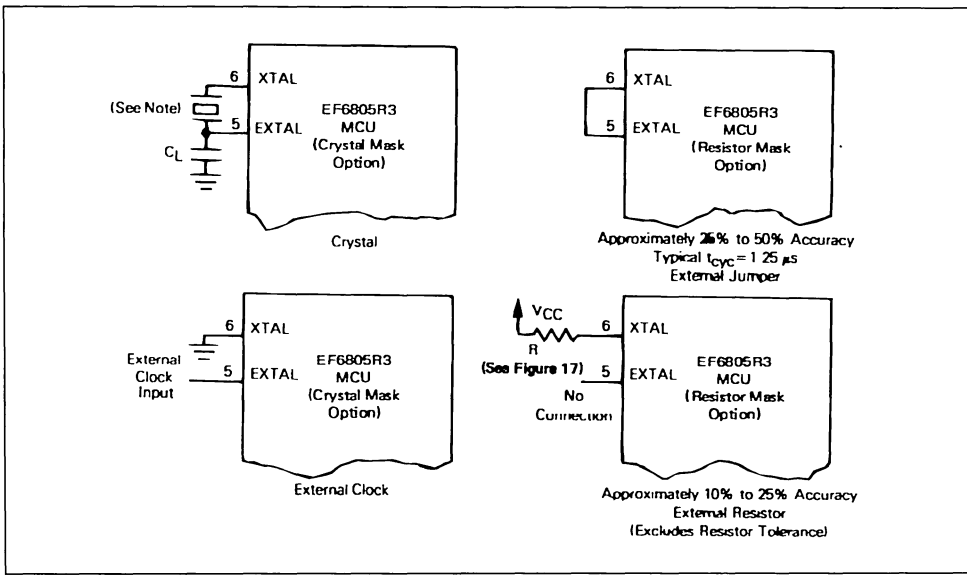
When utilizing the on-board oscillator, the MCU should remain in a reset condition (reset pin voltage below V_{IRES+}) until the oscillator has stabilized at its operating frequency. Several factors are involved in calculating the external reset capacitor required to satisfy this condition : the oscillator start-up voltage, the oscillator stabilization time, the minimum V_{IRES+}, and the reset charging current specification.

Once V_{CC} minimum is reached, the external $\overline{\text{RESET}}$ capacitor will begin to charge at a rate dependent on the capacitor value. The charging current is supplied from V_{CC} through a large resistor, so it appears almost like a constant current source until the reset voltage rises above V_{IRES+}. Therefore, the $\overline{\text{RESET}}$ pin will charge at approximately :

$$(V_{IRES+}) \cdot C_{ext} = IRES \cdot t_{RHL}$$

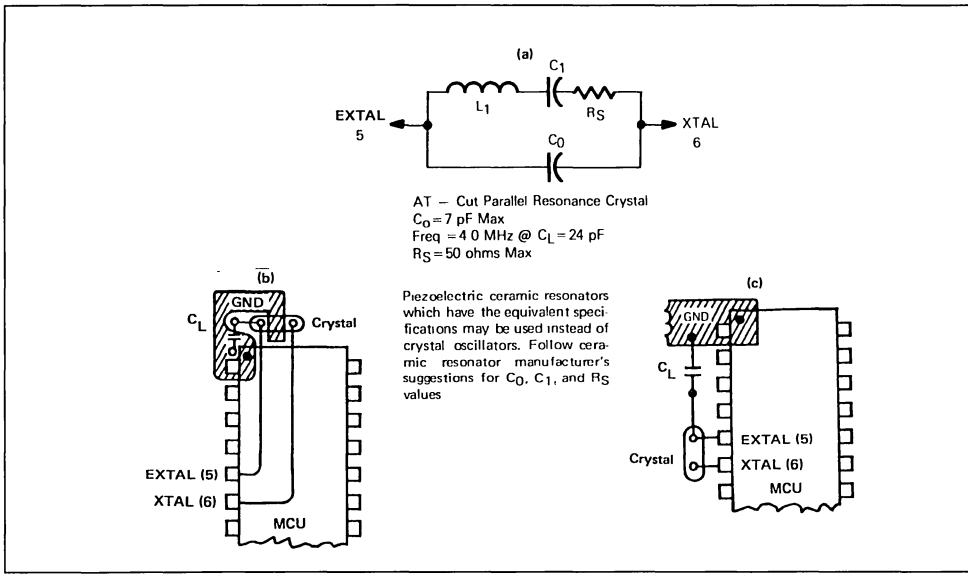
Assuming the external capacitor is initially discharged.

Figure 15 : Clock Generator Options.



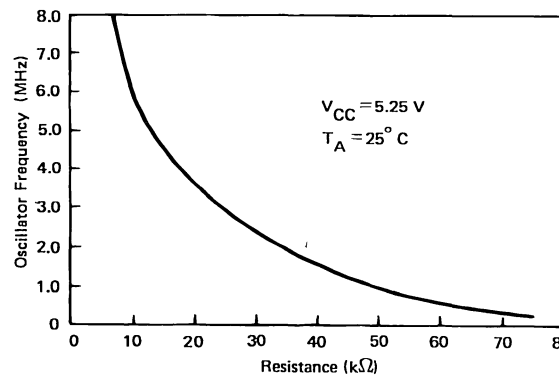
Note : The recommended C_L value with a 4.0MHz crystal is 27pF, maximum, including system distributed capacitance. There is an internal capacitance of approximately 25pF on the XTAL pin. For crystal frequencies other than 4MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2MHz crystal, use approximately 50pF on EXTAL and approximately 25pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

Figure 16 : Crystal Monotional Arm Parameters and Suggested PC Board Layout.



Note : Keep crystal leads and circuit connections as short as possible.

Figure 17 : Typical Frequency Selection for Resistor (oscillator option).



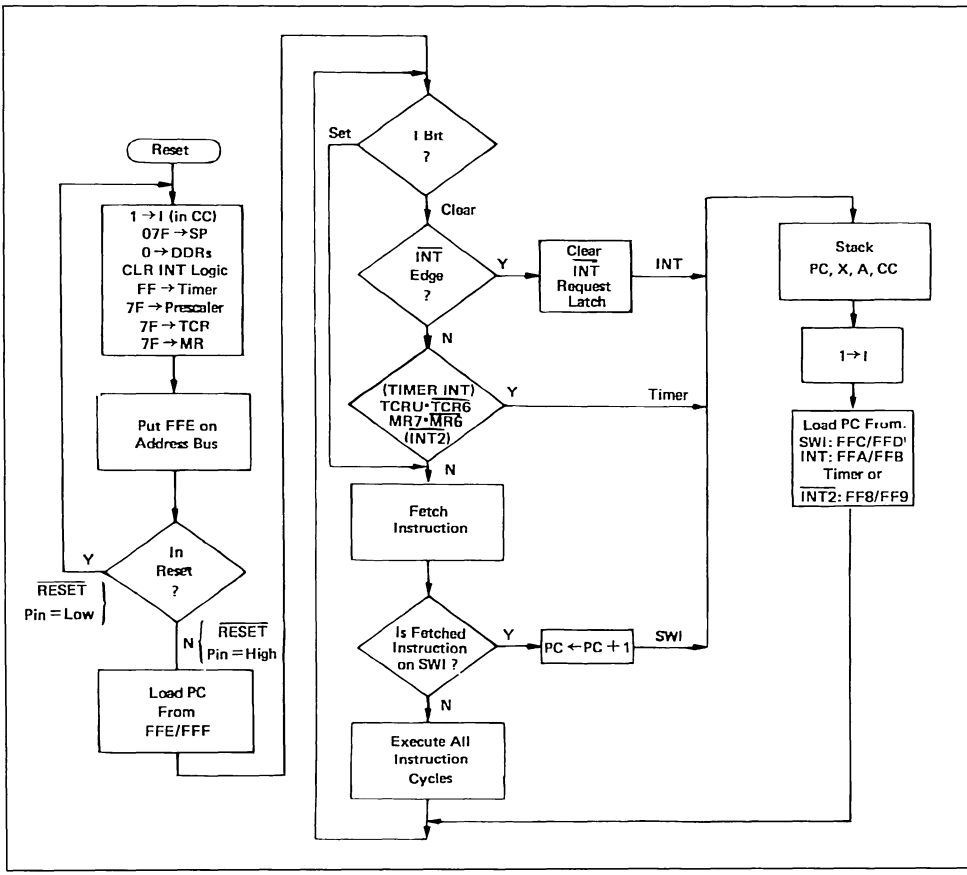
INTERRUPTS

The microcomputers can be interrupted four different ways : through the external interrupt (INT) input pin, the internal timer interrupt request, the external port D bit 6 (INT2) input pin, or the software interrupt instruction (SWI). When any interrupt occurs : the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU register, setting the I bit, and vector fetching require a total of 11 t_{cyc} periods for completion. A flowchart of the interrupt sequence is shown in figure 18. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing ; otherwise the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

Figure 18 : RESET and Interrupt Processing Flochward.



NOTE

The timer and $\overline{INT2}$ interrupts share the same vector address. The interrupt routine must determine the source by examining the interrupt request bits (TCR b7 and MR b7). Both TCR b7 and MR b7 can only be written to zero by software.

The external interrupt, $\overline{INT}$ and $\overline{INT2}$, are synchronized and then latched on the falling edge of the input signal. The $\overline{INT2}$ interrupt has an interrupt request bit (bit 7) and a mask bit (bit 6) located in the miscellaneous register (MR). The $\overline{INT2}$ interrupt is inhibited when the mask bit is set. The $\overline{INT2}$ is always read as a digital input on port D. The $\overline{INT2}$ and timer interrupt request bits, if set, cause the MCU to process an interrupt when the condition code I bit is clear.

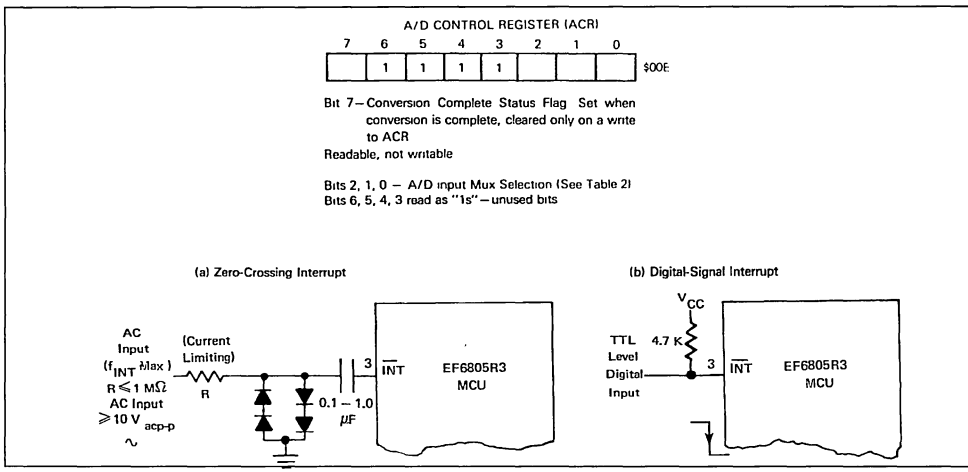
A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock. See figure 19.

NOTE

The $\overline{INT}$ (pin 3) is internally biased at approximately 2.2V due to the internal zero-crossing detection.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. SWIs are usually used as break-points for debugging or as system calls.

Figure 19 : Typical Interrupt Circuits.



INPUT/OUTPUT CIRCUITRY

There are 32 input/output pins. The $\overline{INT}$ pin may be polled with branch instructions to provide an additional input pin. All pins on ports A, B, and C are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). See below I/O port control registers configuration. The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset all the DDRs are initialized to a logic zero state, placing the ports in the input mode. The port output registers are not initialized on reset and should be initialized by software before changing the DDRs from input to output. A read operation on a port programmed as an output will read the contents of the output latch regardless of the logic levels at the output pin, due to output loading. Refer to figure 20.

PORT DATA REGISTER

7 0

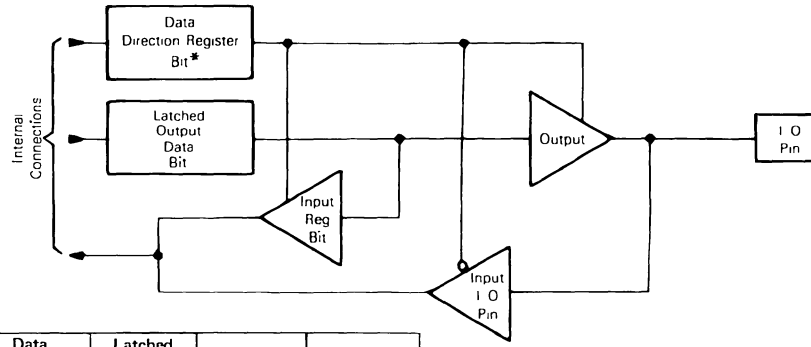
Port A Addr = $000
Port B Addr = $001
Port C Addr = $002
Port D Addr = $003

PORT DATA DIRECTION REGISTER (DDR)

7 0

(1) Write Only, reads as all "1s"
(2) 1 = Output, 0 = Input Cleared to 0 by Reset
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006

Figure 20 : Typical Port I/O Circuitry.



Data Direction Register Bit	Latched Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

*DDR is a write-only register and reads as all "1s"
** Ports B, and C are three-state ports.
Port A has optional internal pullup devices to provide CMOS data drive capability. See Electrical Characteristics tables for complete information

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs (mask option) while port B, C, and D lines are CMOS compatible as inputs. Port D lines are input only ; thus, there is no corresponding DDR. When programmed as outputs, port B is capable of sinking 10 milliamperes and sourcing 1 milliampere on each pin.

Port D provides the reference voltage, $\overline{INT2}$, and multiplexed analog inputs. All of these lines are shared with the port D digital inputs. Port D may always be used as digital inputs and may also be used as analog inputs providing V_{RH} and V_{RL} are connected to the appropriate reference voltages. The V_{RL} and V_{RH} lines (PD4 and PD5) are internally connected to the A/D resistor. Analog inputs may be prescaled to attain the V_{RL} and V_{RH} recommended input voltage range.

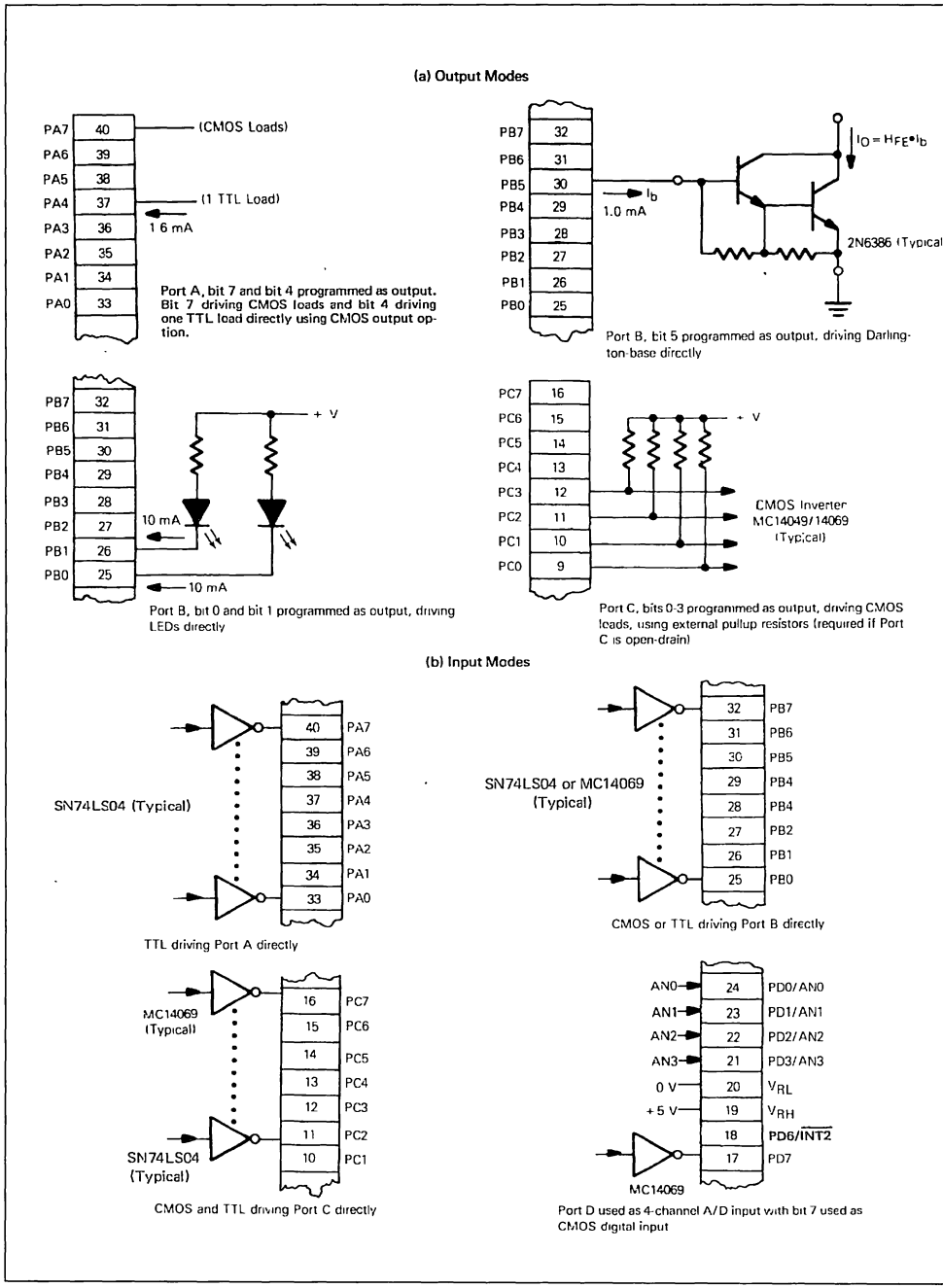
The address map (figure 6) gives the addresses of data registers and data direction registers. Figure 21 provides some examples of port connections.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see figure 20) must always be written. Therefore, any write to a port writes all of its data bits even though the port DDR is set to input. This may be used to initialize the data register and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions, since the data read corresponds to the pin level if the DDR is an input (zero) and corresponds to the latched output data when the DDR is an output (one).

Figure 21 : Typical Port Connections.



ANALOG-TO-DIGITAL CONVERTER

The EF6805R2 has an 8-bit analog-to-digital (A/D) converter implemented on the chip using a successive approximation technique, as shown in figure 22. Up to four external analog inputs, via port D, are connected to the A/D through a multiplexer. Four internal analog channels may be selected for calibration purposes (V_{RH}-V_{RL}, V_{RH}-$V_{RL}/2$, V_{RH}-$V_{RL}/4$, and V_{RL}). The accuracy of these internal channels will not necessarily meet the accuracy specifications of the external channels.

The multiplexer selection is controlled by the A/D control register (ACR) bits 0, 1, and 2 ; see table 1. This register is cleared during any reset condition.

Figure 22 : A/D Block Diagram.

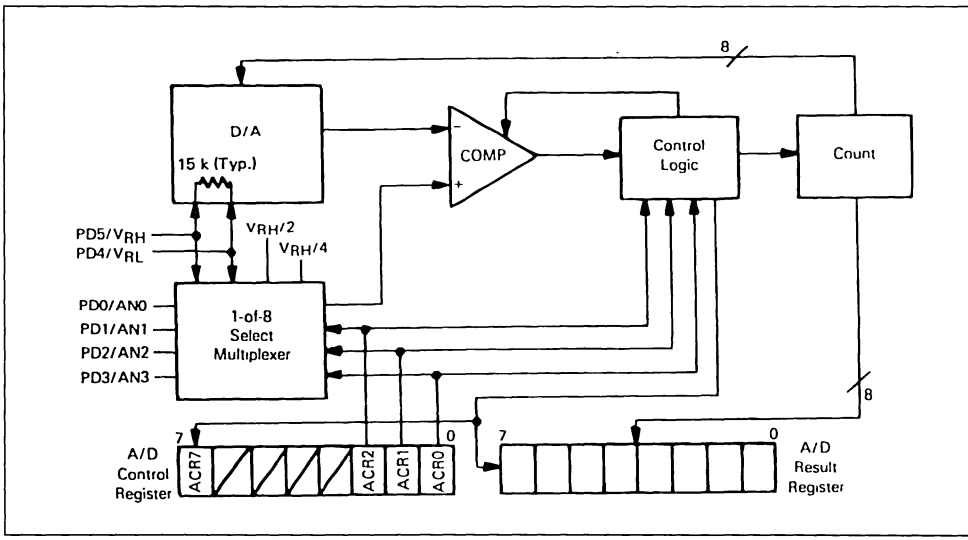


Table 1 : A/D Input Mux Selection.

A/D Control Register			Input Selected	A/D Output (hex)		
ACR2	ACR1	ACR0		Min.	Typ.	Max.
0	0	0	AN0			
0	0	1	AN1			
0	1	0	AN2			
0	1	1	AN3			
1	0	0	V_{RH}*	FE	FF	FF
1	0	1	V_{RL}*	00	00	01
1	1	0	$V_{RH/4}$*	3F	40	41
1	1	1	$V_{RH/2}$*	7F	80	81

* Internal (calibration) levels.

MISCELLANEOUS REGISTER (MR)

7	6						0	
		1	1	1	1	1	1	$00A

A/D RESULT REGISTER (ARR)

7							0	
MSB							LSB	$00F

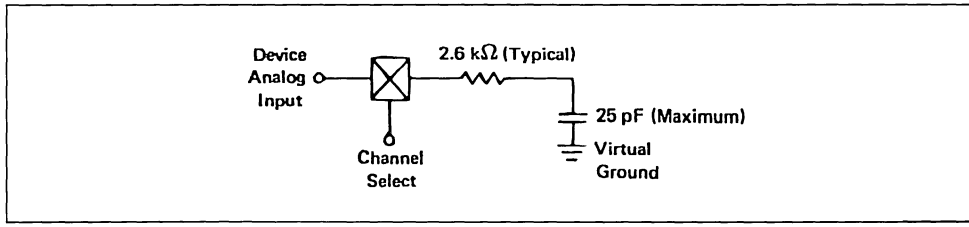
MR7 Bit 7 - $\overline{INT2}$ interrupt request bit : set when falling edge detected on INT2 pin, must be cleared by software. Cleared to 0 by reset.

MR6 Bit 6 - $\overline{INT2}$ interrupt mask bit : 1 = INT2 interrupt masked (disabled). Set to 1 by reset.

MR Bits 5, 4, 3, 2, 1, 0 - Read as "1s" - unused bits.

Whenever the ACR is written, the conversion in progress is aborted, the conversion complete flag (ACR bit 7) is cleared, and the selected input is sampled for five machine cycles and held internally. During these five cycles, the analog input will appear approximately like a 25 picofarad (maximum) capacitor (plus approximately 10pF for packaging) charging through a 2.6 kiloohm resistor (typical). Refer to figure 23.

Figure 23 : Effective Analog Input Impedance (during sampling only).



The converter operates continuously using 30 machine cycles to complete a conversion of the sampled analog input. When the conversion is complete, the digitized sample of digital value is placed in the A/D result register (ARR), the conversion complete flag is set, the selected input is sampled again, and a new conversion is started.

The A/D is ratiometric. Two reference voltages (V_{RH} and V_{RL}) are supplied to the converter via port D pins. An input voltage equal to V_{RH} converts to $FF (full scale) and an input voltage equal to V_{RL} converts to $00. An input voltage greater than V_{RH} converts to SFF and no overflow indication is provided. Similarly, an input voltage less than V_{RL}, but greater than V_{SS} converts to $00. Maximum and minimum ratings must not be exceeded. For ratiometric conversion, the source of each analog input should use V_{RH} as the supply voltage and be referenced to VRL. To maintain the full accuracy on the A/D, V_{RH} should be equal to or less than V_{DD}, V_{RL} should be equal to or greater than V_{SS} but less than the maximum specification and (V_{RH}-V_{RL}) should be equal to or greater than 4 volts.

The A/D has a built-in 1/2 LSB offset intended to reduce the magnitude of the quantizing error to 1/2 LSB, rather than + 0, - 1 LSB with no offset. This implies that, ignoring errors, the transition point from $00 to $01 occurs at 1/2 LSB above VRL. Similarly, the transition from $FE to $FF occurs 1 1/2 LSB below VRH, ideally. Refer to figure 24 and 25.

On release of reset, the A/D control register (ACR) is cleared therefore after reset, channel zero will be selected and the conversion complete flag will be clear.

Figure 24 : Ideal Converter Transfer Characteristic.

