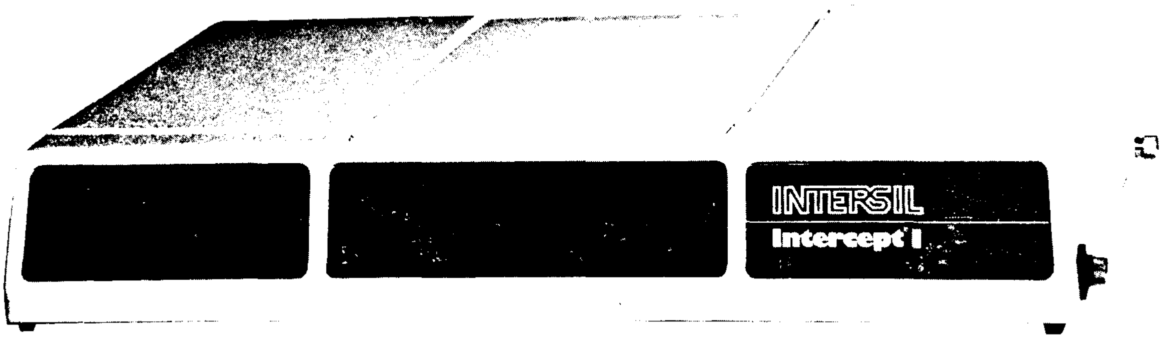
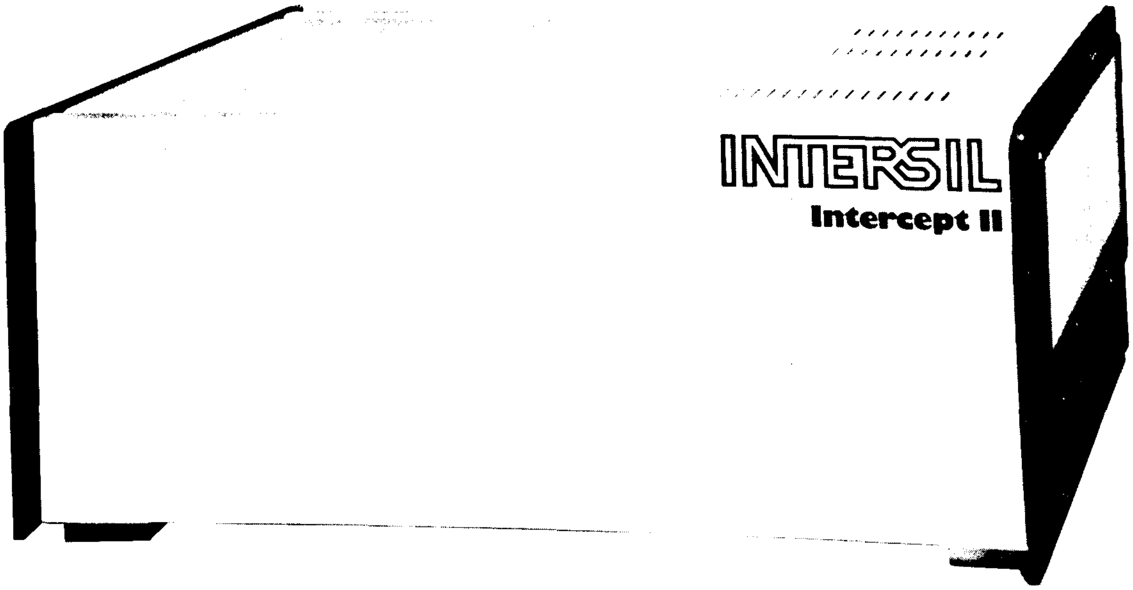
INTERSIL

$20.00

# Intercept® I/II Microcomputer Development System User's Manual





# TABLE OF CONTENTS

## *APPENDICES*

APPENDIX PAGE

# CHAPTER 1

## INTRODUCTION

The INTERCEPT is a general purpose microcomputer based on INTERSIL'S IM6100 CMOS microprocessor. It is software compatible with Digital Equipment Corporation's PDP-8/E minicomputer. The INTERCEPT is also a sophisticated design tool for developing systems using INTERSIL'S rapidly expanding line of CMOS LSI components.

This manual consists of ten chapters. Chapter 2 discusses the operation of the control panel/debugger. This information is essential to the effective use of INTERCEPT for software development.

Chapter 3 discusses the IM6100 microprocessor as implemented in the INTERCEPT. The chapter does not apply to user developed systems, because the IM6100 data book fully documents the microprocessor and its family of support devices. Chapter 3 also defines the INTERCEPT bus.

Chapters 4 and 5 describe the software considerations for using the INTERCEPT's hardware features. The INTERCEPT includes two serial I/O ports, an extended memory controller, a real time clock, and other features.

Chapter 6 summarizes the software options available with INTERCEPT. INTERSIL offers a complete floppy disk based operating system (IFDOS) for efficient and rapid program development. Also, PDP-8 software runs properly on the INTERCEPT with very few exceptions (see Chapter 4).

Chapter 7 presents INTERCEPT hardware options including memory up to 32K, floppy disk, power fail/auto-restart, etc.

Detailed descriptions of INTERCEPT's standard hardware modules (the 6912 CPU and the 6901 4K X 12 memory) are in Chapters 8 and 9. Appropriate schematic diagrams are included.

Chapter 10 outlines simple service and maintenance procedures for qualified service technicians.

Finally, Appendices A - I supply ASCII character codes, control panel program flowcharts, and other useful information. The last Appendix, "Getting Started with INTERCEPT", gives a step-by-step procedure for initially starting the system.

# CHAPTER 2

## INTERCEPT CONTROL PANEL/DEBUGGER

The "control panel memory" space of the IM6100 is completely separate from the normal user memory (or "main memory") space. As long as power is supplied to the INTERCEPT, it is always running, either executing control panel or main memory instructions. Control panel entry is forced at power failure (if the system includes the 6913 Power Failure/Auto-Restart Module), at power restoration, whenever a HALT, OSR, or "trapped IOT" is executed in main memory, or when the user presses the BREAK key at the terminal keyboard.

Upon control panel entry, the control panel resident firmware assumes control. In some cases, the status of the PC, MD, AC, L, and D registers is displayed, and the control panel program enters command mode, waiting for user commands. For OSR and most trapped IOTs, the instruction's function is performed and main memory execution resumes.

Command mode allows the user to issue commands for memory examination and modification, program control, and numerous other functions in a simple, concise control language. Memory addresses and contents are entered and displayed in octal, eliminating user interaction with cumbersome binary numbers.

The control panel program has many advantages over the traditional computer console. Not only are normal console functions provided, but a sophisticated high speed debugger and other capabilities are included in the control panel firmware. The final advantage of the firmware approach is the elimination of costly console lamps, switches, and associated hardware, and the resulting system cost reduction.

The "control panel/debugger" firmware supports some 46 commands, but less than ten are really necessary to control the operation of the computer. The other commands are powerful tools that will be appreciated by more experienced users.

The control/debugger can assume various roles in controlling user program execution. If necessary, the firmware can provide extensive program status information after each user instruction. Alternatively, only sensitive portions of a program can be monitored. Or, the firmware can be completely invisible to user programs. The user selects the mode with program control commands.

For example, the single instruction command enables the user to execute one main memory instruction at a time, receive a status display of the computer registers after each instruction, and return to command mode.

The trace command repeatedly executes main memory instructions, updating the status display with each instruction. A trace counter, under user control, governs the number of instructions executed before returning to command mode.

Alternatively, up to eight breakpoints may be set anywhere in the executable portion of a program. A breakpoint is actually a "trapped IOT" instruction which causes immediate control panel entry when executed. Each instruction replaced by a breakpoint is saved in an internal breakpoint table for later restoration.

During command mode, the user's instructions are temporarily restored to make the breakpoints totally "invisible" to memory examination commands. When execution continues from a breakpoint, the user instruction is executed before the breakpoint instruction replaces it again.

If the user wishes to update the status display at every breakpoint without returning to command mode, snapshot mode can be invoked.

Finally, user programs can run freely with no control panel intervention.

Obviously, breakpoints cannot be used for debugging ROM based programs. However, the single instruction and trace features may be used freely with any type of memory configuration.

## CONTROL PANEL/DEBUGGER COMMANDS

### MEMORY EXAMINATION/MODIFICATION

Whenever a memory location is examined, it is said to be "open." The user may modify the location and close it, or simply close it without modification. Only one location may be open at a time. Opening a new location automatically closes the currently open location, if any. All command arguments must be in octal.

nnnnn/ Open and display location nnnnn, where address nnnnn ranges from 0 to 77777. The contents of nnnnn are printed in octal. If the user types a number on the same line and closes the location, the number typed replaces the previous contents of the location. Typing / with no argument reopens the last location displayed.

<CR> Close the currently open location.

<LF> Close the current location and open (and display) the next sequential location. <LF> does not cross field boundaries.

nnnnn\ Display the contents of memory from location nnnnn to the end of nnnnn's memory page. No locations are opened. Typing \ with no argument types the contents of memory from the last displayed location to the end of that page.

SHIFT/N Close the current location, use its contents as a memory reference instruction, and open (and display) the referenced location. If the original location was modified by the user, its new contents are used by SHIFT/N.

SHIFT/O Similar to SHIFT/N, except the contents of the current location, after closing it, are used as the 12-bit address of the location to open. The Data Field register determines the extended address of the new location.

SHIFT/P Close the current location and reopen the location that was open before the last SHIFT/N or SHIFT/O command.

nnnn+ Close current location, use nnnn as a positive address offset from current address, and open that location. (nnnn may range from 0 to 7777).

nnnn- Similar to + command, except nnnn is a negative address offset.

; Same as <LF>, without any memory display. This command can quickly modify several locations on one command line. If n semicolons are typed without intervening arguments, n-1 locations are left unmodified.

## INTERNAL REGISTER EXAMINATION/MODIFICATION

The control panel/debugger maintains several registers and tables accessable to the user. Some of these registers represent the status of the user's registers as of the last control panel entry. Also, some registers are kept in "real-time." The commands below open locations exactly as the "/" command, but they reference specific control panel locations and therefore take no arguments.

A — Open and display the register containing the user's Accumulator while in control panel.

L — Open and display the register containing user's Link while in control panel. Only bit 11 of this register is significant.

D — Open and display the register containing the user's Data Field while in control panel. The contents of D appear as multiples of 10 octal, therefore only bits 6-8 are significant. (i.e., 10 = field 1, 20 = field 2, etc.)

S — Open and display the switch register.

Q — Open (and display) the user's MQ register.

M — Open (and display) the search mask register. This register defines target values for the W, X, and CTRL/W commands (see W command). On reset, M is set to 7777.

H — Open (and display) the high limit register. This register, with the low limit register, defines the range of the W, X, CTRL/W, and O commands. When H is open, a <LF> opens the low limit register (see W and O commands). On reset, the high and low limits are set to 7777 and 0 respectively.

F — Open (and display) the register defining the extended address for the W, X, CTRL/W, I and O commands. Like D, only bits 6-8 of F are significant (see W, I, and O commands).

R — Open (and display) the register defining the replacement value for the CTRL/W command (see CTRL/W command).

K — Open (and display) the iteration counter of the default breakpoint (see breakpoint section).

N — Open (and display) breakpoint identification register. Only bits 9-11 of the register are significant (see breakpoint section).

J — Open (and display) first word of the "junk table." The contents of J define the length of the primary port transmit flag time-out. Every time control panel is entered and the firmware displays information on the terminal, the firmware must "time-out" the transmit flag for one "character time" so previously sent characters are not garbled. This time-out value should be set according to terminal baud rate. (See Table 2-1)

When the time-out register is open, a <LF> opens the second word of the "junk" table -- the null count. Some terminals require one or more null characters to be output following a carriage return. The null register contains the eights complement minus one of the desired number of nulls. (7777 = 0 null characters, 7776 = 1 null, 7775 = 2 nulls, etc.)

## PROGRAM CONTROL

These commands start or resume user program execution in various modes. Once a user program is started, the control panel program will not accept further commands until one of the following occurs: the user types the BREAK key, a HALT instruction is executed, or a breakpoint is reached which causes its respective breakpoint counter to expire. The 5-digit arguments for the G, C, E, T, and P commands enable the user to set the Instruction Field and Program Counter for program execution.

nnnnnG — Start execution at address nnnnn with the AC and LINK cleared and the Data Field equal to the Instruction Field. G must be preceded by an address.

nnnnnC — Continue program execution at address nnnnn. If no argument is specified, execution continues from the address in the current PC (i.e., the PC of the last control panel entry).

nnnnnE — Execute instruction at address nnnnn and return to command mode. If the argument is omitted, the instruction pointed to by the current PC is executed.

nnnnnT — Begin instruction tracing at location nnnnn, updating the status display after every instruction. If no argument is specified, tracing begins at the current PC. Tracing continues until the trace counter expires (see "." command).

nnnnnP — Proceed in snapshot mode from nnnnn. Similar to the C command, except that every time a breakpoint is encountered, the program status display is updated and execution continues. If no argument is specified, execution begins at the current PC.

nnnnCTRL/G — Start control panel program execution at control panel memory location nnnn. Used improperly, CTRL/G can "lock" the system. Only a hardware reset can "unlock" the system. CTRL/G should only be used after reading Chapter 5 carefully.

xxxx. — Set the instruction trace counter to xxxx.

xxxx: — Set the breakpoint counter to xxxx for the breakpoint whose number is in the "N register."

## BREAKPOINT SUPPORT

The control panel program has numerous commands for setting and deleting up to 8 breakpoints and performing related operations.

Each breakpoint has a unique number between 0 and 7 inclusive. Breakpoint commands may reference any of the 8 breakpoints. The breakpoint identification register, or N, selects a breakpoint for a given operation. The breakpoint whose number is in N is called the default breakpoint. If no breakpoint number is explicitly designated in the commands below, the default breakpoint is used. When a breakpoint occurs during program execution, that becomes the default breakpoint.

b, Make breakpoint b the default breakpoint (b may range from 0 to 7).

nnnnnB Set default breakpoint at location nnnnn. If no argument is present, the default breakpoint is removed.

Y Remove all breakpoints.

# Display the octal memory addresses of all breakpoints currently set. If a breakpoint is not set, its number will be typed rather than its memory address.

N Open the breakpoint identification register (or N register) for possible modification.

K Open the breakpoint iteration counter of the default breakpoint for possible modification.

### BREAKPOINT RESTRICTIONS

1. Breakpoints must be placed on executable instructions.
2. Breakpoints can only be used in read/write memory.
3. Breakpoints should not be placed in locations modified by user programs.
4. Never set two or more breakpoints at the same memory location.

## MEMORY SEARCH/SEARCH AND REPLACE

The control panel program can search all or part of memory for specific values or bit patterns. Three search parameters control the search.

The search mask M is one parameter. During a search, sequential memory locations are bitwise "ANDed" with the search mask. The result of the logical "AND" determines whether the memory word matches the word being searched for. Searching begins at the address specified in the low limit register and progresses to the address in the high limit register. The F register provides the extended address for searching. Searching never crosses field boundaries.

The search parameters are modified via the M and H commands, described previously.

Normal searching displays the addresses and contents of all memory locations meeting the search criteria. Additionally, there is a search and replace command which replaces the contents of all locations meeting the search criteria with the contents of the R register.

mmmmW — Display every location within the search limits whose contents, when logically "ANDed" with the search mask, equal mmmm.

mmmmX — Display every location within the search limits whose contents, when logically "ANDed" with the search mask, do not equal mmmm.

mmmmCTRL/W — All memory locations between the search limits are "ANDed" with the search mask and compared with mmmm. Any locations matching mmmm are replaced with the contents of the R register.

## INPUT/OUTPUT

The input/output capabilities of the control panel program include reading and writing binary data in "BIN" format, and bootstrapping Intersil's IFDOS operating system.

I — Accept binary input in "BIN" format (see Appendix B ). Loading address information is encoded with the binary input. However, if no memory field information is encoded, the F register designates the field for loading. The program makes two checks for data validity while loading. Each time a memory location is loaded with data, it is read back and compared with the input data. Any difference results in a "VERIFY ERROR." This prevents loading non-existent memory. At the end of loading, the accumulated checksum of the input data is compared with that encoded with the input. Any difference results in a "CHECKSUM ERROR."

O — Produce binary output in "BIN" format of all memory contents, between the low and high limits inclusive, in the field specified by the F register. About six inches of leader/trailer are generated before and after the binary information. If more leader/trailer is required, it should be generated locally.

1Z Bootstrap the INTERCEPT FLOPPY DISK OPERATING SYSTEM. (See Chapter 6).

## MISCELLANEOUS

Several commands exist which do not fit any of the above catagories.

nnnnnU — Set "User register" to address nnnnn. When computer status is displayed, the contents of address nnnnn will be displayed along with the PC, MD, AC, L and DF.

U — Disable user address display.

V — Display interrupt system status as of last control panel entry. The interrupt enable bit (IE), interrupt inhibit bit (II), and interrupt request bit (IR) are displayed as ones (enabled) and zeros (disabled).

? — Display status as of last control panel entry. If the AC, LINK, Data Field, or User location were altered since the last status display, the updated contents are shown.

x<ESC> or x<ALT> Change system characteristics. The argument x may range from 0 to 7 or be omitted entirely.

0<ESC> or <ESC> Reference main memory with memory examination/modification, memory search, and input/output commands.

1<ESC> Accept control panel commands from Primary Port and direct control panel output to Primary Port.

2<ESC> Accept control panel commands from either Primary or Secondary Port and direct all control panel output to Secondary Port.

3<ESC> Reference control panel memory with memory examination/modification, memory search, and input/output commands.

4<ESC> Perform error check on ROM addresses 4000-5777.

5<ESC> Perform error check on ROM addresses 6000-7777.

7<ESC> Perform firmware reset of system.

xxxx" Set Secondary Port baud rate. The argument xxxx must be a value from the left column of the table below, which corresponds to the adjacent baud rate value. If no argument xxxx is specified, the value representing the Secondary Port baud rate in the table below is displayed.

TABLE 2-1

| COMMAND ARGUMENT | SECONDARY PORT BAUD RATE | PRIMARY PORT TIME-OUT VALUE | PRIMARY PORT BAUD RATE |
|---|---|---|---|
| 0 | 38,400 | 7750 | 19,200 |
| 400 | 38,400 | 7720 | 9,600 |
| 1000 | 50 | 7640 | 4,800 |
| 1400 | 75 | 7500 | 2,400 |
| 2000 | 134.5 | 7400 | 1,800 |
| 2400 | 200 | 7170 | 1,200 |
| 3000 | 600 | 6360 | 600 |
| 3400 | 2,400 | 4750 | 300 |
| 4000 | 9,600 | 3330 | 200 |
| 4400 | 4,800 | 1710 | 150 |
| 5000 | 1,800 | 1140 | 134.5 |
| 5400 | 1,200 | 100 | 110 |
| 6000 | 2,400 | 0 | 75 |
| 6400 | 300 | 0 | 50 |
| 7000 | 150 | | |
| 7400 | 110 | | |

CONTROL PANEL/DEBUGGER ERRORS

ERROR! The user has specified an illegal argument or used a command improperly. The command is ignored and further user input is awaited.

ZAP! The user has specified an illegal command. The command is ignored and further user input is awaited. Illegal commands can be used to "erase" a mistaken command entry. (e.g., RUBOUT, SPACE BAR)

# CHAPTER 3

# INTERCEPT PROCESSOR ARCHITECTURE

## MEMORY ORGANIZATION

The INTERCEPT has an addressing capacity of 32K 12-bit words. The memory is organized into 4096-word "fields." The first 4K words are in Field 0. If a full 32K of memory is installed, the uppermost memory field is numbered 7. In any given memory field every location has a unique 4-digit octal (12-bit binary) address, $0000_8$ to $7777_8$ ($0000_{10}$ to $4095_{10}$). Each memory field is divided into 32 pages of 128 words each. Memory pages are numbered sequentially from Page $00_8$, containing addresses $0000_8$ - $0177_8$, to Page $37_8$, containing addresses $7600_8$ - $7777_8$. The first five bits of a 12-bit memory address define the page, and the low order 7 bits specify the address of the memory location within the given page, called the Page Address.*

To select the proper memory field from among the eight possible Fields, the IM6102 provides a three-bit extension to the memory addressing word generated by the CPU Program Counter. Normally these three bits come from the Instruction Field register on the IM6102. However, during the execute cycle of an indirectly addressed AND, TAD, ISZ, or DCA instruction, when the DATAF line is asserted by the CPU; the three-bit extension is derived from the Data Field register in the IM6102.

## INSTRUCTION SET

The instruction set is divided into three categories: Memory Reference Instructions (MRI), Operate Instructions (OPI) and Input/Output Instructions (IOT). The high order three bits (on bus lines DX0 - DX2) denote the instruction type. MRI's begin with $0_8$, $1_8$, $2_8$, $3_8$, $4_8$, or $5_8$. All IOT's begin with $6_8$, and all OPI's begin with $7_8$. This first octal digit in the instruction code is called the "Opcode."

*A detailed discussion of PDP-8 (and IM6100) programming is found in "Introduction to Programming," Digital Equipment Corporation, Maynard, Massachusetts, 1973. Also see the IM6100 Data Book.

Table 3-1 details the required machine cycles, and the T-states required in each cycle, for each type of instruction.