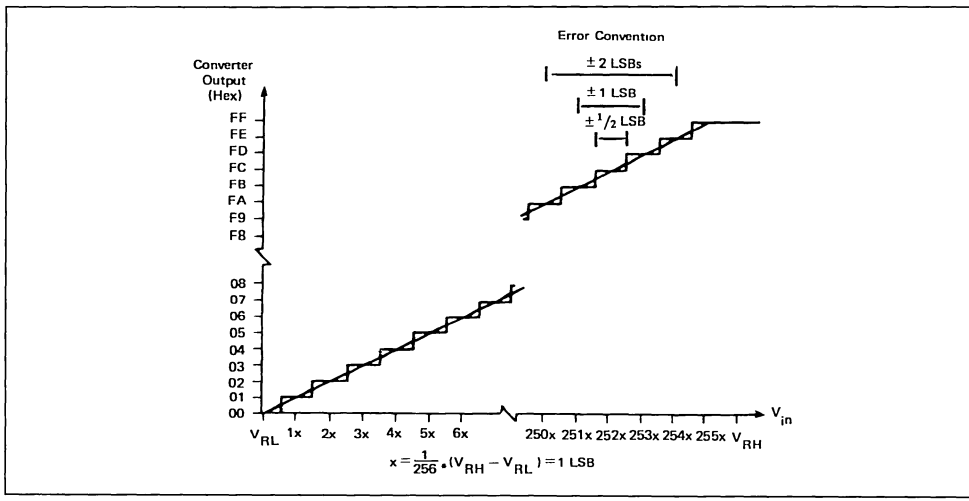
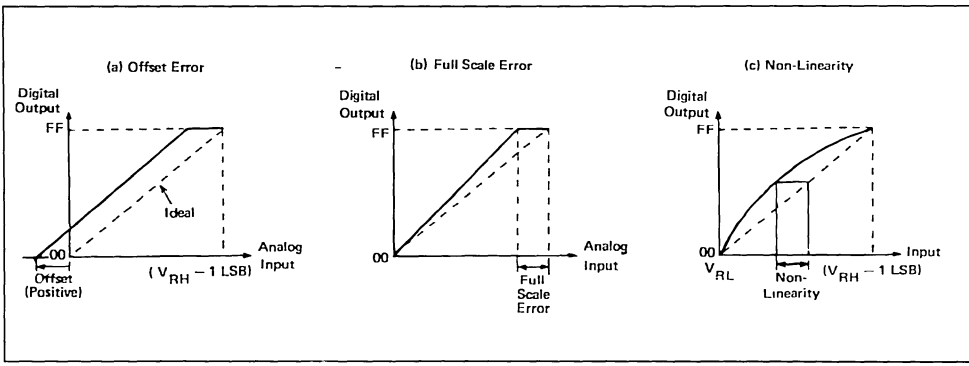


Figure 25 : Types of Conversion Errors.



BIT MANIPULATION

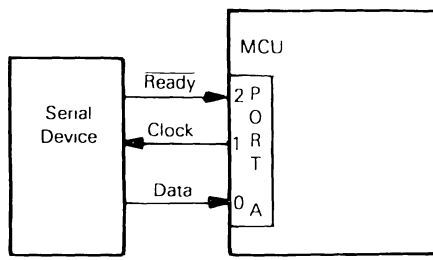
The EF6805R3 has the ability to set or clear any single RAM or I/O bit (except the data direction registers) with a single instruction (BSET, BCLR) (see Caution below). Any bit in page zero can be tested using the BRSET and BRCLR instructions and the program branches as a result of its state. The carry bit equals the value of the bit references by BRSET or BRCLR. The capability to working with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines.

CAUTION

The corresponding data direction registers for ports A, B, and C are write-only registers (locations $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set a data direction register bit (all "unaffected" bits would be set). It is recommended that all data direction register bits in a port be written using a single-store instruction.

The coding examples shown in figure 26 illustrate the usefulness of the bit manipulation and test instruction. Assume that the microcomputer is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line to clock data one bit at a time, least significant bit first, out of the device. The microcomputer waits until the data is ready, clocks the external device, picks up the data in the carry flag, clears the clock line, and finally accumulates the data bit in a random-access memory location.

Figure 26 : Bit Manipulation Example.




```
*       BRSET   2, PORTA, *     WAIT FOR READY

        BSET    1, PORTA        CLOCK NEXT BIT IN
        BRCLR   0, PORTA, NEXT  PICKUP BIT IN C-BIT
NEXT    BCLR    1, PORTA        RETURN CLOCK LINE HIGH
        ASR     RAMLOC          MOVE C-BIT INTO RAM
```

ADDRESSING MODES

The EF6805R3 MCU has ten addressing modes available for use by the programmer. They are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the EF6805 Family Users Manual.

The term "effective address" (EA) is used in describing the addressing modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE. In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT. In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single 2-byte instruction. This address area includes all on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED. In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE. The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if, and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from – 126 to + 129 from the opcode address. The programmer need not worry about calculating the correct offset if he uses the assembler, since it calculates the proper offset and checks to see if it is within the span of the branch.

INDEXED, NO OFFSET. In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET. In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET. In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended, the assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR. In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

BIT TEST AND BRANCH. The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit which is to be tested and condition (set or clear) is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from – 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code registers.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

INHERENT. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instructions with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The EF6805R3 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS. Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to table 1.

READ-MODIFY-WRITE INSTRUCTIONS. These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register ; see Caution under Input/Output section. The test for negative or zero (TST) instruction is included in the read-modify-write instruction though it does not perform the write. Refer to table 2.

BRANCH INSTRUCTIONS. The branch instructions cause a branch from the program when a certain condition is met. Refer to table 3.

BIT MANIPULATION INSTRUCTIONS. The instructions are used on any bit in the first 256 bytes of the memory ; One group either sets or clears. The other group performs the bit test and branch operations. Refer to table 4.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

CONTROL INSTRUCTION. The control instructions control the MCU operations during program execution. Refer to table 5.

ALPHABETICAL LISTING. The complete instruction set is given in alphabetical order in table 6.

OPCODE MAP. Table 8 is an opcode map for the instruction used on the MCU.

Table 1 : Register/memory Instructions.

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (no offset)			Indexed (8 bit offset)			Indexed (16 bit offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA				B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX				BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (logical compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP				BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR				BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 2 : Read-modify-write.

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (no offset)			Indexed (8 bit offset)		
Function	Mnem	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 3 : Branch Instructions.

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
Branch IFF Higher or Same	BHS	24	2	4
Branch IFF Carry Set	BCS	25	2	4
Branch IFF Lower	BLO	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF interrupt mask bit is clear.	BMC	2C	2	4
Branch IFF interrupt mask bit is set.	BMS	2D	2	4
Branch IFF interrupt line is low.	BIL	2E	2	4
Branch IFF interrupt line is high.	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 4 : Bit Manipulation Instructions.

Function	Mnemonic	Addressing Modes					
		Bit Set/clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is set	BRSET n (n = 0... 7)				2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0... 7)				01 + 2 • n	3	10
Set Bit n	BSET n (n = 0... 7)	10 + 2 • n	2	7			
Clear Bit n	BCLR n (n = 0... 7)	11 + 2 • n	2	7			

Table 5 : Control Instructions.

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-operation	NOP	9D	1	2

Table 6 : Instruction Set.

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			∧	●	∧	∧	∧
ADD		X	X	X		X	X	X			∧	●	∧	∧	∧
AND		X	X	X		X	X	X			●	●	∧	∧	●
ASL	X		X			X	X				●	●	∧	∧	∧
ASR	X		X			X	X				●	●	∧	∧	∧
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	∧	∧	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	∧
BRSET										X	●	●	●	●	∧
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLL	X										●	●	●	●	0

Condition Code Symbols :

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
∧ Test and Set if True, Cleared Otherwise
• Not Affected

Table 6 : Instruction Set (continued).

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/clear	Bit Test & Branch	H	I	N	Z	C
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	^	^	^
COM	X		X			X	X				●	●	^	^	1
CPX		X	X	X		X	X	X			●	●	^	^	^
DEC	X		X			X	X				●	●	^	^	●
EOR		X	X	X		X	X	X			●	●	^	^	●
INC	X		X			X	X				●	●	^	^	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	^	^	●
LDX		X	X	X		X	X	X			●	●	^	^	●
LSL	X		X			X	X				●	●	^	^	^
LSR	X		X			X	X				●	●	0	^	^
NEQ	X		X			X	X				●	●	^	^	^
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	^	^	●
ROL	X		X			X	X				●	●	^	^	^
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols :

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
^ Test and Set if True, Cleared Otherwise
• Not Affected
? Load CC Register From Stack

EF6805 HMOS FAMILY

Features	EF6805P2	EF6805P6	EF6805R2	EF6805R3	EF6805U2	EF6805U3
Technology	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	40	40	40
On-chip RAM (bytes)	64	64	64	112	64	112
On-chip User ROM (bytes)	1100	1796	2048	3776	2048	3776
External Bus	None	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	24	24	24
Unidirectional I/O Lines	None	None	6 Inputs	6 Inputs	8 Inputs	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, A/D	Timer	Timer
External Interrupt Inputs	1	1	2	2	2	2
STOP and WAIT	No	No	No	No	No	No

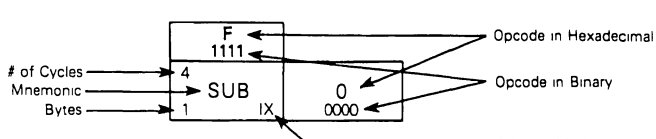
Table 7 : 6805 HMOS Family Instruction Set Opcode Map.

Hi / Low	Bit Manipulation BTB 0 0000	Bit Manipulation BSC 1 0001	Branch REL 2 0010	Read-Modify-Write DIR 3 0011	Read-Modify-Write INH 4 0100	Read-Modify-Write INH 5 0101	Read-Modify-Write IX1 6 0110	Read-Modify-Write IX 7 0111	Control INH 8 1000	Control INH 9 1001	Register/Memory IMM A 1010	Register/Memory DIR B 1011	Register/Memory EXT C 1100	Register/Memory IX2 D 1101	Register/Memory IX1 E 1110	Register/Memory IX F 1111	Hi / Low
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DTR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

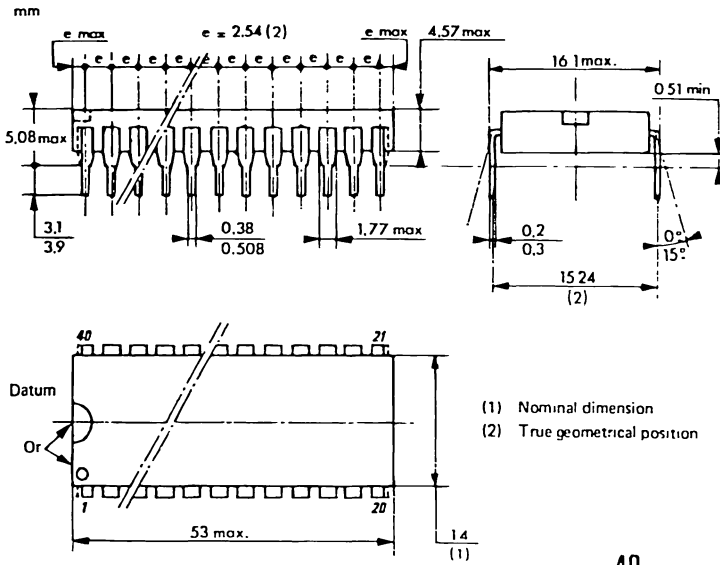
INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

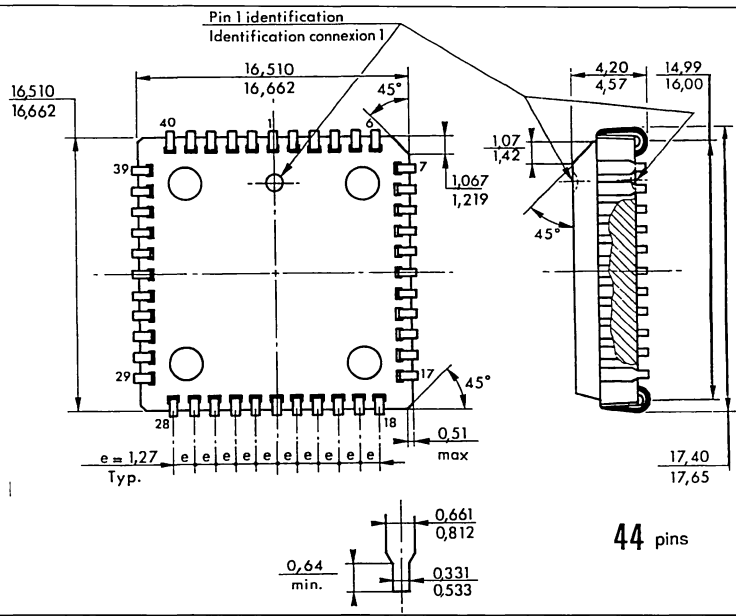


PACKAGE MECHANICAL DATA

P SUFFIX - PLASTIC PACKAGE



FN SUFFIX - PLCC44



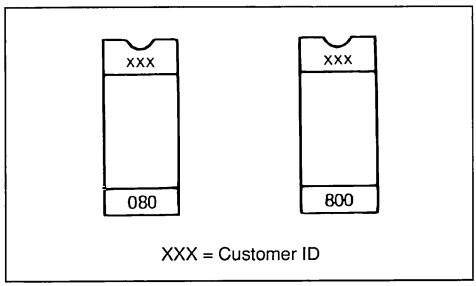
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS-THOMSON on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local SGS-THOMSON representative or distributor.

EPROMs

Two 2716 or one 2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS THOMSON. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, SGS-THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by SGS THOMSON. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6805 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

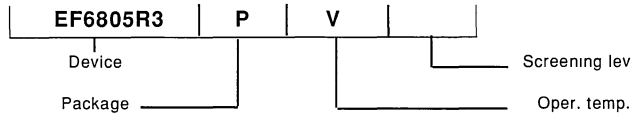
EFDOS is SGS-THOMSON' Disk Operating System available on development systems such as DEVICE,...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser,...

* Requires prior factory approval.

Whenever ordering a custom MCU is required, please contact your local SGS-THOMSON representative or SGS-THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS-THOMSON Microelectronics representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device	Package					Oper. Temp			Screening Level			
	C	J	P	E	FN	L*	V	T	Std	D	G/B	B/B
EF6805R3			●		●	●	●	●	●	●		
Examples : EF6805R3P, EF6805R3FN, EF6805R3PV, EF6805R3FNV.												

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : * : 0°C to +70°C, **V :** –40°C to +85°C, **T :** –40°C to +105°C, *** :** may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B.

EXORciser is a registered trademark of MOTOROLA Inc.

EF6805 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....
.....

ROM capacity required : | | | | bytes

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 40°C / + 85°C
- ❑ – 40°C / + 105°C

Package
- ❑ Plastic
- ❑ PLCC

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
- ❑ SGS-THOMSON Microelectronics reference
  .....
- ❑ Special customer data reference*
  .....

Number of interrupt vector : | |

Quality level :
- ❑ STD
- ❑ D
- ❑ Other* (customer's quality specification ref.) :
  .....

Software developped by :
- ❑ SGS-THOMSON Microelectronics application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purposes) :
- ❑ EPROM Reference :
- ❑ EFDOS/MDOS* disk file
  - ❑ 8" floppy
  - ❑ 5" 1/4 floppy
- ❑ Other *

OPTION LIST

-Oscillator input
- ❑ Xtal
- ❑ RC

- Low voltage inhibit
- ❑ Enabled
- ❑ Disabled

- Port A output drive :
- ❑ CMOS and TTL
- ❑ TTL only

- Port C output drive :
- ❑ TTL
- ❑ Open drain

* Requires prior factory approval

Yearly quantity forecast :
- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME : DATE : SIGNATURE :

SGS-THOMSON MICROELECTRONICS

EF6805U2

8-BIT MICROCOMPUTER

ADVANCE DATA

HARDWARE FEATURES

- 32 TTL/CMOS COMPATIBLEI/O LINES
- 24 BIDIRECTIONAL (8 lines are LED compatible)
- 8 INPUT-ONLY
- 2048 BYTES OF USER ROM
- 64 BYTES OF RAM
- SELF-CHECK MODE
- ZERO-CROSSING DETECT/INTERRUPT
- INTERNAL 8-BIT TIMER WITH 7-BIT MASK PROGRAMMABLE PRESCALER AND CLOCK SOURCE
- 5V SINGLE SUPPLY

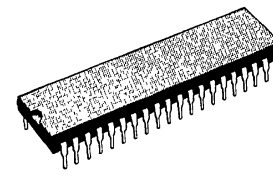
SOFTWARE FEATURES

- 10 POWERFUL ADDRESSING MODES
- BYTE EFFICIENT INSTRUCTION SET WITH TRUE BIT MANIPULATION, BIT TEST, AND BRANCH INSTRUCTIONS
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- POWERFUL INDEXED ADDRESSING FOR TABLES
- FULL SET OF CONDITIONAL BRANCHES
- MEMORY USABLE AS REGISTER/FLAGS
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE®

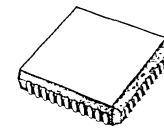
USER SELECTABLE OPTIONS

- INTERNAL 8-BIT TIMER WITH SELECTABLE CLOCK SOURCE (External Timer Input or Internal Machine Clock)
- TIMER PRESCALER OPTION (7 Bits, 2^n)
- 8 BIDIRECTIONAL I/O LINES WITH TTL OR TTL/CMOS INTERFACE OPTION
- 8 BIDIRECTIONAL I/O LINES WITH TTL OR OPEN-DRAIN INTERFACES OPTION
- CRYSTAL OR LOW-COST RESISTOR OSCILLATOR OPTION
- LOW VOLTAGE INHIBIT OPTION
- VECTORED INTERRUPTS TIMER, SOFTWARE, AND EXTERNAL
- USER CALLABLE SELF-CHECK SUBROUTINES

INICE® is SGS-THOMSON's development/emulation tool.

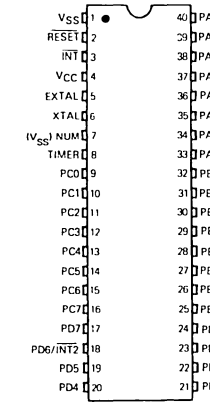


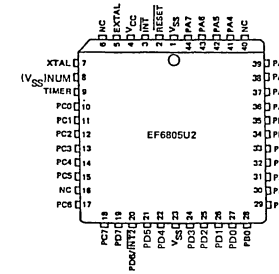
CB-182
P SUFFIX
(Plastic Package)



CB-521
L SUFFIX
PLCC44

PIN CONNECTION

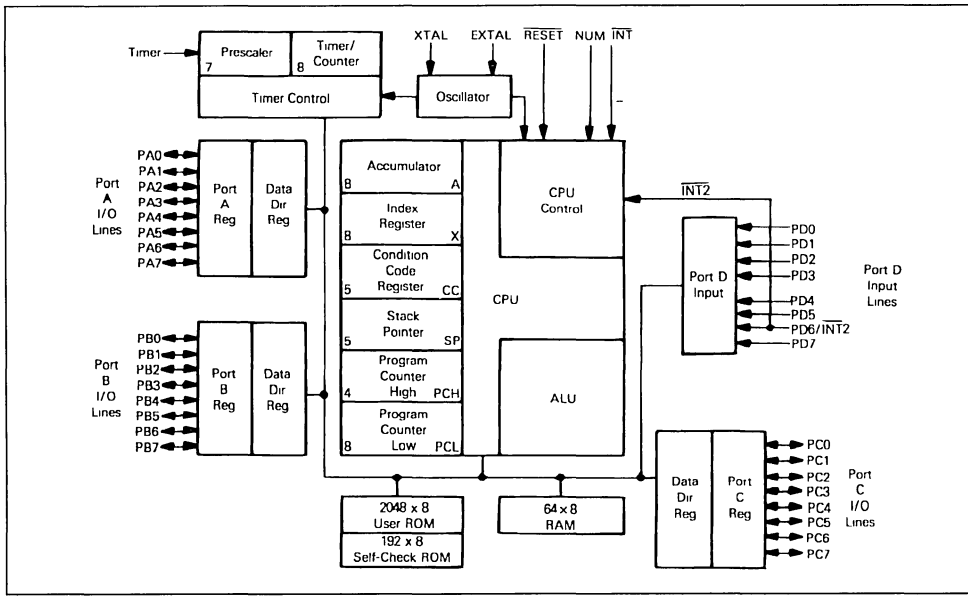




DESCRIPTION

The EF6805U2 Microcomputer Unit (MCU) is a member of the 6805 Family of low-cost single-chip Microcomputers. The 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of several members of the 6805 Family of microcomputers is shown at the end of this data sheet. The following are some of the hardware and software highlights of the EF6805U2 MCU.

Figure 1 : EF6805U2 HMOS Microcomputer Block Diagram.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage (except TIMER in self-check mode and open-drain inputs)	– 0.3 to + 7.0	V
V_{in}	Input Voltage (open-drain pins, TIMER pin in self-check mode)	– 0.3 to + 15.0	V
T_A	Operating Temperature Range V Suffix (T_L to T_H) - T Suffix	0 to + 70 – 40 to + 85 – 40 to + 105	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_j	Junction Temperature Plastic Package PLCC	150 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electrical fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. Reliability of operation is enhanced if unused inputs except EXTAL are tied to an appropriate logic voltage level (e g. either V_{SS} or V_{CC}).

THERMAL DATA

θJA	Thermal Resistance	Plastic PLCC	50 80	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Juntion-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} \leq P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$$P_D = K + (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives :

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

Where K is a constant pretaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ELECTRICAL CHARACTERISTICS (V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage				V
	RESET (4.75 ≤ V_{CC} ≤ 5.75)	4.0		V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5		V_{CC}	
	INT (4.75 ≤ V_{CC} ≤ 5.75)	4.0	•	V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5	•	V_{CC}	
	All Other (except timer)	2.0		V_{CC}	
V_{IH}	Input High Voltage Timer				V
	Timer Mode	2.0		V_{CC} + 1.0	
	Self-check Mode	9.0	10.0	15.0	
V_{IL}	Input Low Voltage				V
	RESET	V_{SS}		0.8	
	INT	V_{SS}	•	1.5	
	All Other	V_{SS}		0.8	
V_{IRES+} V_{IRES-}	RESET Hystereris Voltages (see figures 10, 11 and 12)				V
	"Out of Reset"	2.1		4.0	
	"Into Reset"	0.8		2.0	
V_{INT}	INT Zero Crossing Input Voltage, Through a Capacitor	2		4	$V_{ac\ p-p}$
P_D	Power Dissipation - (no port loading, V_{CC} = 5.75V)				mW
	T_A = 0°C		520	740	
	T_A = – 40°C		580	800	
C_{in}	Input Capacitance				pF
	EXTAL		25		
	All Other		10		
V_{LVR}	Low Voltage Recover			4.75	V
VL_{VI}	Low Voltage Inhibit	2.75	3.75	4.70	V
I_{in}	Input Current				µA
	TIMER (V_{in} = 0.4V)			20	
	INT (V_{in} = 2.4V to V_{CC})		20	50	
	EXTAL (V_{in} = 2.4V to V_{CC} - crystal option)			10	
	(V_{in} = 0.4V - crystal option)			– 1600	
	RESET (V_{in} = 0.8V) - External Capacitor Charging Current	– 40.0		– 40	

* Due to internal biasing this input (when unused) floats to approximately 2.2V.

SWITCHING CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency	0.4		4.2	MHz
t_{cyc}	Cycle Time ($4/f_{osc}$)	0.95		10	µs
t_{WL}, t_{WH}	$\overline{INT}$, $\overline{INT2}$, and TIMER Pulse Width (see interrupt section)	t_{cyc} + 250			ns
t_{RWL}	$\overline{RESET}$ Pulse Width	t_{cyc} + 250			ns
f_{INT}	$\overline{INT}$ Zero-crossing Detection Input Frequency	0.03		1	kHz
	External Clock Input Duty Cycle (EXTAL)	40	50	60	%
	Crystal Oscillator Start-up Time*			100	ms

* See figure 16 for typical crystal parameters

PORT ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

PORT A WITH CMOS DRIVE ENABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage (I_{Load} = 1.6mA)			0.4	V
V_{OH}	Output High Voltage I_{Load} = – 100µA I_{Load} = – 10µA	 2.4 V_{CC} – 1.0			V
V_{IH}	Input High Voltage (I_{Load} = – 300µA max.)	2.0		V_{CC}	V
V_{IL}	Input Low Voltage (I_{Load} = – 500µs max.)	V_{SS}		0.8	V
I_{IH}	High Z State Input Current (V_{in} = 2.0V to V_{CC})			– 300	µA
I_{IL}	High Z State Input Current (V_{in} = 0.4V)			– 500	µA

PORT B

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage I_{Load} = 3.2mA I_{Load} = 10mA (sink)			 0.4 1.0	V
V_{OH}	Output High Voltage I_{Load} = – 200µA	2.4			V
I_{OH}	Darlington Current Drive (source) V_O = 1.5V	– 1.0		– 10	mA
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	High Z State Input Current		< 2	10	µA

PORT C AND PORT A WITH CMOS DRIVE DISABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage I_{Load} = – 100µA	2.4			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	High Z State Input Current		< 2	10	µs

PORT ELECTRICAL CHARACTERISTICS (continued)

PORT C (open-drain option)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage	2.0		13.0	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{LOD}	Input Leakage Current		< 3	15	µA
V_{OL}	Output Low Voltage I_{Load} = 1.6mA			0.4	V

PORT D

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{in}	Input Current		< 1	5	µA

Figure 2 : TTL Equivalent Test Load (port B).

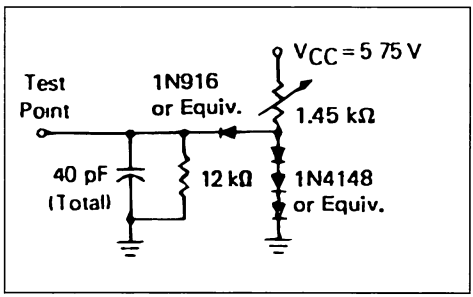


Figure 3 : CMOS Equivalent Test Load (port A).

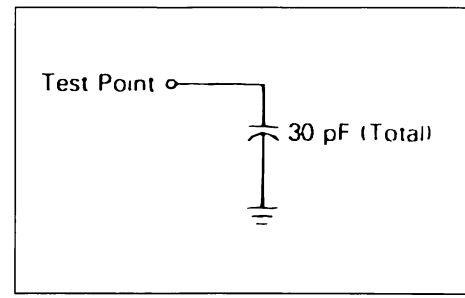


Figure 4 : TTL Equivalent Test Load (ports A and C).

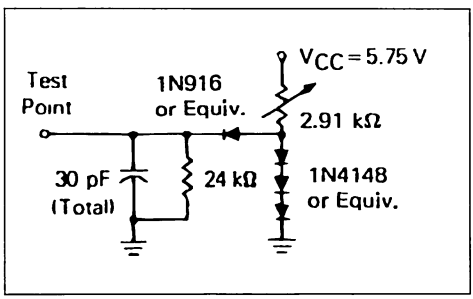
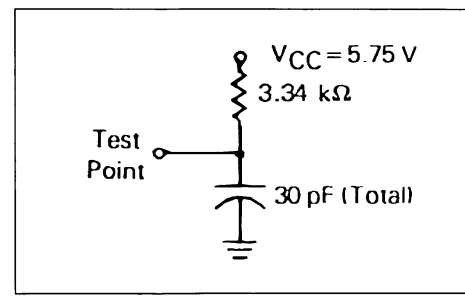


Figure 5 : Open-drain Equivalent Test Load (port C).



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in figure 1, are described in the following paragraphs.

V_{CC} AND V_{SS} - Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

$\overline{INT}$ - This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts Section for additional information.

XTAL AND EXTAL - These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on user selectable manufacturing mask option, can be connected to these pins to provide a system clock with various degrees of stability/cost tradeoffs. Lead length and stray capacitance on these two pins

should be minimized. Refer to Internal Clock Generator Options Section for recommendations about these inputs.

NUM (NON USER MODE) - This pin is not for user application and must be connected to V_{SS}.

TIMER - The pin allows an external input to be used to control the internal timer circuitry and also to initiate the self test program. Refer to Timer Selection for additional information about the timer circuitry.

$\overline{\text{RESET}}$ - This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. The MCU can be reset by pulling $\overline{\text{RESET}}$ low. Refer to Resets Section for additional information.

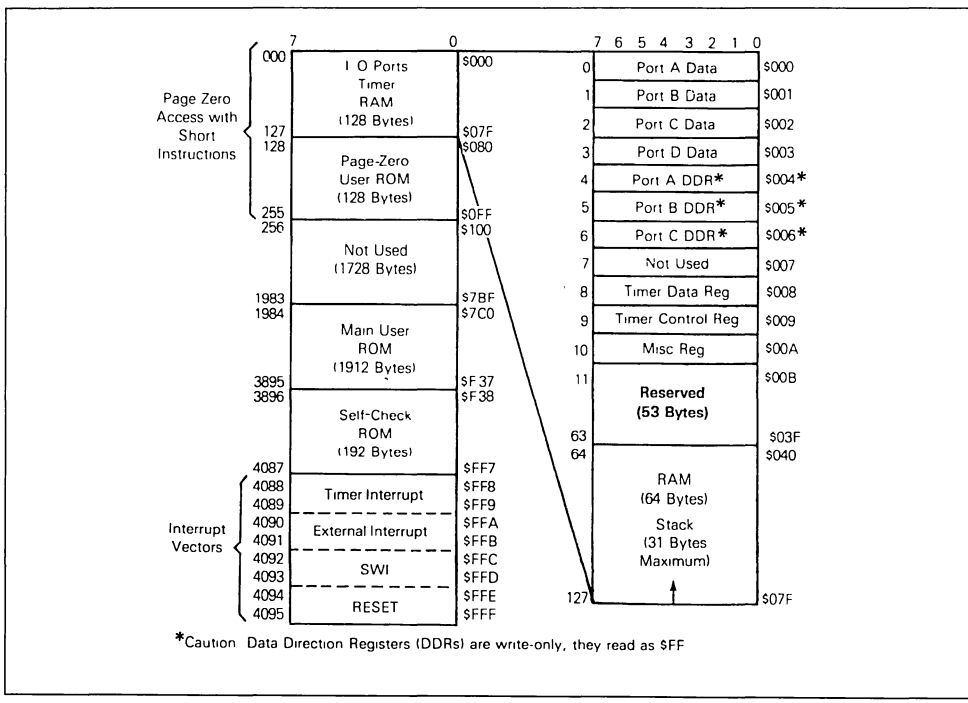
INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) - These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction registers (DDRs). Port D is for digital input only and bit 6 may be used for a second interrupt INT2. Refer to Input/Output Section and Interrupts Section for additional information.

MEMORY - The MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The EF6805U2 MCU has implemented 2314 of these bytes. This consists of 2048 user ROM bytes, 192 self-check ROM bytes, 64 user RAM bytes, 7 port I/O bytes, 2 timer registers, and a miscellaneous register ; see figure 6 for the Address map. The user ROM has been split into three areas. The main user ROM area is from $080 to $OFF and from $7CO to $F37. The last 8 user ROM locations at the bottom of memory are for the interrupt vectors.

The MCU reserves the first 16 memory locations for I/O features, of which 10 have been implemented. These locations are used for the ports, the port DDRs, the timer and the INT2 miscellaneous register, and the 64 RAM bytes, 31 bytes are shared with the stack area. The stack must be used with care when data shares the stack area.

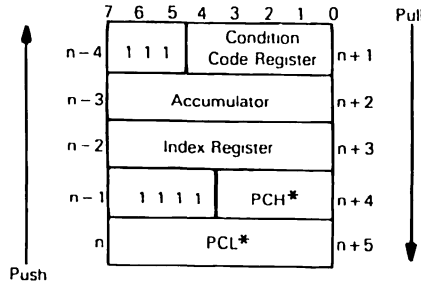
The shared stack area is used during the processing of an interrupt or subroutine calls to save the contents of the CPU state. The register contents are pushed onto the stack in the order shown in figure 7. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program.

Figure 6 : EF6805U2 MCU Address Map.



*Caution Data Direction Registers (DDRs) are write-only, they read as $FF

Figure 7 : Interrupt Stacking Order.



* For subroutine calls, only PCH and PCL are stacked

CENTRAL PROCESSING UNIT

The CPU of the EF6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

REGISTERS

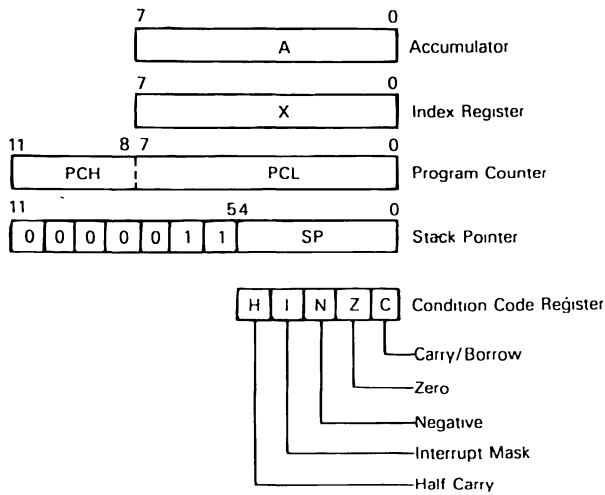
The 6805 Family CPU has five registers available to the programmer. They are shown in figure 8 and are explained in the following paragraphs.

ACCUMULATOR (A) - The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDES REGISTER (X) - The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions. The Index Register may also be used as a temporary storage area.

PROGRAM COUNTER (PC) - The Program Counter is a 12 bit register that contains the address of the next instruction to be executed.

Figure 8 : Programming Model.



STACK POINTER (SP) - The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is then pulled from the stack. The seven most significant bits of the stack pointer are permanently set to 0000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

CONDITION CODE REGISTER (CC) - The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

Half Carry (H) - Set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I) - When this bit is set, the timer and external interrupts ($\overline{INT}$ and $\overline{INT2}$) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical "1").

Zero (Z) - When set, this bit indicates that the result of the last arithmetic, or data manipulation was zero.

Carry/Borrow (C) - When set, this bit indicates that a carry or borrow out of the Arithmetic Logic Unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The timer circuitry for the MC6805U2 is shown in figure 10. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (or prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine (see **RESET, CLOCK, AND INTERRUPT STRUCTURE SECTIONS**). The timer interrupt request bit must be cleared by software. The interrupt request bit must be cleared by software. The TIMER and INT2 share the same interrupt vector. The interrupt routine must check the request bits to determine the source of the interrupt.

The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal phase two signal. Three machine cycles are required for a change in state of the TIMER pin to decrement the timer prescaler. The maximum frequency of a signal that can be recognized by the TIMER pin logic is dependent on the parameter labeled t_{WH}. The pin logic that recognizes the high state on the pin must also recognize the low state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows (assumes 50/50 duty cycle for a given period) :

$$t_{cyc} \times 2 + 250ns = period = \frac{1}{freq}$$

The periods is not simply $t_{WH} + t_{WL}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 nanoseconds times two).

When the phase two signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. The source of the clock input is one of the mask options that is specified before manufacture of the MCU.

NOTE

For ungated phase two clock input to the timer prescaler, the TIMER pin should be tied to V_{CC}.

A prescaler option, divide by 2^n, can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture. To avoid truncation errors, the prescaler is cleared when bit 3 of the timer control register is written to a logic one (this bit always reads a logic zero). See figure 9.

Figure 9 : Timer Control Register (TCR).

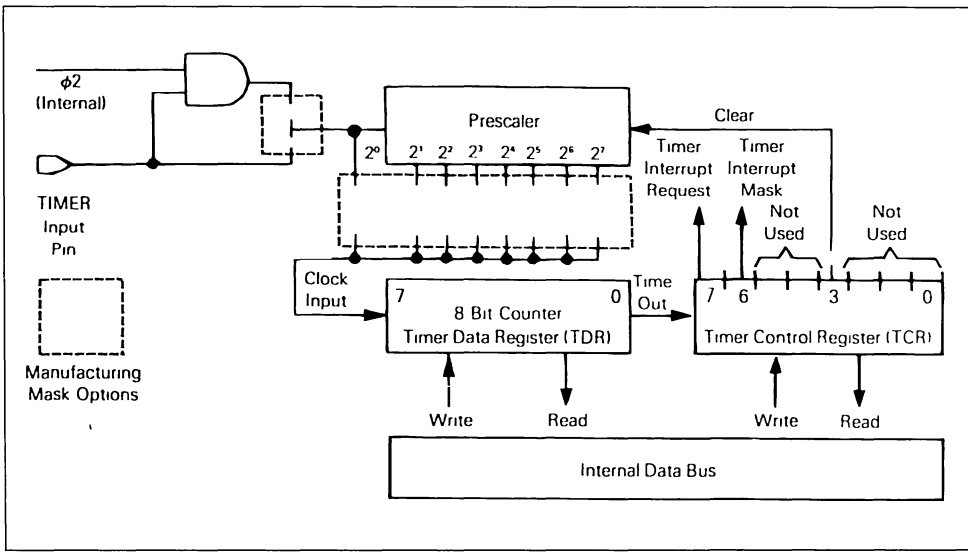
7	6	5	4	3	2	1	0	
		1	1		1	1	1	$009

TCR7 - Timer Interrupt Request Status Bit Set when TDR goes to zero, must be cleared by software Cleared to 0 by Reset
TCR6 - Timer Interrupt Mask Bit : 1 = timer interrupt masked (disabled) Set to 1 by Reset
TCR3 - Clear prescaler always reads as a 0, clears prescaler when written to a logic "1"
TCR Bits 5, 4, 2, 1, 0 reads "1s" - unused bits

The timer continues to count past zero, falling through to $FF from $00 and then continuing the count down. Thus the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occured, and not disturb the counting process.

At power up or reset, the prescaler and counter are initialized with all logic ones ; the timer interrupt request bit (bit 7) is cleared and the timer interrupt mask bit (bit 6) is set.

Figure 10 : Timer Block Diagram.



SELF-CHECK

The self-check capability of the EF6805U2MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 11 and monitor the output of Port C bit 3 for an oscillation of approximately 7Hz. A 10-volt level (through a 10k resistor) on the timer input, pin 8 and pressing then releasing the RESET button, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, interrupts, and I/O ports.

Several of the self-check subroutines can be called by a user program with a JSR or BSR instruction. They are the RAM, ROM, tests. The timer routine may also be called if the timer input is the internal ∅ 2 clock.

To call those subroutines in customer applications, please contact your SGS-THOMSON Microelectro-

nics sales office in order to obtain the complete discription of the self-check program and the entrance/exit conditions.

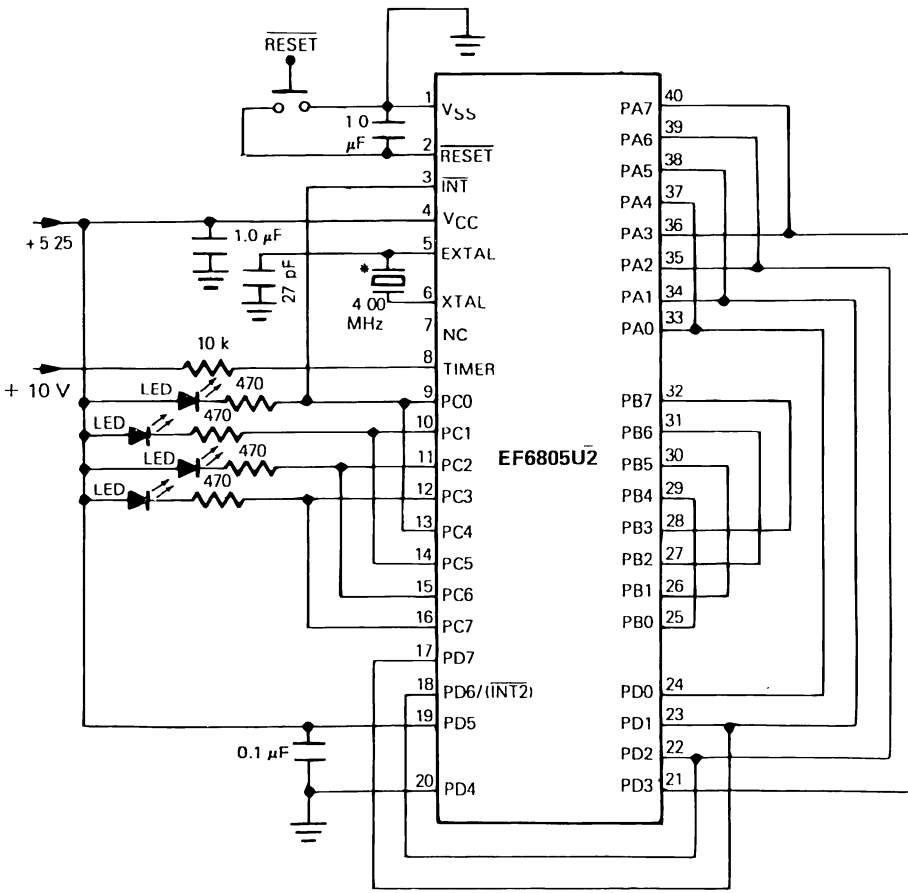
RAM SELF-CHECK SUBROUTINE - The RAM self-check is called at location $F6F and returns with the Z bit clear if any error is detected ; otherwise the Z bit is set. The walking diagnostic pattern method is used on the EF6805U2.

The RAM test must be called with the stack pointer at $07F. When run, the test checks every RAM cell except for $07F and $07E which are assumed to contain the return address.

The A and X registers and all RAM locations except $07F and $07E are modified.

ROM CHECKSUM SUBROUTINE - The ROM self-check is called at location $F8A. If any error is detected, it returns with the Z bit cleared ; otherwise Z = 1, X = 0 on return, and A is zero if the test passes. RAM location $040 to $043 is overwritten. The checksum is the complement of the execution OR of the contents of the user ROM.

Figure 11 : Self-check connections.



* This connection depends on clock oscillator user selectable mask option Use jumper if the RC mask option is selected

LED MEANINGS

PC0	PC1	PC2	PC3	Remarks (1 : LED ON ; 0 : LED OFF)
1	0	1	0	Bad I/O
0	0	1	0	Bad Timer
1	1	0	0	Bad RAM
0	1	0	0	Bad ROM
0	0	0	0	Bad Interrupts or Request Flag
All Flashing				Good Device

Note : Anything else bord Device, bad Port C, etc.

TIMER SELF-CHECK SUBROUTINE - The timer self-check is called at location $FCF and returns with the Z bit cleared if any error was found ; otherwise Z = 1.

In order to work correctly as a user subroutine, the internal Ø 2 clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock is running and the interrupt mask is not set so the caller must protect from interrupts if necessary.

The A and X register contents are lost. The timer self-check routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of 2 since the prescaler is a power of 2. If not, the timer is probably not counting correctly. The routine also detects a timer which is not running.

RESET

The MCU can be reset three ways : by initial power-up, by the external reset input ($\overline{\text{RESET}}$) and by an optional internal low-voltage detect circuit. The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger which senses the $\overline{\text{RESET}}$ line logic level. A typical reset Schmitt trigger hysteresis curve is shown in figure 12. The Schmitt trigger provides an internal reset voltage if it senses a logical zero on the $\overline{\text{RESET}}$ pin.

Power-On Reset (POR) - An internal reset is generated upon powerup that allows the internal clock generator to stabilize. A delay of t_{RHL} milliseconds is required before allowing the $\overline{\text{RESET}}$ input to go high. Refer to the power and reset timing diagram of figure 13. Connecting a capacitor to the $\overline{\text{RESET}}$ input (as illustrated in figure 14) typically provides sufficient delay. During poerup, the Schmitt trigger switches on (remove reset) when $\overline{\text{RESET}}$ rises to V_{IRES+}.

Figure 12 : Typical Reset Schmitt Trigger Hysteresis

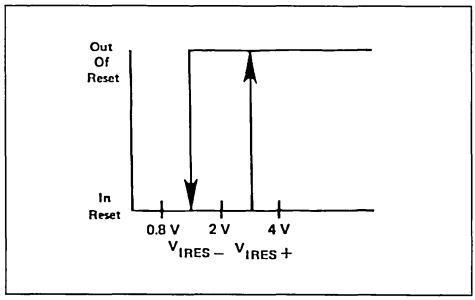


Figure 13 : Power and Reset Timing.

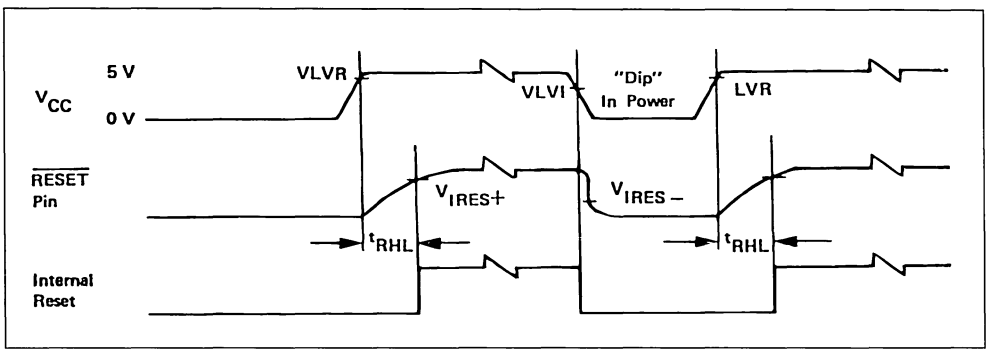
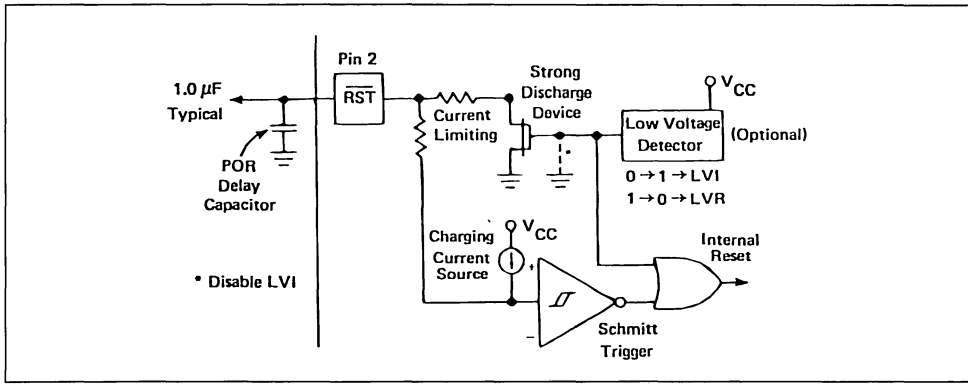


Figure 14 : $\overline{\text{RESET}}$ Configuration.



External Reset Input - The MCU will be reset if a logical zero is applied to the $\overline{\text{RESET}}$ input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES} to provide an internal voltage.

Low-voltage Inhibit (LVI) - The optional low-voltage detection circuit causes a reset of the MCU if the power supply voltage falls below a certain level (V_{LVI}). The only requirement is that V_{CC} remains at or below the V_{LVI} threshold for one t_{cyc} minimum. In typical applications, the V_{CC} bus filter capacitor will eliminate negative-going voltage glitches of less than one t_{cyc}. The output from the low voltage detector is connected directly to the internal reset circuitry. It also forces the $\overline{\text{RESET}}$ pin low via a strong discharge device through a resistor. The internal reset will be removed once the power supply voltage rises above a recovery level (V_{LVR}), at which time a normal power-on-reset occurs.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost trade-offs. The oscillator frequency is internally divided by four to produce the internal system clocks. A manufacturing mask option is used to select crystal or resistor operation.

The different connection methods are shown in figure 15. Crystal specifications and suggested PC board layouts are given in figure 16. A resistor selection graph is given in figure 17.

The crystal oscillator start-up time is a functions of many variables : crystal parameters (especially R_S), oscillator load cpacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator start up, neither the crystal characteristics nor the load capacitances should exceed recommendations.

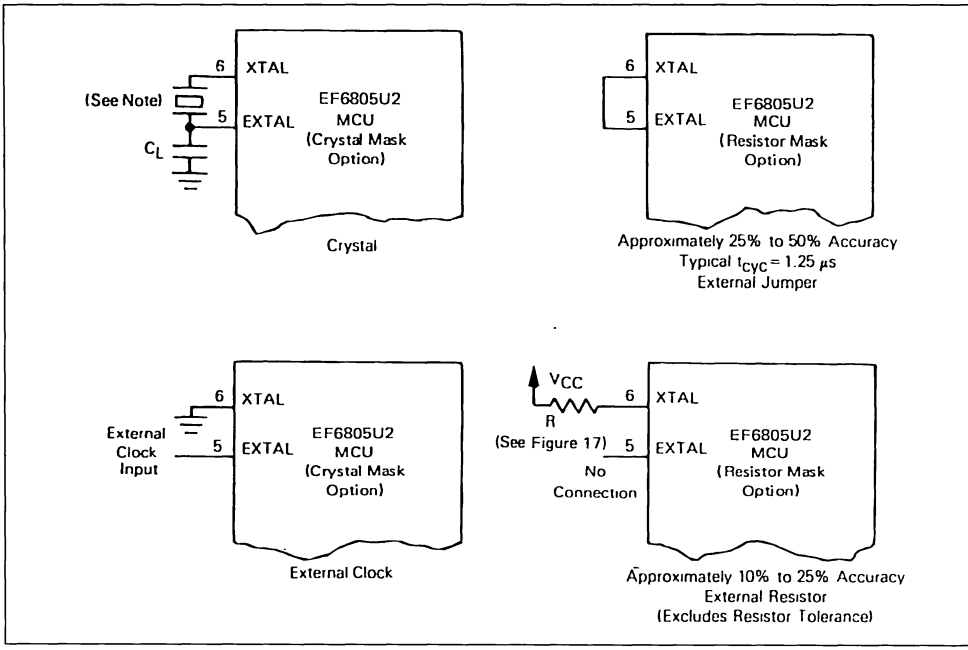
When utilizing the on-board oscillator, the MCU should remain in a reset condition (reset pin voltage below V_{IRES+}) until the oscillator has stabilized at its operating frequency. Several factors are involved in calculating the external reset capacitor required to satisfy this condition : the oscillator start-up voltage, the oscillator stabilization time, the minimum V_{IRES+}, and the reset charging current specification.

Once V_{CC} minimum is reached, the external $\overline{\text{RESET}}$ capacitor will begin to charge at a rate dependent on the capacitor value. The charging current is supplied from V_{CC} through a large resistor, so it appears almost like a constant current source until the reset voltage rises above V_{IRES+}. Therefore, the $\overline{\text{RESET}}$ pin will charge at approximately :

$$(V_{IRES+}) \bullet C_{ext} = I_{RES} \bullet t_{RHL}$$

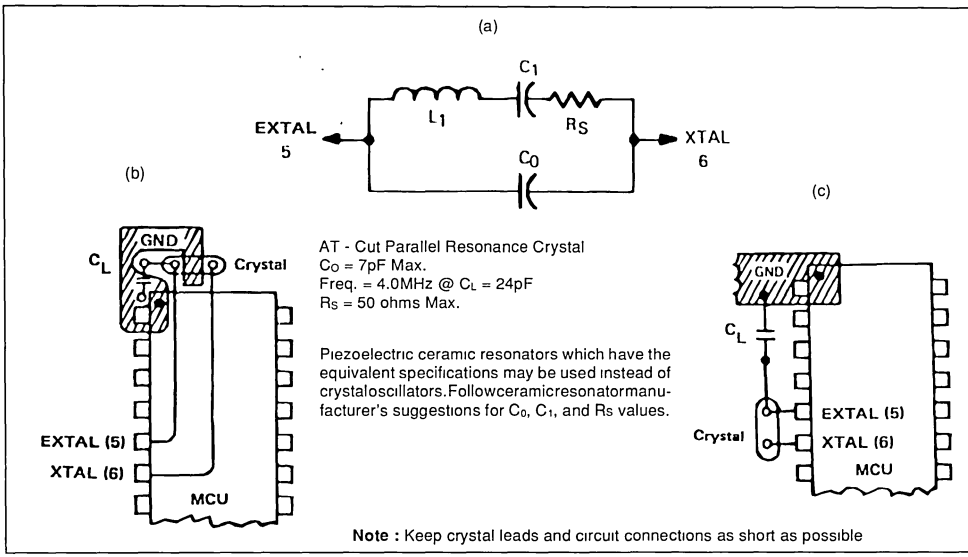
Assuming the external capacitor is initially discharged.