TABLE 3-1

Required Machine Cycles and T-States for Each Instruction Type

| INSTRUCTION TYPE | OPCODE | REQUIRED CYCLES | | | |
|---|---|---|---|---|---|
| | | first | second | third | fourth |
| | | T-states | T-states | T-states | T-states |
| AND | $0_8$ | | | | |
| directly addressed | | 5 (IFETCH) | 5 (execute) | | |
| indirectly addressed | | | 5 (indirect) | 5 (execute) | |
| auto-indexed | | | 6 (auto-index) | 5 (execute) | |
| TAD | $1_8$ | | | | |
| directly addressed | | | 5 (execute) | | |
| indirectly addressed | | | 5 (indirect) | 5 (execute) | |
| auto-indexed | | | 6 (auto-index) | 5 (execute) | |
| ISZ | $2_8$ | | | | |
| directly addressed | | | 6 (execute) | 5 (execute) | |
| indirectly addressed | | | 5 (indirect) | 6 (execute) | 5 (execute) |
| auto-indexed | | | 6 (auto-index) | 6 (execute) | 5 (execute) |
| DCA | $3_8$ | | | | |
| directly addressed | | | 6 (execute) | | |
| indirectly addressed | | | 5 (indirect) | 6 (execute) | |
| auto-indexed | | | 6 (auto-index) | 6 (execute) | |
| JMS | $4_8$ | | | | |
| directly addressed | | | 6 (execute) | | |
| indirectly addressed | | | 5 (indirect) | 6 (execute) | |
| auto-indexed | | | 6 (auto-index) | 6 (execute) | |
| JMP | $5_8$ | | | | |
| directly addressed | | | 5 (execute) | | |
| indirectly addressed | | | 5 (indirect) | 5 (execute) | |
| auto-indexed | | | 6 (auto-index) | 5 (execute) | |
| IOT | | | | | |
| IOT | $6_8$ | | 7 (execute) | 6 (execute) | |
| OPI | $7_8$ | | | | |
| 2-cycle OPI | | | 5 (execute) | | |
| 3-cycle OPI | | | 5 (execute) | 5 (execute) | |

## MEMORY REFERENCE INSTRUCTIONS

The memory reference instructions operate on the contents of a memory location or use the content of a memory location to operate on the Accumulator or Program Counter.

Operation of each of the MRI's is detailed in Table 3-2 and figure 3-1. Each of these instructions may be directly addressed, the absolute address (bits 5-11) of the operand is embedded in the instruction itself, so only two cycles are required for a complete fetch and execution.

When an MRI is indirectly addressed, the second machine cycle is an "indirect" cycle, which is used to pick up the desired absolute address of the operand from memory. Execution of the instruction takes place in the third (and fourth for an ISZ) cycle. This mode of addressing is used when the desired address of the operand is not on the current page or on Page $00_8$.

TABLE 3-2

Operation of Memory Reference Instructions

| MNEMONIC | OPCODE | OPERATION |
|---|---|---|
| AND | $0_8$ | Logical AND. Operand is AND'ed with contents of Accumulator. Result remains in Accumulator. |
| TAD | $1_8$ | Binary ADD. Operand is added to Accumulator contents; result remains in Accumulator. Carry out complements the Link. Can be used for Accumulator load if Accumulator is initially cleared. |
| ISZ | $2_8$ | INCREMENT, AND SKIP IF ZERO. Operand is incremented and restored. Next instruction is skipped if result was zero. |
| DCA | $3_8$ | DEPOSIT TO MEMORY, AND CLEAR ACCUMULATOR. Contents of Accumulator are deposited in operand address, then Accumulator is cleared. |
| JMS | $4_8$ | JUMP TO SUBROUTINE. Contents of Program Counter are deposited in operand address. Then Program Counter is set to operand address +1. |

JMP $5_8$ UNCONDITIONAL JUMP. Program Counter is set to operand address.

During an instruction fetch cycle, the IM6100 fetches the instruction pointed to by the Program Counter (PC). The contents of the PC are transferred to the Memory Address Register (MAR). The PC is incremented by 1. The PC now contains the address of the 'next' sequential instruction. The MAR contains the address of the 'current' instruction which must be fetched from memory. Bits 0-4 of the MAR identify the CURRENT PAGE, that is, the Page from which the the current instruction was fetched and bits 5-11 of the MAR identify the location within the Current Page. MAR is an internal processor register which is not user accessable.

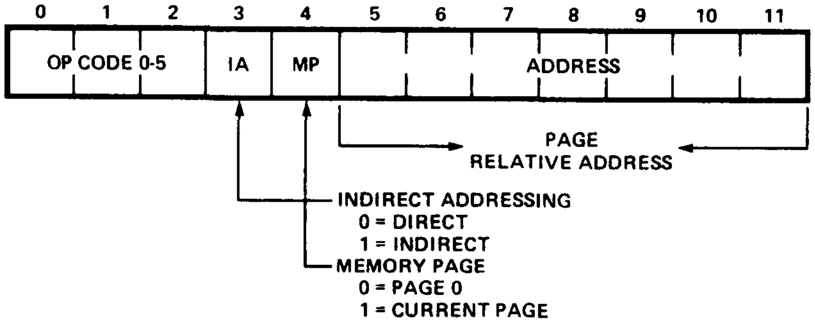


FIGURE 3-1

MEMORY REFERENCE INSTRUCTION FORMAT

Bits 5-11, the PAGE ADDRESS, identify the location of the OPERAND on a given page, but they do not identify the page itself. The page is specified by bit 4, called the CURRENT PAGE OR PAGE 0 BIT. If bit 4 is a 0, the page address is interpreted as a location on Page 0 (By definition, locations $0000_8$ to $0177_8$). If bit 4 is a 1, the page address specified is interpreted as a Current Page location.

For example, if bits 5 through 11 represent $123_8$ and bit 4 is a 0, the location referenced is the absolute address $0123_8$. However, if bit 4 is a 1 and the current instruction is in a memory location whose absolute address is $4610_8$ the page address $123_8$ designates the absolute address $4723_8$ as shown below.

$4610_8$ = 100 110 001 000 = PAGE 10 011 = PAGE $23_8$

Location $4610_8$ is in PAGE $23_8$. Location $123_8$ in PAGE $23_8$, CURRENT PAGE, will be:

10 011 1 010 011 = 100 111 010 011 = $4723_8$

| PAGE NUMBER | PAGE ADDRESS |
|---|---|
| $23_8$ | $123_8$ |

Therefore, 256 locations are directly addressable; 128 on PAGE 0 and 128 on the CURRENT PAGE. Other locations must be addressed indirectly. This is done by setting bit 3 in the instruction and using a 12 bit "pointer" address on PAGE 0 or the CURRENT PAGE as the address of the operand. An indirect memory reference instruction operates on the contents of the location identified by the pointer location. It should be noted that locations $0010_8$ - $0017_8$ in PAGE 0 of each Field are AUTO-INDEXED. If these locations are addressed indirectly with the memory page bit set to 0, the contents are incremented by 1 and restored before they are used as the operand address. These locations are often used for indexing applications.

## IOT INSTRUCTIONS

The IOT instructions all have an Opcode of $6_8$. IOT's initiate the operation of peripheral devices and transfer data between peripherals and the processor. Input/output is achieved by programmed data transfers, interrupt initiated transfers, or direct memory access. Programmed data transfers are the simplest way to move data to/from peripheral devices, but also the slowest.

In an IOT instruction, bits 0-2 are always set to 110 ($6_8$). Unless the selected device interface employs the Intersil IM6101 PIE device, bits 3-8 are the device selection code used to specify the peripheral device, and bits 9-11 specify the operation to be performed with the selected peripheral. (When the PIE device is used, bits 3-7 specify the PIE and bits 8-11 the operation to be performed. Please refer to the data sheet on the IM6101 for more details). The device selection code 000 000 ($00_8$) in bits 3-8 is reserved for processor IOT's. There are eight of these: $6000_8$ - $6007_8$. They are used by the processor for certain "housekeeping" functions associated with the interrupt system. The operation of each of the processor IOT's is detailed in Table 3-3.

A programmed data transfer begins when the CPU fetches an instruction from memory and recognizes it as an IOT instruction. The processor sequences the IOT instruction through a 2-cycle execute phase referred to as IOTA and IOTB. See Figure 3-2.

The instruction is latched into the device interface during IOTA, using the trailing edge of the LXMAR pulse. DEVSEL is the active SELect line for all IOT instructions. The selected peripheral device controls the processor during the data transfer by means of the C0, C1, C2 and SKP/INT lines on the bus. The type of data transfer is specified by the peripheral device interface by asserting the control lines as shown in Table 3-4.

TABLE 3-3

PROCESSOR IOT INSTRUCTIONS

| MNEMONIC | OCTAL | OPERATION |
|---|---|---|
| SKON | 6000 | SKIP IF INTERRUPT ON. The next instruction is skipped if the processor Interrupt Enable flip-flop is set, then this flip-flop is reset. |
| ION | 6001 | INTERRUPTS ON. The processor Interrupt Enable flip-flop is set immediately after fetching the next instruction. (At least one more instruction is executed before the first interrupt is recognized if interrupt enable was previously cleared). |
| IOF | 6002 | INTERRUPTS OFF. Immediately resets the processor Interrupt Enable flip-flop, so no more interrupts are allowed. |

SRQ 6003 SKIP IF INT REQUEST. The next instruction is skipped if the INTREQ pin is asserted low.

GTF 6004 GET FLAGS. The following flag bits are read into the AC:

AC (0) <- Link flip-flop (AC (0) = 1 if Link set)
AC (2) <- INTREQ pin (pin 8) on IM6100 (1 if pin 8 low)
AC (3) <- Interrupt Inhibit flip-flop in IM6102 (1 if IIFF set)
AC (4) <- CPU Interrupt Enable flip-flop (1 if IEFF set)
AC (6-11) <- Save Field register in IM6102.

RTF 6005 RETURN FLAGS. Link is set by AC (0). Interrupt Inhibit flip-flop on IM6102 is unconditionally set until the next JMP or JMS instruction is executed. CPU Interrupt Enable flip-flop is unconditionally set, as in ION instruction. Instruction Buffer (IB) register on IM6102 is loaded from AC (6-8), and Data Field register is loaded from AC (9-11). IB register will be transferred to IF register as next JMP or JMS is being executed.

6006 Not used.

CAF 6007 CLEAR ALL FLAGS. Accumulator and Link are cleared. Interrupt Enable flip-flop is reset. This instruction is also decoded by some device interfaces for initialization purposes.

Except for processor the above IOT instructions are non-specific because the operation that they perform is not "known" by the processor. Rather, the interface designer must define what each IOT instruction does by the logic built into the interface. The IOT instructions work in conjunction with the C0, C1, C2 and SKP lines to the processor. For example: for a PDP-8 compatible terminal interface, it is necessary that instruction $6034_8$ cause the TTY keyboard data to be OR'ed into the Accumulator. To cause device data to be OR'ed into the Accumulator, it is necessary to assert control line C1 low while C0 and C2 remain high. See Table 3-4. The interface logic must recognize the arrival of the $6034_8$

instruction and assert C1 low. Similarly, instruction $6031_8$ must cause the next instruction to be skipped if the Keyboard Data Ready Flag is set on the device interface. To accomplish this, the interface logic must, upon arrival of the $6031_8$ instruction, test the Data Ready Flag, and then if (and only if) it is set, assert the SKP line low.

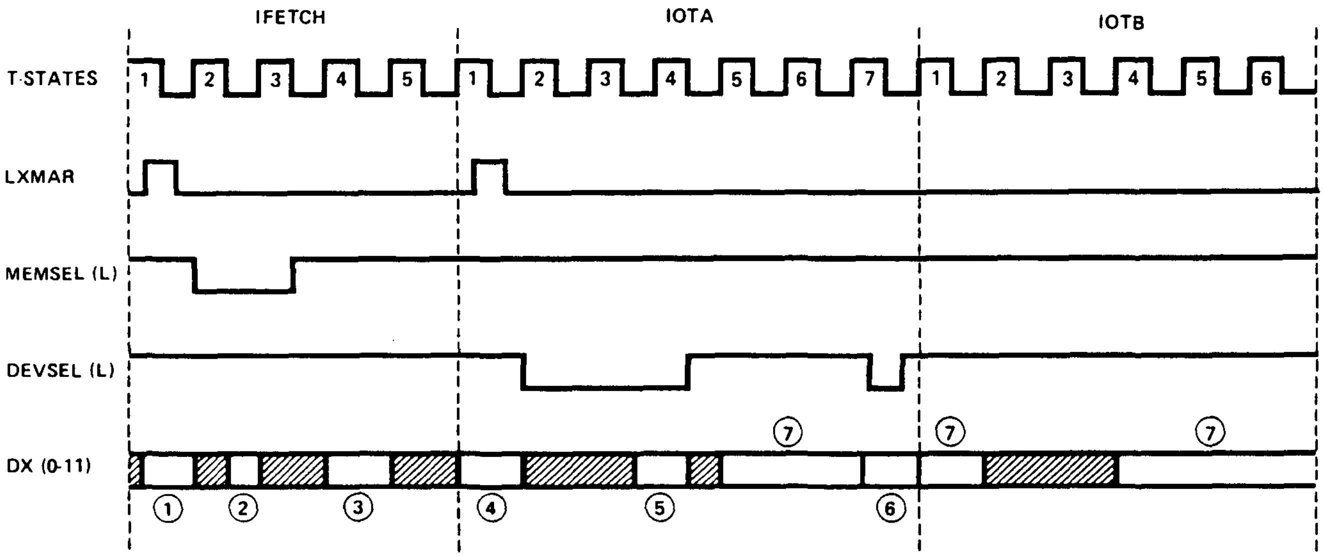
The system designer has nearly complete freedom with the IOT instructions. The designer first decides what a given IOT must do, then builds the necessary "interpretive" logic into the interface. If the IM6100 family interface elements are used for interfacing, all control codes are preassigned, but device addresses are generally variable.

TABLE 3-4

CONTROL LINES (CO, C1, C2) OPERATION

| CONTROL LINES | | | |
|---|---|---|---|
| CO | C1 | C2 | OPERATION |
| H | H | H | Accumulator (AC) contents written into device. |
| L | H | H | Accumulator contents written into device, then AC cleared. |
| H | L | H | Device data OR'd into Accumulator. |
| L | L | H | Device data read into Accumulator (jam transfer). |
| * | H | L | Device data added to contents of Program Counter (relative jump). |
| * | L | L | Device data loaded into Program Counter (absolute jump). |

* means don't care



NOTES: (1) Instruction address
(2) Instruction
(3) IM6102 current address register
(4) Device address and control
(5) Device Data In, C0, C1, C2, SKP
(6) AC Out
(7) CPU module forces BUS with $7777_8$

FIGURE 3-2

INPUT-OUTPUT INSTRUCTION TIMING

## OPI INSTRUCTIONS

The third category of instructions are called Operate Instructions, and have the Opcode of 111 ($7_8$). These instructions are all used for processor internal operations, such as conditional and unconditional skips, Accumulator rotates (either left or right, one or two-bit shifts), clearing and setting the Accumulator and Link, transferring data between the MQ register and Accumulator, etc. These instructions use bits 3-11 in the instruction (after the Opcode 111 in bits 0-2) to specify the exact operation to be performed. All these bits are available, of course, since all the operations specified are internal to the processor itself and do not require specification of a memory address or device code.

A complete listing and discussion of the OPI's is given in the IM6100 data book. It should be pointed out that these instructions are actually termed micro-instructions, since by setting or not setting given bits in the instruction word, they can be combined with one another. This reduces the number of individual steps necessary in a program. It is possible, for example, to use a single instruction to produce CLL followed by RTL, which will clear the Link and then rotate the Accumulator two positions to the left.

There is one unique OPI which is particularly noteworthy, since it acts somewhat like an IOT instruction. This is the OSR instruction, which OR's the control panel Switch Register into the processor's Accumulator.

## BASIC MACHINE TIMING

The timing for the most fundamental processor lines is illustrated in Figures 3-3 and 3-4. The T-state square wave shown is internal to the IM6100; it is not available externally as a timing reference. However, all machine timing is derived from this wave form, so it is an important reference point in processor timing discussions. Note that the frequency of this waveform is one-half the frequency of the system clock.

All machine cycles are composed of either five, six, or seven T-states (often referred to simply as "states").

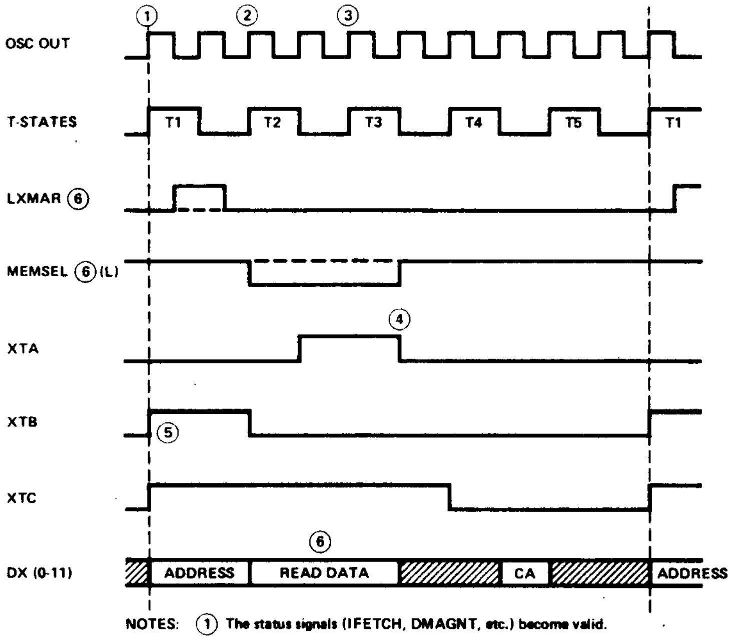
Each instruction requires 2, 3 or 4 cycles to be fetched from memory and completely executed. The first cycle is always an instruction fetch cycle consisting of five states. The remaining cycles can consist of either 5, 6, or 7 states each. Thus, a complete fetch and execution can consist of 10, 11, 15, 16, 17, 18, 21 or 22 states. Table 3-1 details the number of cycles and states in each type of instruction.

An instruction fetch and execution begins with an instruction fetch cycle (see Figure 3-3). The IFETCH line is asserted high throughout the duration of this cycle to indicate an instruction is being fetched. The processor puts the address of the instruction on the DX lines throughout the first T-state. This address is then latched into the memory modules, by using the trailing edge of the LXMAR pulse. (The trailing edge is used to allow time for the address to settle at the latch inputs on the peripheral/memory modules). Next, the MEMSEL line is asserted by the processor to allow the selected memory module to drive the DX lines with the instruction data from the addressed location. This data is picked up from the DX lines by the processor on the rising edge of T3. The rest of the cycle is then used by the processor for internal operations.

The next cycle may be the first (and possibly the only) execute cycle, or it may be an "indirect" cycle. The latter type of cycle is entered when the instruction fetched is an indirectly addressed memory reference instruction (MRI). If the instruction is auto-indexed, the indirect cycle will consist of six states; otherwise an indirect cycle has just five states. Execute cycles may consist of either five (MRI and Operate instructions) or six ( MRI and IOT instructions) or seven (IOT instructions) states. Some instructions require one, and some two, execute cycles.

The six state cycle shown in Figure 3-4 is similar to the five state cycle, except that the cycle has been extended one state so that the processor can write data into memory or a peripheral device. To accomplish this operation the processor puts the data on the DX lines throughout state T6. A SELect line (MEMSEL, DEVSEL or CPSEL) is then asserted by the processor to actuate the "write" operation. The data is typically strobed into the memory or peripheral device on the trailing edge of the SELect pulse. When the SELect line is asserted by the processor, the logic in the memory or device interface differentiates between a "read" and "write" operation by monitoring the XTC line. When this line is high, an assertion of a SELect line calls for a "read," when low a "write." Note that every "write" operation is preceded by a "read," which in most cases is ignored by the processor. During an auto-indexed indirect cycle, however, the processor does use the information picked up in the "read" part of the cycle, to determine the "pointer" address.

While the timing diagrams given in Figures 3-3 and 3-4 adequately detail the relationships between several of the time-critical lines on the processor, it should not be inferred that all edges of the illustrated signals are perfectly coincident in time, as shown. When the user designs custom interfaces for the INTERCEPT (or the IM6100 as a stand alone processor), timing differentials that appear between the various processor signals must be considered. The data book on the IM6100 family should be consulted.



NOTES: (1) The status signals (IFETCH, DMAGNT, etc.) become valid.

(2) Request lines are sampled if this is last cycle of an instruction execution.

(3) Wait line is sampled and if it is active low, the $\overline{T2}$ state is extended. Read data is sampled.

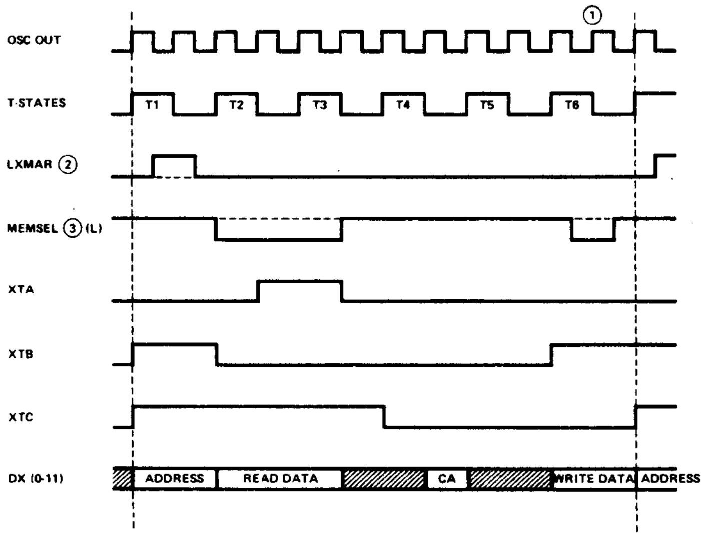
(4) 6912 samples instruction for trap functions during IFETCH. VREAD becomes valid.

(5) XTB remains high if the previous cycle was a 6-State cycle.

(6) Not present for non-memory-reference cycles.

FIGURE 3-3

BASIC 5-STATE CYCLE TIMING



NOTES: (1) Wait line is sampled and if active low, the T6 State is extended.

(2) Not present for IM6100 internal CPU cycles.

(3) Present only for external memory references.

FIGURE 3-4

BASIC 6-STATE CYCLE TIMING

Figure 3-5 shows some typical timing differentials between the illustrated signals, as measured at the INTERCEPT bus. Because of the point of measurement, these figures include the delays caused by the bus interfacing logic.