Figure 15 : Clock Generator Options.



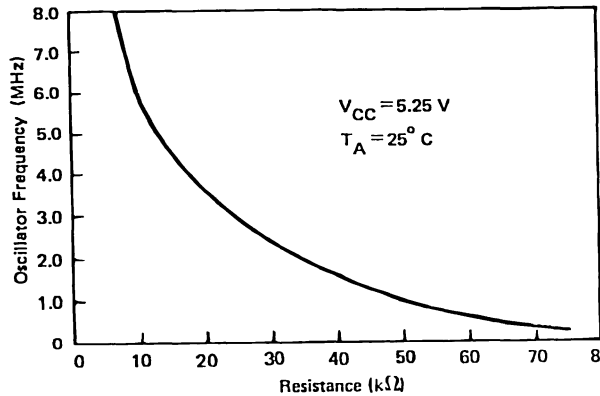
Note : The recommended C_L value with a 4.0MHz crystal is 27pF maximum, including system distributed capacitance There is an internal capacitance of approximately 25pF on the XTAL pin For crystal frequencies other than 4MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2MHz crystal, use approximately 50pF on EXTAL and approximately 25pF on XTAL The exact value depends on the Motional-Arm parameters of the crystal used.

Figure 16 : Crystal Motional ARM Parameters and Suggested PC Board Layout.



Note : Keep crystal leads and circuit connections as short as possible

Figure 17 : Typical Frequency Selection for Resistor (oscillator option).



INTERRUPTS

The microcomputers can be interrupted four different ways : through the external interrupt (INT) input pin, the internal timer interrupt request, the external port D bit 6 (INT2) input pin, or the software interrupt instruction (SWI). When any interrupt occurs : the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (1) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU register, setting the I bit, and vector fetching require a total of 11 t_{cyc} periods for completion. A flowchart of the interrupt sequence is shown in figure 18. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

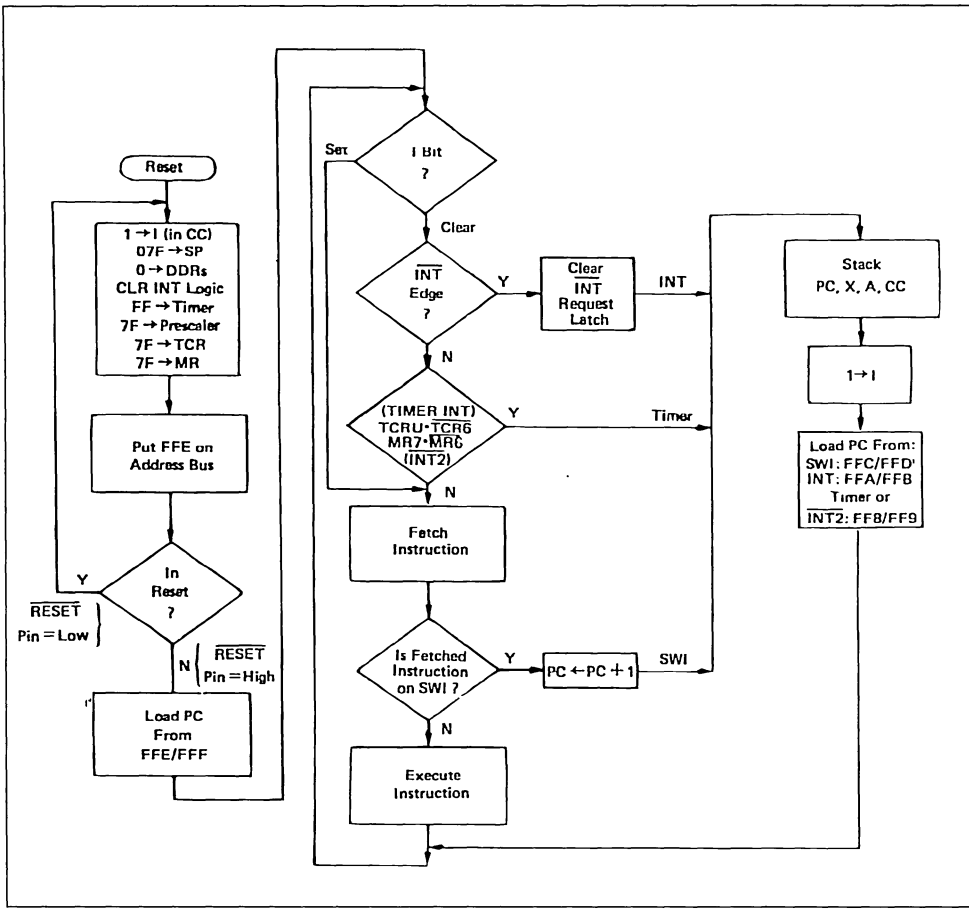
When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing ; otherwise the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

NOTE

The timer and $\overline{INT2}$ interrupts share the same vector address. The interrupt routine must determine the source by examining the interrupt request bits (TCR b7 and MR b7). Both b7 and MR b7 can only be written to zero by software.

Figure 18 : Reset and Interrupt Processing Flowchard.



The external interrupt, $\overline{INT}$ and $\overline{INT2}$, are synchronized and then latched on the falling edge of the input signal. The $\overline{INT2}$ interrupt has an interrupt request bit (bit 7) and a mask bit (bit 6) located in the miscellaneous register (MR). The $\overline{INT2}$ interrupt is inhibited when the mask bit is set. The $\overline{INT2}$ is always read as a digital input on port D. The $\overline{INT2}$ and timer interrupt request bits, if set, cause the MCU to process an interrupt when the condition code I bit is clear.

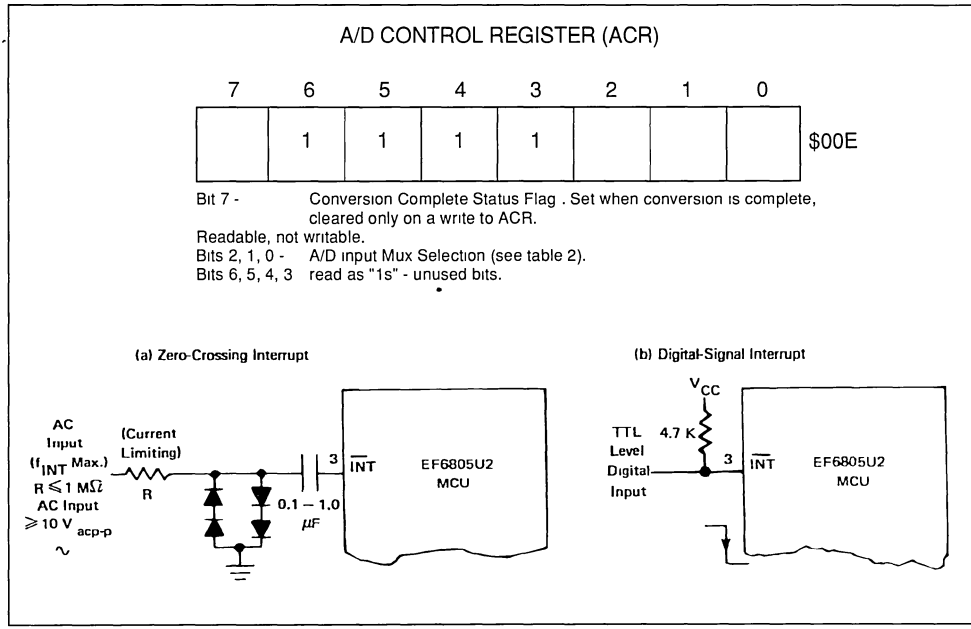
A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock. See figure 19.

NOTE

The $\overline{INT}$ (pin 3) is internally biased at approximately 2.2V due to the internal zero-crossing detection.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. SWIs are usually used as break-points for debugging or as systems calls.

Figure 19 : Typical Interrupt Circuits.

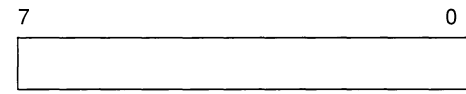


INPUT/OUTPUT CIRCUITRY

There are 32 input/output pins. The $\overline{INT}$ pin may be polled with branch instructions to provide an additional input pin. All pins on ports A, B, and C are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). See below I/O port control registers configuration. The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset all the DDRs are initialized to a logic zero state, placing the ports in the input mode.

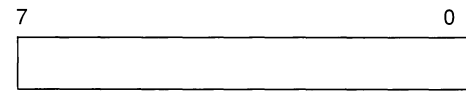
The port output registers are not initialized on reset and should be initialized by software before changing the DDRs from input to output. A read operation on a port programmed as an output will read the contents of the output latch regardless of the logic levels at the output pin, due to output loading. Refer to figure 20.

PORT DATA REGISTER



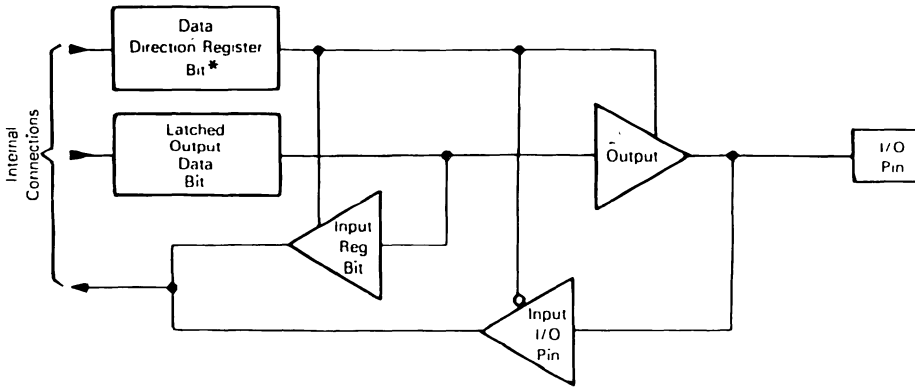
Port A Addr = $000
Port B Addr = $001
Port C Addr = $002
Port D Addr = $003

PORT DATA DIRECTION REGISTER (DDR)



(1) Write only , reads as all "1s"
(2) 1 = Output, 0 = Input Cleared to 0 by Reset
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006

Figure 20 : Typical Port I/O Circuitry.



Data Direction Register Bit	Latched Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

* DDR is a write-only register and reads as all "1s"
** Ports B, and C are three-state ports
Port A has optional internal pullup devices to provide CMOS data drive capability. See Electrical Characteristics tables for complete information

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs (mask option) while port B, C, and D lines are CMOS compatible as inputs. Port D lines are input only ; thus, there is no corresponding DDR. When programmed as outputs, port B is capable of sinking 10 milliamperes and sourcing 1 milliampere on each pin.

The address map (figure 6) gives the addresses of data registers and data direction registers. Figure 21 provides some examples of port connections.

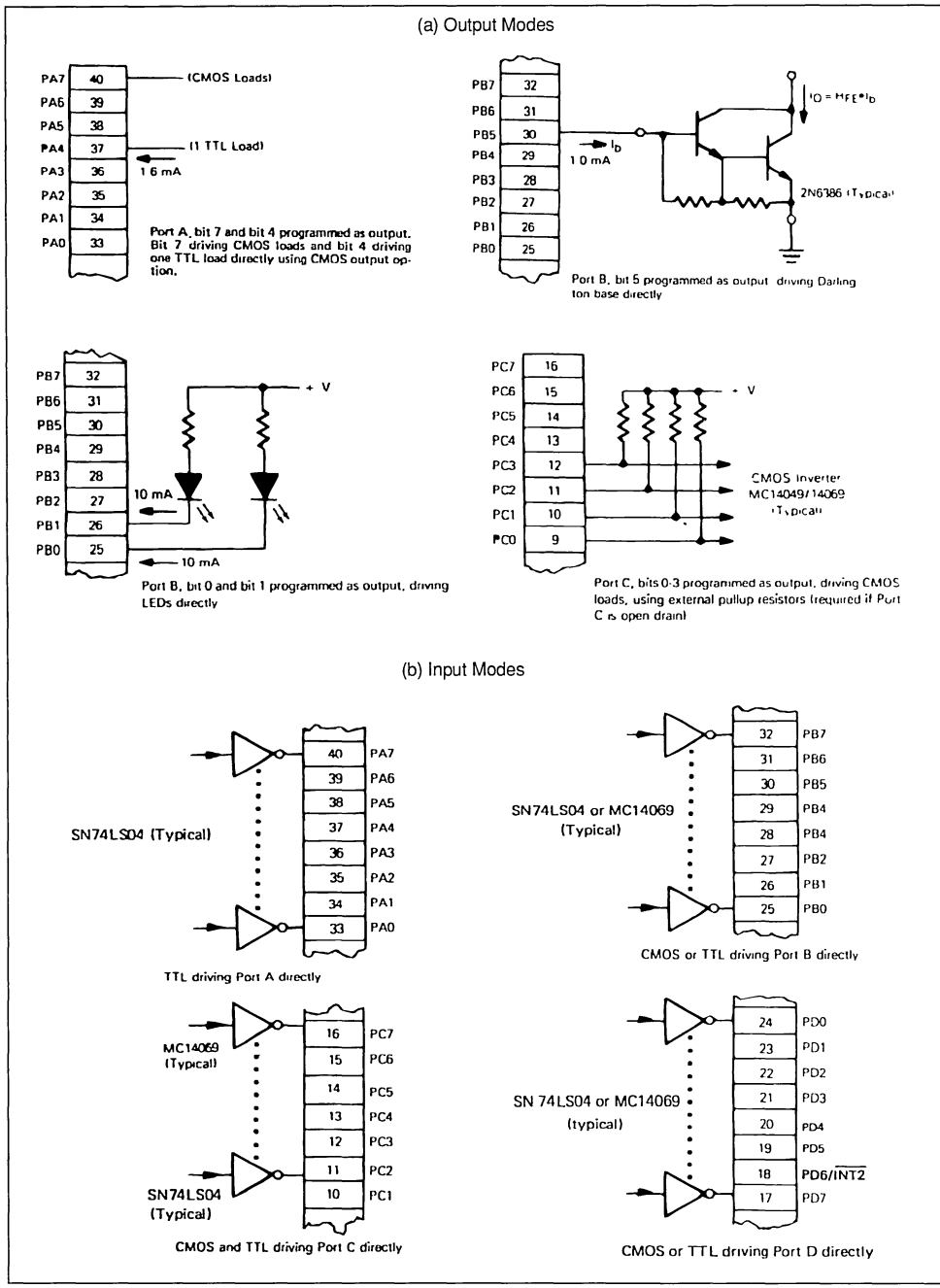
CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined

Since BSET and BCLR are read-modify write in function, they cannot be used or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see figure 20) must always be written. Therefore, any write to a port writes all of its data bits even though the port DDR is set to input. This may be used to initialize the data register and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions, since the data read corresponds to the pin level if the DDR is an input (zero) and corresponds to the latched output data when the DDR is an output (one).

Figure 21 : Typical Port Connections.



BIT MANIPULATION

The EF6805U2 as the ability to set or clear any single RAM or I/O bit (except the data direction registers) with a single instruction (BSET, BCLR) (see Caution below). Any bit in page zero can be tested using the BRSET and BRCLR instructions and the program branches as a result of its state. The carry bit equals the value of the bit references by BRSET or BRCLR. The capability to working with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines.

CAUTION

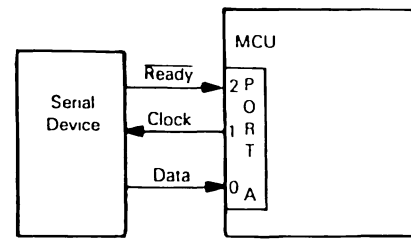
The corresponding data direction registers for ports A, B, and C are write-only registers (locations $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set a data direction register bit (all "unaffected" bits would be set). It is recommended that all data direction register bits in a port be written using a single-store instruction.

The coding examples shown in figure 22 illustrates the usefulness of the bit manipulation and test instruction. Assume that the microcomputer is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line to clock data one bit at a time, least significant bit first out of the device. The microcomputer waits until the data is ready, clocks the external device, picks up the data in the carry flag, clears the clock line, and finally accumulates the data bit in a random-access memory location.

Figure 22 : Bit Manipulation Example.

```
*       BRSET   2, PORTA, *     WAIT FOR READY

        BSET    1, PORTA        CLOCK NEXT BIT IN
        BRCLR   0, PORTA, NEXT  PICKUP BIT IN C-BIT
NEXT    BCLR    1, PORTA        RETURN CLOCK LINE HIGH
        ASR     RAMLOC          MOVE C-BIT INTO RAM
```



ADDRESSING MODES

The EF6805U2 MCU has ten addressing modes available for use by the programmer. They are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the EF6805 Family Users Manual.

The term "effective address" (EA) is used in describing the addressing modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

Immediate - In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct - In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single 2-byte instruction. This address area includes all on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

Extended - In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects ths hortest form of the instruction.

Relative - The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if, and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from - 126 to + 129 from the opcode address. The programmer need not worry about cal-

culating the correct offset if he uses the assembler, since it calculates the proper offset and checks to see if it is within the span of the branch.

Indexed, no Offset - In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

Indexed, 8-bit Offset - In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

Indexed, 16-bit Offset - In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended, the assembler determines the shortest form of indexed addressing.

Bit Set/clear - In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

Bit Test and Branch - The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit which is to be tested and condition (set or clear) is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from -125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code registers.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

Inherent - In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instructions with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The EF6805U2 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

Register/memory Instructions - Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to table 1.

Read-modify-write Instructions - These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register ; see Caution under Input/Output section. The test for negative or zero (TST) instruction is included in the read-modify-write instruction though it does not perform the write. Refer to table 2.

Branch Instructions - The Branch Instructions cause a branch from the program when a certain condition is met. Refer to table 3.

Bit Manipulation Instructions - The instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 4.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

Control Instruction - The control instructions control the MCU operations during program execution. Refer to table 5.

Alphabetical Listing - The complete instruction set is given in alphabetical order in table 6.

Opcode Map - Table 7 is an opcode map for the instruction used on the MCU.

Table 1 : Register/Memory Instructions.

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (no offset)			Indexed (8 bit offset)			Indexed (16 bit offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	AG	2	2	R6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA				B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX				BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OS Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (logical compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP				BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR				BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 2 : Read-Modify-Write Instructions.

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (no offset)			Indexed (8 bit offset)		
Function	Mnem	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2'S complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 3 : Branch Instructions.

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF interrupt mask bit is clear.	BMC	2C	2	4
Branch IFF interrupt mask bit is set.	BMS	2D	2	4
Branch IFF interrupt line is low.	BIL	2E	2	4
Branch IFF interrupt line is high.	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 4 : Bit Manipulation Instructions.

Function	Mnemonic	Addressing Modes					
		Bit Set/clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF bit n is set.	BRSET n (n = 0.....7)				2 • n	3	10
Branch IFF bit n is clear.	BRCLR n (n = 0.....7)				01 + 2 • n	3	10
Set Bit n	BSET n (n = 0.....7)	10 + 2 • n	2	7			
Clear Bit n	BCLR n (n = 0.....7)	11 + 2 • n	2	7			

Table 5 : Control Instructions.

		Inherent		
Function	**Mnemonic**	**Op Code**	**# Bytes**	**#. Cycles**
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-operation	NOP	9D	1	2

Table 6 : Instruction Set.

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			^	●	^	^	^
ADD		X	X	X		X	X	X			^	●	^	^	^
AND		X	X	X		X	X	X			●	●	^	^	●
ASL	X		X			X	X				●	●	^	^	^
ASR	X		X			X	X				●	●	^	^	^
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	^	^	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	^
BRSET										X	●	●	●	●	^
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLL	X										●	●	●	●	0

Condition Code Symbols :

H	Half Carry (from bit 3)	Z	Zero
I	Interrupt Mask	C	Carry/borrow
N	Negative (sign bit)	^	Test and Set if True, Cleared Otherwise
		•	Not Affected

Table 6 : Instruction Set (continued).

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/clear	Bit Test & Branch	H	I	N	Z	C
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	^	^	^
COM	X		X			X	X				●	●	^	^	1
CPX		X	X	X		X	X	X			●	●	^	^	^
DEC	X		X			X	X				●	●	^	^	●
EOR		X	X	X		X	X	X			●	●	^	^	●
INC	X		X			X	X				●	●	^	^	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	^	^	●
LDX		X	X	X		X	X	X			●	●	^	^	●
LSL	X		X			X	X				●	●	^	^	^
LSR	X		X			X	X				●	●	0	^	^
NEQ	X		X			X	X				●	●	^	^	^
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	^	^	●
ROL	X		X			X	X				●	●	^	^	^
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols :

H	Half Carry (from bit 3)	Z	Zero
I	Interrupt Mask	C	Carry/borrow
N	Negative (sign bit)	^	Test and Set if True Cleared Otherwise
		•	Not Affected

EF6805 HMOS FAMILY

Features	EF6805P2	EF6805P6	EF6805R2	EF6805R3	EF6805U2	EF6805U3
Technology	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	40	40	40
On-chip RAM (bytes)	64	64	64	112	64	112
On-chip User ROM (bytes)	1100	1796	2048	3776	2048	3776
External Bus	None	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	24	24	24
Unidirectional I/O Lines	None	None	6 Inputs	6 Inputs	8 Inputs	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, A/D	Timer	Timer
External Interrupt Inputs	1	1	2	2	2	2
STOP and WAIT	No	No	No	No	No	No

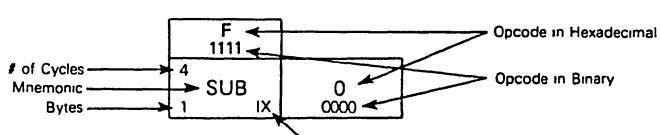
Table 7 : 6805 HMOS Family Instruction Set Opcode Map.

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory					
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DIR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX

Abbreviations for Address Modes

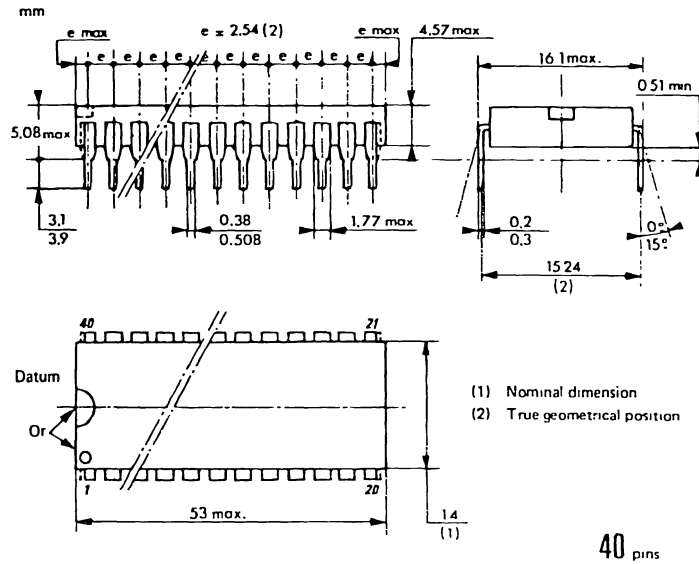
INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

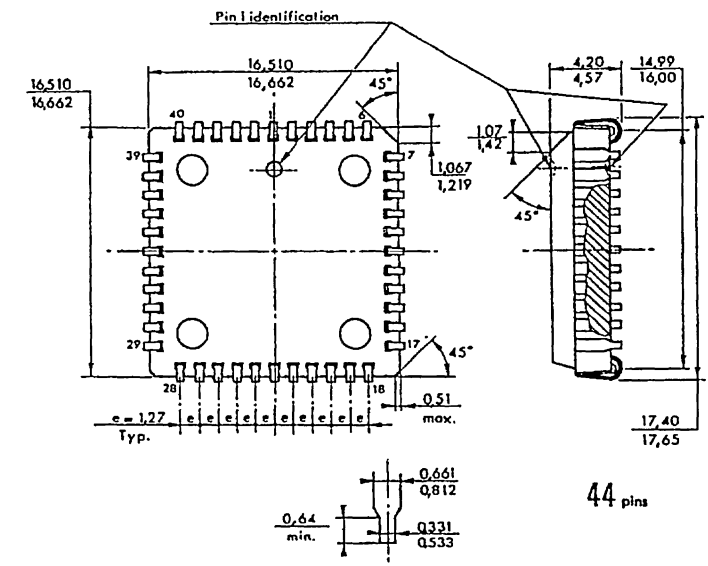


PACKAGE MECHANICAL DATA

CB-182 PLASTIC



CB-521



ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS THOMSON on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local SGS–THOMSON representative or distributor.

EPROMs

Two 2716 or one 2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filled for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS–THOMSON. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, SGS THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by SGS THOMSON. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided. EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6805 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS–THOMSON factory representatives.

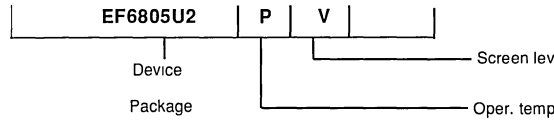
EFDOS is SGS–THOMSON Disk Operating System available on development systems such as DEVICE...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser...