INTERCEPT BUS TIMING, @ Vcc = 5.ov, Ta = +45°C

| Parameter | Symbol | Spec. | Parameter | Symbol | Spec. |
|---|---|---|---|---|---|
| Major State Time | TS | 600 NS | Address Hold Time | TAH | 100 NS |
| LXMAR Pulse Width | TL | 260 NS | Data Hold Time | TDH | 100 NS |
| Read Wait Sample Time | TWI | 100 NS | Data Set Up Time | TDS | 120 NS |
| Write Wait Sample Time | TW2 | 1.1 µS | Status Signals Valid Time | TST | 400 NS |
| Access Time from LXMAR | TAL | 450 NS | Request Input Set Up Time | TRS | 50 NS |
| Output Enable Time | TEN | 250 NS | Request Input Hold Time | TRH | 200 NS |
| Read Pulse Width | TRP | 900 NS | XTA Pulse Width | TAP | 600 NS |
| Write Pulse Width | TWP | 235 NS | XTB Pulse Width | TBP | 600 NS |
| Address Set Up Time | TAS | 80 NS | XTC Pulse Width | TCP | 1.8 µS |

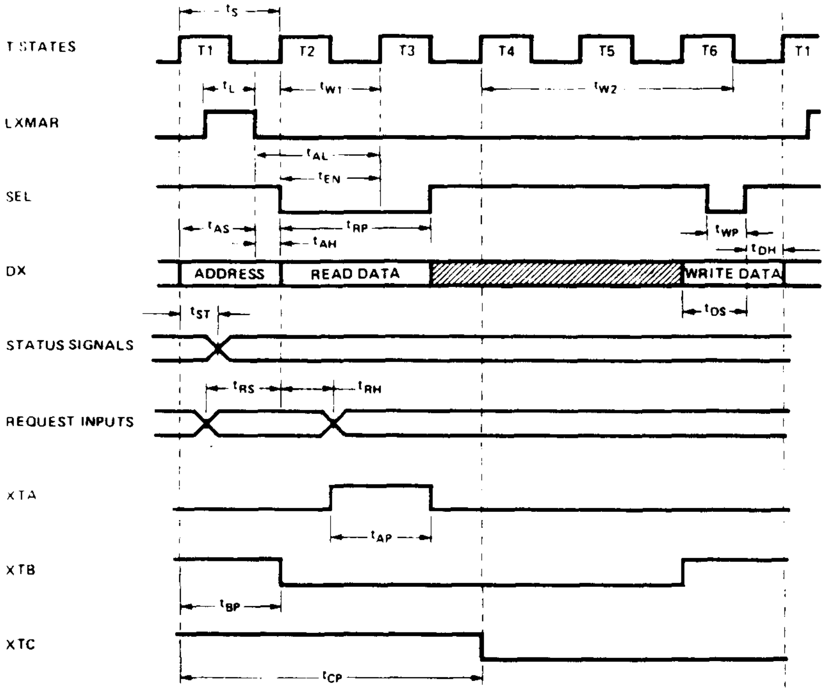


FIGURE 3-5

INTERCEPT BUS TIMING

Note that the first signal to appear in each cycle is the rising edge of XTC. (The timing lines XTA, XTB and XTC are actually used by the IM6100 to develop the other signals shown, so the latter are bound to be delayed with respect to the 'X' lines.) Particularly noteworthy is the delay on the IFETCH line. It actually overlaps into the next cycle before again going low. (This delay in IFETCH, however, is not detrimental, since this line is seldom, if ever, needed for critical timing applications.)

## INTERNAL PRIORITY STRUCTURE

As indicated in Figures 3-3 nad 3-4, RESET, control panel interrupt, DMAREQ and INTREQ are sampled on the rising edge of T2 during the last execute cycle of each instruction. If any of these actions are pending, the processor performs the requested operation according to the priority hierarchy below. If none or these actions is impending, the processor fetches and executes the next sequential instruction, and again samples the request lines, etc.

The priority hierarchy is:

RESET - If the RESET line is asserted at the sample time, the processor immediately sets its Program Counter to $7777_8$, clears the Accumulator and Link, and puts the processor in the HALT state. While halted, the processor continues to cycle and generate the timing signals XTA, XTB and XTC. When the IM6100 is powered up, the RESET pulse must span at least 58 clock pulses to be recognized, since the sequencer takes a maximum of 34 clock pulses to be initialized and a maximum of 24 clock cycles may elapse before the request line is sampled by the sequencer. Deactivating RESET forces control panel entry due to logic on the 6912 CPU board.

CONTROL PANEL INTERRUPT - If RESET is not asserted and a control panel interrupt condition exists (Framing Error, trapped instruction, HLT, etc.) the processor transfers control to the control panel firmware so the condition may be serviced.

DMAREQ - If none of the above actions is indicated, and the processor finds the DMAREQ line asserted, it grants the DMA request at the end of the current cycle.

INTREQ - If neither RESET, control panel interrupt, nor DMA action is indicated, and the INTREQ line is found asserted, the processor will grant the device interrupt request at the end of the current instruction if the interrupt system is enabled.

IFETCH - If none of the above actions is indicated, the processor will fetch the next sequential instruction, in the next cycle.

The above priority heirarchy is supplemented by internal and external logic and program software. During the processing of a control panel interrupt, device interrupt requests and DMA requests are ignored by the processor. When the processor grants a device interrupt request, it ignores further interrupt requests until the interrupt system is re-enabled by an ION instruction.

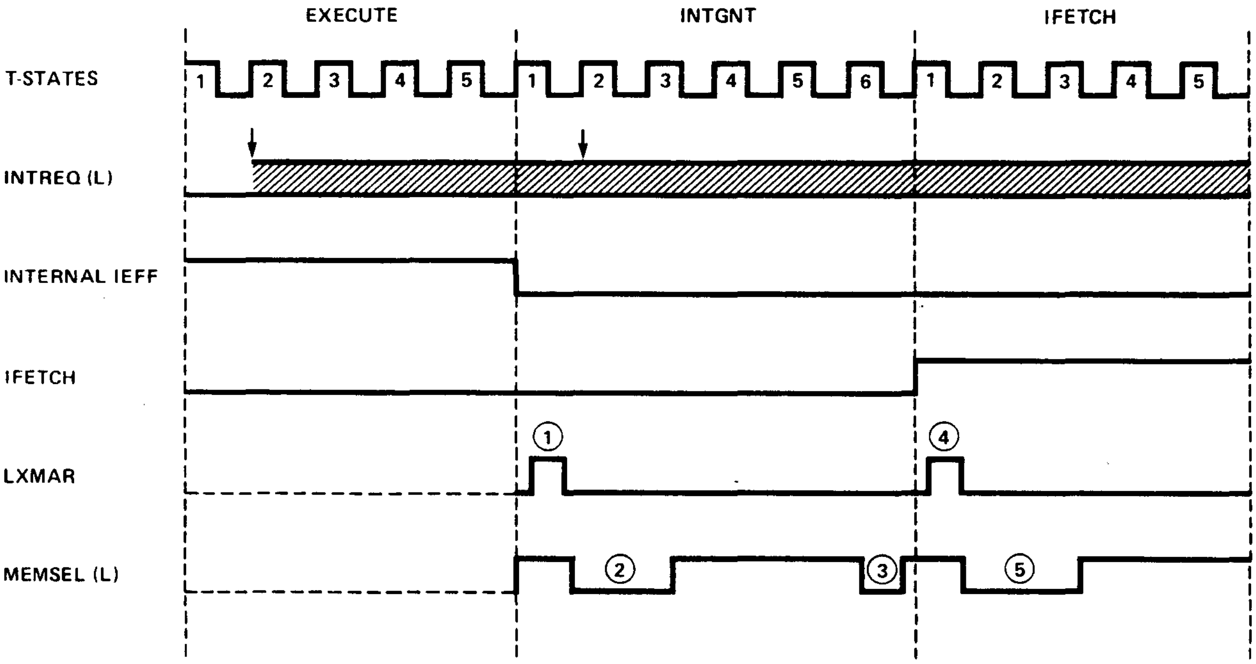
## DEVICE INTERRUPT TRANSFERS

The program interrupt method is used to transfer data between processor and peripheral devices when it is unacceptable to have the processor wait for the device to indicate that it is ready to output or accept new data. Using the interrupt system, the processor is free to execute a "background" program until the external device indicates it is ready for a transfer by requesting an interrupt. This avoids putting the processor into a waiting loop.

An external device requests an interrupt by asserting the INTREQ line to the low state. If no higher priority request (e.g. a RESET action, control panel interrupt request, or direct memory access request) is active when the computer finishes executing the current instruction, the machine grants the interrupt request at that time if the interrupt system is enabled. (The interrupt system is enabled whenever the Interrupt Enable Flip-Flop (IEFF) in the IM6100 is set, and the IM6102 Interrupt Inhibit Flip-Flop is cleared).

The timing diagram for an interrupt request/grant is shown in Figure 3-6. In the first cycle after an interrupt is granted, the processor stores the current Program Counter in memory location $0000_8$ of Field 0. (This location holds the "return address" the computer needs so that it can return to the "background" program at the end of the interrupt service routine.) In the next cycle, the machine fetches the first instruction in the interrupt service routine from location $0001_8$.

When the program counter is stored in location $0000_8$ the IEFF is reset and the IM6102 resets the extended address bits (EMA0 - EMA2) on the bus to the low state. This causes the initial instruction(s) in the interrupt service routine to be fetched from memory field zero. The Instruction Field register on the IM6102 is held in the cleared state until changed by the software. See Figure 3-7, the timing diagram for device vectoring. The IM6101 PIE uses the VREAD signal to implement a hardwired priority vectoring scheme.



NOTES: (1) Address $0000_8$ Field 0

(2) Don't care read

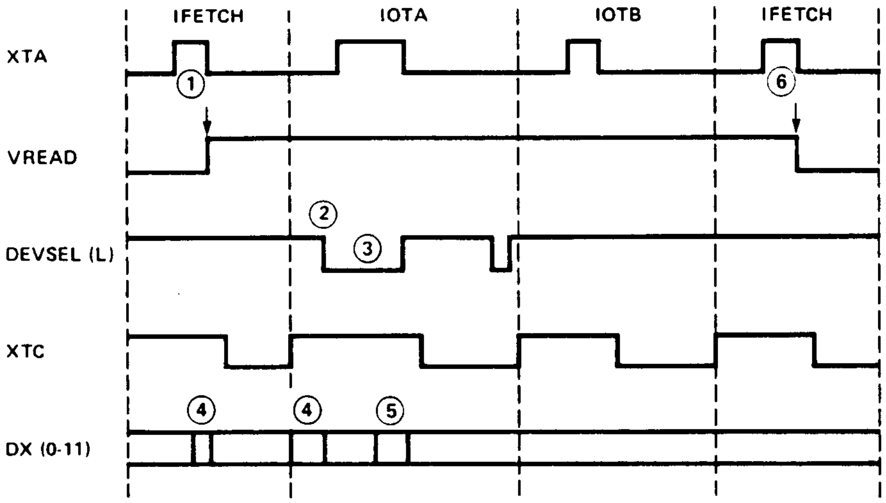
(3) PC written in LOC $0_8$ Field 0

(4) Address $0001_8$ Field 0

(5) Instruction fetch from location $0001_8$ Field 0

FIGURE 3-6

DEVICE INTERRUPT GRANT TIMING



NOTES: ① 6912 decodes vector instruction (IOT 6047).

② Bus priority "Daisy Chain" stabilizes.

③ Highest priority requesting device presents unique vector and drives C1 and C2 low.

④ IOT 6047.

⑤ Vector from device.

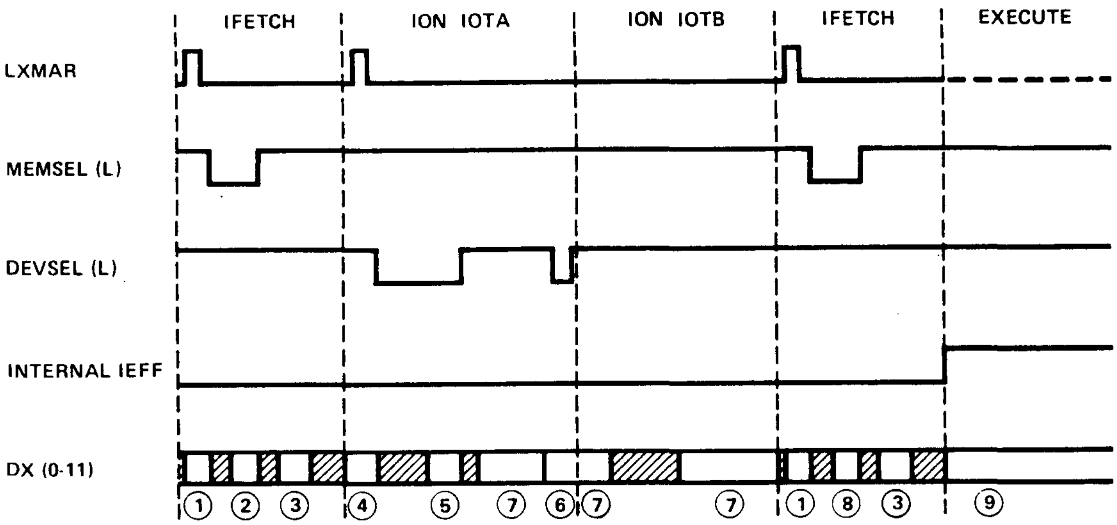
⑥ 6912 decodes non-vector instruction from location ⑤

FIGURE 3-7

DEVICE VECTOR TIMING

After an interrupt is granted, the IEFF is not set again until an ION or RTF instruction is executed. This gives the processor time to do its housekeeping chores before another interrupt is recognized. Typically, these chores include storing the Accumulator, Link and Program Counter for restoration later and executing a skip chain or vector instruction (IOT 6047) to find which device requested the interrupt. A timing diagram for setting the IEFF with an ION instruction is shown in Figure 3-8. Note that the IEFF is not actually set until the processor has fetched the next instruction after the ION. This guarantees one more instruction is executed after an ION before the next interrupt is recognized.

The IM6101 PIE device provides for a hardware prioritized interrupt system. This is implemented by logically connecting the three primary vector control signals (PRIN, PROUT, VREAD) to the IM6101.



NOTES:
① Instruction address
② Instruction (IOT 6001 in this case)
③ IM6102 current address register
④ Device address and control ($6001_8$ in this case)
⑤ Don't care device read, sample C0,C1,C2,SKP
⑥ Don't care device write
⑦ $7777_8$
⑧ Next instruction
⑨ DX data depends on next instruction

FIGURE 3-8

ION INSTRUCTION EXECUTION

## CONTROL PANEL INTERRUPTS

Control panel interrupts take several forms on the INTERCEPT. The resident control panel firmware interprets each control panel entry according to hardware indications on the CPU module. Although the user need not know the detailed operations of the firmware, a familiarity with the various types of control panel entry, discussed later in the chapter, is useful.

During the processing of a control panel interrupt, the processor ignores DMA and device interrupt requests. It also ignores further control panel interrupt requests.

The control panel interrupt system is not affected by the processor's Interrupt Enable Flip-Flop (IEFF); this logic applies only to device interrupts. Further, the processor IOT instructions ION and IOF do not affect the control panel interrupt system. (In fact, the instructions ION and IOF, if executed during a control panel interrupt, do not even affect the device interrupt system. IOF is totally disabled during control panel interrupts. ION has a special purpose, as described below .

A control panel interrupt can only be requested by logic on the CPU module asserting the IM6100 CPREQ pin. Since CPREQ is not available on the INTERCEPT bus, other modules cannot request control panel interrupts. The CPREQ pin is sampled by the processor in the last execute cycle of every instruction. If CPREQ is found asserted, the processor grants the control panel request by setting its internal Control Panel Flip-Flop (CNTRL F/F). While CNTRL F/F is set, further CPREQs are ignored, as are DMA requests and device interrupt requests.

When the control panel interrupt is granted the processor's Program Counter is stored in control panel memory location $0000_8$. This forms the return address so the processor can return to the main memory program at the end of the control panel interrupt service routine. The Program Counter is then set to $7777_8$, and the first instruction in the control panel routine is fetched from this control panel memory location during the next cycle.

During the execution of a control panel program, all instructions are fetched from control panel memory. Also, all directly addressed memory reference instructions and indirect JMS and JMP instructions reference control panel memory. Indirect AND,' TAD, ISZ and DCA instructions can either reference control panel memory or main memory. Memory selection for these instructions is accomplished by logic on the CPU board, under software program control (see Chapter 5, MAINM and CPMEM IOT instructions.)

The control panel routine ends with the execution of an ION instruction followed immediately by an indirect JMP, usually through control panel memory location $0000_8$. (See Figure 3-12.) The ION instruction has no effect on the device interrupt system, since it is executed in control panel memory, but it resets the CNTRL F/F midway through execution of the next instruction (the JMP I 0). Execution of an RTF instruction (IOT 6005) also resets the CNTRL F/F and can be used like ION to exit a control panel routine.

Assertion of the RESET line on the INTERCEPT bus (e.g., by activating the RESET switch on the rear panel) also forces an exit from control panel since RESET has a higher priority than CPREQ.

The following four paragraphs describe the various forms of control panel interrupts.

UART Framing Error

A Framing Error received from either serial I/O port of the INTERCEPT causes a control panel interrupt. Framing Errors result either from pressing the "BREAK" key on a terminal or incorrectly transmitting (e.g., at the wrong baud rate) serial data to the INTERCEPT. The control panel firmware displays system status and awaits user command input. See Figure 3-9.

Trapped Instructions

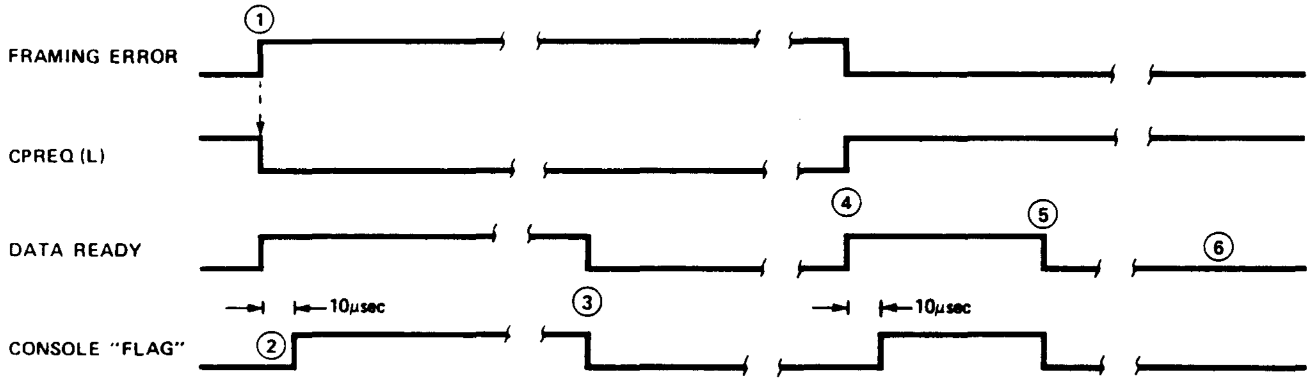
IOT instructions $6007_8$ through $6046_8$ and OSR cause a control panel interrupt when executed in main memory. The trapped IOTs include the DEC console terminal instructions and several utility instructions discussed in Chapter 5. The control panel firmware performs a specified function for each trapped instruction. The firmware emulates an OSR instruction, for example, by inclusively ORing the contents of the AC with the software switch register, leaving the result in the AC, and returning to main memory. See Figure 3-10.

Power-up, RESET and HLT

All forms of HLT executed in main memory cause a control panel interrupt immediately after the OPR2B cycle (see Figure 3-11). Also, when the RESET line is asserted low, the RUN pin on the IM6100 is pulled low forcing a control panel interrupt as soon as RESET is released. The RESET may be initiated by the hardware at power-up or asynchronously by the user. The control panel firmware differentiates between a HLT entry and a RESET entry and acts appropriately. If RESET, the firmware initializes PIE control registers and the control panel RAM it uses, clears the PIE sense inputs, and outputs status on the Primary Port. If HLT, the firmware outputs status and awaits user command input.

Power-failure (only with 6913 PFARM module)

Bus Power Requests generated by the 6913 cause an immediate control panel interrupt. The firmware saves important system status and branches to the user's power-fail vector location, where user software must complete the task of shutting down the system in a recoverable fashion. See the 6913 module documentation.



NOTES: (1) Input Data ("Break Key") with Framing Error immediately causes CPREQ.

(2) Terminal "Flag" delayed by 10μsec so control panel interrupt occurs before main memory programs can sense Flag.

(3) Control panel firmware recognizes Framing Error and clears "Flag" and Data Ready.

(4) User types command to control panel firmware, automatically resetting CPREQ and Framing Error while setting Data Ready and "Flag" (after 10μsec).

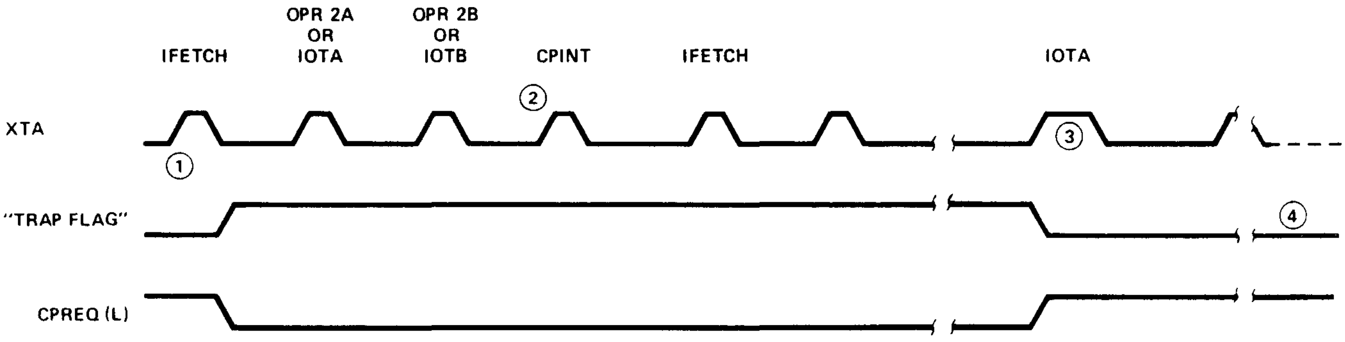
(5) Firmware reads command resetting Data Ready and "Flag".

(6) Main memory program resumes. Control panel entry invisible to main memory program since all signals are restored to pre-entry conditions.

NOTE: 10μsec delay is implemented for Primary Port Framing Errors only, although either port may cause Framing Error Control Panel Interrupt.

FIGURE 3-9

FRAMING ERROR CONTROL PANEL INTERRUPT



NOTES: (1) Main memory program Fetches OSR or 6007 ≤ IOT ≤ 6046 or any instruction in single instruction mode.
6912 decodes fetched instruction and sets the "Trap Flag" causing immediate CPREQ.

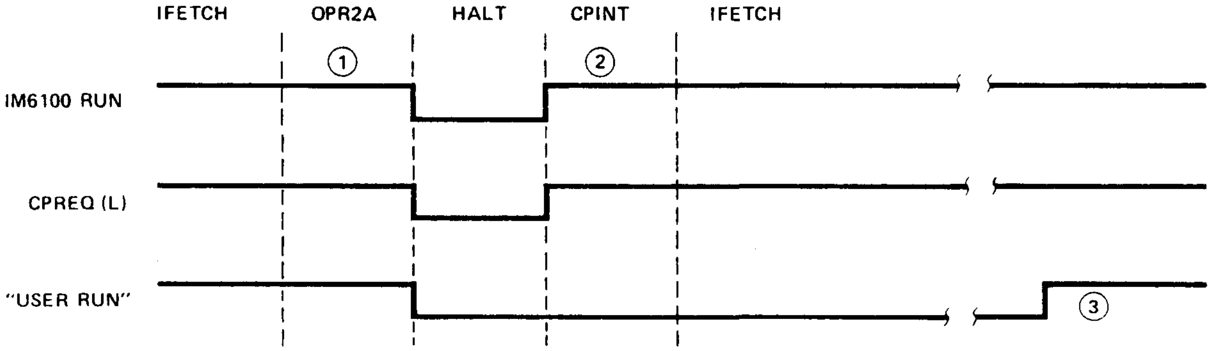
(2) Control panel entered, firmware assumes control.

(3) Firmware resets "Trap Flag" and CPREQ and processes trapped instruction (e.g., OSR emulation).

(4) Main memory program resumes after trapped instruction processed.

FIGURE 3-10

TRAPPED INSTRUCTION CONTROL PANEL INTERRUPT



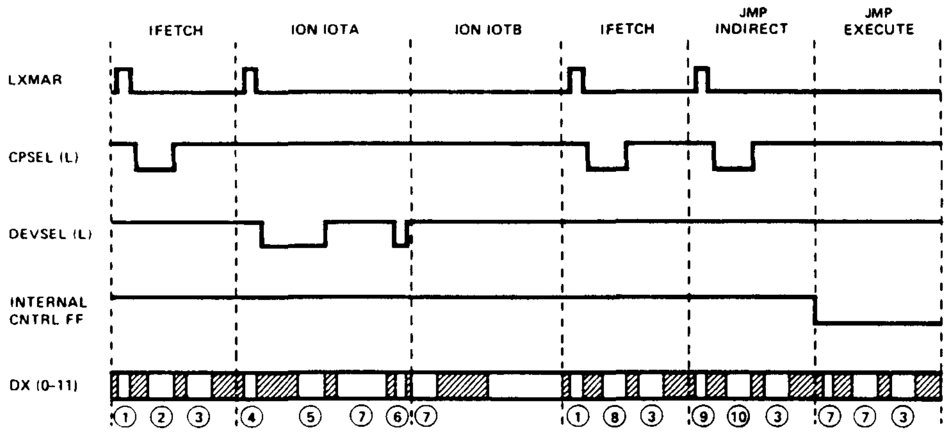
NOTES: (1) HLT Instruction (e.g., OPR 7402) Fetched and Executed in main memory immediately causes CPREQ.

(2) Control panel firmware assumes control. If CPU is Reset (e.g., at Power Up), this action is deferred (by increments of 5-State Cycles) until Reset is released.

(3) The Control panel firmware sets the "User Run" FF before main memory execution resumes.

FIGURE 3-11

HLT INSTRUCTION CONTROL PANEL INTERRUPT



NOTES: ① Instruction Address (CP Memory)
② Instruction (IOT $6001_8$ In This Case)
③ IM6102 Current Address Register
④ Device Address and Control ($6001_8$ In This Case)
⑤ Don't Care Device Read, Sample C0, C1, C2, SKP
⑥ Don't Care Device Write
⑦ $7777_8$
⑧ Instruction (MRI $5400_8$ In This Case)
⑨ Pointer Address ($0000_8$)
⑩ Effective Address (Contents of Pointer Address)

NOTE: CPSEL is an optional signal on the intercept bus, and not present in standard systems.

FIGURE 3-12

"ION; JMP I $0000_8$" EXECUTION IN CONTROL PANEL ROUTINE

## DIRECT MEMORY ACCESS OPERATIONS

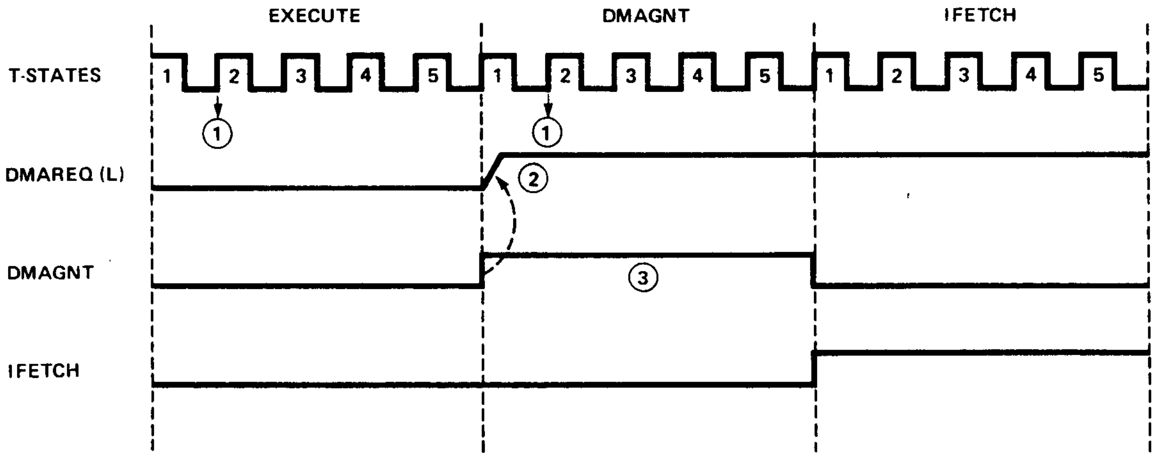
Very fast peripheral devices, such as conventional disk memories, usually transfer their data to and from the computer using direct memory access (DMA).

DMA port logic in a peripheral device interface requests DMA action by asserting the bus line DMAREQ to a low level. This line is sampled by the processor in the last execute cycle of each instruction. If the processor finds the DMAREQ line asserted, and there are no higher priority requests active (e.g., a RESET or control panel interrupt request), the processor suspends program execution at the beginning of the next cycle, and asserts the DMAGNT line on the bus.

When the DMAGNT line is asserted, the processor tri-states all bus control lines with which it normally drives the memory modules: all DX lines, MEMSEL, XTC, and LXMAR. Also, the logic tri-states the EMA0-EMA2 bus lines. The DMA port logic is then able to assert these lines to read or write data to/from any main memory location. (When the XTC line is tri-stated, it is pulled high by resistors on the memory module boards to prevent an inadvertent memory "write").

During each machine cycle of a DMA, the processor continues to sample the DMAREQ line on the rising edge of T2. When it finds this line has been released by the DMA port, the processor resumes execution of the program in the next cycle. Figure 3-13 shows the case where a single cycle is used by the DMA port in a single DMA action.

The IM6102 MEDIC device utilizes the 'unused' time slots in the processor cycle (i.e., states in which the processor is guaranteed not to use the DX lines) to provide a 'transparent' DMA port to read, write or refresh memory. This DMA access is guaranteed to occur at least once for every instruction since the IM6100 DX lines are tri-stated during the second half of the IFETCH cycle. The IM6102 also provides the 3 bits of extended address to expand main memory from 4K to 32K words. The IM6102 also has an on-chip crystal controlled programmable real time clock. For additional details refer to the data sheet on the IM6102 MEDIC.



NOTES: (1) DMAREQ sampled here.

(2) DMAREQ removed after DMAGNT

(3) While DMAGNT is high, the following Bus signals are tri-stated: DX0-11, XTC, XTC*, LXMAR, LXMAR*, MEMSEL, MEMSEL*, VREAD, UP, PROUT, EMA0-2.

DMA transfers actually entail some fairly complex cooperation between program software and DMA port logic. It should be noted that the DMAREQ bus line can be used by external devices as a level sensitive "pause" line. Asserting the DMAREQ line simply causes the processor to suspend program execution for an integral number of cycles until the line is released. Since the processor continues to drive the INTERCEPT bus with the timing signals XTA and XTB during a DMA or "pause" action, either of these pulses may be counted to determine the exact number of cycles during the pause.

## RESET

Activating the RESET switch on the INTERCEPT causes the processor Accumulator and Link to be cleared, the Program Counter to be set to $7777_8$, and the IM6102 Data Field and Instruction Field to be reset. All SELect lines remain high. The RESET switch asserts the RESET line bus low. Executing a CAF ($6007_8$) instruction also asserts the RESET line low.

## INTERCEPT BUS

The INTERCEPT bus contains 72 lines connected in parallel to every edge connector. The bus includes two groups of signals. One group is driven by the 6912 module with 74365 buffers and received with LS type gates. The other group of signals is pulled up to +5V with 1K resistors and driven by LS open collector gates. The only two exceptions are M DISABLE (bus pin 37) and the optional CPSEL (bus pin 35), which are driven by LS gates only.

The following is a summary of bus signals and their functions. The list is in alphabetical order.

C0<br>
C1<br>
C2 — During the execute phase of an IOT instruction these lines are asserted by the device interface logic to define the I/O operation taking place.

CPSEL* — Note a jumper must be installed to connect this signal to the bus. (See Chapter 10). The IM6100 asserts this line low to read data from or write data to control panel memory.

DEVSEL During an IOT execute cycle the processor asserts this line low to read from or write to a peripheral device interface.

DMAEN A peripheral device asserts this line to enable an IM6102 DMA transfer.

DMAGNT The processor asserts this line high when it grants a DMA cycle request. The assertion of DMAGNT tri-states the data bus and primary memory control signals and disables the DMA function of the IM6102.

DMAREQ Peripheral devices assert this line low to request bus control for DMA transfers not controlled by the IM6102.

DX0-DX11 These 12 multiplexed bi-directional lines carry addresses, instructions, and data between the processor, memory, and device interfaces. DX0 is the most significant bit and DX11 the least significant bit.

EMA0<br>
EMA1<br>
EMA2 The IM6102 defines these lines as a three bit extension to the IM6100 Program Counter and Memory Address Register enabling up to 32K of memory to be addressed.

IFETCH The processor asserts this line high throughout an instruction fetch cycle.

LXMAR The processor asserts this line high. The falling edge of LXMAR is used by memory and interfaces to latch addresses and/or instructions.

LXMAR* The IM6102 asserts this line high. The falling edge of LXMAR* is used by memory to latch addresses when the IM6102 DMA is used. (Used by memory only).

MEM DIS Asserted low to disable main memory reads and writes.

MEMSEL The processor asserts this line low to read from or write to main memory.

MEMSEL* The IM6102 asserts this line low to read from or write to main memory when the IM6102 DMA is used. (Used by memory only).

POW IN PROG The 6913 Powerfail Module asserts this line low while a power transition is processed by control panel or user software.

POW RES This line is asserted low by the "on-board" reset switch, the external reset switch, or the 6913 Powerfail Module after powerfail software finishes or when DC power is insufficient.

PREQ The 6913 Powerfail Module asserts this line low when it detects power failure or restoration.

PRIN
PROUT Daisy chained priority signals for the IM6101 PIE.

RESET This line is asserted low by the reset switches, or by the control panel program or the Powerfail Module after status has been saved.

SKP/INT Peripheral devices assert this line low to request interrupts. Also, peripheral interfaces assert this line low during an IOT instruction to increment the Program Counter (skipping the next instruction).

XTA
XTB
XTC The processor asserts these three lines high during various parts of each cycle. The status of these lines indicates the instantaneous state of the processor.

XTC* The IM6102 asserts this line high to indicate a memory read rather than a memory write. This signal is used instead of XTC on systems using the IM6102 DMA feature. (XTC* is used by memory and DMA interfaces).

UP DMA transfer signal generated by the IM6102.

V1 -12 volts.

V2 +12 volts.

VREAD The processor asserts this line high when it executes the IOT 6047. Bus peripherals should use this signal instead of INTGNT (IM6101). IOT 6047 is a unique instruction for propagating and evaluating the computer's prioritized vectoring system.

WAIT A peripheral device or memory module can assert this line low, causing the processor to pause for an integral number of system clock cycles while the peripheral or memory "catches up" with the processor. WAIT is also asserted by the 6912 during each IOT instruction to increase available read time by one state.

Table 3-5 shows the edge connector pin assignments for the INTERCEPT bus. A three or four character designation follows each signal name. The first character is either H or L depending upon whether the active level of the signal is HIGH or LOW. The second character is either O for "open collection" or T for "totem pole." The last characters signify the number of equivalent LS input loads presented to the signal by the 6912 module, independent of pull-ups:

Table 3-5

| | COMPONENT SIDE | | | REVERSE SIDE | |
|---|---|---|---|---|---|
| PIN | SIGNAL NAME | DESIGNATION | PIN | SIGNAL NAME | DESIGNATION |
| 1 | PROUT | HTO | 2 | PRIN | HTO |
| 3 | +5V | | 4 | +12V | |
| 5 | +5V | | 6 | XTB | HT2 |
| 7 | XTA | HT2 | 8 | UNUSED | |
| 9 | SKP/INT | LOO | 10 | SKP/INT | LOO |
| 11 | DEVSEL | LT1 | 12 | GND | |
| 13 | DMAGNT | HT16 | 14 | RESET | LO2 |
| 15 | PREQ | LO1 | 16 | RESERVED | |
| 17 | C2 | LOO | 18 | CO | LOO |
| 19 | DMAREQ | LOO | 20 | C1 | LOO |
| 21 | +12V | | 22 | GND | |
| 23 | +12V | | 24 | +5V | |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 25 | EMA2 | HT0 | 26 | MEMSEL* | LT1 |
| 27 | XTC | HT6 | 28 | WAIT | L00 |
| 29 | EMA1 | HT0 | 30 | MEMSEL | LT1 |
| 31 | RESERVED |  | 32 | GND |  |
| 33 | EMA0 | HT0 | 34 | VREAD | HT0 |
| 35 | RESERVED |  | 36 | +5V |  |
| 37 | MEM DIS | LT0 | 38 | LXMAR | HT0 |
| 39 | UNUSED |  | 40 | GND |  |
| 41 | UP | LT1 | 42 | DX8 | HT2 |
| 43 | DX7 | HT2 | 44 | DX9 | HT2 |
| 45 | DX0 | HT2 | 46 | GND |  |
| 47 | DX10 | HT2 | 48 | DX1 | HT2 |
| 49 | DMAEN | L00 | 50 | DX11 | HT2 |
| 51 | DX6 | HT2 | 52 | GND |  |
| 53 | RESERVED |  | 54 | DX2 | HT2 |
| 55 | CPSEL* | L02 | 56 | DX3 | HT2 |
| 57 | UNUSED |  | 58 | GND |  |
| 59 | DX4 | HT2 | 60 | LXMAR* | HT0 |
| 61 | IFETCH | HT1 | 62 | DX5 | HT2 |
| 63 | POW IN PROG | L01 | 64 | GND |  |
| 65 | UNUSED |  | 66 | RESERVED |  |
| 67 | POW RES | L00 | 68 | XTC* | HT2 |
| 69 | -12V |  | 70 | +5V |  |
| 71 | -12V |  | 72 | +5V |  |

# CHAPTER 4

# EXTENDED MEMORY, REAL TIME CLOCK, AND DMA PROGRAMMING

## INTRODUCTION

The IM6102 extends the addressing space of the system from 4K to 32K words. To perform this function, the EXTENDED MEMORY CONTROLLER maintains a 3 bit extended address which is decoded by the memory modules to select 1 of 8 memory fields each containing 4096 words of storage. These 4K fields start with FIELD 0 and progress to FIELD 7 when 32K of memory is used. All software communication with the controller is via programmed IOT instructions for which a summary is included in Table 8-1. For a more detailed description see the IM6102-MEDIC data sheet.

## MEMORY EXTENSION CONTROLLER

A simplified. block diagram of the Extended Memory Controller is represented in Figure 4-1. The diagram shows two 3-bit field registers: the Instruction Field, which defines the extended address for instructions and directly obtained operand addresses, and the Data Field, which extends indirectly obtained operand addresses. Programs can, therefore, use one field for instructions and address pointers and another field for data. The selection between Instruction and Data Fields is controlled by the DATAF signal generated by the IM6100. A discussion of the various register functions follows.

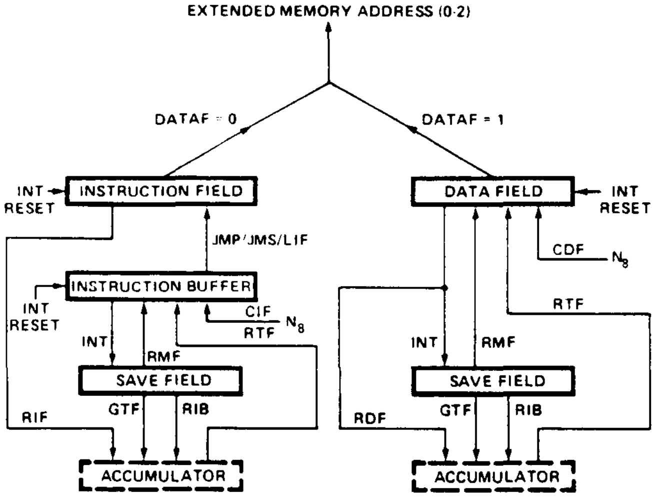


FIGURE 4-1

EXTENDED MEMORY CONTROLLER BLOCK DIAGRAM

## INSTRUCTION FIELD REGISTER (IF)

The IF is a 3-bit register that extends the Program Counter (PC). The IF, however, is not incremented when the PC goes from $7777_8$ to $0000_8$. The contents of the IF determine the field from which all instructions are taken. Operands for all directly addressed memory reference instructions also come from the Instruction Field. The pointer for all indirectly addressed memory reference instructions resides in the Instruction Field.

## DATA FIELD REGISTER (DF)

The DF is a 3-bit register which determines the memory field of operands in indirectly addressed AND, TAD, ISZ, or DCA instructions. The branch address for indirectly addressed JMS or JMP instructions is obtained from the Instruction Field. The Data Field register may be modified under program control.

## INSTRUCTION BUFFER REGISTER (IB)

The IB is a 3-bit register which serves as an input buffer for the Instruction Field (IF) register. All programmed modifications of the IF register are made through the IB register. The transfer from IB to IF takes place during the execute phase of the "next" JMP or JMS instruction or immediately upon execution of an LIF instruction. Using this feature, a program segment can execute an instruction to modify the IF and then "exit" the program segment before the actual modification of the IF. If CIF could change the IF directly, it would be impossible to execute the "next" sequential instruction following a Change IF instruction. The IB to IF transfer is inhibited if the JMP/JMS instruction is fetched from control panel memory, which is restricted to 4K, but the LIF instruction is used here to load the IF register from the IB register. This allows the control panel routines to execute transparently while the IB and IF differ and also allows the panel to extract or alter the status of the primary EMC registers.

## SAVE FIELD REGISTER (SF)

The SF is a 6-bit register in which the IB and DF registers are saved during an Interrupt Grant.

When an Interrupt occurs, the contents of IB and DF are automatically stored in SF (0-2) and SF (3-5), respectively, and the IF, IB and DF registers are cleared. The INTGNT (Interrupt Grant) cycle stores the "current" Program Counter (PC) in location $0000_8$ of Memory Field $0_8$ and the CPU resumes operation in location $0001_8$ of Memory Field $0_8$. The Instruction Field and Data Field of the program segment being executed by the CPU before the interrupt was acknowledged, are available in the SF register.

## INTERRUPT INHIBIT FLIP-FLOP

The INTREQ (Interrupt Request) line to the IM6100 is "gated" by the Interrupt Inhibit Flip-Flop so that, when the Instruction Field is changed under program control, interrupts are disabled until a JMP or JMS instruction is executed. Since the actual modification of the Instruction Field takes place only after the "next" JMP/JMS, inhibition of INTREQ ensures that the program begins operation in the "new" memory field before an Interrupt Request is granted.

Since Interrupt Requests are asynchronous in nature, a situation may arise in which an INTREQ would be generated when the IF and IB bits are different. The Interrupt Inhibit FF guarantees the structural integrity of the program segment.

## OPERAND FETCHING

Instructions are accessed from the current Instruction Field. For indirect AND, TAD, ISZ, or DCA instructions the operand address refers first to the Instruction Field to obtain an Effective Address which in turn refers to a location in the current Data Field.

Thus, DF is active only in the Execute phase of an AND, TAD, ISZ or DCA when it is directly preceded by an Indirect phase.

| ADDRESS MODE | IF | DF | AND, TAD, ISZ or DCA |
|---|---|---|---|
| Direct | m | n | Operand in field m |
| Indirect | m | n | Absolute address of operand in field m; operand in field n |

MNEMONIC OCTAL CODE OPERATION

GTF $6004_8$ GET FLAGS

Operation:

AC (0) <- LINK
AC (1) <- 0
AC (2) <- INTREQ Line
AC (3) <- INT INHIBIT FF
AC (4) <- INT ENABLE FF
AC (5) <- 0
AC (6-11) <- SF (0-5)

Description: LINK, INTREQ and INT ENABLE FF are internal to the CPU. The INT INHIBIT FF and SF are in the Memory Extension Controller.

Implementation: The CPU accepts the device data available on DX (0-11) and then bits 0, 2 and 4 are modified by the respective internal flags before the data is loaded into the Accumulator (AC).

The Memory Extension Controller must drive the C-lines (C0 = L, C1 = L). AC (1) and AC (4) are determined externally.

RTF $6005_8$ RETURN FLAGS

Operation:

LINK <- AC (0)
IB <- AC (6-8)
DF <- AC (9-11)

Description: LINK is restored. All AC bits are available externally during IOTA T6 to restore other flag bits. The internal Interrupt System is enabled. However, the Interrupt Inhibit FF is made active until the "next" JMS/JMP/LIF. The IB is transferred to IF after the "next" JMS/JMP/LIF. Note that the AC is not cleared.

CDF $62N1_8$ CHANGE DATA FIELD

Operation: DF <- $N_8$

Description: Change DF register to N ($0_8$-$7_8$). AC is unaffected.