* Requires prior factory approval.

Whenever ordering a custom MCU is required, please contact your local SGS–THOMSON representative or SGS–THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS–THOMSON Microelectronics representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device	Package					Oper. Temp			Screening Level			
	C	J	P	E	FN	L*	V	T	Std	D	G/B	B/B
EF6805U2			●		●	●	●	●	●	●		
Examples : EF6805U2P, EF6805U2FN, EF6805U2PV, EF6805U2FNV												

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC.
Oper. temp. : L* : 0°C to + 70°C, **V :** – 40°C to + 85°C, **T :** – 40°C to + 105°C, *** :** may be omitted
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B

EXORciser is a registered trademark of MOTOROLA Inc.

EF6805 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....

.....

ROM capacity required : | | | | bytes

Temperature range :
- ❑ 0°C / + 70°C
- ❑ – 40°C / + 85°C
- ❑ – 40°C / + 105°C

Package
- ❑ Plastic
- ❑ PLCC

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
- ❑ SGS-THOMSON Microelectronics reference
  .....
- ❑ Special customer data reference*
  .....

Number of interrupt vector : | |

Quality level :
- ❑ STD
- ❑ D
- ❑ Other* (customer's quality specification ref.) :
  .....

Software developped by :
- ❑ SGS-THOMSON Microelectronics application lab.
- ❑ External lab.
- ❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
- ❑ EPROM Reference :
- ❑ EFDOS/MDOS* disk file
  - ❑ 8" floppy
  - ❑ 5" 1/4 floppy
- ❑ Other *

OPTION LIST

-Oscillator input
- ❑ Xtal
- ❑ RC

TTL

- Port A output drive :
- ❑ CMOS and TTL

- Timer clock source
- ❑ Internal ϕ2 clock
- ❑ Timer input pin

- Timer prescaler

❑ 1 ❑ 4 ❑ 16 ❑ 64
❑ 2 ❑ 8 ❑ 32 ❑ 128

- Low voltage inhibit
- ❑ Enabled
- ❑ Disabled

- Port C output drive :
- ❑ TTL

* Requires prior factory approval

Yearly quantity forecast :

- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME DATE : SIGNATURE :

SGS-THOMSON MICROELECTRONICS

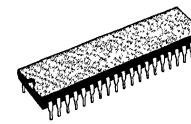
EF6805U3

8-BIT MICROCOMPUTER UNIT

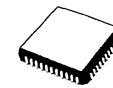
ADVANCE DATA

HARDWARE FEATURES

- 32 TTL/CMOS COMPATIBLE I/O LINES
  - 24 BIDIRECTIONAL (8 lines are LED compatible)
  - 8 INPUT-ONLY
- 3776 BYTES OF USER ROM
- 112 BYTES OF RAM
- SELF-CHECK MODE
- ZERO-CROSSING DETECT/INTERRUPT
- INTERNAL 8-BIT TIMER WITH 7-BIT SOFTWARE PROGRAMMABLE PRESCALER AND CLOCK SOURCE
- 5V SINGLE SUPPLY



P
(Plastic Package)



FN
(PLCC 44)

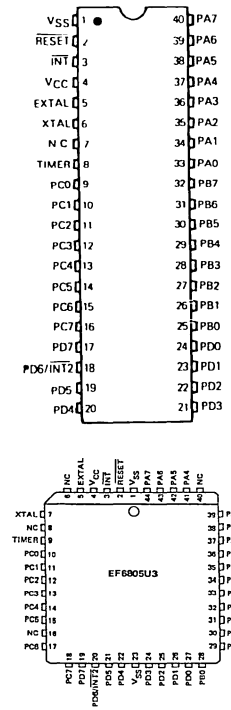
SOFTWARE FEATURES

- 10 POWERFUL ADDRESSING MODES
- BYTE EFFICIENT INSTRUCTION SET WITH TRUE BIT MANIPULATION, BIT TEST, AND BRANCH INSTRUCTIONS
- SINGLE INSTRUCTION MEMORY EXAMINE/CHANGE
- POWERFUL INDEXED ADDRESSING FOR TABLES
- FULL SET OF CONDITIONAL BRANCHES
- MEMORY USABLE AS REGISTER/FLAGS
- COMPLETE DEVELOPMENT SYSTEM SUPPORT ON INICE®

USER SELECTABLE OPTIONS

- 8 BIDIRECTIONAL I/O LINES WITH TTL OR TTL/CMOS INTERFACE OPTION
- 8 BIDIRECTIONAL I/O LINES WITH TTL OR OPEN-DRAIN INTERFACE OPTION
- CRYSTAL OR LOW-COST RESISTOR OSCILLATOR OPTION
- LOW VOLTAGE INHIBIT OPTION
- VECTORED INTERRUPTS : TIMER, SOFTWARE, AND EXTERNAL
- USER CALLABLE SELF-CHECK SUBROUTINES

PIN CONNECTIONS



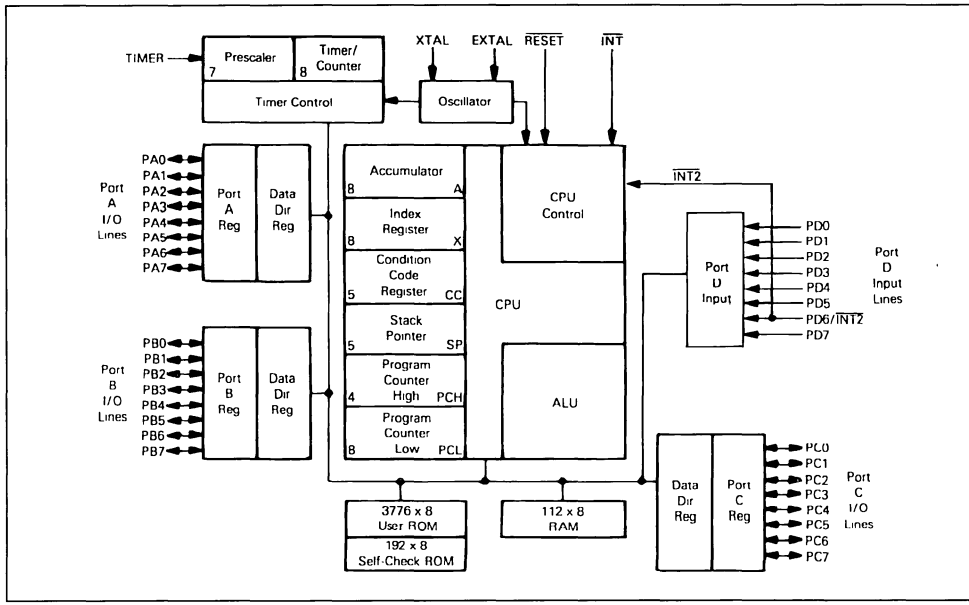
DESCRIPTION

The EF6805U3 Microcomputer Unit (MCU) is a member of the 6805 Family of low-cost single-chip Microcomputers. The 8-bit microcomputer contains

INICE® is SGS-THOMSON development/emulation tool

a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the 6800-based instruction set. A comparison of the key features of several members of the 6805 Family of Microcomputers is shown at the end of this data sheet. The following are some of the hardware and software highlights of the EF6805U3 MCU.

Figure 1 : EF6805U3 HMOS Microcomputer Block Diagram.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{in}	Input Voltage (except TIMER in self-check mode and open-drain inputs)	– 0.3 to + 7.0	V
V_{in}	Input Voltage (open-drain pins, TIMER pin in self-check mode)	– 0.3 to + 15.0	V
T_A	Operating Temperature Range (T_L to T_H) V Suffix T Suffix	0 to + 70 – 40 to + 85 – 40 to + 105	°C
T_{stg}	Storage Temperature Range	– 55 to + 150	°C
T_J	Junction Temperature Plastic Package PLCC	 150 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electrical fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. Reliability of operation is enhanced if unused inputs except EXTAL are tied to an appropriate logic voltage level (e g , either V_{SS} or V_{CC}).

THERMAL DATA

θ_{JA}	Thermal Resistance	Plastic PLCC	50 80	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$T_J = T_A + (P_D \cdot \theta_{JA})$ (1)

Where :

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts - User Determined

For most applications $P_{PORT} << P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is :

$P_D = K + (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives :

$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2$ (3)

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage				V
	$\overline{\text{RESET}}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0		V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5		V_{CC}	
	$\overline{\text{INT}}$ (4.75 ≤ V_{CC} ≤ 5.75)	4.0	*	V_{CC}	
	(V_{CC} < 4.75)	V_{CC} – 0.5	*	V_{CC}	
	All Other (except timer)	2.0		V_{CC}	
V_{IH}	Input High Voltage Timer				V
	Timer Mode	2.0		V_{CC} + 1.0	
	Self-check Mode	9.0	10.0	15.0	
V_{IL}	Input Low Voltage				V
	$\overline{\text{RESET}}$	V_{SS}		0.8	
	$\overline{\text{INT}}$	V_{SS}	*	1.5	
	All Other	V_{SS}		0.8	
V_{IRES+} V_{IRES-}	$\overline{\text{RESET}}$ Hystereris Voltages (see figures 10, 11 and 12)				V
	"Out of Reset"	2.1		4.0	
	"Into Reset"	0.8		2.0	
V_{INT}	$\overline{\text{INT}}$ Zero Crossing Input Voltage, Through a Capacitor	2		4	$V_{ac\ p\text{-}p}$
P_D	Power Dissipation - (no port loading, V_{CC} = 5.75V)				mW
	T_A = 0°C		520	740	
	T_A = – 40°C		580	800	
C_{in}	Input Capacitance				pF
	EXTAL	*	25		
	All Other		10		
V_{LVR}	Low Voltage Recover			4.75	V
V_{LVI}	Low Voltage Inhibit	2.75	3.75	4.70	V
I_{in}	Input Current				µA
	$\overline{\text{TIMER}}$ (V_{in} = 0.4V)			20	
	$\overline{\text{INT}}$ (V_{in} = 2.4V to V_{CC})		20	50	
	EXTAL (V_{in} = 2.4V to V_{CC} - crystal option)			10	
	_____ (V_{in} = 0.4V - crystal option)			– 1600	
	$\overline{\text{RESET}}$ (V_{in} = 0.8V) - External Capacitor Charging Current	– 4.0		– 40	

* Due to internal biasing this input (when unused) floats to approximately 2.2V.

SWITCHING CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Symbol	Parameter	Min.	Typ.	Max.	Unit
f_{osc}	Oscillator Frequency	0.4		4.2	MHz
t_{cyc}	Cycle Time ($4/f_{osc}$)	0.95		10	µs
t_{WL}, t_{WH}	$\overline{INT}$, $\overline{INT2}$, and TIMER Pulse Width (see interrupt section)	t_{cyc} + 250			ns
t_{RWL}	$\overline{RESET}$ Pulse Width	t_{cyc} + 250			ns
f_{INT}	$\overline{INT}$ Zero-crossing Detection Input Frequency	0.03		1	kHz
	External Clock Input Duty Cycle (EXTAL)	40	50	60	%
	Crystal Oscillator Start-up Time*			100	ms

* See figure 16 for typical crystal parameters

PORT ELECTRICAL CHARACTERISTICS

(V_{CC} = + 5.25Vdc ± 0.5Vdc, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

PORT A WITH CMOS DRIVE ENABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage (I_{Load} = 1.6mA)			0.4	V
V_{OH}	Output High Voltage I_{Load} = – 100µA I_{Load} = – 10µA	 2.4 V_{CC} – 1.0			V
V_{IH}	Input High Voltage (I_{Load} = – 300µA max.)	2.0		V_{CC}	V
V_{IL}	Input Low Voltage (I_{Load} = – 500µs max.)	V_{SS}		0.8	V
I_{IH}	High Z State Input Current (V_{in} = 2.0V to V_{CC})			– 300	µA
I_{IL}	High Z State Input Current (V_{in} = 0.4V)			– 500	µA

PORT B

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage I_{Load} = 3.2mA I_{Load} = 10mA (sink)			 0.4 1.0	V
V_{OH}	Output High Voltage I_{Load} = – 200µA	2.4			V
I_{OH}	Darlington Current Drive (source) V_O = 1.5V	– 1.0		– 10	mA
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	High Z State Input Current		< 2	10	µA

PORT C AND PORT A WITH CMOS DRIVE DISABLED

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{OL}	Output Low Voltage I_{Load} = 1.6mA			0.4	V
V_{OH}	Output High Voltage I_{Load} = – 100µA	2.4			V
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{TSI}	High Z State Input Current		< 2	10	µs

PORT C (open-drain option)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage	2.0		13.0	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{LOD}	Input Leakage Current		< 3	15	µA
V_{OL}	Output Low Voltage I_{Load} = 1.6mA			0.4	V

PORT D

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input High Voltage	2.0		V_{CC}	V
V_{IL}	Input Low Voltage	V_{SS}		0.8	V
I_{in}	Input Current		< 1	5	µA

Figure 2 : TTL Equivalent Test Load (port B).

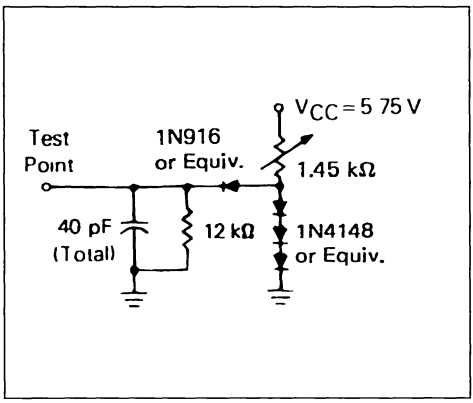


Figure 3 : CMOS Equivalent Test Load (port A).

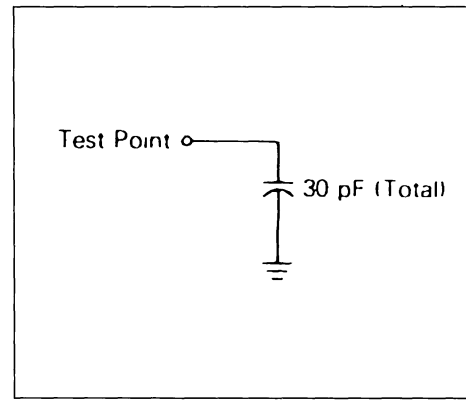


Figure 4 : TTL Equivalent Test Load (port A and C).

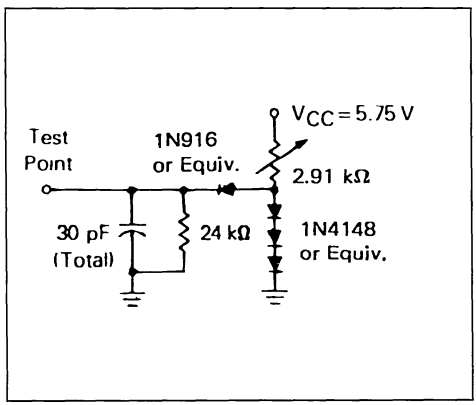
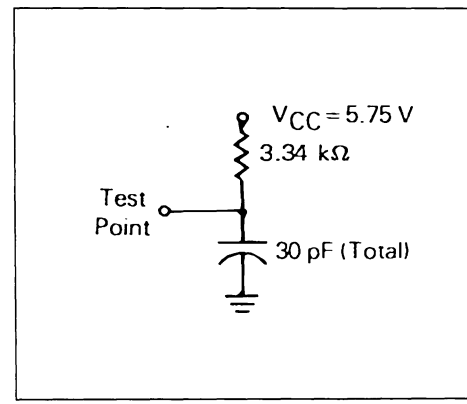


Figure 5 : Open-drain Equivalent Test Load (port C).



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in figure 1, are described in the following paragraphs.

V_{CC} AND V_{SS} - Power is supplied to the MCU using these two pins. V_{CC} is power and V_{SS} is the ground connection.

$\overline{INT}$ - This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts Section for additional information.

XTAL AND EXTAL - These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on user selectable manufacturing mask option, can be connected to these pins to provide a system clock with various degrees of stability/cost tradeoffs. Lead length and stray capacitance on these two pins should be minimized. Refer to Internal Clock Generator Options Section for recommendations about these inputs.

NOTE : Pin 7 in DIL package/pin 8 in PLCC package is connected to internal protection.

TIMER - The pin allows an external input to be used to control the internal timer circuitry and also to initiate the self test program. Refer to Timer Section for additional information about the timer circuitry.

$\overline{RESET}$ - This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. The MCU can be reset by pulling $\overline{RESET}$ low. Refer to Resets Section for additional information.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) - These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction registers (DDRs). Port D is for digital input only and bit 6 may be used for a second interrupt $\overline{INT2}$. Refer to Input/Output Section and Interrupts Section for additional information.

MEMORY - The MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The EF6805U3 MCU has implemented 4090 of these bytes. This consists of : 3776 user ROM bytes, 192 self-check ROM bytes, 112 user RAM bytes, 7 port I/O bytes, 2 timer registers, and a miscellaneous register ; see figure 6 for the Address map. The user ROM has been split into two areas. The main user ROM area is from $080 to $F37. The last 8 user ROM locations at the bottom of memory are for the interrupt vectors.

The MCU reserves the first-16 memory locations for I/O features, of which 10 have been implemented. These locations are used for the ports, the port DDRs, the timer and the $\overline{INT2}$ miscellaneous register, and the 112 RAM bytes, 31 bytes are shared with the stack area. The stack must be used with care when data shares the stack area.

The shared stack area is used during the processing of an interrupt or subroutine calls to save the contents of the CPU state. The register contents are pushed onto the stack in the order shown in figure 7. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first, then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack since the stack pointer increments when it pulls data from the stack. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack ; the remaining CPU registers are not pushed.

Figure 6 : EF6805U3 MCU Address Map.

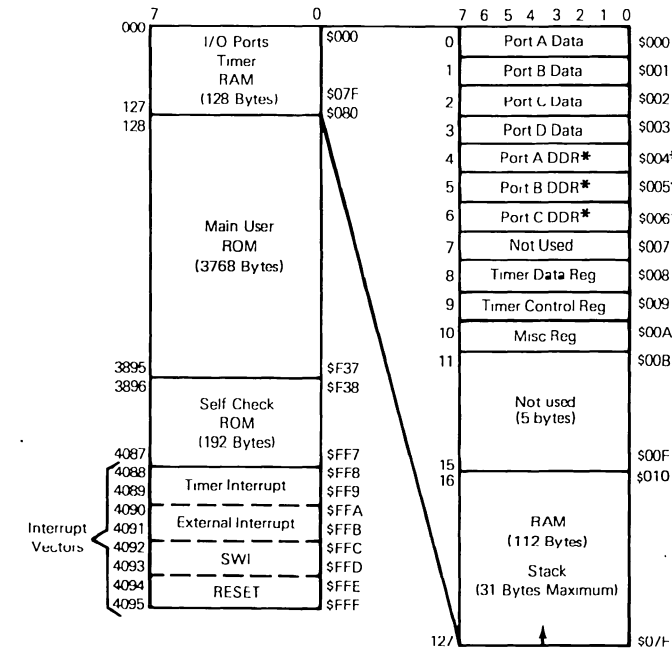
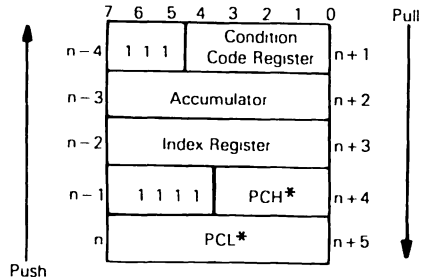


Figure 7 : Interrupt Stacking Order.



*For subroutine calls, only PCH and PCL are stacked

CENTRAL PROCESSING UNIT

The CPU of the EF6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

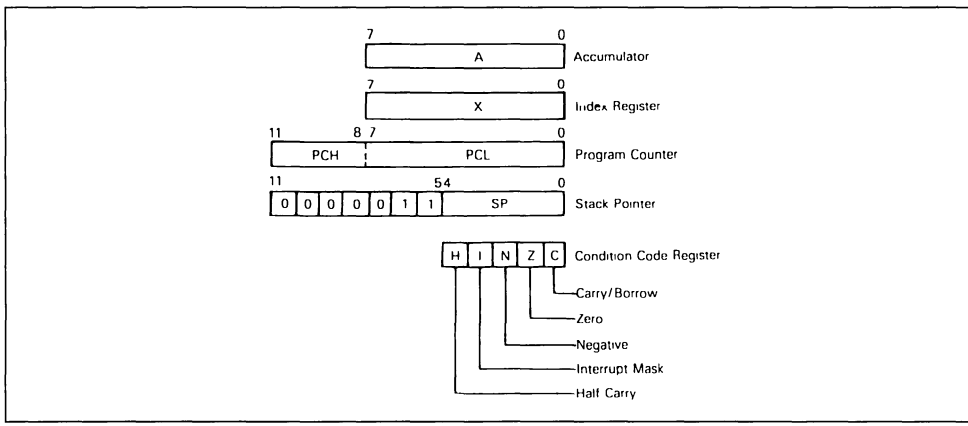
REGISTERS

The 6805 Family CPU has five registers available to the programmer. They are shown in figure 8 and are explained in the following paragraphs.

ACCUMULATOR (A) - The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X) - The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions. The Index Register may also be used as a temporary storage area.

Figure 8 : Programming Model.



PROGRAM COUNTER (PC) - The Program Counter is a 12 bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP) - The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is then pulled from the stack. The seven most significant bits of the stack pointer are permanently set to 0000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

CONDITION CODE REGISTER (CC) - The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

Half Carry (H) - Set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I) - When this bit is set, the timer and external interrupts ($\overline{INT}$ and $\overline{INT2}$) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical "1").

Zero (Z) - When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C) - When set, this bit indicates that a carry or borrow out of the Arithmetic Logic Unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The timer circuitry for the EF6805U3 is shown in figure 10. The timer contains a single 8-bit software programmable counter with a 7-bit software selectable prescaler. The counter may be preset under program control and decrements toward zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the timer control register (TCR), is set. Then if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer interrupt vector from locations $FF8 and $FF9 in order to begin servicing the interrupt.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle and do not change during the read. The timer interrupt request bit remains set until cleared by the software. If a write occurs before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler ; however, its contents are cleared to all zeros by the write operation into TCR when bit 3 of the written data equals one, which allows for truncation-free counting.

The timer input can be configured for three different operating modes, plus a disable mode, depending on the value written to the TCR4 and TCR5 control bits. For further information see figure 9.

Timer Input Mode 1 - If TCR5 and TCR4 are both programmed to a zero, the input to the timer is from an internal clock and the external TIMER input is disabled. The internal clock mode can be used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock.

Timer Input Mode 2 - With TCR5 = 0 and TCR4 = 1, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse widths.

Timer Input Mode 3 - If TCR5 = 1 and TCR4 = 0, then all inputs to the timer are disabled.

Timer Input Mode 4 - If TCR5 = 1 and TCR4 = 1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The external TIMER pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

7	6	5	4	3	2	1	0	
TCR7	TCR6	TCR5	TCR4	TCR3*	TCR2	TCR1	TCR0	$009

* Wnte only (read as zero).

TCR7 - Timer Interrupt Request Bit :
1 - Set when TDR goes to zero, or under program control
0 - Cleared on external Reset, Power-On-Reset, or under Program Control.

TCR6 - Timer Interrupt Mask Bit :
1 - Timer Interrupt masked (disabled) Set on external Reset, Power-On-Reset, or under Program Control
0 - Cleared under Program Control.

TCR5 - External or Internal Clock Source Bit :
1 - External Clock Source. Set on external Reset, Power-On-Reset, or under Program Control
0 - Cleared under Program Control.

TCR4 - External Enable Bit :
1 - Enable external TIMER pin. Set on external Reset, Power-On-Reset, or under Program Control.
0 - Cleared under Program Control.

TCR5	TCR4	Result
0	0	Internal Clock to Timer
0	1	AND of Internal Clock and TIMER Pin to Timer
1	0	Input to timer disabled.
1	1	TIMER Pin to Timer

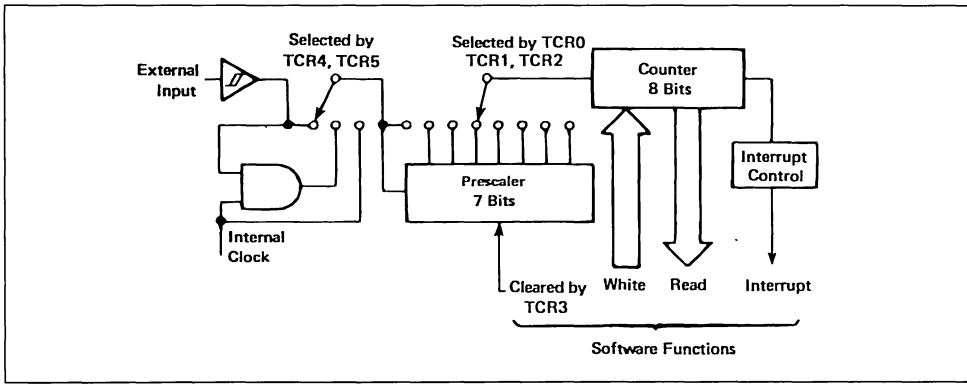
TCR3 - Timer prescaler reset bit : A read of TCR3 always indicates a zero.
1 - Set on external Reset, Power-On-Reset or under Program Control.
0 - Cleared under Program Control.
TCR2, TCR1, and TCR0 - Prescaler address bits :
1 - All set on external Reset, Power-On-Reset or under Program Control.
0 - Cleared under Program Control.

Figure 9 : Timer Control Register (TCR).