CIF $62N2_8$ CHANGE INSTRUCTION FIELD

Operation: IB <-$N_8$

Description: Change IB to N ($0_8$-$7_8$). Transfer IB to IF after the "next" JMP/JMS/ LIF. The Interrupt Inhibit FF is active until the "next" JMP/JMS/LIF. AC is unaffected.

CDF, CIF $62N3_8$ CHANGE DF, IF

Operation: DF <-$N_8$

IB <-$N_8$

Description: Combination of CDF and CIF.

RDF $6214_8$ READ DATA FIELD

Operation: AC (6-8) <- AC (6-8) V DF

Description: OR's the contents of DF into 6-8 of the AC. All other bits are unaffected.

Implementation: DataX (0-5) and DataX (9-11) must be 0's. Drive C1 = L.

RIF $6224_8$ READ INSTRUCTION FIELD

Operation: AC (6-8) <- AC (6-8) V IF

Description: OR's the contents of IF into bits 6-8 of the AC. All other bits of the AC are unaffected.

Implementation: Same as RDF.

RIB $6234_8$ READ INTERRUPT BUFFER<br>
READ SAVE FIELD

Operation: AC (6-11) <- AC (6-11) V SF

Description: OR's the contents of SF into bits 6-11 of the AC. All other bits of the AC are unaffected.

RMF $6244_8$ RESTORE MEMORY FIELD

Operation: IB <-SF (0-2)

DF <-SF (3-5)

Description: The SF register saves the contents of the IB and DF when an interrupt occurs. This command is used to restore IB and DF when "exiting" from the interrupt service routine in another field.

Transfer IB to IF after the next JMP/JMS/LIF. The Interrupt Inhibit Flip-Flop is active until the next JMP/JMS/LIF.

LIF $6254_8$ LOAD INSTRUCTION FIELD

Operation: IF <-IB

Description: Transfer IB to IF and clear the Interrupt Inhibit FF.

## PROGRAMMABLE REAL TIME CLOCK

The programmable real time clock offers several ways to accurately measure and count intervals or events. Applications include real time data acquisition and process control systems.

Figure 4-2 shows a simplified block diagram of the Real Time Clock. There are three 12-bit registers: The Clock Enable register controls clock operation, counting occurs in the Clock Counter register, and communication between the AC and the Clock Counter register is via the Clock Buffer register.

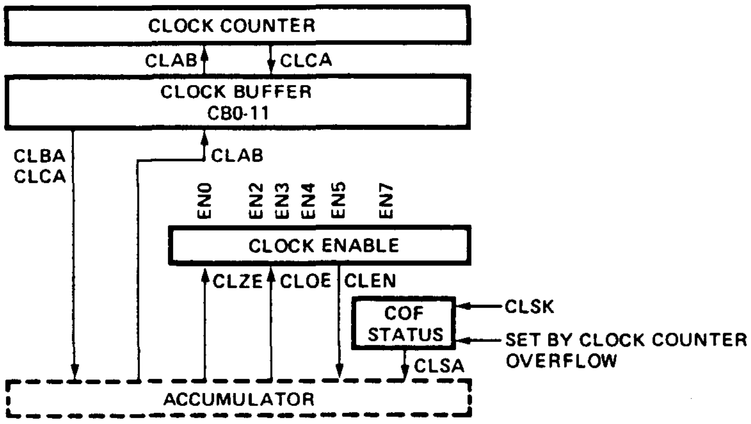


FIGURE 4-2

RTC REGISTERS

## CLOCK ENABLE REGISTER

This register controls the mode of operation, rate of counting, and the interrupt enable flip-flop for the real time clock. The clock enable register is loaded from the AC under program control. Bit assignments are described in Table 4-1.

TABLE 4-1

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EN0 | • | EN2 | EN3 | EN4 | EN5 | • | EN7 | • | • | • | • |

* Don't care for write and zero for read.

TABLE 4-1 continued

Where EN0 - When set to 1, enables clock overflow (COF flag) to cause an interrupt. Cleared by RESET, CAF.

EN2 - When reset to a 0-counter runs at selected rate. Overflow occurs every 4096 ($2^{12}$) counts. COF flag remains set until cleared by IOT 6135 (CLSA), CAF, RESET. When set to a 1, counter runs at selected rate. Overflow causes clock buffer to be transferred to the clock counter which continues to run. COF flag remains set until cleared with IOT 6135 (CLSA). Also cleared by RESET, CAF.

EN3, 4, 5 - Assuming 4 MHz crystal oscillator. Cleared by RESET, CAF. See below.

| BITS 3,4,5 | OCTAL | PERIOD | FREQUENCY |
|---|---|---|---|
| 0 0 0 | 0 | STOP | 0 |
| 0 0 1 | 1 | STOP | 0 |
| 0 1 0 | 2 | 20 msec | 50 Hz |
| 0 1 1 | 3 | 2 msec | 0.5 kHz |
| 1 0 0 | 4 | 200 μ sec | 5 kHz |
| 1 0 1 | 5 | 20μ sec | 50 kHz |
| 1 1 0 | 6 | 2 μ sec | 0.5 MHz |
| 1 1 1 | 7 | STOP | 0 |

EN7 - Inhibits clock prescaler when set to 1. Cleared by RESET, CAF

## CLOCK COUNTER REGISTER

This register is a 12-bit binary counter. It is loaded via the clock buffer register from the AC, and can be read into the AC via the clock buffer. Clock counter overflow is sensed by a skip instruction, a read status instruction, and/or an interrupt request.

## CLOCK BUFFER REGISTER

This 12-bit register is the communications link between the AC and the Clock Counter register. The Buffer is loaded from the AC with an initial value for the counter, then, this value is transferred to the counter once or upon every counter overflow, per enable register bit 2. The contents of the clock counter can also be read into the AC via the buffer register.

## CLOCK OVERFLOW FLAG

This flag is set by clock counter overflow. It causes an interrupt request and/or a "CLSK" skip depending on enable register bit 0. Its complement provides bit 11 of the clock interrupt vector. The flag is cleared by CAF, CLSA, and RESET.

| MNEMONIC | OCTAL CODE | OPERATION |
|---|---|---|
| CLZE | $6130_8$ | CLEAR ENABLE REGISTER PER AC<br>Description: Clear the bits in the clock enable register corresponding to those bits set in the AC. The AC is not changed. |
| CLSK | $6131_8$ | SKIP ON CLOCK INTERRUPT<br>Description: Skip the next instruction if clock interrupt condition exists. |
| CLDE | $6132_8$ | SET ENABLE REGISTER PER AC<br>Description: Set the bits in the clock enable register corresponding to those bits set in the AC. The AC is not changed. |

CLAB $6133_8$ TRANSFER AC TO CLOCK BUFFER

Description: Transfer the contents of the AC to the Clock Buffer, then transfer the contents of the Clock Buffer to the Clock Counter. The AC is not changed.

CLEN $6134_8$ READ CLOCK ENABLE REGISTER

Description: Transfer contents of the Clock Enable Register to the AC.

CLSA $6135_8$ READ CLOCK STATUS

Description: Clear AC, then transfer COF into AC bit 0 and clear COF.

CLBA $6136_8$ READ CLOCK BUFFER

Description: Clear the AC, then transfer the contents of the Clock Buffer to the AC.

CLCA $6137_8$ READ CLOCK COUNTER

Description: Clear the AC, transfer the contents of the Clock Counter to the Clock Buffer, then transfer the contents of the Clock Buffer to the AC.

CAF $6007_8$ CLEAR ALL FLAGS

Description: Clear COF flag (and also F7E, WOF flags), clock enable and clock buffer registers.

## SIMULTANEOUS DIRECT MEMORY ACCESS (SDMA)

The Simultaneous Direct Memory Access channel of the IM6102 can read, write, or refresh memory without affecting the performance of the INTERCEPT. The SDMA channel performs memory input/output or refresh by using the bus simultaneously with the microprocessor.

A simplified block diagram of the SDMA registers is shown in Figure 4-3. More detailed information on the SDMA channel is found in the IM6102 data sheet.

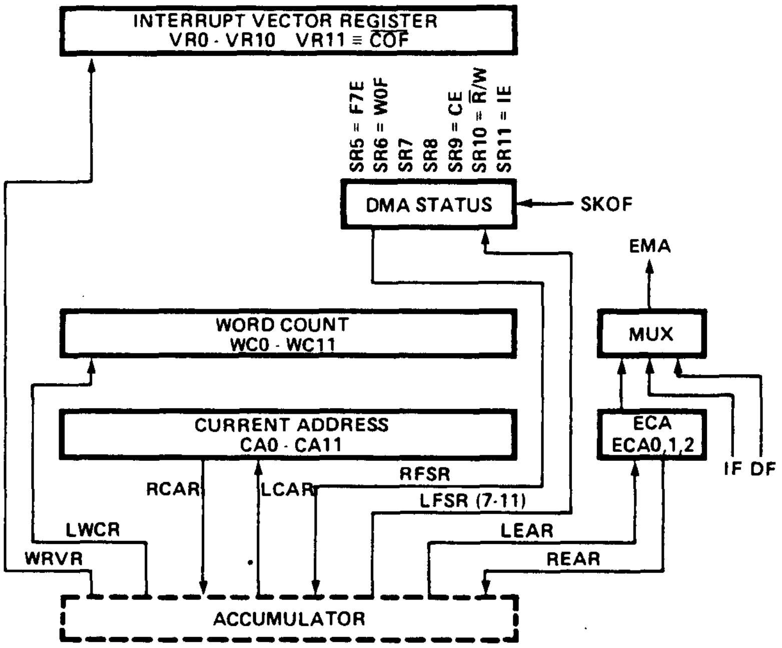


FIGURE 4-3

SDMA REGISTERS

A program communicates with the SDMA channel via four IM6102 registers. These registers are read and written via the AC.

CURRENT ADDRESS REGISTER (CA)

The Current Address Register is a 12-bit presettable binary counter. Prior to an SDMA transfer, it must be set to the first location accessed by the transfer. After each direct memory access, the contents of CA are incremented by one and the new contents of CA specify the address of the next direct memory access.

EXTENDED CURRENT ADDRESS REGISTER (ECA)

This register is a 3-bit extension of the Current Address Register. Prior to an SDMA transfer, the ECA must be set to the memory field of the transfer. Depending upon the setting of DMA Flags and Status Register (see Table 4-2) bit 9, the ECA register can be incremented when the CA register overflows.

## WORD COUNT REGISTER

The Word Count Register is a 12-bit presettable counter. Prior to a SDMA transfer, a program must load the WC with the two's complement of the number of words to be transferred. The SDMA operation begins when the WC is loaded unless Flags and Status Register bits 7 and 8 are ones (see below). After each direct memory access, the contents of WC are incremented by one. When WC overflow occurs, Flags and Status Register bit 11 is set and Interrupt Request becomes active if Flags and Status Register bit 11 is one (see Table 4-2).

## SDMA FLAGS AND STATUS REGISTER (FSR)

This register controls SDMA modes of operation and provides status information about the SDMA channel. The bit assignments are as follows:

TABLE 4-2

DMA FLAGS AND STATUS REGISTER BIT ASSIGNMENTS

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| • | • | • | • | • | F7E | WOF | SR7 | SR8 | CE | $\overline{R}$/W | IE |

where * = don't care for write and zero for read.

F7E Field 7 wrap around carry error; cleared by CAF, RFSR and RESET.

WOF Logic one indicates word counter overflow; cleared by CAF, LWCR and RESET

CE Carry enable from CA(0-11) to ECA; cleared by RESET

$\overline{\text{R}}$/W Logic one indicates DMA write (Port to Memory transfer). Cleared (DMA Read) by RESET

IE Enable interrupt when WC overflows or Field 7 error occurs; cleared by RESET

SR7, 8 00 Refresh mode

01 Normal mode

10 Burst mode

11 Stops DMA

## SDMA MODES

### REFRESH MODE

Reset clears FSR bits 7 and 8, forcing Refresh Mode. Used for refreshing dynamic memory.

### NORMAL MODE

Used for normal SDMA operation with static memory. A program must initialize CA and WC (ECA and vector registers are optional) to begin SDMA.

### BURST MODE

Used for SDMA operations with concurrent dynamic memory refresh. Same as Normal Mode, except SDMA hardware reverts to Refresh Mode whenever Word Count Overflow occurs.

### STOP MODE

No SDMA operations take place. This is the only mode where the WC register can be loaded without starting SDMA operations.

| MNEMONIC | OCTAL CODE | OPERATION |
|---|---|---|
| LCAR | $6205_8$ | LOAD CURRENT ADDRESS REGISTER (CA)<br>Description: The contents of the AC replace the contents of the CA and the AC is cleared. |
| RCAR | $6215_8$ | READ CURRENT ADDRESS REGISTER<br>Description: Contents of CA transferred to AC. |
| LWCR | $6225_8$ | LOAD WORD COUNT REGISTER (WC)<br>Description: Contents of AC are transferred to the WORD COUNT REGISTER, the AC is cleared WORD COUNT OVERFLOW (WOF) is cleared and DMA operation started. |
| LEAR | $62N6_8$ | LOAD IMMEDIATE TO EXTENDED CURRENT ADDRESS REGISTER (ECA)<br>Description: Field N of the IOT instruction is transferred to the Extended current address register. |
| REAR | $6235_8$ | READ EXTENDED CA<br>Description: Extended current address register contents OR's into bits 6, 7, 8, of AC. |
| LFSR | $6245_8$ | LOAD DMA FLAGS and STATUS REGISTER<br>Description: AC bits 7-11 are transferred to the DMA STATUS REGISTER and the AC is cleared. |
| RFSR | $6255_8$ | READ DMA FLAGS and STATUS REGISTER<br>Description: DMA Flags and Status Register bits are OR transferred into AC bits 5-11 and Field 7 wraparound error (F7E) is cleared. |
| SKOF | $6265_8$ | SKIP ON OVERFLOW INTERRUPT CONDITION |

Description: The PC is incremented by 1 if a word count register overflow interrupt condition exists causing next instruction to be skipped.

WRVR $6275_8$ WRITE VECTOR REGISTER

Description: AC bits 0-10 are transferred to the Vector Register and the AC is cleared.

CAF $6007_8$ CLEAR ALL FLAGS

Description: Clears F7E and WOF (and also COF), clock enable and clock buffer

# CHAPTER 5

# INTERCEPT INPUT/OUTPUT PROGRAMMING

The DEC console terminal emulation capabilities of the control panel program allow much existing PDP-8 software to run on the INTERCEPT without modification. The 6912 CPU module communicates with the console terminal via a PIE-UART serial interface. (See IM6100 Data Book for PIE-UART interface description). However, the INTERCEPT CPU produces a control panel trap each time a DEC terminal IOT is executed, and the control panel program then emulates the function using the PIE-UART interface.

This chapter first discusses the DEC terminal instructions and other trapped IOT's. Then, all IOT instructions recognized by the two INTERCEPT PIE's are listed.

## DEC TERMINAL INSTRUCTIONS

For compatibility with the DEC terminal interface the INTERCEPT's primary port is interrupt enabled when the system is powered-up, or upon a hardware reset, software reset, or CAF instruction. More detailed information on the DEC interface is found in DEC's "Small Computer Handbook 1973."

TABLE 5-1

| MNEMONIC | OCTAL | FUNCTION |
|---|---|---|
| KCF | 6030 | Reset keyboard data ready flag. |
| KSF | 6031 | Skip next instruction if keyboard data ready flag is set. |
| KCC | 6032 | Reset keyboard data ready flag, advance paper tape reader (if applicable), and clear the AC. |
| KRS | 6034 | Inclusive OR keyboard/reader data into AC. |
| KIE | 6035 | Enable or disable terminal interrupts per AC bit 11. (i.e., AC(11)=1 means enable. AC(11)=0 means disable). |
| KRB | 6036 | Clear AC, read keyboard/reader data into AC, reset data ready flag and advance reader (if applicable). |

| | | |
|---|---|---|
| TFL | 6040 | Set printer/punch transmit ready flag. |
| TSF | 6041 | Skip next instruction if printer/punch ready flag is set. |
| TCF | 6042 | Reset printer/punch ready flag. |
| TPC | 6044 | Output the AC to the printer/punch. AC is unaffected. |
| TSK | 6045 | Skip next instruction if terminal is interrupt enabled and the keyboard data ready flag and/or the printer ready flag is set. |
| TLS | 6046 | Output the AC to the printer/punch and clear the printer/punch ready flag. The flag is automatically set when the printer/punch can accept another character. |

Although the following instruction is not a terminal IOT, it affects the terminal operation.

| | | |
|---|---|---|
| CAF | 6007 | Reset the keyboard data ready flag and the printer ready flag. The CAF also resets the bus and some IM6100 and IM6102 registers. (See Chapter 3). |

## OTHER CONTROL PANEL TRAPS

The following instructions cause an immediate trap to the control panel. Some of the traps perform predefined operations and return to main memory while others transfer control to control panel RAM memory, where user written routines must take over.

TABLE 5-2

| MNEMONIC | OCTAL | FUNCTION |
|---|---|---|
| BRK | 6010 | Breakpoint trap instruction. Used by the control panel program to breakpoint user instructions. 6010 should not be executed by the user. |
| U200 | 6011 | Trap to control panel RAM location 200 octal. The user's registers are saved in the following control panel locations: |

| | |
|---|---|
| AC | location 20 |
| LINK | location 21 |
| DATA FIELD | location 22 |
| SWITCH REGISTER | location 23 |
| MQ | location 24 |

All other registers are left intact.

U201 6012 Trap to control panel location 201. See U200 instruction.

U202 6013 Trap to control panel location 202. See U200.

U203 6014 Trap to control panel location 203. See U200.

U204 6015 Trap to control panel location 204. See U200.

U205 6016 Trap to control panel location 205. See U200.

U206 6017 Trap to control panel location 206. See U200.

RCP 6020 Transfer contents of control panel memory to main memory as specified by three parameters following RCP instruction. General Form:

```
     RCP       /CONTROL PANEL TRAP
     PARAM1    /MAIN MEMORY BUFFER ADDRESS
               (IN FIELD OF CALL)
     PARAM2    /CONTROL PANEL BUFFER ADDRESS
     PARAM3    /NEGATIVE OF NUMBER OF WORDS
               TO TRANSFER
               /MAIN MEMORY PROGRAM RESUMES
               HERE
```

WCP 6021 Transfer contents of main memory to control panel memory as specified by three parameters following WCP instruction. Parameters are the same as for RCP instruction. Example:

```
     WCP       /CONTROL PANEL TRAP
     1000      /TRANSFER 200 WORDS FROM
      200      /MAIN MEMORY ADDRESS 1000 TO
     -200      /CONTROL PANEL ADDRESS 200.
               /MAIN MEMORY PROGRAM RESUMES
               HERE
```

CCP 6022 Call control panel subroutine at address specified by parameter following CCP instruction. General form:

```
     CCP        /CONTROL PANEL TRAP
     PARAM1     /ADDRESS OF CONTROL PANEL
                SUBROUTINE
                /MAIN MEMORY PROGRAM RESUMES
                HERE
```

JCP 6023 Jump to control panel address specified by parameter following JCP instruction. General form same as CCP instruction. Example:

```
     JCP        /CONTROL PANEL TRAP
     300        /JUMP TO CONTROL PANEL 300
```

PDN 6024 Power down trap. PDN should only be used with the 6913 Power Fail Module when a power failure condition exists. See the 6913 PFARM documentation.

PUP 6025 Power up trap. PUP should only be used with the 6913 Power Fail Module when a power restoration condition exists. See the 6913 PFARM documentation.

STC 6033 Used to improve IFDOS operating system performance at low clock frequencies. 6033 should never be executed by the user.

## INTERCEPT INPUT/OUTPUT INSTRUCTIONS

The INTERCEPT CPU uses two IM6101 PIE devices to communicate with the two serial I/O ports and support numerous control functions. These devices, referred to as "PIE A" and "PIE B," respond to IOT instructions in the range 6060 to 6117. The IM6101 PIE is documented fully in the IM6101 data sheet.

PIE A mainly controls the two serial I/O ports. Additionally, two "flag bits" in control register A are used for other functions. PIE A works with two IM6402 UART devices to form the two I/O ports.

On input, the UART receives serial data from a peripheral device and converts it to parallel data in the Receiver Register. The Data Ready (DR) signal becomes active, which is sensed by PIE A, and the UART data can be read into the AC with a PIE READ instruction.

On output, data from the AC is written to the UART Transmit Buffer Register in parallel form, and later transferred to the Transmit Register to be serialized and sent to a peripheral device. After the Transmit Buffer Register to Transmit Register transfer is made by the hardware, the Transmit Buffer Register Empty (TBRE) signal becomes active, which is sensed by PIE A, and more data can be written to the UART. Output is thus buffered by one character.

PIE A INSTRUCTIONS

TABLE 5-3

| MNEMONIC | OCTAL | FUNCTION |
|---|---|---|
| READS | 6100 | Inclusively OR Secondary Port UART Receiver Register data into AC. |
| WRITES | 6101 | Write AC data to Secondary Port UART Transmit Buffer Register. AC is unaffected. |
| SKIPRS | 6102 | Skip if Secondary Port Data Ready signal is active, meaning the UART has input data ready to be read into the AC. |
| SKIPXS | 6103 | Skip if Secondary Port Transmit Buffer Register Empty signal is active, meaning output data can be written to the UART from the AC. |
| RCRAA | 6104 | Inclusively OR control register A PIE A into the AC. |
| WCRAA | 6105 | Write AC to control register A PIE A. AC is unaffected. |
| MAINM | 6106 | Set Flag 1 of PIE A control register A, allowing control panel programs to reference main memory with indirect AND, TAD, ISZ, and DCA instructions. 6106 has no effect when executed from main memory. |
| CPMEM | 6107 | Clear Flag 1 PIE A control register A, forcing all memory fetches and references to control panel memory. 6107 should not be executed from main memory. |
| READP | 6110 | Inclusively OR Primary Port UART Receiver Register data into the AC. |
| WRITEP | 6111 | Write AC data to Primary Port Transmit Buffer Register. AC is unaffected. |

| | | |
|---|---|---|
| SKIPRP | 6112 | Skip if Primary Port Data Ready signal is active, meaning the UART has input data waiting to be read into the AC. |
| SKIPXP | 6113 | Skip if Primary Port Transmit Buffer Register Empty signal is active, meaning output data can be written to the Transmit Buffer Register from the AC. |
| WCRBA | 6115 | Write AC to control register B PIE A. AC is unaffected. |

PIE A IOT EXAMPLES

1). Reading a character into the AC from the Primary Port:

```
CLA                /MAKE SURE AC=0
SKIPRP             /CHARACTER READY?
JMP       .-1      /NO, WAIT
READP              /YES, READ CHARACTER
```

Note: Change SKIPRP and READP instructions to SKIPRS and READS for Secondary Port input.