TCR2	TCR1	TCR0	Result
0	0	0	+ 1
0	0	1	+ 2
0	1	0	+ 4
0	1	1	+ 8

TCR2	TCR1	TCR0	Result
1	0	0	+ 16
1	0	1	+ 32
1	1	0	+ 64
1	1	1	+ 128

Figure 10 : Timer Block Diagram.



Notes : 1. Prescaler and 8-bit counter are clocked on the falling edge of the internal clock (AS) or external input.
2 Counter is written to during data strobe (DS) and counts down continuously.

SELF-CHECK - The self-check capability of the EF6805U3 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in figure 11 and monitor the output of Port C bit 3 for an oscillation of approximately 7Hz. A 10-volt level (through a 10k resistor) on the timer input, pin 8 and pressing then releasing the RESET button, energizes the ROM-based self-check feature. The self-check program exercices the RAM, ROM, TIMER, interrupts, and I/O ports.

Several of the self-check subroutines can be called by a user program with a JSR or BSR instruction. They are the RAM, ROM. The timer routine may also be called if the timer input is the internal $\phi 2$ clock.

To call those subroutines in customer application, please contact your local SGS-THOMSON sales office in order to obtain the complete description of the self-check program and the entrance/exit conditions.

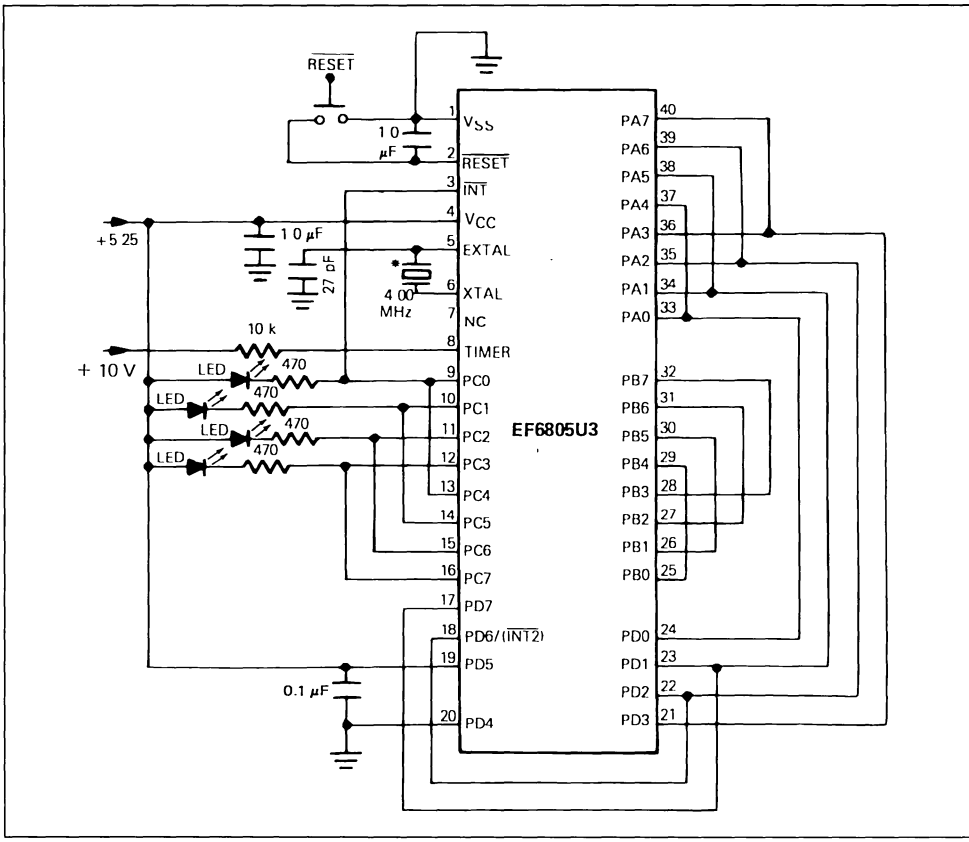
RAM SELF-CHECK SUBROUTINE - The RAM self-check is called at location $F84 and returns with the Z bit clear if any error is detected ; otherwise the Z bit is set. The RAM test causes each byte to count from 0 up to 0 again with a check after each count.

The RAM test must be called with the stack pointer at $07F and A = 0. When run, the test checks every RAM cell except for $07F and $07E which are assumed to contain the return address.

The A and X registers and all RAM locations except $07F and $07E are modified.

ROM CHECKSUM SUBROUTINE - The ROM self-check is called at location $F95. The A register should be cleared before calling the routine. If any error is detected, it returns with the Z bit cleared ; otherwise Z = 1, X = 0 on return, and A is zero if the test passes. RAM location $040 to $043 is overwritten. The checksum is the complement of the execution OR of the contents of the user ROM.

Figure 11 : Self-check Connections.



* This connection depends on clock oscillator user selectable mask option. Use jumper if the RC mask option is selected

LED MEANINGS

PC0	PC1	PC2	PC3	Remarks (1 : LED ON ; 0 : LED OFF)
1	0	1	0	Bad I/O
0	0	1	0	Bad Timer
1	1	0	0	Bad RAM
0	1	0	0	Bad ROM
0	0	0	0	Bad Interrupts or Request Flag
All Flashing				Good Device

Anything else bad device. Bad Port C, etc

TIMER SELF-CHECK SUBROUTINE - The timer self-check is called at location $F6D and returns with the Z bit cleared if any error was found ; otherwise Z = 1.

In order to work correctly as a user subroutine, the internal $\phi 2$ clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock is running and the interrupt mask is not set so the caller must protect from interrupts if necessary.

The A and X register contents are lost. This routine sets the prescaler for divide-by-128 and the timer data register is cleared. The X register is configured to count down the same as the timer data register. The two registers are then compared every 128 cycles until they both count down to zero. Any mismatch during the count down is considered as an error. The A and X registers are cleared on exit from the routine.

RESET

The MCU can be reset three ways : by initial power-up, by the external reset input (RESET) and by an optional internal low-voltage detect circuit. The RESET input consists mainly of a Schmitt trigger which senses the RESET line logic level. A typical reset Schmitt trigger hysteresis curve is shown in figure 12. The Schmitt trigger provides an internal reset voltage if it senses a logical zero on the RESET pin.

POWER-ON RESET (POR) - An internal reset is generated upon powerup that allows the internal clock generator to stabilize. A delay of t_{RHL} milliseconds is required before allowing the RESET input to go high. Refer to the power and reset timing diagram of figure 13. Connecting a capacitor to the RESET input (as illustrated in figure 14) typically provides sufficient delay. During powerup, the Schmitt trigger switches on (removes reset) when RESET rises to V_{IRES+}.

Figure 12 : Typical Reset Schmitt Trigger Hysteresis.

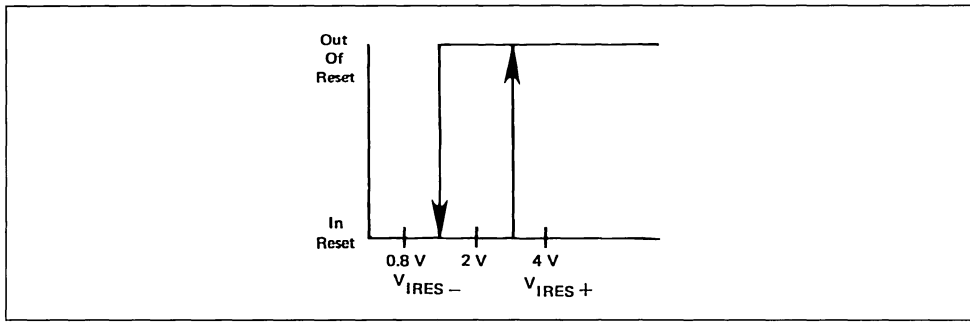


Figure 13 : Power and Reset Timing.

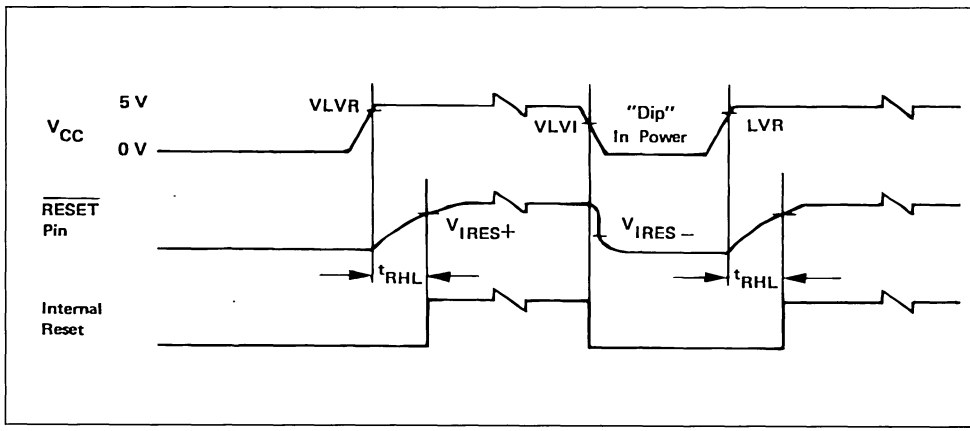
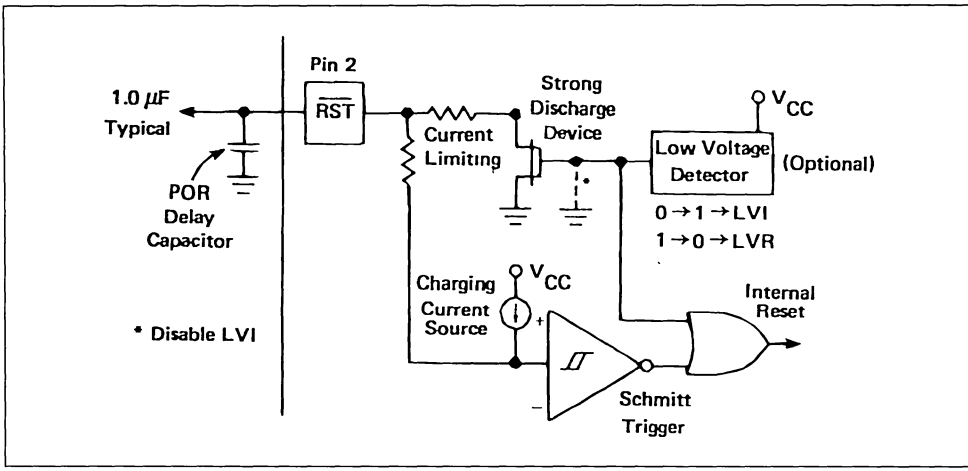


Figure 14 : $\overline{\text{RESET}}$ Configuration.



EXTERNAL RESET INPUT - The MCU will be reset if a logical zero is applied to the $\overline{\text{RESET}}$ input for a period longer than one machine cycle (t_{cyc}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

LOW-VOLTAGE INHIBIT (LVI) - The optional low-voltage detection circuit causes a reset of the MCU if the power supply voltage falls below a certain level (V_{LVI}). The only requirement is that V_{CC} remains at or below the V_{LVI} threshold for one t_{cyc} minimum. In typical applications, the V_{CC} bus filter capacitor will eliminate negative-going voltage glitches of less than one t_{cyc}. The output from the low-voltage detector is connected directly to the internal reset circuitry. It also forces the $\overline{\text{RESET}}$ pin low via a strong discharge device through a resistor. The internal reset will be removed once the power supply voltage rises above a recovery level (V_{LVR}), at which time a normal power-on-reset occurs.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. The oscillator frequency is internally divided by four to produce the internal system clocks. A manufacturing mask option is used to select crystal or resistor operation.

The different connection methods are shown in figure 15. Crystal specifications and suggested PC board layouts are given in figure 16. A resistor selection graph is given in figure 17.

The crystal oscillator start-up time is a function of many variables : crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator start up, neither the crystal characteristics nor the load capacitances should exceed recommendations.

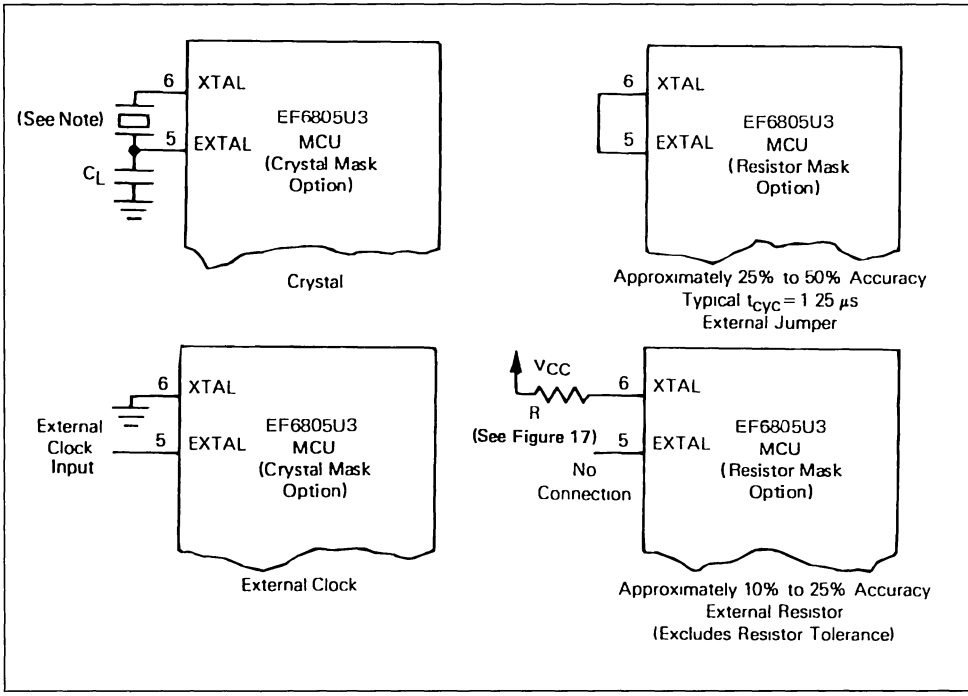
When utilizing the on-board oscillator, the MCU should remain in a reset condition (reset pin voltage below V_{IRES+}) until the oscillator has stabilized at its operating frequency. Several factors are involved in calculating the external reset capacitor required to satisfy this condition : the oscillator start-up voltage, the oscillator stabilization time, the minimum V_{IRES+}, and the reset charging current specification.

Once V_{CC} minimum is reached, the external $\overline{\text{RESET}}$ capacitor will begin to charge at a rate dependent on the capacitor value. The charging current is supplied from V_{CC} through a large resistor, so it appears almost like a constant current source until the reset voltage rises above V_{IRES+}. Therefore, the $\overline{\text{RESET}}$ pin will charge at approximately :

$(V_{IRES+}) \cdot C_{ext} = IRES \cdot t_{RHL}$

Assuming the external capacitor is initially discharged.

Figure 15 : Clock Generator Options.



Note : The recommended C_L value with a 4.0MHz crystal is 27pF, maximum, including system distributed capacitance. There is an internal capacitance of approximately 25pF on the XTAL pin For crystal frequencies other than 4MHz, the total capacitance on each pin should be scaled as the inverse of the frquency ratio. For example, with a 2MHz crystal, use approximately 50pF on EXTAL and approximately 25pF on XTAL The exact value depends on the Motional-Arm parameters of the crystal used.

Figure 16 : Crystal Motional Arm Parameters and Suggested PC Board Layout.

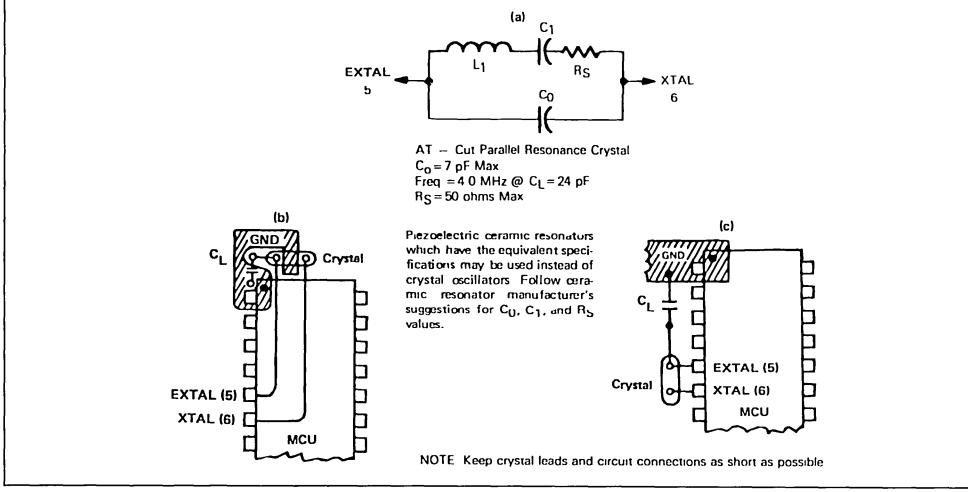
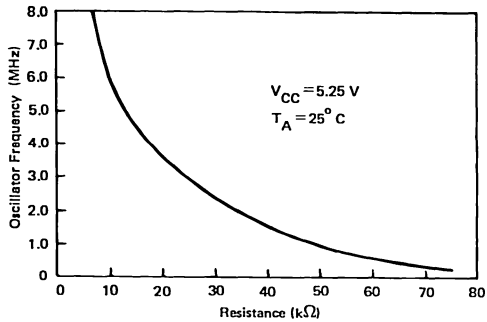


Figure 17 : Typical Frequency Selection for Resistor (oscillator option).



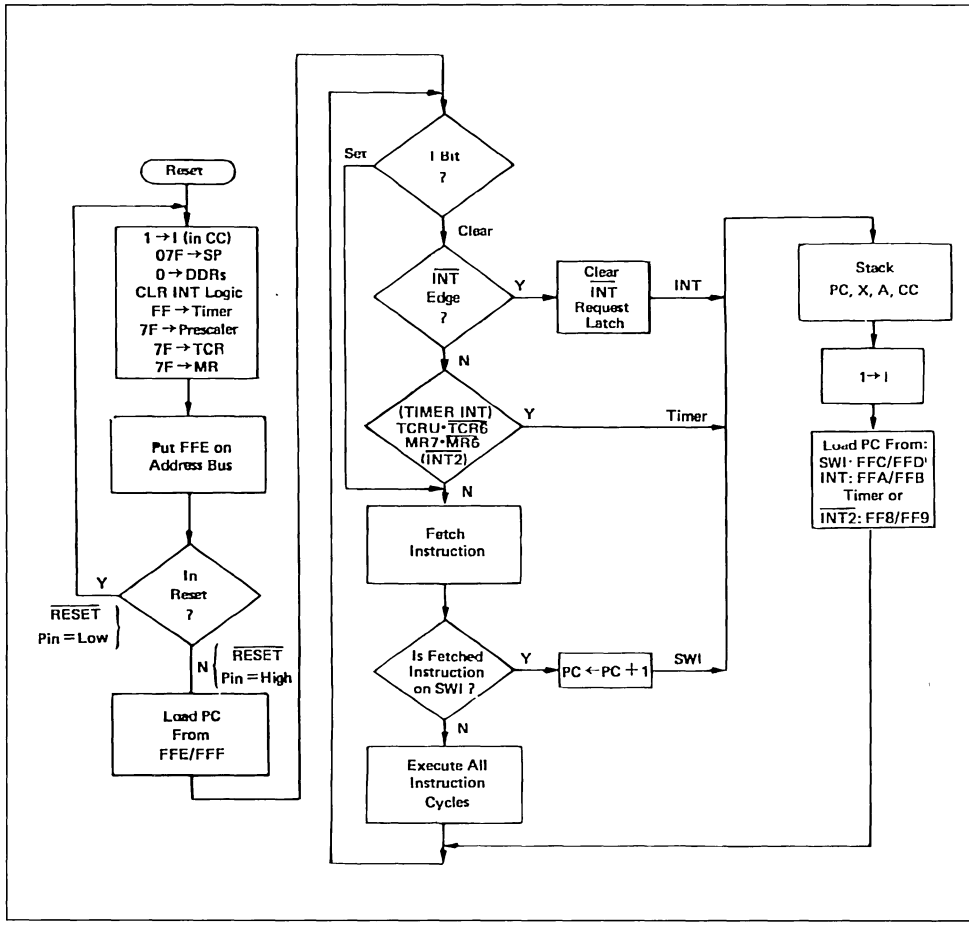
INTERRUPTS

The microcomputers can be interrupted four different ways : through the external interrupt (INT) input pin, the internal timer interrupt request, the external port D bit 6 (INT2) input pin, or the software interrupt instruction (SWI). When any interrupt occurs : the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU register, setting the I bit, and vector fetching require a total of 11 t_{cyc} periods for completion. A flowchart of the interrupt sequence is shown in figure 18. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing ; otherwise the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

Figure 18 : $\overline{\text{RESET}}$ and Interrupt Processing Flowchard.



NOTE

The timer and $\overline{\text{INT2}}$ interrupts share the same vector address. The interrupt routine must determine the source by examining the interrupt request bits (TCR b7 and MR b7). Both TCR b7 and MR b7 can only be written to zero by software.

The external interrupt, $\overline{\text{INT}}$ and $\overline{\text{INT2}}$, are synchronized and then latched on the falling edge of the input signal. The $\overline{\text{INT2}}$ interrupt has an interrupt request bit (bit 7) and a mask bit (bit 6) located in the miscellaneous register (MR). The $\overline{\text{INT2}}$ interrupt is inhibited when the mask bit is set. The $\overline{\text{INT2}}$ is always read as a digital input on port D. The $\overline{\text{INT2}}$ and timer interrupt request bits, if set, cause the MCU to process an interrupt when the condition code I bit is clear.

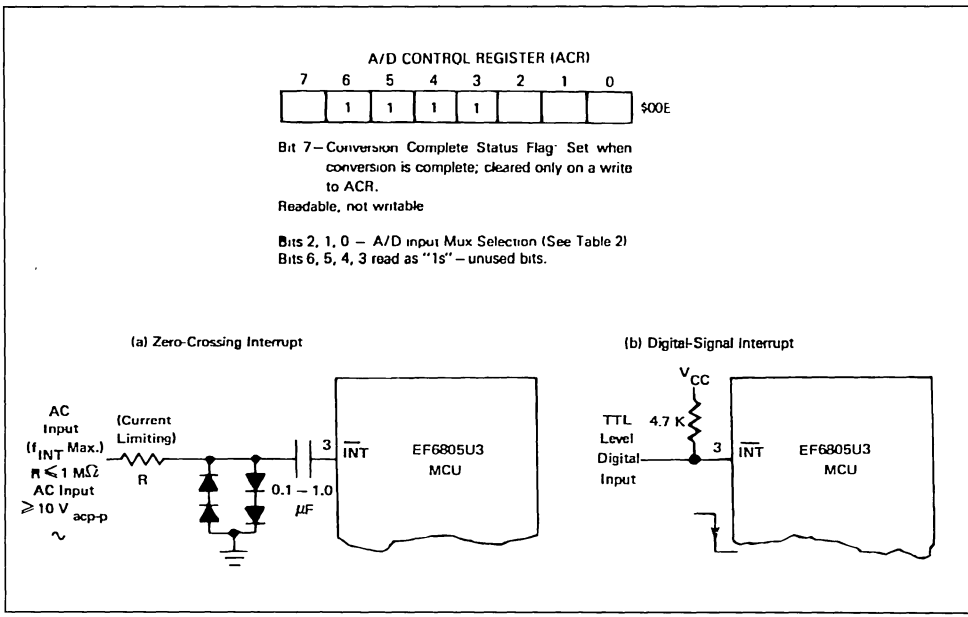
A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock. See figure 19.

NOTE

The $\overline{\text{INT}}$ (pin 3) is internally biased at approximately 2.2V due to the internal zero-crossing detection.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. SWIs are usually used as break-points for debugging or as system calls.

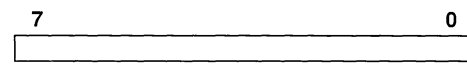
Figure 19 : Typical Interrupt Circuits.



INPUT/OUTPUT CIRCUITRY

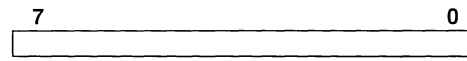
There are 32 input/output pins. The $\overline{INT}$ pin may be polled with branch instructions to provide an additional input pin. All pins on ports A, B, and C are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). See below I/O port control registers configuration. The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic one for output or a logic zero for input. On reset all the DDRs are initialized to a logic zero state, placing the ports in the input mode. The port output registers are not initialized on reset and should be initialized by software before changing the DDRs from input to output. A read operation on a port programmed as an output will read the contents of the output latch regardless of the logic levels at the output pin, due to output loading. Refer to figure 20.

PORT DATA REGISTER



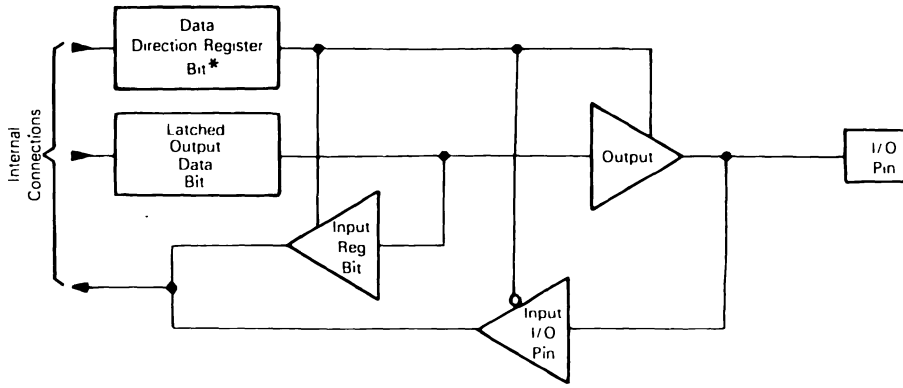
Port A Addr = $000
Port B Addr = $001
Port C Addr = $002
Port D Addr = $003

PORT DATA DIRECTION REGISTER (DDR)



(1) Write only , reads as all "1s"
(2) 1 = Output. 0 = Input Cleared to 0 by Reset
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006

Figure 20 : Typical Port I/O Circuitry.