2). Writing a character from the AC to the Primary Port:

```
SKIPXP             /READY FOR OUTPUT
                   CHARACTER?
JMP       .-1      /NO, WAIT
WRITEP             /YES, WRITE THE
                   CHARACTER
```

Note: Change SKIPXP and WRITEP instructions to SKIPXS and WRITES for Secondary Port output.

PIE A CONTROL REGISTERS

Figure 5-1 shows the bit configurations of PIE A control registers A and B. Control register A can be read or written under program control, while control register B can be written only. Both registers are initialized by control panel firmware with every hardware or software reset. The initial values shown correspond to the control register settings when control panel is exited.

User programs changing PIE control register values should use the following techniques to avoid disturbing other bits. To set a bit:

```
        CLA             /CLEAN UP
        TAD BIT         /GET BIT TO SET
        RCRAA           /"OR" IN CRA
        WCRAA           /WRITE NEW VALUE
```

To clear a bit:

```
        CLA             /CLEAN UP
        RCRAA           /GET CRA CONTENTS
        AND BITMASK     /MASK OUT UNWANTED BIT
        WCRAA           /WRITE NEW VALUE
```

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FL4 | FL3 | FL2 | FL1 | WP2 |  | WP1 |  | IE4 | IE3 | IE2 | IE1 |

CONTROL REGISTER A

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| SL4 | SL3 | SL2 | SL1 | SP4 | SP3 | SP2 | SP1 |

CONTROL REGISTER B

FIGURE 5-1

CONTROL REGISTERS A AND B

## CONTROL REGISTER A

FL4 Secondary Port RS-232 handshake. If 0, RS-232 signal Carrier Detect (Secondary Port connector pin 8) active.

FL3 Secondary Port RS-232 handshake. If 0, RS-232 signal Clear to Send (Secondary Port connector pin 5) active. Secondary Port connector pins 11 (RS-232 Supervisory Transmit) and 20 (RS-232 Data Terminal Ready) must be logical HI (+3V to +25V) or PIE A does not sense TBRE on the Secondary Port.

FL2 Switch strap select. If 0, OSR references 6912 CPU switch strap settings. If 1, OSR references software switch register. (Set by "S" command). Initially 1.

FL1 Control Panel enable. If 0, all memory fetches and references are from control panel memory. If 1, indirect AND, TAD, ISZ, and DCA instructions executed in control panel reference main memory. Initially 1.

WP2 Primary UART write pulse polarity. Must always remain 0.

WP1 Secondary UART write pulse polarity. Must always remain 0.

IE4 Primary Port transmit interrupt enable. If 0, Transmit Buffer Register Empty signal does not generate Interrupt Request. If 1, TBRE signal does generate Interrupt Request. Initially 1 for DEC interface compatibility.

IE3 Primary port receive interrupt enable. If 0, Data Ready signal does not generate Interrupt Request. If 1, DR signal generates Interrupt Request. Initially 1 for DEC interface compatibility.

IE2 Secondary Port transmit interrupt enable. See IE4. Initially 0.

IE1 Secondary Port receive interrupt enable. See IE3. Initially 0.

## CONTROL REGISTER B

SL4 Primary Port transmit sensing definition. If 0, the PIE is edge sensitive to the UART TBRE signal. If 1, the PIE is level sensitive to TBRE. Initially 0.

SL3 Primary Port receive sensing definition. If 0, the PIE is edge sensitive to the UART DR signal. If 1, the PIE is level sensitive to DR. Initially 0.

SL2 Secondary Port transmit sensing definition. See SL4. Initially 0.

SL1 Secondary Port receive sensing definition. See SL3. Initially 0.

SP4 Primary Port transmit sensing polarity. If 0, the PIE senses a low UART TBRE signal, and if 1, a high TBRE is sensed. Initially 1.

SP3 Primary Port receive sensing polarity. If 0, the PIE senses a low UART DR signal, and if 1, a high DR is sensed. Initially 1.

SP2 Secondary Port transmit sensing polarity. See SP4. Initially 1.

SP1 Secondary Port receive sensing polarity. See SP3. Initially 1.

PIE B supports various control functions of the CPU. For example, several conditions can create control panel interrupts. If the user types "BREAK" on a terminal connected to an I/O port, the UART generates a Framing Error, which in turn creates a control panel interrupt. The Framing Error signal is also sensed by PIE B so the control panel firmware can determine the nature of the control panel entry.

PIE B also controls the baud rate of the Secondary Port, the Primary Port Reader Run Relay, the hardware Single Step function, and sensing of Power Request, control panel traps, and the Switch Select signal for OSR emulation.

PIE B INSTRUCTIONS

TABLE 5-4

| MNEMONIC | OCTAL | FUNCTION |
|---|---|---|
| CLRTRP | 6060* | Clear control panel trap requests. Control panel traps occur when a DEC terminal interface instruction, OSR, or other trapped instruction. (See Tables 5-1 and 5-2). |
| RUNHLT | 6061* | Toggle the IM6100 RUN/HALT flip-flop. This allows control panel firmware to ensure the run condition when main memory programs are started. |
| SKPTRP | 6062* | Skip on control panel trap request. The trap request is not cleared. See CLRTRP instruction. |
| SNPRQ | 6063* | Skip if no Power-fail Request signal is present. The Power-fail Module, if present, generates a Power-fail Request when it detects deficient AC power. |
| RCRAB | 6064 | Inclusively OR control register A PIE B into the AC. |
| WCRAB | 6065 | Write AC to control register A PIE B. AC is unaffected. |
| RESET | 6070 | Issue Bus Reset. The IM6100, IM6102, and IM6402's are <u>not</u> reset. AC is affected in an undefined manner. |
| SETRDR | 6071 | Enable Reader Relay. |
| SKPOSR | 6072* | Skip on OSR control panel trap request. The OSR trap is cleared. |
| SKPFRE | 6073* | Skip on Framing Error control panel interrupt. Typing "BREAK" on a terminal connected to either I/O Port causes a Framing Error on the respective UART, which forces control panel entry. Framing Error is cleared when any "non-BREAK" character is typed. |
| WCRBB | 6075 | Write AC to control register B PIE B. AC is unaffected. |

* Instructions followed by asterisks should only be used in control panel programs.

## PIE B CONTROL REGISTERS

Figure 5-2 shows the bit assignments for PIE B control registers A and B. Control register A is "read/write," while control register B is "write-only." Both registers are initialized during a hardware or software reset by the control panel firmware. The initial values shown correspond to the control register settings during control panel exit.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FL4 | FL3 | FL2 | FL1 | WP2 |  | WP1 |  | IE4 | IE3 | IE2 | IE1 |

CONTROL REGISTER A

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| SL4 | SL3 | SL2 | SL1 | SP4 | SP3 | SP2 | SP1 |

CONTROL REGISTER B

FIGURE 5-2

CONTROL REGISTERS A AND B

## CONTROL REGISTER A

FL4-FL1 Secondary Port baud rate, according to the following table:

TABLE 5-5

TRUTH TABLE FOR RATE SELECT BITS

| FL4 | FL3 | FL2 | FL1 | Output Rate |
|---|---|---|---|---|
| L | L | L | L | 38400 Baud |
| L | L | L | H | 38400 Baud |

|  |  |  |  |  |
|---|---|---|---|---|
| L | L | H | L | 50 Baud |
| L | L | H | H | 75 Baud |
| L | H | L | L | 134.5 Baud |
| L | H | L | H | 200 Baud |
| L | H | H | L | 600 Baud |
| L | H | H | H | 2400 Baud |
| H | L | L | L | 9600 Baud |
| H | L | L | H | 4800 Baud |
| H | L | H | L | 1800 Baud |
| H | L | H | H | 1200 Baud |
| H | H | L | L | 2400 Baud |
| H | H | L | H | 300 Baud |
| H | H | H | L | 150 Baud |
| H | H | H | H | 110 Baud |

Initially, FL4 to FL1 are set to 1011 for 1200 baud.

WP2 Reader Relay write pulse polarity. If 0, relay is active low. If 1, relay is active high. Initially 0.

WP1 Single Step write pulse polarity. If 0, control panel entry is forced after exactly one main memory execution. (Hardware single instruction). If 1, normal main memory execution is allowed. Initially 1.

IE4-IE1 Not used, set to 0.

CONTROL REGISTER B

SL4 Framing error sensing definition. If 0, the PIE is edge sensitive to the UART FE signal. If 1, the PIE is level sensitive to FE. Initially 0.

SL3 Switch Select sensing definition. If 0, the PIE is edge sensitive to the IM6100 SWSEL signal. If 1, the PIE is level sensitive to SWSEL. Initially 0.

SL2 Power fail Request sensing definition. If 0, the PIE is edge sensitive to the 6913-PFARM PREQ signal. If 1, the PIE is level sensitive to PREQ. Initially 1.

SL1 Trap Request sensing definition. If 0, the PIE is edge sensitive to Trap Requests. If 1, the PIE is level sensitive. Initially 1.

SP4 Framing Error sensing polarity. Initially 0. If changed to 1 a false Framing Error could be sensed upon the next control panel entry, and the proper operation of the system may be impaired.

SP3 Switch Select (OSR) sensing polarity. Initially 0. If changed to 1, a false OSR will be sensed upon the next control panel trap.

SP2 Power-fail Request sensing polarity. Initially 1. If changed to 0, a false power failure will be sensed upon the next control panel entry.

SP1 Trap Request sensing polarity. Must be 0, or control panel entries are misinterpreted. Initially 0.

# CHAPTER 6

## SOFTWARE

The economies offered by low cost high performance microprocessors have opened new fields of dedicated computer applications. However, because of the lack of adequate software and applications support for microprocessors, microcomputer based systems have required substantial engineering investment on the part of the user. The lack of adequate software, general utility programs, mathematical routines and executive system software, increases the cost and lengthens the development time of a system by at least an order of magnitude when compared with minicomputer based systems. Since the investment in cost and time associated with the extensive engineering development must be off-set, microprocessors have been most cost effective only in large production volume systems.

The architecture, design and technology features of the IM6100 Microprocessor overcomes many of the problems associated with the current microprocessor designs. The IM6100 recognizes the instruction set of the DEC PDP-8/E minicomputer. The PDP-8 instruction set was chosen for a variety of reasons; the software support, efficient memory utilization, straightforward, yet, powerful instruction set and flexible input-output instructions. The DEC PDP-8 has the most well known machine organization and instruction set with more software support than any existing minicomputer system. System designers, familiar with conventional minicomputer hardware can now develop new microcomputer systems using the IM6100 with a minimum of time and effort.

The Digital Equipment Corporation Distribution Centers maintain a library or more than one thousand fully documented and developed programs for the PDP-8 family of minicomputers. A list of available software for the PDP-8 can be obtained from the Software Distribution Centers. Additional programs and applications packages are available from DECUS, the DEC User's Society. DECUS is a nonprofit user's group--the second largest such group, worldwide--that sponsors technical symposia, publishes a periodic newsletter and maintains a library of more than 1200 programs for the various DEC computers. A complete catalog of available programs may be obtained from the society.

The INTERCEPT and the PDP-8/E are software compatible. However, the Extended Arithmetic Element, EAE, and the user Flag, UF, options of the PDP-8/E cannot be used with the IM6100. The EAE is used for hardware multiply/divide and the UF for timesharing. Like the PDP-8, an Extended Memory Control element can be used with the IM6100 to extend its addressing capacity from 4K to 32K. The extended memory control element is standard on INTERCEPT systems.

For more information on DECUS Software, please refer to the application note, "DCAN001--DECUS PDP-8 Software Program Library".

## INTERCEPT SOFTWARE

### INTERCEPT FLOPPY DISC OPERATING SYSTEM -6970-IFDOS

The software components of the IFDOS consist of the following:

A file system which maintains a catalog of user files on floppy disc and performs file handling and input/output operations as specified by the user.

A Keyboard monitor which provides communication between the user and the operating system thereby enabling simple commands to enter and delete files in the user catalog, transfer files between memory and mass storage, print the user file catalog and call system programs.

An easy to learn text editor which allows the user to create and modify ASCII text at the console terminal.

An extremely fast and flexible assembler which accepts source programs created by the editor and produces binary output for subsequent loading and execution.

A binary loader which loads and executes assembler output files and facilitates loading of existing binary paper tapes.

A flexible "BATCH" program which executes files of operating system commands and greatly simplifies repetitive operations.

Numerous utility programs for absolute block copying and dumping of floppy discs, system date handling, control of system parameters and printing of system program catalogs.

## INTERCEPT DIAGNOSTIC SOFTWARE

Programs to test the processor, extended memory controller, memory and the Teletype interface (IDIAG-1) are supplied with the INTERCEPT system.

All the option modules (6970-IFDOS, etc.) have their own diagnostic programs, which are supplied as part of the hardware.

## PDP-8 SOFTWARE

This section contains brief descriptions of a selection of PDP-8/E programs and software packages. This is not, by any means, an extensive summary of all available software. It gives emphasis only to the standard PDP-8 programs which can run on a standard 4K INTERCEPT.

### PDP-8/E EXTENDED SOFTWARE KIT (QF081-AC)

The basic PDP-8/E Paper Tape Software Kit assists the user to create and edit programs and to debug and correct programs after assembly or compilation. Two handbooks, "PDP-8/E Small Computer Handbook" and "Introduction to Programming," are available with this software package. The Small Computer Handbook provides extensive technical information concerning hardware options, interfacing and system operation of the PDP-8 family of computers. Introduction to Programming deals specifically with the fundamentals of machine and assembly language programming on a small machine. A detailed description of the various PDP-8 loaders, verifiers, duplicators, conversion and printing routines is given in the PDP-8 family utility routines handbook, available with the basic software package.

#### SYMBOLIC EDITOR

The Symbolic Editor is used to create and modify symbolic (source) program tapes from the Teletype keyboard eliminating the tedious task of preparing source program tapes off-line. The Editor is fully interactive. The editing changes may be verified and recorrected, if necessary. The Editor includes a search feature to scan the text for occurrences of a specified character. Other commands permit blocks of text to be inserted, deleted, appended, listed or changed. The Editor is documented in Chapter 5 of the Introduction to Programming.

## PAL III ASSEMBLER

PAL III is a three pass Assembler designed for the PDP-8 family of computers with 4K words of memory. During the first pass of the assembly, all user symbols are defined and placed in the Assembler Symbol Table. During the second pass, the binary equivalent of the input source language, is generated and punched. The Assembler's third pass, which is optional, produces a printed assembly listing of the program instructions with the location, generated binary and source code side by side on each line. The binary tape output of the second pass can be loaded into the computer for execution.

The DEC manual, entitled "4K Assemblers," contains descriptions of two PDP-8 4K Assemblers, the most basic of which is PAL III. In addition to PAL III, the document also discusses the MACRO-8 Assembler, which is similar to the PAL III with some additional features such as user defined macros, double precision integers, floating point constants, arithmetic and Boolean operations, literals, text facilities, etc. However, the MACRO-8 does not have as large a symbol table capacity as the PAL III.

## MATHEMATICAL ROUTINES

The 23-bit Floating Point Package (FPP) provides an easy means of performing basic arithmetic operations such as addition, subtraction, multiplication and division using floating point numbers. It also provides extended function capabilities for the computation of natural logarithms, exponential functions, basic trigonometric functions and the like. The 23-bit FPP maintains a high degree of precision and greatly facilitates I/O operations in floating point notation. Chapter 8 of the Introduction to Programming describes the functional features of the 23-bit FPP.

## ADVANCED PROGRAMMING LANGUAGES

### FOCAL-8 (DEC-8E-LFOCA-A-PB, DEC-08-LFL8A-A-D) (IS-LFOCA)

FOCAL-8 is an interactive algebraic language developed specifically for the PDP-8/E。 FOCAL's desk calculator mode of operation makes the full computational power of the computer available to the user in response to simple sentence structured keyboard commands。 FOCAL is similar to BASIC and FORTRAN in many respects, however, it is more easily learned. The dynamic combination of computational capability and simplicity makes FOCAL-8 an ideal language for on-line problem solving without having to master a complex programming language. FOCAL requires only 4K words of memory, yet, it offers a full range of mathematical functions, extendable I/O and versatile self-editing capabilities.

### FORTRAN

DEC makes available two forms of paper tape FORTRAN for the PDP-8. One is for 4K machines and the other for 8K or larger machines.

### BASIC

DEC offers a standalone paper tape 8K BASIC interpreter. Other versions of BASIC are available from DECUS.

### ALGOL

A 4K ALGOL is also available from DECUS.

The user should reference the DEC PDP-8 software catalog and the DECUS PDP-8 software catalog for a list of all the available software.

# CHAPTER 7

## HARDWARE OPTIONS

The following additional hardware modules are available from Intersil.

### 6905-WIREWP

The wirewrap module permits the user to prototype and incorporate user interfaces to the INTERCEPT system. The module provides for all standard dual-in-line pin spacings.

### 6906-EXTEND

The extender module enables the user to extend any INTERCEPT cards for servicing, testing and debugging.

### 6909-RRELAY

This module provides remote control of the Teletype paper tape reader (Appendix G).

### 6970-IFDOS

The floppy disc operating system, designated 6970-IFDOS, together with the 4096 words of memory provided with INTERCEPT and an ASCII terminal (RS232 or 20ma) enable the user to rapidly develop software for an IM6100 CMOS microprocessor based system.

The hardware components of 6970-IFDOS consist of two completely interfaced flexible disc drive mechanisms with all electronics, power supplies, and cables necessary to add over four (4) million bits of "on line" mass storage capability to the INTERCEPT prototyping system. All components are contained in a single covered enclosure which can be rack mounted or placed on any flat surface. The interface module is inserted directly into the INTERCEPT bus and is connected to the disc system via a multi-conductor ribbon cable.

Some of the features of the system are:

IBM 3740 compatible media with multiple sources

Software compatible with the DEC RX8 Floppy Disc System for the PDP-8/E

Intelligent disc drive/controller formatter/interface communications with the ability to:

> Detect, identify, and correct errors resulting from mechanical, electrical, media or human means
>
> Completely format a diskette within industry standards

Automatic transparent self tests on disc related equipment are performed at times when system throughput is least affected

Flexible Programmed Input/Output for applications that require direct communications between user programs and the storage system.

For more detailed descriptions of the hardware and software features of the IFDOS system, the user is referred to the Intercept D10 Diskette Memory System Hardware Manual and the IFDOS Software Handbook.

# CHAPTER 8

## 6912-CPUCTRL: CENTRAL PROCESSOR UNIT WITH DUAL SERIAL I/O AND FIRMWARE CONTROL PANEL - HARDWARE DESCRIPTION

### INTRODUCTION

The 6912-CPUCTRL module is the nucleus of the INTERCEPT system. The 6912 is also a one-board stand-alone microcomputer. For clarity, the major functions of the 6912 are discussed separately in this chapter. The board positions of components mentioned are in parentheses. Throughout this chapter, the reader should refer to the 6912 schematic diagram at the end of the chapter.

### IM6100 and IM6102

The IM6100 and IM6102 perform most important functions. The IM6100 (1C) performs computation and generates most of the system's addressing and control signals. The IM6102 (2C) augments the addressing information of the IM6100 and creates many system control signals.

The system clock is generated by ICM 7209 (1AB), a 3.3 MHz crystal (Y3) and two capacitors (C7 and C8).

The clock for the IM6102 is provided by a 2 MHz crystal (Y1), a trim capacitor (C5), and two capacitors (C4 and C6).

Most of the 6912 internal bus signals are driven by CMOS. Pull-up resistors provide solid "one" levels on internal bus signals driven by Bipolar.

The external bus of the INTERCEPT is strictly Bipolar. Most 6912 external bus outputs are driven by 74365 buffers. External bus inputs are routed through 74LS257 devices for multiplexing OSR instructions, which can read all zeros or the auto-start straps.

Devices 4A, 7B, and 5C cause a wait cycle for all IOT instructions at read time (DEVSEL · XTC), allowing peripheral devices more time to respond to input signals.

Device 1AA controls bidirectional internal bus inputs. Its output is active whenever the IM6100 expects input from a bus module, and inactive when input is from internal 6912 circuitry.

Devices 4A, 3A, 2A, 7A, and 3B generate the BOUT signal, which controls bus outputs. BOUT is active for all addresses generated by the IM6100 or IM6102 and whenever the IM6100 is presenting write data at its DX pins.

Resistor pack 1B pulls up the internal DX lines, assuring true CMOS "1" levels to all 6912 CMOS devices.

Input/output control signals INTREQ, SKIP, CO, C1, and C2 generated on the 6912 module are buffered to the bus with low power Schottky open collector devices. The "on-board" bare drain outputs for these signals are pulled up to VCC by 2.7K resistors (4B).

## CONTROL PANEL

The "on-board" memory of the 6912 consists of 2048 words of ROM and 256 words of RAM. This memory resides in the unique control panel memory space of the IM6100. In INTERCEPT systems, the control panel memory program provides the only means of communication between the INTERCEPT and the human operator. The control panel memory also allows the 6912 module to "stand-alone" as a single-board microcomputer.

The control panel memory consists of two IM6312 1K x 12 ROMs (3E and 4E) and three IM6561 256 x 4 RAMs (3F, 4F, and 5F). Device 4E does address decoding for the RAMs via its RSEL (pin 1) output. Devices 6B and 5B, along with the F1 output (pin 32) of the IM6101 (2E) provide the chip select control for the control panel memory. Programming the F1 output to logic "0" forces all memory references to control panel by ANDing the IM6100 MEMSEL signal into the control panel memory chip selects. Main memory is simultaneously disabled via the MDISABLE bus signal (bus pin 37). This output is driven with a 74LS32 device. MDISABLE is forced inactive whenever IM6100 XTC is low to allow IM6102 Direct Memory Access to main memory while all IM6100 memory references are forced to control panel.

The control panel memory of the 6912 module may be disabled and the chip select routed to bus pin 55, making an "off-board" control panel memory possible.