Data Direction Register Bit	Latched Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

* DDR is a write-only register and reads as all "1s".
** Ports B, and C are three-state ports
Port A has optional internal pullup devices to provide CMOS data drive capability. See Electrical Characteristics tables for complete information

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs (mask option) while port B, C, and D lines are CMOS compatible as inputs. Port D lines are input only ; thus, there is no corresponding DDR. When programmed as outputs, port B is capable of sinking 10 milliamperes and sourcing 1 milliampere on each pin.

The address map (figure 6) gives the addresses of data registers and data direction registers. Figure 21 provides some examples of port connections.

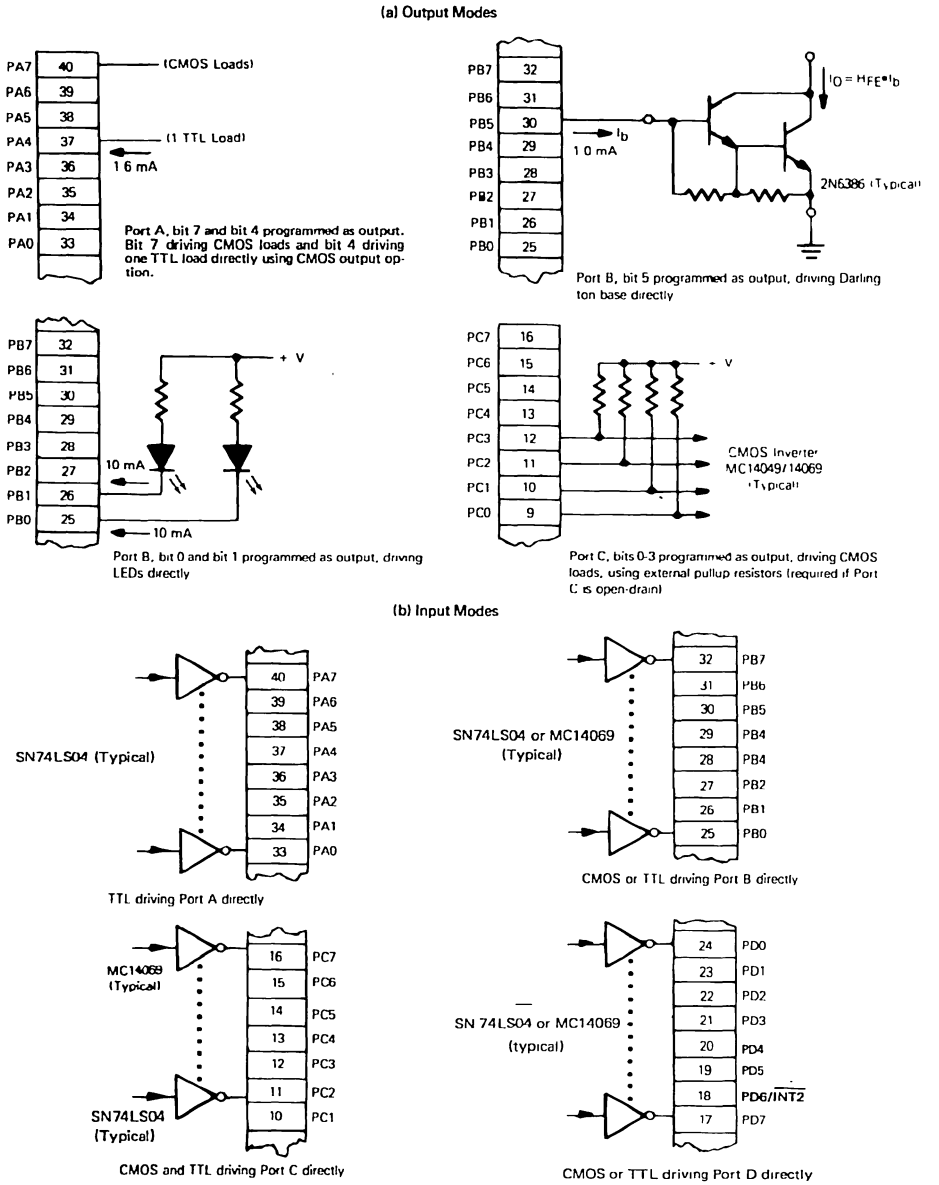
CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at **$004, $005, $006**).

A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see figure 20) must always be written. Therefore, any write to a port writes all of its data bits even though the port DDR is set to input. This may be used to initialize the data register and avoid undefined outputs ; however, care must be exercised when using read-modify-write instructions, since the data read corresponds to the pin level if the DDR is an input (zero) and corresponds to the latched output data when the DDR is an output (one).

Figure 21 : Typical Port Connections.



BIT MANIPULATION

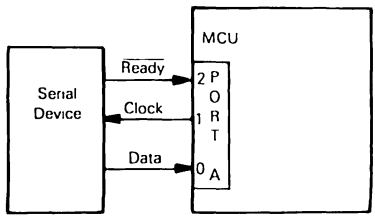
The EF6805U3 has the ability to set or clear any single RAM or I/O bit (except the data direction registers) with a single instruction (BSET, BCLR) (see Caution below). Any bit in page zero can be tested using the BRSET and BRCLR instructions and the program branches as a result of its state. The carry bit equals the value of the bit references by BRSET or BRCLR. The capability to working with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines.

CAUTION

The corresponding data direction registers for ports A, B, and C are write-only registers (locations $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set a data direction register bit (all "unaffected"' bits would be set). It is recommended that all data direction register bits in a port be written using a single-store instruction.

The coding examples shown in figure 22 illustrate the usefulness of the bit manipulation and test instruction. Assume that the microcomputer is to communicate with an external serial device. The external device has a data ready signal, a data output line, and a clock line to clock data one bit at a time, least significant bit first out of the device. The microcomputer waits until the data is ready, clocks the external device, picks up the data in the carry flag, clears the clock line, and finally accumulates the data bit in a random-access memory location.

Figure 22 : Bit Manipulation Example.




```
*       BRSET,  2, PORTA, *     WAIT FOR READY
        BSET    1, PORTA        CLOCK NEXT BIT IN
        BRCLR   0, PORTA, NEXT  PICKUP BIT IN C-BIT
NEXT    BCLR    1, PORTA        RETURN CLOCK LINE HIGH
        ASR     RAMLOC          MOVE C-BIT INTO RAM
```

ADDRESSING MODES

The EF6805U3 MCU has ten addressing modes available for use by the programmer. They are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the EF6805 Family Users Manual.

The term "effective address" (EA) is used in describing the addressing modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE - In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

DIRECT - In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single 2-byte instruction. This address area includes all on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED - In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE - The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if, and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from – 126 to + 129 from the opcode

address. The programmer need not worry about calculating the correct offset if he uses the assembler, since it calculates the proper offset and checks to see if it is within the span of the branch.

INDEXED, NO OFFSET - In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET - In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET - In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended, the assembler determines the shortest form of indexed addressing.

BIT SET/CLEAR - In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

BIT TEST AND BRANCH - The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit which is to be tested and condition (set or clear) is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3–byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from – 125 to + 130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code registers.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

INHERENT - In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instructions with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The EF6805U3 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 usable opcodes. They can be divided into five different types : register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS - Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to table 1.

READ-MODIFY-WRITE INSTRUCTIONS - These instructions read a memory location or a register, modify or test its contents, and write the modified

value back to memory or to the register ; see Caution under Input/Output section. The test for negative or zero (TST) instruction is included in the read-modify-write instruction though it does not perform the write. Refer to table 2.

BRANCH INSTRUCTIONS - The branch instructions cause a branch from the program when a certain condition is met. Refer to table 3.

BIT MANIPULATION INSTRUCTIONS - The instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to table 4.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

CONTROL INSTRUCTION - The control instructions control the MCU operations during program execution. Refer to table 5.

ALPHABETICAL LISTING - The complete instruction set is given in alphabetical order in table 6.

OPCODE MAP - Table 7 is an opcode map for the instruction used on the MCU.

Table 1 : Register/Memory Instructions.

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (no offset)			Indexed (8 bit offset)			Indexed (16 bit offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA				B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX				BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (logical compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP				BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR				BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 2 : Read Modify-Write Instructions.

Function	Mnem	Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (no offset)			Indexed (8 bit offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2'S complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 3 : Branch Instructions.

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF interrupt mask bit is clear.	BMC	2C	2	4
Branch IFF interrupt mask bit is set.	BMS	2D	2	4
Branch IFF interrupt line is low.	BIL	2E	2	4
Branch IFF interrupt line is high.	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 4 : Bit Manipulation Instructions.

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF bit n is set.	BRSET n (n = 0 7)				2 • n	3	10
Branch IFF bit n is clear.	BRCLR n (n = 0 7)				01 + 2 • n	3	10
Set Bit n	BSET n (n = 0 7)	10 + 2 • n	2	7			
Clear Bit n	BCLR n (n = 0 7)	11 + 2 • n	2	7			

Table 5 : Control Instructions.

		Inherent		
Function	**Mnemonic**	**Op Code**	**# Bytes**	**# Cycles**
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No Operation	NOP	9D	1	2

Table 6 : Instruction Set.

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			^	●	^	^	^
ADD		X	X	X		X	X	X			^	●	^	^	^
AND		X	X	X		X	X	X			●	●	^	^	●
ASL	X		X			X	X				●	●	^	^	^
ASR	X		X			X	X				●	●	^	^	^
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	^	^	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	^
BRSET										X	●	●	●	●	^
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLL	X										●	●	●	●	0

Condition Code Symbols :

H Half Carry (from bit 3).
I Interrupt Mask.
N Negative (sign bit).
Z Zero.
C Carry/Borrow.
^ Test and Set if True. Cleared Otherwise.
• Not Affected.

Table 6 : Instruction Set (continued).

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (no offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/clear	Bit Test & Branch	H	I	N	Z	C
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	^	^	^
COM	X		X			X	X				●	●	^	^	1
CPX		X	X	X		X	X	X			●	●	^	^	^
DEC	X		X			X	X				●	●	^	^	●
EOR		X	X	X		X	X	X			●	●	^	^	●
INC	X		X			X	X				●	●	^	^	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	^	^	●
LDX		X	X	X		X	X	X			●	●	^	^	●
LSL	X		X			X	X				●	●	^	^	^
LSR	X		X			X	X				●	●	0	^	^
NEQ	X		X			X	X				●	●	^	^	^
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	^	^	●
ROL	X		X			X	X				●	●	^	^	^
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols :

H Half Carry (from bit 3)
I Interrupt Mask.
N Negative (sign bit).
Z Zero
C Carry/Borrow.
^ Test and Set if True. Cleared Otherwise
• Not Affected.
? Load CC Register from Stack.

EF6805 HMOS FAMILY

Features	EF6805P2	EF6805P6	EF6805R2	EF6805R3	EF6805U2	EF6805U3
Technology	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	40	40	40
On-chip RAM (bytes)	64	64	64	112	64	112
On-chip User ROM (bytes)	1100	1796	2048	3776	2048	3776
External Bus	None	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	24	24	24
Unidirectional I/O Lines	None	None	6 Inputs	6 Inputs	8 Inputs	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, A/D	Timer	Timer
External Interrupt Inputs	1	1	2	2	2	2
STOP and WAIT	No	No	No	No	No	No

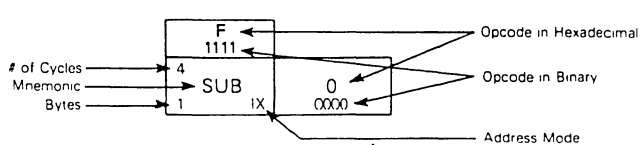
Table 7 : 6805 HMOS Family Instruction Set Opcode Map.

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register / Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DIR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed 2 Byte (16-Bit) Offset

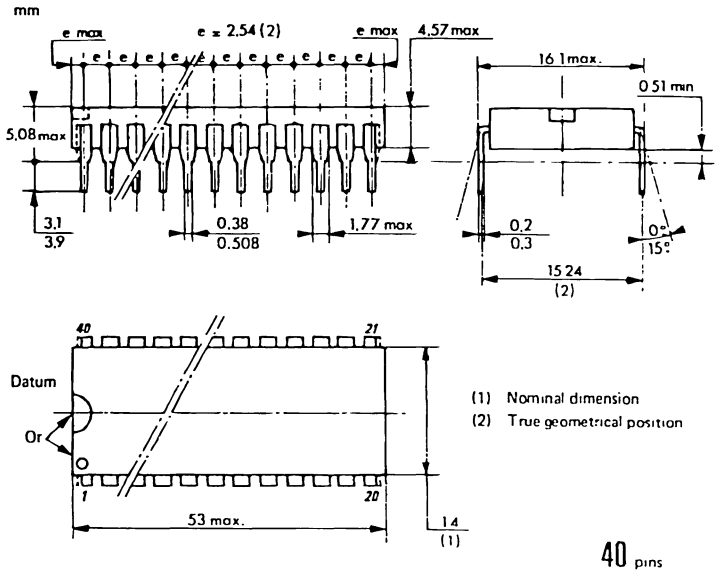
LEGEND



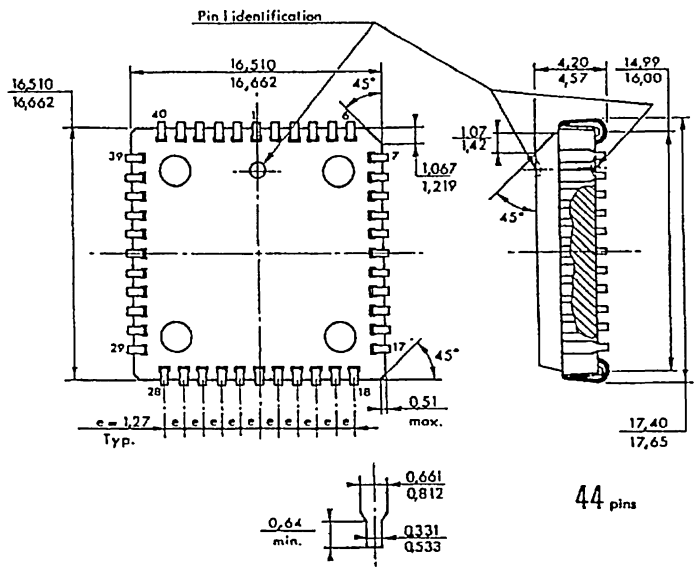
SGS-THOMSON MICROELECTRONICS

PACKAGE MECHANICAL DATA

40 PINS – PLASTIC PACKAGE



44 PINS – PLASTIC PACKAGE



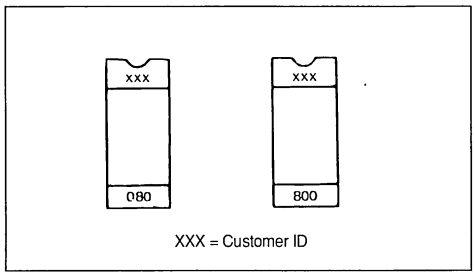
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to SGS-THOMSON on EPROM(s) or an EFDOS/MDOS* disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local SGS-THOMSON representative or distributor.

EPROMs

Two 2716 or one 2732 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below :



After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disk) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed, and returned to SGS-THOMSON. The signed verification form constitutes the contractual agreement for creation of the customer mask. If desired, SGS-THOMSON will program on blank EPROM from the data file used to create the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask but are for the purpose of ROM verification only. For expediency they are usually unmarked, packaged in ceramic, and tested only at room temperature and 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by SGS-THOMSON. Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, EFDOS/MDOS* compatible floppies.

The customer must write the binary file name and company name on the disk with a felt-tip-pen. The minimum EFDOS/MDOS* system files, as well as the absolute binary object file (Filename .LO type of file) from the 6805 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files : filename .LX (DEVICE/EXORciser loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from SGS-THOMSON factory representatives.

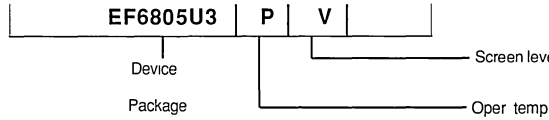
EFDOS is SGS-THOMSON' Disk Operating System available on development systems such as DEVICE...

MDOS* is MOTOROLA's Disk Operating System available on development systems such as EXORciser, ...

* Requires prior factory approval.

Whenever ordering a custom MCU is required,please contact your local SGS-THOMSON representative or SGS-THOMSON distributor and/or complete and send the attached "MCU customer ordering sheet" to your local SGS-THOMSON Microelectronics. representative.

ORDER CODES



The table below horizontally shows all available suffix combinations for package, operating temperature and screening level. Other possibilities on request.

Device	Package					Oper. Temp			Screening Level			
	C	J	P	E	FN	L*	V	T	Std	D	G/B	B/B
EF6805U3			●		●	●	●	●	●	●		
Examples : EF6805U3P, EF6805U3FN, EF6805U3PV, EF6805U3FNV												

Package : C : Ceramic DIL, **J :** Cerdip DIL, **P :** Plastic DIL, **E :** LCCC, **FN :** PLCC
Oper. temp. : L* : 0°C to + 70°C, **V :** – 40°C to + 85°C, **T :** – 40°C to + 105°C, *** :** may be omitted.
Screening level : Std : (no-end suffix), **D :** NFC 96883 level D,
G/B : NFC 96883 level G, **B/B :** NFC 96883 level B and MIL-STD-883C level B

EXORciser is a registered trademark of MOTOROLA Inc

EF6805 FAMILY - MCU CUSTOMER ORDERING SHEET

Commercial reference : | | | | | | | | |

Customer's marking : | | | | | | | | |

Application : .....

.....

Customer name : .....
Company : .....
Address : .....
Phone : .....

Specification reference ;
❑ SGS-THOMSON Microelectronics reference
.....
❑ Special customer data reference*
.....

ROM capacity required : | | | | bytes

Number of interrupt vector : | | |

Temperature range :
❑ 0°C / + 70°C
❑ – 40°C / + 85°C
❑ – 40°C / + 105°C*

Quality level :
❑ STD
❑ D
❑ Other* (customer's quality specification ref.) :
.....

Package
❑ Plastic
❑ PLCC

Software developed by :
❑ SGS-THOMSON Microelectronics application lab.
❑ External lab.
❑ Customer

PATTERN MEDIA (a listing may be supplied in addition for checking purpose) :
❑ EPROM Reference :
❑ EFDOS/MDOS* disk file
  ❑ 8" floppy
  ❑ 5" 1/4 floppy
❑ Other *

OPTION LIST

-Oscillator input
❑ Xtal
❑ RC

- Low voltage inhibit
❑ Enabled
❑ Disabled

- Port A output drive :
❑ CMOS and TTL
❑ TTL only

- Port C output drive :
❑ TTL
❑ Open drain

* Requires prior factory approval

Yearly quantity forecast :

- start of production date :
- for a shipment period of :

CUSTOMER CONTACT NAME DATE : SIGNATURE :

DEVELOPMENT TOOLS

SGS-THOMSON MICROELECTRONICS

TSTIN48

DEVELOPMENT AND EMULATION TOOL

TSTIN48 is a support for :

- EF6805, EF6804 and TS68HC04 MCU Families

SOFTWARE DEVELOPMENT ON IBM-PC

DEVELOPMENT WITH TSTIN48

TSTIN48 is a powerful, lowcost and user friendly stand alone development tool, fully dedicated to support the complete range of 4 and 8 bit MCU's manufactured by SGS-THOMSON Microelectronics.

TSTIN48 offers 2 RS 232 C Serial communication lines :

– one to interface to a host computer IBM-PC which provides functions of human interface and software development.

– the other is used for auxiliary systems such as a line printer. Eprom programmer or centralized mainframe.

The whole development station comprises the host computer IBM-PC, TSTIN48 and a probe dedicated for MCU family.

SOFTWARE DEVELOPMENT ON IBM-PC

The choice of a standard personnal computer for Software purposes (MS DOS operating system) and for hardware (IBM-PC bus) is an important key to obtaining a full evolutive and open system.

It offers access to a very large Software library.

Full screen editor : word, wordstar... (2)

Full range of development software

– MPU and MCU cross assembler

– linking editor.

(2) Software available in library for IBM-PC.

TSTIN48 INTERACTIVE SOFTWARE

Software supplied to control TSTIN48 offers the ability to use the advantages of the powerful architecture hardware.

- Meaningful mnemonics
- Wide choice for characterising values (Decimal, Hexa, ASCII)
- Aids to the operator when in difficulty
- Full range of debugging aides
  - On line assembler/disassembler
  - 8 hardware breakpoints
  - 8 software breakpoints
  - Logic analyser triggered breakpoints
  - Execution program step by step in RAM and ROM area
  - Echo on line printer
  - Load and save program through RS 232 C connection
  - Memory check
  - Operator keyboard entry request can be chained in a command file
  - Self test of the entire system TSTIN48 + emulator probe at power up
  - Possibility to temporarily leave control of emulation session to access DOS routines, through the emulation session keeps acting.

TSTMU 94

EMULATION WITH TSTIN48 (*)

An advanced architecture for total transparency and dynamic emulation.

Thanks to the dual processor architecture (System bus and emulation bus), the emulated MPU or MCU operates independently of the other system processors.

Therefore, it is possible during execution, to list or modify emulator probe control parameters.

REALTIME EMULATION

With the transparency feature, the emulated processor operates at 100% of the maximum speed specified by the manufacturer.

(*) Each emulator probe is delivered with an interactive software specific to each MPU or MCU family.

OPEN ENDED

- Switching from one MPU or MCU to another by changing only the emulator probe
- Program execution without existing final application
- Shared overlay memory (16 Kbytes) in blocks of 256 bytes
- Write protect facility for individual blocks
- 8 hardware and 8 software breakpoints over entire emulation space.

LOGIC ANALYSER

The logic analyser operates with 1024 words of 32 bits (16 address, 8 data bits, 8 available to user).

With sampling and trigger or recognition of maskable words. Five operating modes are available.

USER'S SELECTION GUIDE

Family	MCU	Development System	Emulator Chip	Emulator Board	Interactive Software*	Emulator Probe
EF6805	EF6805P2	TSTIN48	MC68705P3		TSR6805	TSTMUPS
			MC68705P3			
	EF6805U2		MC68705U3			
	EF6805R2		MC6805R2			
	EF6805U3		MC68705U3			
	EF6805R3		MC68705R3			
EF6804	EF6804P2		MC68704P3		TSR6804	EFTMUP4
	EF68HC04P3			TSTEV04		

* Cross assembler included.

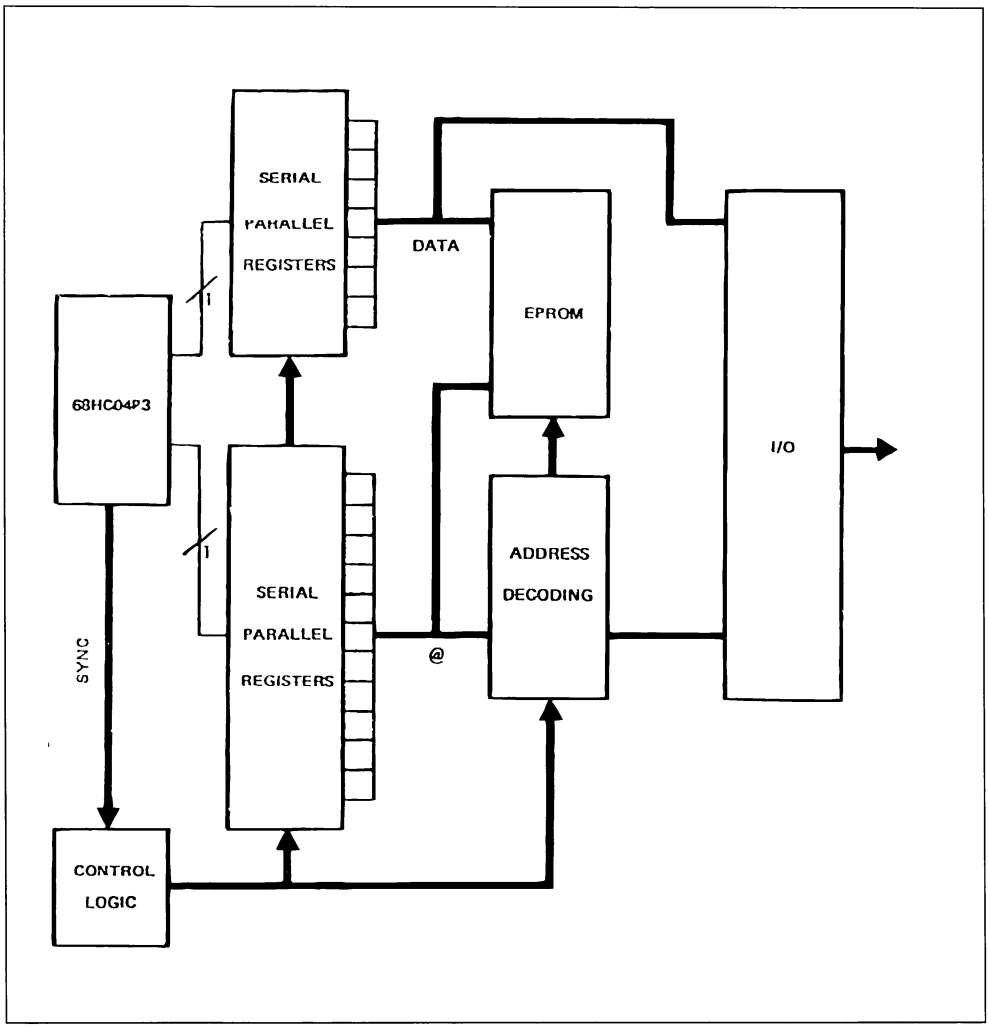
PIGGY-BACK EMULATION TOOL

The EMU-04 is a printed board module, built around EF68HC04P3, and a standard EPROM.

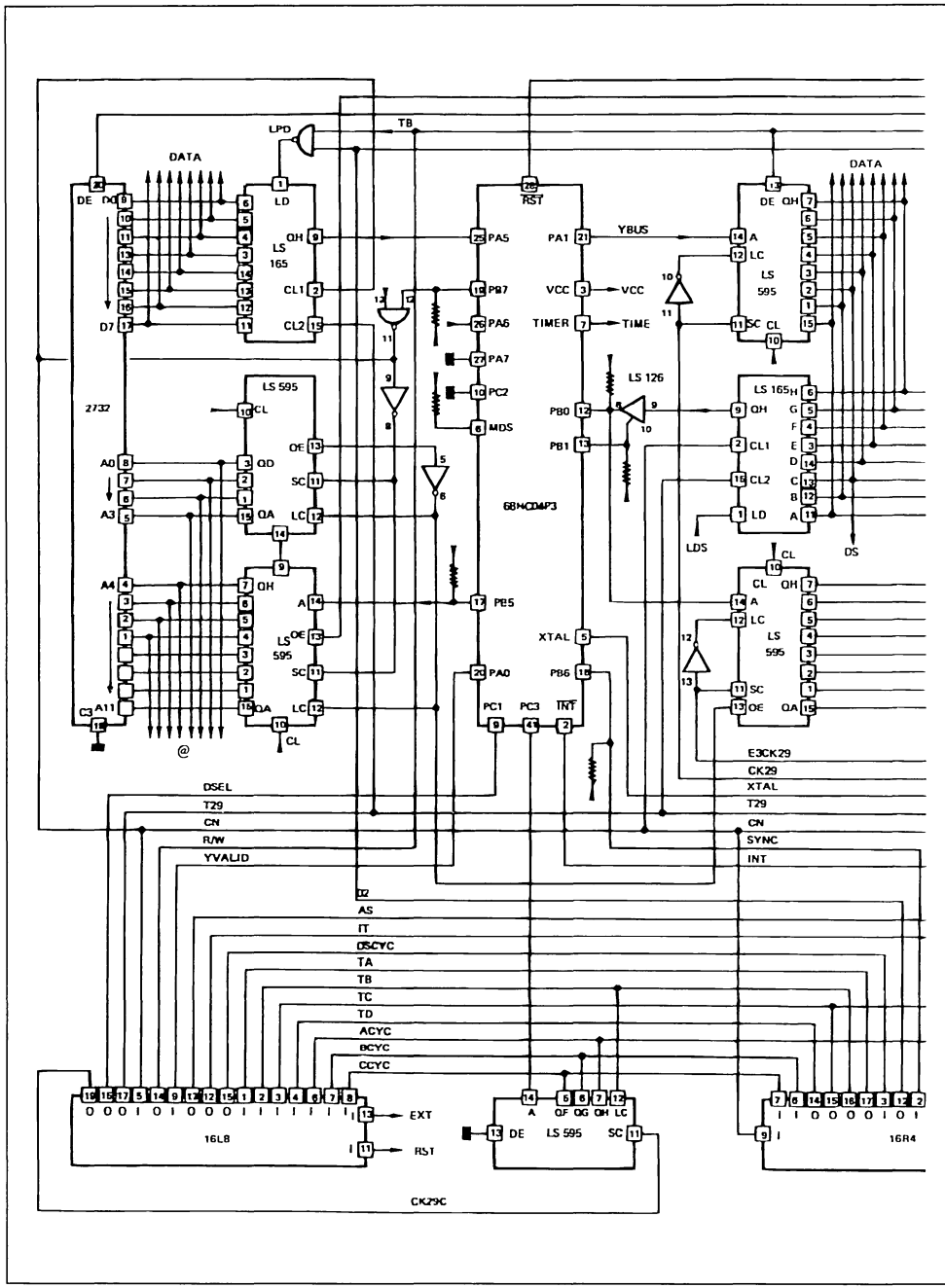
It fully emulates de EF68HC04P3 in the final applications, with the customer program stored in the board EPROM.

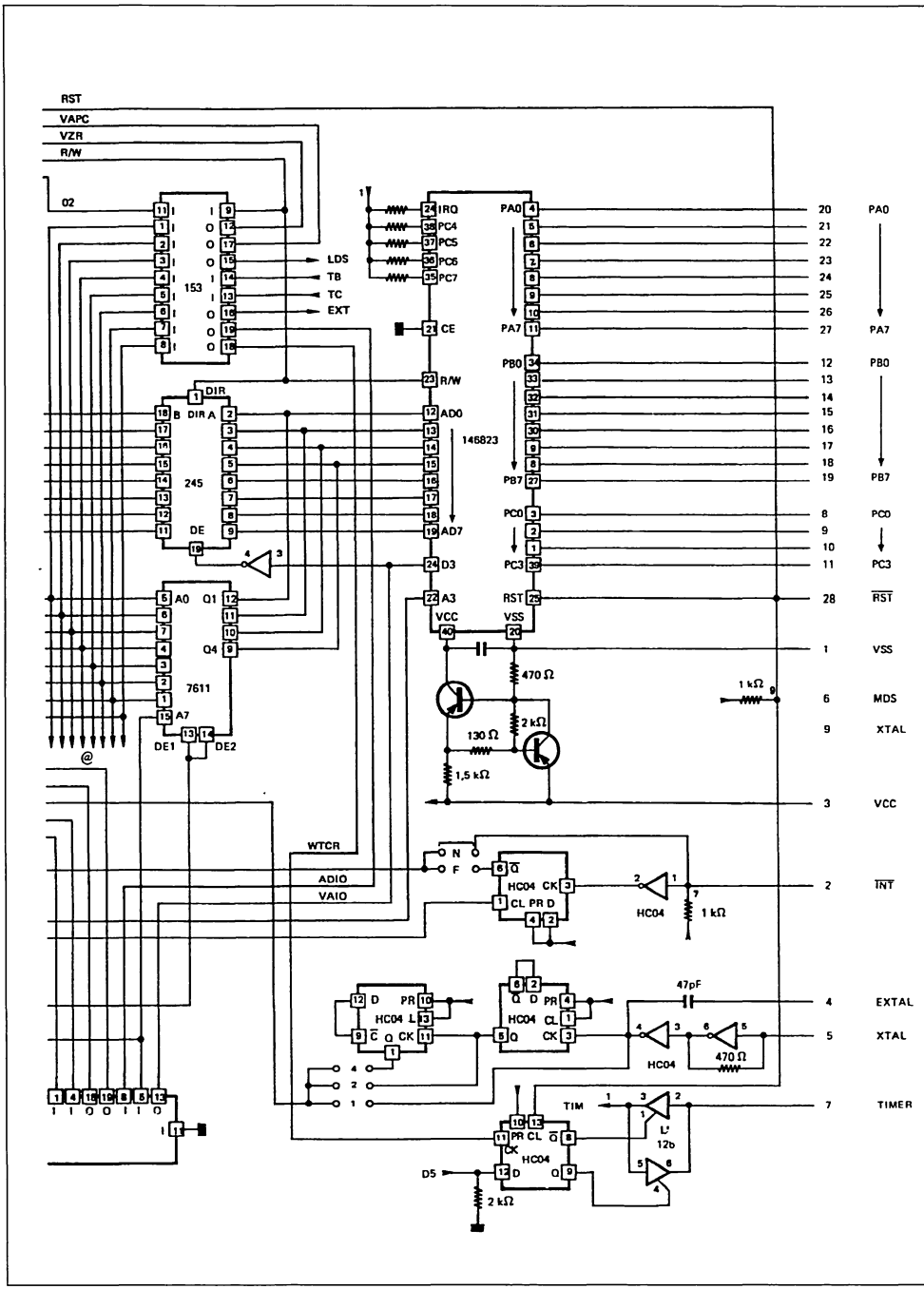
The EMU-04 includes MICRO, RAM, STACK, I/O parts EPROM Data and Program Space.

The following block diagram describes it.



BLOCK DIAGRAM




RST
VAPC
VZR
R/W
02
153
LDS
TB
TC
EXT
DIR
245
DE
A0
Q1
Q4
7611
A7
DE1
DE2
@
IRQ
PC4
PC5
PC6
PC7
CE
R/W
AD0
146823
AD7
D3
A3
VCC
VSS
PA0
PA7
PB0
PB7
PC0
PC3
RST
470 Ω
130 Ω
2 kΩ
1,5 kΩ
1 kΩ
WTCR
ADIO
VAIO
HC04
47pF
TIM
D5
12b
20 PA0
21
22
23
24
25
26
27 PA7
12 PB0
13
14
15
16
17
18
19 PB7
8 PC0
9
10
11 PC3
28 RST
1 VSS
6 MDS
9 XTAL
3 VCC
2 INT
4 EXTAL
5 XTAL
7 TIMER

SOFTWARE

Instructions STOP & WAIT correctly work on piggy-back. Due to PAL & LS components of P.C. board, supply current does not meet Micro's spec. It stands around 800mA as it does in RUN mode.

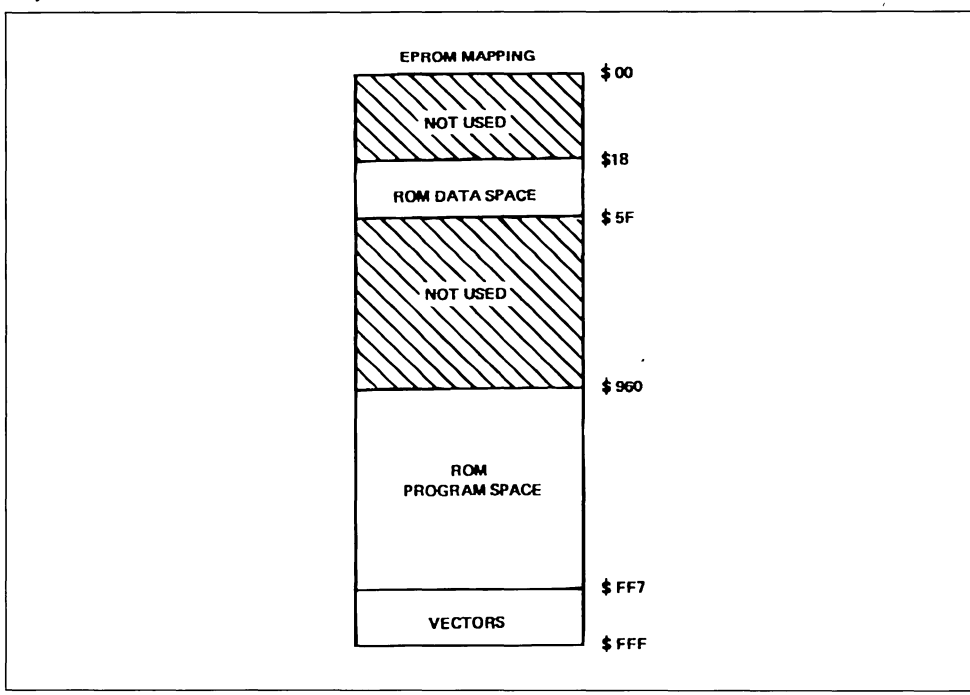
2732 EPROM have been chosen (access time <X300nS) but 2764 or 27128 ones may be used, taking into account that :

- upper address bit must be correctly setted
- program codes must stand in the lower 4Kbytes space
- one have to wire a socket adaptor between EPROM & piggy-back EPROM socket.

DESCRIPTION

All EF68HC04P3 fonctions are restored. Thus application is emulated in real time and user program may be modified.

The DATA space and PROGRAM space are in EPROM space. ADDRESS $00 to $17 and $60 to $95F are unknown.



CLOCK

Instead of using EF68HC04P3 one can plug the piggy-back and use application clock, either quartz (pin EXTAL-XTAL) or CMOS levels (pin XTAL).

SELECTABLE OPTIONS

As EF68HC04P3 device you can choose maskable option on piggy-back board :

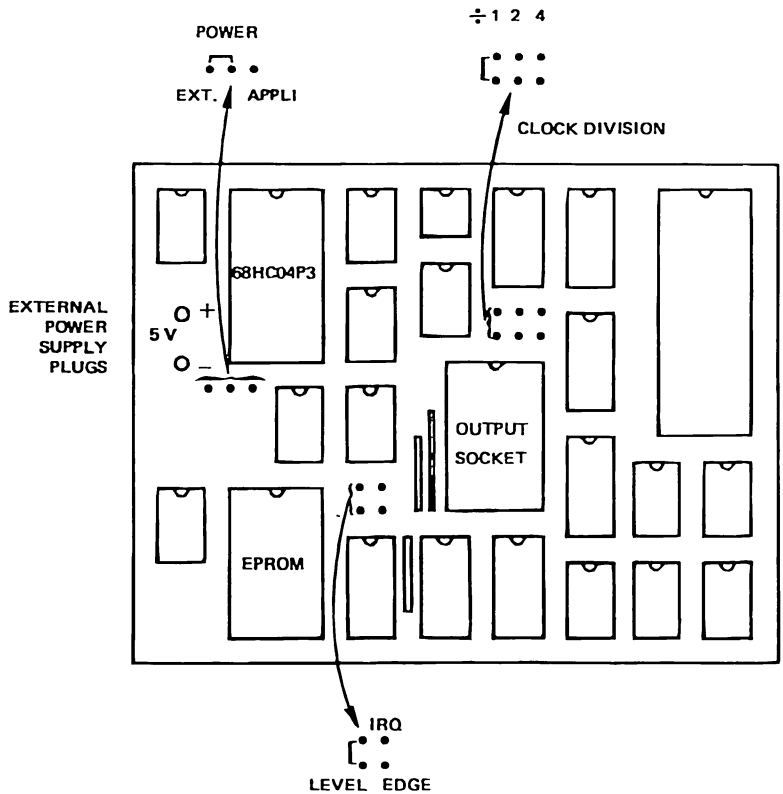
- CLOCK : division by 1, 2, 4
- IRQ : level or negative going edge.

Two ways are available to supply the piggy-back :

- Application supply (pin 3 of EF68CH02P3 socket)
- External power supply through DC supply plugs on piggy-back board.

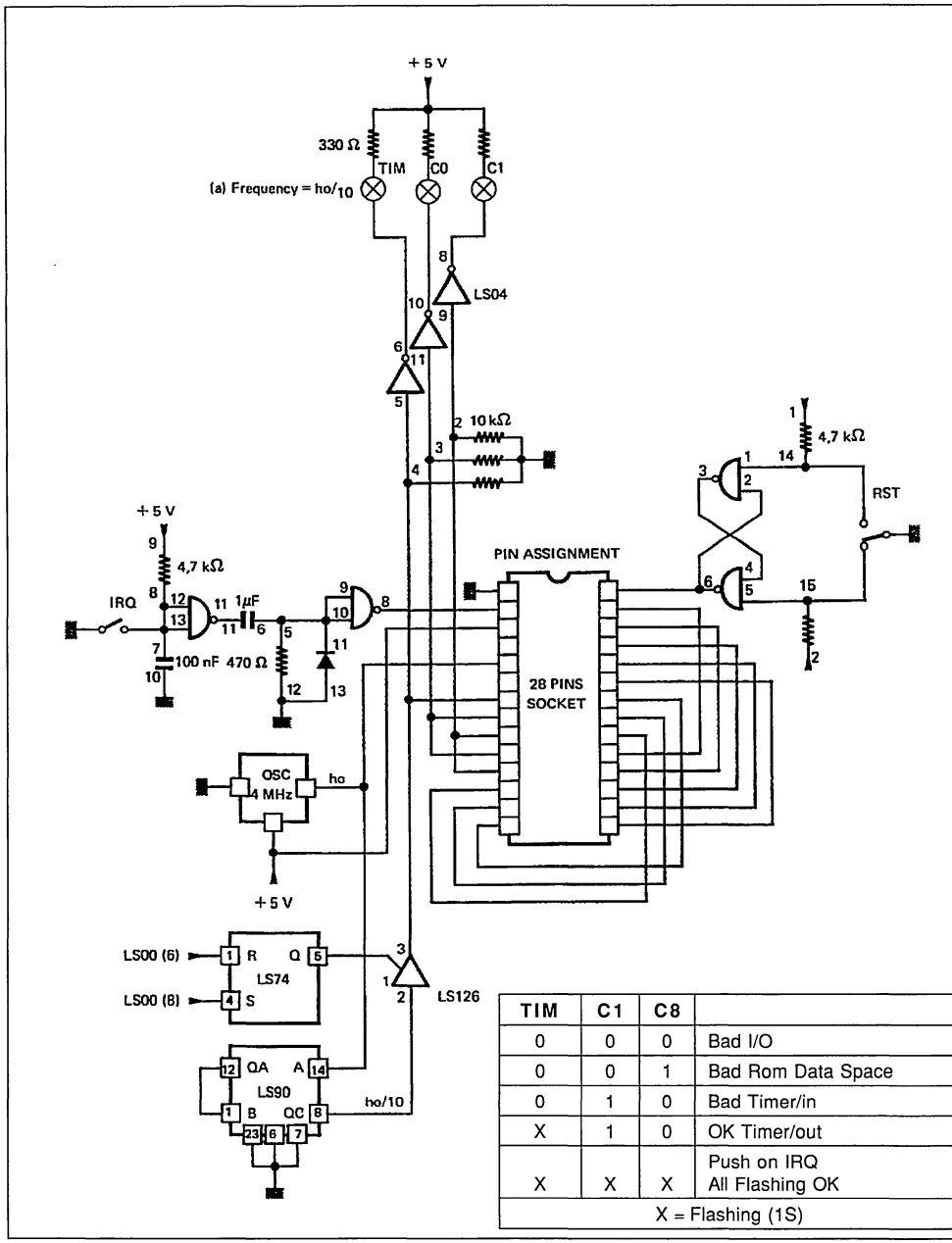
Due to bipolar components V supply must be 5 volts. (0,8A).

The following diagram shows where jumpers are located.



SELF-CHECK

A self-check program is available in EPROM to verify piggy-back. To use it you must connect pins according to the following diagram. (Piggy-back self-check program is different than monochip one).



TIM	C1	C8	
0	0	0	Bad I/O
0	0	1	Bad Rom Data Space
0	1	0	Bad Timer/in
X	1	0	OK Timer/out
X	X	X	Push on IRQ All Flashing OK
X = Flashing (1S)			

NOTES

SALES OFFICES

AUSTRALIA

NSW 2027 EDGECLIFF
Suite 211, Edgecliff centre
203-233, New South Head Road
Tel (61-2) 327 39 22
Telex 071 126911 TCAUS
Telefax (61-2) 327 61 76

BRAZIL

05413 SÃO PAULO
R Henrique Schaumann 286-CJ33
Tel (55-11) 883-5455
Telex (39-11) 37988 "UMBR BR"

CANADA

BRAMPTON, ONTARIO
341 Main St North
Tel (416) 455-0505
Telefax 416-455-2606

CHINA

BEIJING
Beijing No 5 Semiconductor
Device Factory
14 Wu Lu Tong Road
Da Shang Mau Wai
Tel (861) 2024378
Telex 222722 STM CH

DENMARK

2730 HERLEV
Herlev Torv, 4
Tel (45-2) 94 85 33
Telex 35411
Telefax (45-2) 948694

FRANCE

94253 GENTILLY Cedex
7 - avenue Galliéni - BP 93
Tel (33-1) 47 40 75 75
Telex 632570 STMHQ
Telefax (33-1) 47 40 79 10

67000 STRASBOURG
20, Place des Halles
Tel (33) 88 25 49 90
Telex 870001F
Telefax (33) 88 22 29 32

HONG KONG

WANCHAI
22nd Floor - Hopewell centre
183 Queen's Road East
Tel (852-5) 8615788
Telex 60955 ESGIES HX
Telefax (852-5) 8656589

INDIA

NEW DELHI 110048
Liason Office
S114, Greater Kailash Part 2
Tel (91) 6414537
Telex 31-62000 SGSS IN

ITALY

20090 ASSAGO (MI)
V le Milanofiori - Strada 4 - Palazzo A/4/A
Tel. (39-2) 89213 1 (10 linee)
Telex 330131 - 330141 SGSAGR
Telefax (39-2) 8250449

40033 CASALECCHIO DI RENO (BO)
Via R Fucini, 12
Tel (39-51) 591914
Telex 512442
Telefax (39-51) 591305

00161 ROMA
Via A Torlonia, 15
Tel (39-6) 8443341/2/3/4/5
Telex 620653 SGSATE I
Telefax (39-6) 8444474

JAPAN

TOKYO 108
Nisseki - Takanawa Bld 4F
2-18-10 Takanawa
Minato-Ku
Tel (81-3) 280-4121
Telefax (81-3) 280-4130

KOREA

SEOUL 121
8th floor Shinwon Building
823-14, Yuksam-Dong
Kang-Nam-Gu
Tel (82-2) 552-0399
Telex SGSKOR K29998
Telefax (82-2) 552-1051

NETHERLANDS

5612 AM EINDHOVEN
Dillenburgstraat 25
Tel (31-40) 550015
Telex 51186
Telefax (31-40) 528835

SINGAPORE

SINGAPORE 2056
28 Ang Mo Kio - Industrial Park 2
Tel (65) 4821411
Telex RS 55201 ESGIES
Telefax (65) 4820240

SPAIN

08021 BARCELONA
Calle Platon, 6 4th Floor, 5th Door
Tel (34-3) 2022017-2020316
Telefax (34-3) 2021461

28027 MADRID
Calle Albacete, 5
Tel (34-1) 4051615
Telex 27060 TCCEE
Telefax (34-1) 4031134

SWEDEN

S-16421 KISTA
Borgarfjordsgatan, 13 - Box 1094
Tel (46-8) 7939220
Telex 12078 THSWS
Telefax (46-8) 7504950

SWITZERLAND

1218 GRAND-SACONNEX (GENÈVE)
Chemin François-Lehmann, 18/A
Tel (41-22) 7986462
Telex 415493 STM CH
Telefax (41-22) 7984869

TAIWAN

TAIPEI
12th Floor
571, Tun Hua South Road
Tel (886-2) 755-4111
Telex 10310 ESGIE TW
Telefax (886-2) 755-4008

UNITED KINGDOM

MARLOW, BUCKS
Planar House, Parkway
Globe Park
Tel (44-628) 890800
Telex 847458
Telefax (44-628) 890391

SALES OFFICES

U.S.A.

NORTH & SOUTH AMERICAN
MARKETING HEADQUARTERS
1000 East Bell Road
Phoenix, AZ 85022-2699
(1)-(602) 867-6340

SALES COVERAGE BY STATE

ALABAMA
Huntsville - (205) 533-5995

ARIZONA
Phoenix - (602) 867-6340

CALIFORNIA
Irvine - (714) 250-0455
San Josè - (408) 452-8585

COLORADO
Boulder (303) 449-9000

GEORGIA
Norcross - (404) 242-7444

ILLINOIS
Schaumburg - (312) 517-1890

MARYLAND
Columbia - (301) 995-6952

MASSACHUSSETTS
Waltham - (617) 890-6688

NEW JERSEY
Voorhees - (609) 772-6222

OREGON
Tigard - (503) 620-5517

TEXAS
Austin - (512) 339-4191
Carrollton - (214) 466-8844

WASHINGTON
Seattle - (206) 524-6421

FOR RF AND MICROWAVE
POWER TRANSISTORS CONTACT
THE FOLLOWING REGIONAL
OFFICES IN THE U S A

CALIFORNIA
Hawthorne - (213) 675-0742

NEW JERSEY
Totowa - (201) 890-0884

PENNSYLVANIA
Montgomeryville - (215) 362-8500

TEXAS
Carrollton - (214) 466-8844

WEST GERMANY

6000 FRANKFURT M1
Gutleutstrabe 322
Tel (49-69) 237492
Telex 176997 689
Telefax (49-69) 231957
Teletex 6997689 = STVBP

8011 GRASBRUNN
Bretonischer Ring 4
Neukeferloh Technopark
Tel . (49-89) 460060
Telex 528211
Telefax (49-89) 4605454
Teletex 897107 = STDISTR

3000 HANNOVER 1
Eckenerstrasse 5
Tel (49-511) 634191
Telex 175118418
Teletex 5118418 csfbeh
Telefax (49-511) 633552

8500 NÜRNBERG 20
Erlenstegenstrasse, 72
Tel (49-911) 597032
Telex 626243
Telefax (49-911) 5980701

5200 SIEGBURG
Frankfurter Str 22a
Tel (49-2241) 660 84-86
Telex 889510
Telefax (49-2241) 67584

7000 STUTTGART 1
Oberer Kirchhaldenweg 135
Tel (49-711) 692041
Telex 721718
Telefax (49-711) 691408

Cover design by Keith & Koppel, Segrate, Italy
Typesetting and layout on Desk Top Publishing
by AZIMUT, Hénin Bt., France
Printed by Grafiche Moretti, Segrate, Italy

SGS-THOMSON Microelectronics GROUP OF COMPANIES
Australia - Brazil - China - France - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - United Kingdom - U S A - West Germany