## SERIAL INTERFACES

The 6912 CPU module includes two independent serial interfaces using one IM6101 device (2E) and two IM6402 devices (6F and 7F). Baud Rate Generators (3D and 4D), in conjunction with other components(C2, C3, R2, and Y2), provide up to 15 different transmission rates for each interface. Four "DIP" switches on the 6912 control the select inputs to device 4D, generating the clock for UART 7F which formats the data for the Primary Serial Port. The Flag outputs of IM6101 1E control the select inputs to device 3D to control Secondary Port timing. Thus, the baud rate of the Secondary Port is software controlled.

Signal conditioning for RS-232 input/output is generated by devices 1H and 2H. All serial data is routed through J1, a 20 pin ribbon cable connector on the 6912, to the connectors on the INTERCEPT enclosure.

Devices 5E, C10, and C11 cause a timing delay to the write outputs from device 2E, thus optimizing interface performance from a UART viewpoint. Devices 5E and 5C add one gating level to the Transmit Ready circuit of the Secondary Port allowing supervisory control over RS-232 transmissions on that port. IM6101 device 2E Flags 3 and 4 control two additional RS-232 outputs. Device 7B causes a fixed delay on the Primary Port Data Ready signal assuring the transparency of "BREAK" key control panel interrupts to software using the Primary Port. Specifically, Primary Port FRAMING ERROR propagates through 6B, 3A, 1AA, and 2A causing a control panel interrupt before DATA READY is sensed by main memory programs. Both UART control registers are strapped (via component side traces) for no parity and two stop bits per character.

## 20ma CURRENT LOOPS

Both of the serial I/O ports on the 6912 module operate as RS-232 interfaces. However, one port can be converted to 20ma current loop operation by installing jumpers on the external DP-25 connectors. Either port may operate as a current loop interface, but the other must remain RS-232.

The current loop circuit consists of three parts: transmit, receive, and reader current sources. The transmit current source consists of transistor Q2, diode CR3, and resistors R13 and R14. The input to the current source is one of the RS-232 output signals

J1-3 or J1-14. An "off-board" connection must be made between J1-6 and one of the above current source inputs. The circuit converts the voltage function of the RS-232 drivers (LM324, 2H) to a 20ma current when referenced to -12 volts. The 20ma output is routed "off-board" through J1-4.

Transistor Q3 and resistors R15 and R17 generate the receiver current source at J1-9. The keyboard section of a 20ma current loop terminal should be connected between the receiver current source (J1-9) and -12 volts (J1-11). A jumper on the external connector between J1-9 and one of the RS-232 input pins (J1-1 or J1-13) allows sensing of transitions in the interface. Capacitor C12 acts as a low pass filter. C12 must be replaced by a lower capacitance when the current loop interface baud rate exceeds 300 baud.

The reader control current originates from transistor Q1, resistors R11 and R12, and diodes CR1 and CR2. This current is at J1-7, again referenced to -12 volts. When device 7A senses a MARK to SPACE transition, it switches reader control current off.

The reader circuit can respond to either I/O port depending upon the strap option on device 7A pin 6. (See Chapter 10, "Selecting CPU Module Options").

## CONTROL LOGIC

Part of the circuitry on the 6912 module generates and supports control panel interrupts and the VREAD signal for device interrupt vectoring.

Device 6C, a preprogrammed IM5624 512 x 4 Bipolar PROM, along with logic elements on devices 6B, 5A, and 5C, decodes the 12 DX bus lines. Flip-flop 6A is clocked by gate 1AA with each main memory instruction fetch. Output 02 from 6C is active for the IOT instruction 6047. This output is clocked through 6A and buffered to bus pin 34. This VREAD signal, replacing the INTGNT signal on previous systems, acting with PROUT (bus pin 1), propagates the daisy chained interrupt priority network on the bus. All bus devices with vectored interrupt capability must route bus pin 2 into their priority control circuit (PRIN on IM6101). The output of the priority control circuit (POUT on IM6101) must drive bus pin 1. All cards on the bus not using the priority system must short bus pin 1 to bus pin 2. The IM6102 is the only device on the 6912 using the priority chain, and it has the highest priority in the system. The VREAD signal can not drive the INTGNT pin on the IM6102 because the extended memory control uses INTGNT, so the IM6102 responds with a vector upon the first IOT executed after an interrupt. An RS flip-flop (composed of gates in device 5A and a two-input NOR gate in device 3A) disables bus signal C2 generated by the IM6102 while in control panel, so vectoring cannot occur in control panel. This flip-flop, along with a "dummy IOT" in the control panel firmware, cancels the effect of the interrupt, but still allows the control panel/debugger to be used transparently on programs using interrupts.

OUTPUT 01 of device 6C is active for all trapped instructions (OSR and all IOT's in the range 6007-6046). Gate 2A combines output 01 and the hardware single instruction control signal generated by the W1 output of the IM6101 (1E). The output of gate 2A is latched by device 6A, then routed through devices 1AA and 2A to produce a CPREQ to IM6100 (1E) pin 5.

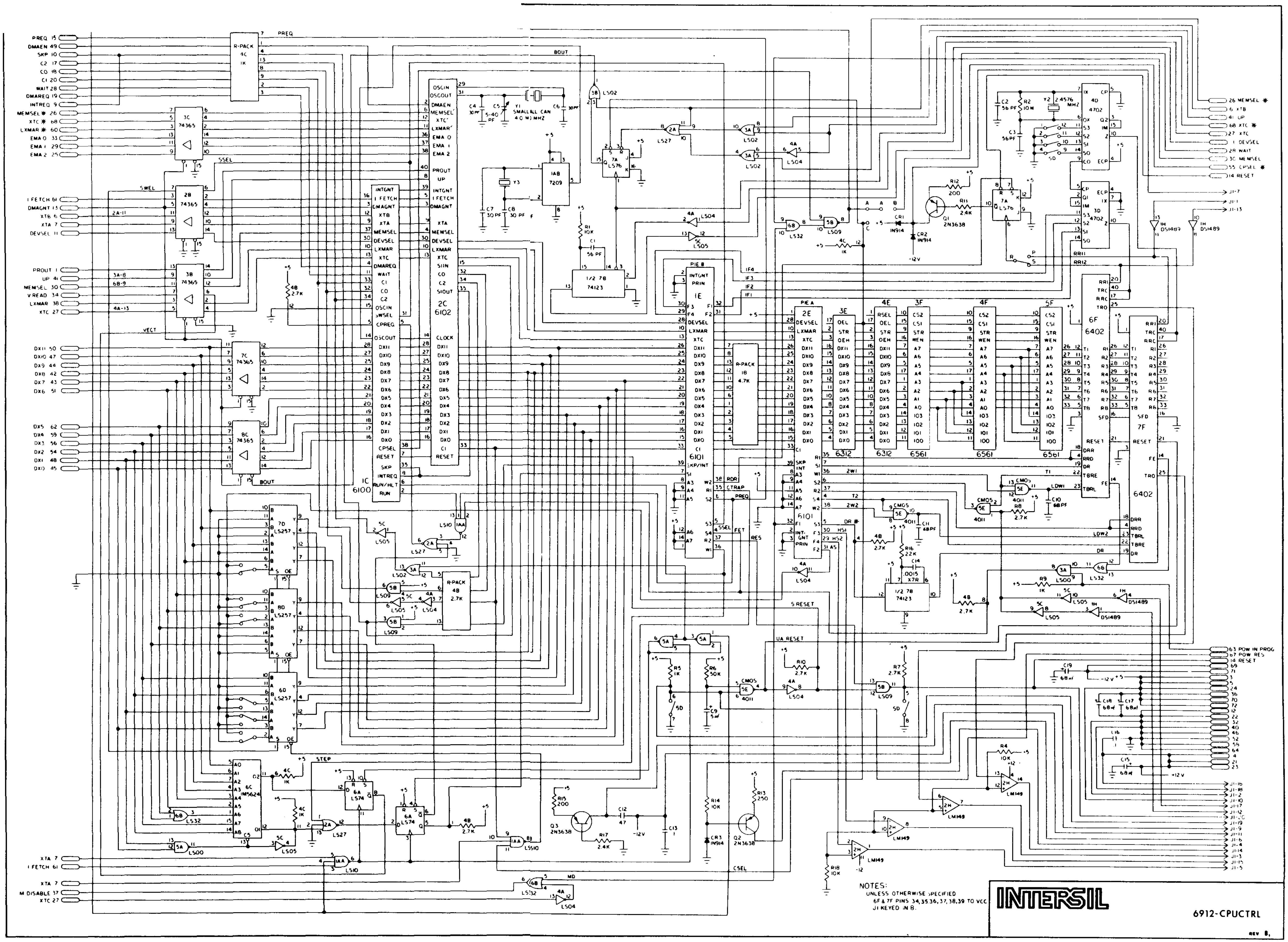
Trapped instructions occur in main memory programs while the hardware single instruction should originate only from control panel programs. The RUN/HALT input of the IM6100 (pin 6) is controlled by the control panel firmware via the same PIE output as the single instruction control.

Other control panel interrupts are caused by Framing Errors from either UART via devices 6B, 3A, 1AA, and 2A; by the IM6100 whenever it is halted via the RUN signal (pin 2); by the bus signal PREQ (bus pin 15). PREQ, with POW IN PROG (bus pin 63), and POWRES (bus pin 67), support the 6913 powerfail/auto-restart module, if present.

The INTERCEPT system can be RESET by the switch in position 5D of the 6912 module, the reset switch on the rear of the enclosure (which shorts J1 pin 19 to GND at J1 pin 2 or J1 pin 18), the 6913 power fail module via bus pin 67, or the RC timer (R6, C9, 5E) upon initial system power-up. The RESET conditions reset the 6912 circuitry and create a Bus Reset on bus pin 14 via logic in devices 4A and 5B. Control panel programs can also issue a Bus Reset with IOT 6070.

## SYSTEM MEMORY VOLATILITY

The 6912 module does not automatically protect main memory when power fails. Memory protection is automatic in systems equipped with the 6913 power-fail module. Users of systems without the 6913 module must ensure that bus pins 14 and 67 are grounded during power transitions for memory nonvolatility. This is accomplished by reseting the switch on the rear of the INTERCEPT enclosure or the switch in position 5D of the 6912 module. Control panel RAM cannot be protected under any circumstances. Power fail software must perform this function by moving the contents of control panel RAM into a protected memory area.



NOTES:
UNLESS OTHERWISE SPECIFIED
6F & 7F PINS 34,35,36,37,38,39 TO VCC
J1 KEYED IN B.

INTERSIL

6912-CPUCTRL

REV B.

# CHAPTER 9

## 6901-M4KX12C NONVOLATILE CMOS MEMORY MODULE - HARDWARE DESCRIPTION

Since the standard CMOS RAMs and ROMs manufactured by Intersil have tristate outputs and internal edge triggered address latches, address, data-in and data-out can be time multiplexed on the same lines, resulting in considerable reduction in the total number of lines bused without degradation in system performance.

### ADDRESS AND FIELD DECODING

The address information (AO-A9) to the IM6508 1K X 1 CMOS RAM devices need be valid only for a short duration when their STR input makes a negative transition. Since the IM6508 address strobe and chip select functions are provided on a single pin (STR), the address strobes are sent to the devices after decoding. The high order address bits (DXO and DX1) are latched into a Quad DFF (74LS75 - U12) when LXMAR makes a negative transition. The 4-input NAND gates (7420 - U13 and U14) then decode the latched address bits. When LXMAR is high, all the strobes are high and all the IM6508's are tristated. When LXMAR goes low, one of the strobes goes low enabling one row of IM6508's if the memory module is selected. A module is selected if the MDIS (line 37) is high and if EMA (0-2) is the complement of the field select switch setting (SW1-3 in U7). If MDIS is low, all main memory operations are disabled. This is a useful feature if one wants to selectively enable/disable main memory operations under control panel program control to implement non-standard control panel routines. The switch settings for field selection are as follows:

| FIELD | SW(1) | SW(2) | SW(3) |
|---|---|---|---|
| 0 | OFF | OFF | OFF |
| 1 | OFF | OFF | ON |
| 2 | OFF | ON | OFF |
| 3 | OFF | ON | ON |
| 4 | ON | OFF | OFF |
| 5 | ON | OFF | ON |
| 6 | ON | ON | OFF |
| 7 | ON | ON | ON |

Note that DX(0) and EMA(0) are the most significant address/data and extended address bits, respectively.

## ADDRESS AND DATA BUFFERING

Bipolar devices are used for buffering and decoding. Battery back-up is provided only for the CMOS devices. When system power is interrupted, transistors Q1 and Q2 turn off, isolating CMOS VCC from System VCC and the CMOS devices are on stand-by with the 3.6V VCC provided by the rechargeable Ni-Cad batteries.

Most low power Schottky devices (54/74 LS) have the interesting property that when their VCC is at GND, their outputs will be at GND also. Since LS devices are used for buffering the address and data-in lines to the memory devices, they are guaranteed to be at GND when the system VCC is off. If inputs to CMOS devices are allowed to float, both P and N channel transistors of the input inverters could be active simultaneously, increasing stand-by power dissipation considerably. Note that the NAND gates used for decoding the address strobes are TTL devices and their outputs float if their VCC is at GND. The strobe lines are pulled up to CMOS VCC with the pull-up resistors R3-R6 (10K).

## WRITE PROTECT

The entire memory can be write protected if the 4K DIS (line 35) is low. Only the upper 3K of memory is write protected when the 3K DIS (line 31) is low. The user can simulate 1K RAM - 3K ROM combinations with these two signals.

## MEMORY STAND-BY

One way to guarantee the contents of IM6508 memory devices in the stand-by mode is to ensure that their STR lines are held at CMOS VCC. CMOS devices U16 (40174) and U15 (4050) guarantee that no transitions occur on the STR lines when the system power goes on/off. The system Reset (line 14) must be low before DC power goes off. This is done by detecting the loss of any cycles on the AC power line. The Q output of U16 is asynchronously set high when Reset is low which, in turn, makes all the decoded outputs of U13 and U14 high. The Reset line must go high only after the DC power has been restored. The rising edge of the very first LXMAR pulse after the Reset line has gone high, enables U13 and U14 outputs. Note that the circuit design of a 54/74 NAND gate guarantees that if at least one of the inputs is held to GND, the output can never "glitch" to GND even if the VCC to the device ramps between +5V and GND. So the following constraints are satisfied to ensure the contents of the IM6508 series of synchronous CMOS RAMs:

1. STR line is held high and no STR transitions occur on stand-by.

2. At no time is the positive STR pulse width less than the specified minimum pulse width.

3. Whenever STR makes a negative transition, the address inputs are settled for the specified minimum set up and hold time.

The user may use SW(4) to hold the Reset line low when the module is removed from the system bus.

## DC CHARACTERISTICS

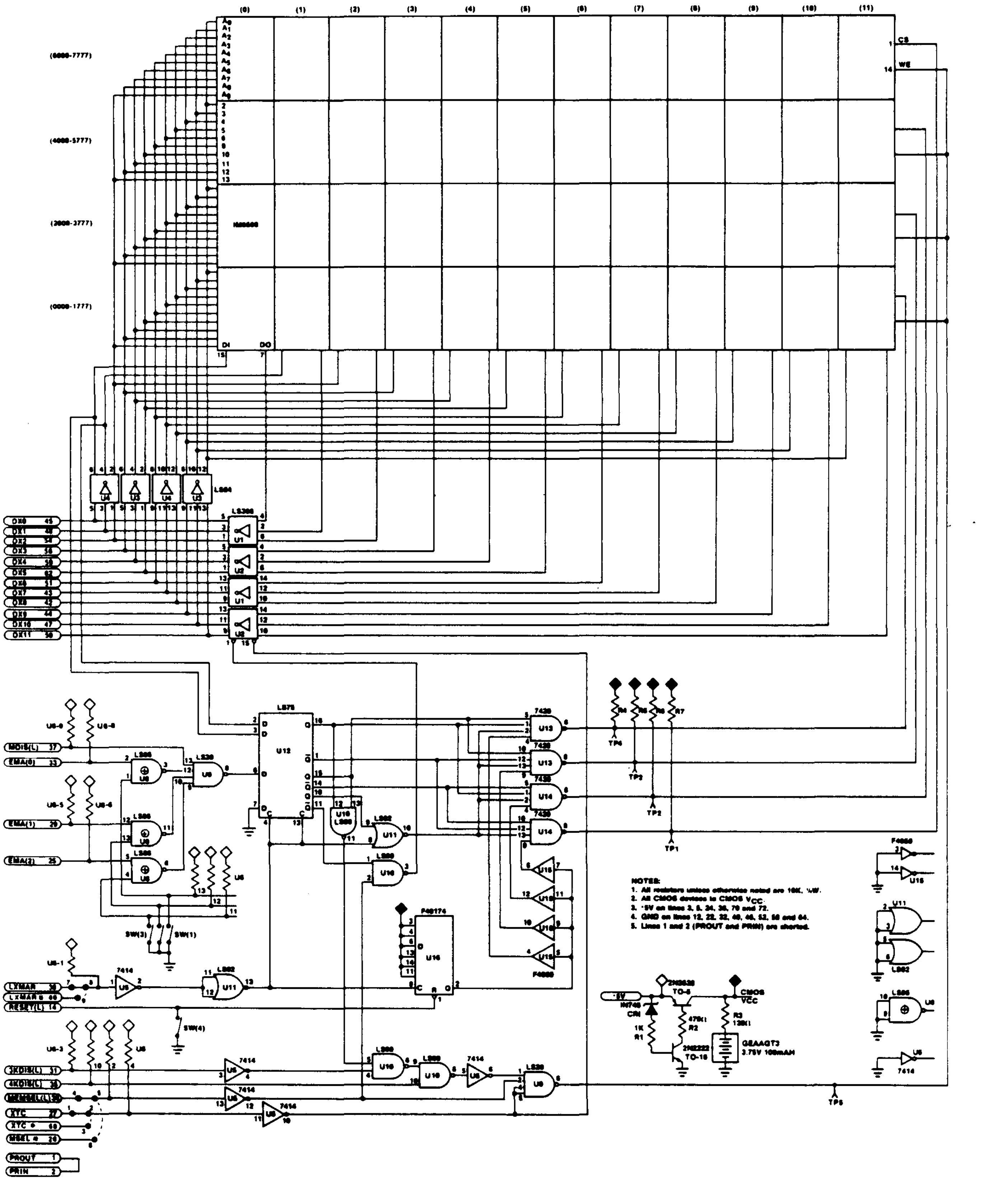
The maximum stand-by current drain of the 4K X 12 memory module at 25°C and 3.6 volts is 50 µA. With the 100 mAH Ni-Cad batteries used in 6901-M4KX12C module, the stand-by period is approximately 2000 hours (80 days). Since the battery storage capacity decreases and the device leakage current increases as the ambient temperature goes up, the stand-by period is a function of ambient temperature.

## AC CHARACTERISTICS

The memory module specification follows closely that of the IM6508/18.

| | VCC = 5.0V | TA = 25°C |
|---|---|---|
| | MIN (ns) | MAX (ns) |
| Access time from LXMAR (TAL) | | 390 |
| Address set-up time (TAS) | 50 | |
| Address hold time (TAH) | 115 | |

|  | VCC = 5.0V | TA = 25°C |
|---|---|---|
|  | MIN (ns) | MAX (ns) |
| LXMAR pulse width<br>(TL) | 235 |  |
| Write pulse width<br>(TWP) | 200 |  |
| Write data set-up time<br>(TDS) | 150 |  |
| Write data hold time<br>(TDH) | 150 |  |
| Output enable time<br>(TEN) | 10 | 50 |
| Output disable time<br>(TDIS) | 10 | 50 |
| ICC at 250 KHz<br>(4 MHz for IM6100) |  | 400 mA |



NOTES:
1. All resistors unless otherwise noted are 10K, ¼W.
2. All CMOS devices to CMOS $V_{CC}$.
3. +5V on lines 3, 5, 24, 36, 70 and 72.
4. GND on lines 12, 22, 32, 40, 46, 52, 58 and 64.
5. Lines 1 and 2 (PROUT and PRIN) are shorted.

INTERSIL 6901-M4KX12 REV C

# CHAPTER 10

# SERVICE AND MAINTENANCE

-WARNING-

> INTERCEPT maintenance and service should be performed by qualified electronic service personnel only.

## INTRODUCTION

The INTERCEPT microcomputer system was designed with flexibility and mechanical simplicity in mind. Many system characteristics may be changed by removing a plug-in printed circuit board and setting/resetting "DIP" switches or soldering/unsoldering wire "straps" between P.C. board traces. No tools are required to remove a P.C. board from the system, and the entire enclosure may be disassembled with only a blade screwdriver and a Phillips screwdriver.

## 6911-INTERCEPT I ENCLOSURE

### COVER REMOVAL

Access to the plug-in printed circuit modules and main power supply is by removal of the top cover. The user should Reset the system and disconnect power before removing cover. The INTERCEPT I cover is removed by unscrewing the four thumb screws along the lower sides of the enclosure. If access to the DC to DC converters is required, the four recessed flat head screws along the front bezel must be removed and the bezel lifted off. Refer to Figure 10-8 for power distribution information.

## 6910-INTERCEPT II ENCLOSURE

### FRONT COVER REMOVAL

Access to the plug-in printed circuit modules is by removal of the front snap-on cover (refer to Figure 10-3). Front cover tabs located under the unit next to the front feet should be used to pull the cover forward and off. Before either inserting or removing P.C. modules the system should be reset and powered off.

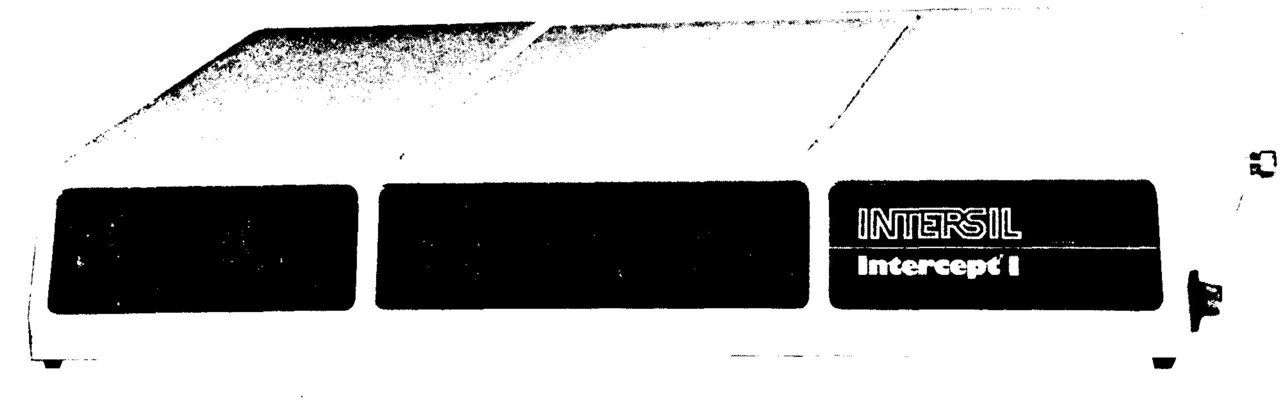


FIGURE 10-1

INTERCEPT I (FRONT)

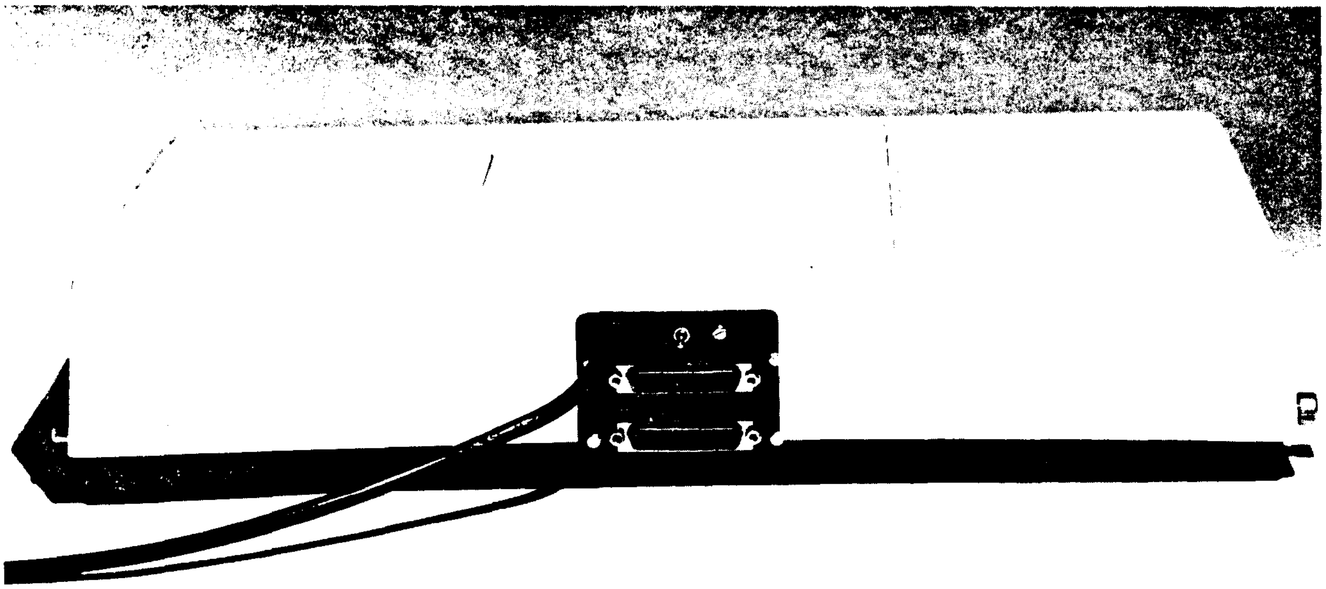


FIGURE 10-2

INTERCEPT I (REAR)

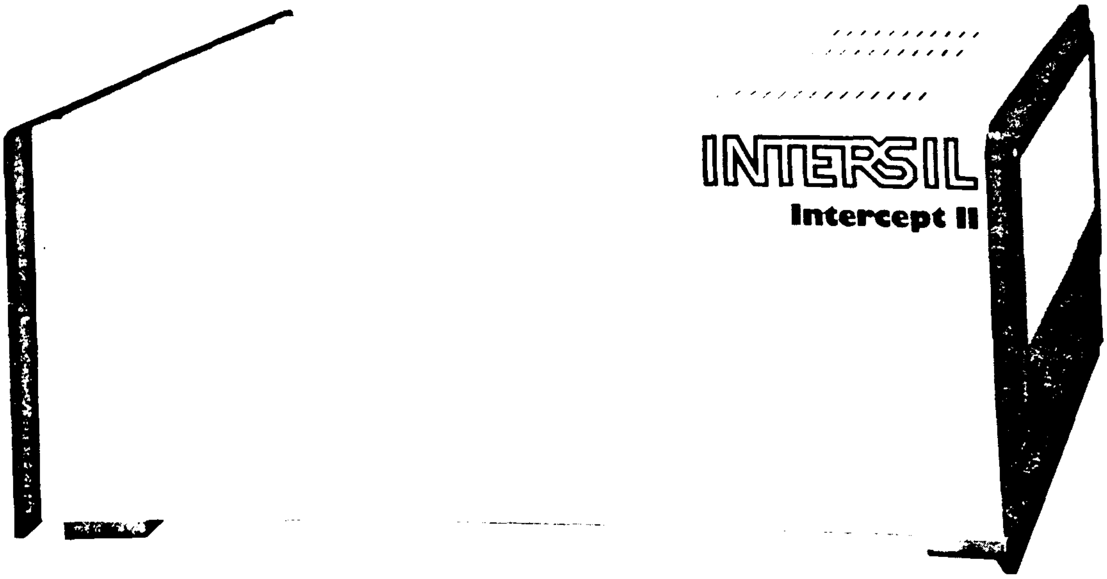


FIGURE 10-3

INTERCEPT II (FRONT)

TOP & BOTTOM PANEL REMOVAL

WARNING: Power should be disconnected completely from system before removing top or bottom panels. Remove snap-on front cover as previously described. While facing the rear panel as in Figure 10-4, remove the two upper-most screws (Phillips head) securing the edge of the top cover. Then, lift the cover and slide it rearward to free it from the front channel. The bottom panel is removed in the same manner by turning the unit upside-down.

NOTE: Power should never be connected to the system with the top and/or bottom panels removed as exposure to hazardous voltages exists in the power distribution areas of the rear panel. (Refer to Figures 10-6 and 10-9.)

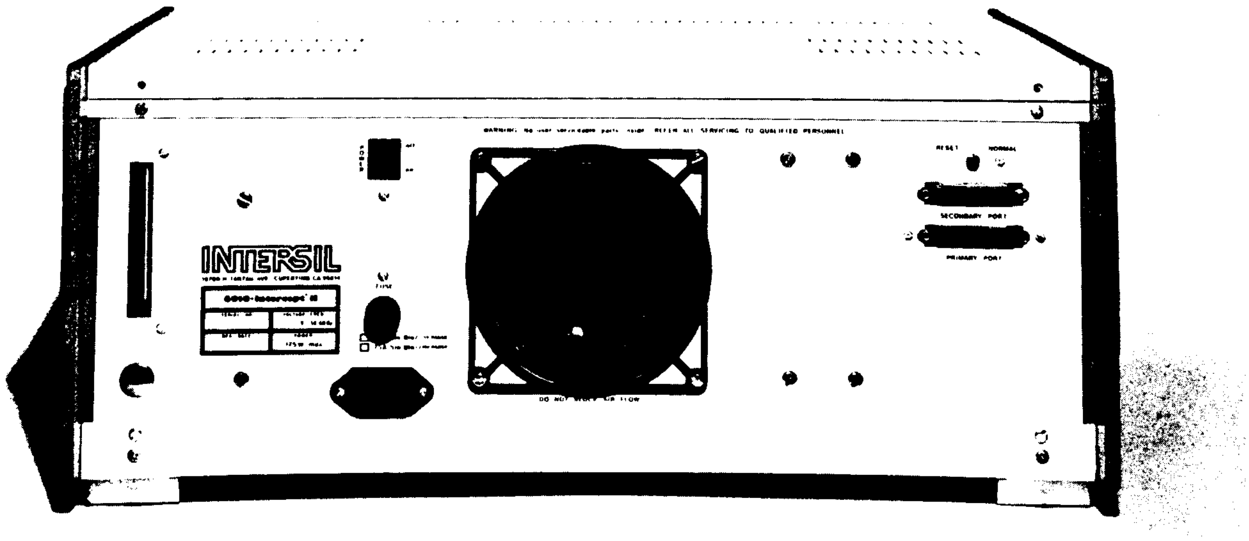


FIGURE 10-4

INTERCEPT II (REAR)

FUSE REPLACEMENT - INTERCEPT I

The main power fuse holder is inside the enclosure (see earlier section on cover removal) within the power supply section. The required fuse is a standard 1/4" X 1/4" 3AG-250 V. The current rating of the fuse depends on the system operating voltage:

230 Volt operation - 0.3 Amp fuse
115 Volt operation - 0.6 Amp fuse

Reset system and disconnect power before examining or replacing this fuse. INTERCEPT I contains no other fuses.

FUSE REPLACEMENT - INTERCEPT II

The main power fuse holder is located on the rear panel directly above the power cord connector. Reset the system and disconnect power before examining or replacing this fuse. The required fuse is a 1/4" X 1/4" 3AG style SLO-BLO. The current rating of the fuse is found on the rear panel directly under the fuse holder. The enclosure contains no other fuses.

## PRINTED CIRCUIT BOARD REMOVAL/INSERTION

Reset and power off the system before removing a P.C. board. Remove the top cover (INTERCEPT I) or the front cover (INTERCEPT II) of the system as described previously. Check for ribbon cables, round cables, etc., which may impair P.C. board removal, and remove these cables if necessary. Then use the plastic card ejectors on the P.C. board to pull the board free of its connector on the INTERCEPT bus. Once free, the boards slide out easily on plastic card guides.

When inserting a P.C. board make sure the component side of the board is facing in the correct direction. Each board is keyed to ensure correct installation. The board must be aligned correctly on the plastic card guides. It should snap easily into its connector on the bus. Never force a P.C. board into its connector. After inserting the board, attach all cables removed previously and replace the enclosure cover.

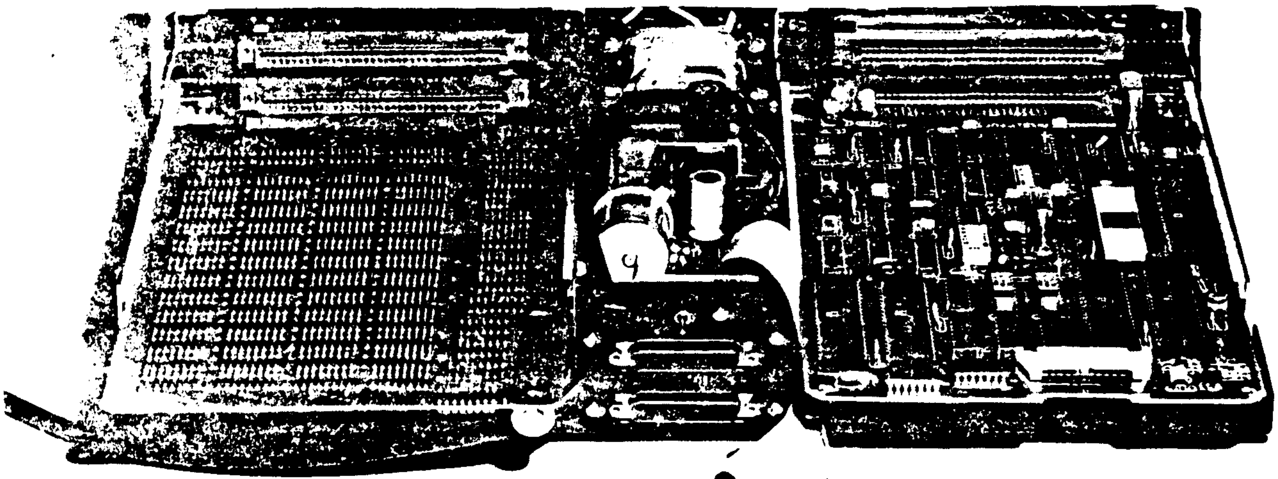


FIGURE 10-5

INTERCEPT I (INTERIOR)

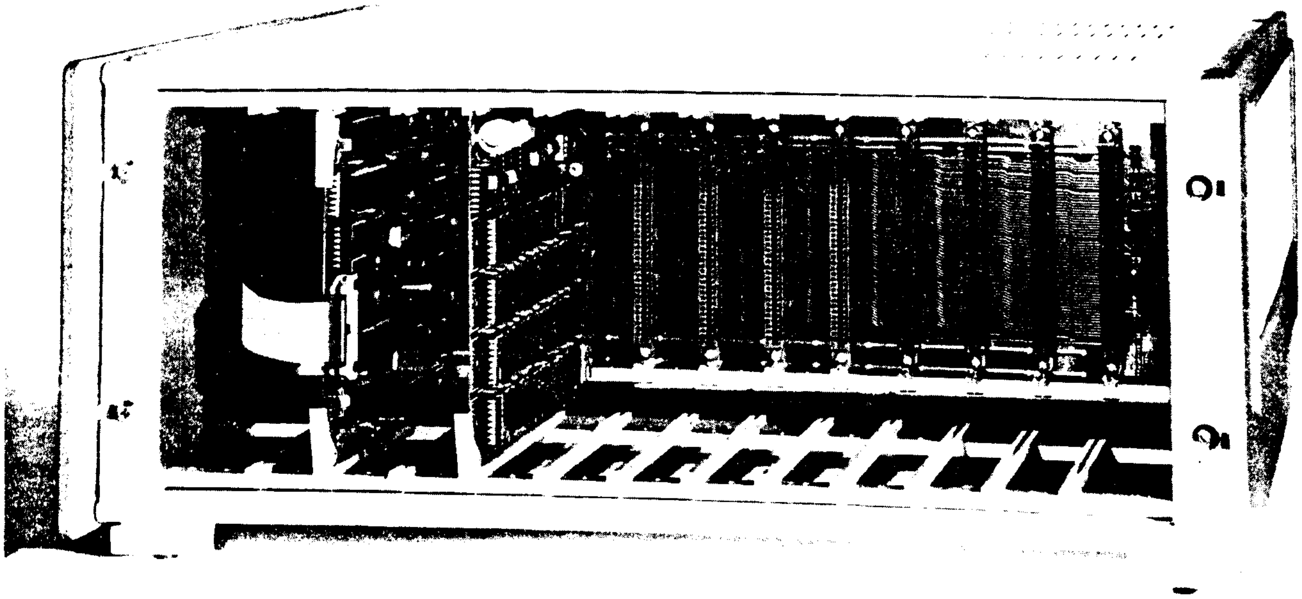


FIGURE 10-6

INTERCEPT II (INTERIOR)

## SELECTING CPU MODULE OPTIONS

### CHANGING PRIMARY PORT BAUD RATE

The Primary Port baud rate is changed by means of the "DIP" switches near the center of the module (see Figure 10-7). Switches 1 through 4 control the baud rate as follows:

TABLE 10-1

| SWITCH 1 | SWITCH 2 | SWITCH 3 | SWITCH 4 | BAUD RATE |
|---|---|---|---|---|
| ON | ON | ON | ON | 19,200 |
| ON | ON | ON | OFF | 19,200 |
| ON | ON | OFF | ON | 50 |
| ON | ON | OFF | OFF | 75 |
| ON | OFF | ON | ON | 134.5 |
| ON | OFF | ON | OFF | 200 |
| ON | OFF | OFF | ON | 600 |
| ON | OFF | OFF | OFF | 2,400 |
| OFF | ON | ON | ON | 9,600 |
| OFF | ON | ON | OFF | 4,800 |

TABLE 10-1
(continued)

| | | | | |
|---|---|---|---|---|
| OFF | ON | OFF | ON | 1,800 |
| OFF | ON | OFF | OFF | 1,200 |
| OFF | OFF | ON | ON | 2,400 |
| OFF | OFF | ON | OFF | 300 |
| OFF | OFF | OFF | ON | 150 |
| OFF | OFF | OFF | OFF | 110 |

RESETTING THE SYSTEM

"DIP" switches 5 and 6 are RESET switches. Switch 5 generates a BUS RESET when switched ON. Switch 6 generates a POWER RESET when switched ON. In general, both switches are switched ON to reset peripheral devices, CPU circuitry, and to ensure main memory non-volatility when the power is turned off. For normal system operation, both switches 5 and 6 should be in the OFF position.

CONNECTING READER RELAY TO PRIMARY PORT

This strap is located just below the leftmost IM6402 device of the two in the upper right corner of the CPU module (see Figure 10-7). Cut the trace connecting R with S and install a wire jumper between R and P.

USING AN "OFF-BOARD" CONTROL PANEL MEMORY

This strap is located about one inch below the "DIP" switches and slightly to the left. The strap should connect A and C for "on-board" control panel and should connect A and B to provide the CPSEL signal to bus pin 55 for an "off-board" control panel memory.

CHANGING AUTO-START STRAPS

The auto-start strap matrix is located just below the rightmost of the two IM6402 devices in the upper right corner of the CPU module. Each of the six horizontal rows of the matrix forms one strap option. From bottom to top, the straps are called AS0, AS2, AS1, AS8, AS6, AS7 respectively. Each of the matrix points in the left vertical column must connect either to a point in the center column or in the right column. If connected to the right column, that strap is forced to zero. If connected to the center column, that strap is forced to one when PIE A Flag 2 is a one (see CPU schematic, Chapter 8).

When the INTERCEPT RESET line is released and the

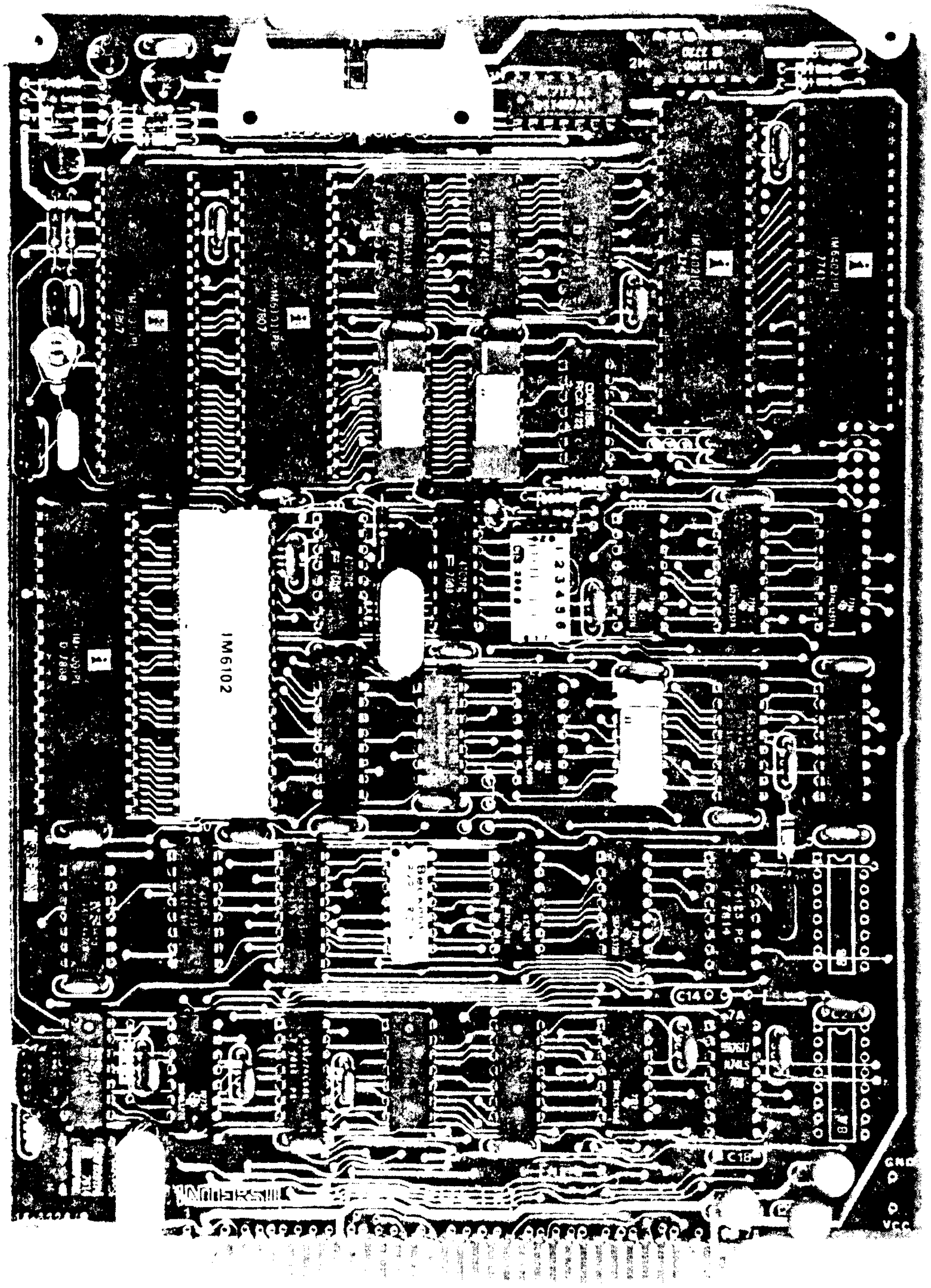


FIGURE 10-7

CPUCTRL PRINTED CIRCUIT BOARD

control panel firmware is entered, the firmware reads the auto-start straps. If AS0 is set to one, AS1 and AS2 become the two most significant bits of the program counter (PC0 and PC1 respectively), and AS6, AS7, and AS8 become the Instruction Field (IF0, IF1, IF2, respectively). The remainder of the program counter is cleared. In this way, control can automatically be transfered to any 1K main memory segment after a RESET. If AS0 is zero, the other auto-start straps are ignored, the firmware outputs machine status and awaits user command input.

## INSTALLATION OF ADDITIONAL CABLES

Provisions have been made for the installation of extra cabling in the INTERCEPT II enclosure. Flat ribbon cables can be drawn through the slot provided at the left side of the rear panel (see Figure 10-4) and secured by the ribbon cable strain relief. This is accomplished as follows:

1. Remove top cover as previously described.
2. Remove strain relief bracket from raar panel (2 screws).
3. Loosen strain relief clanp (2 screws).
4. Feed ribbon cable through rear panel slot and then through strain relief clamp.
5. Adjust internal cable length.
6. Tighten strain relief clamp.
7. Re-install strain relief bracket in rear panel.

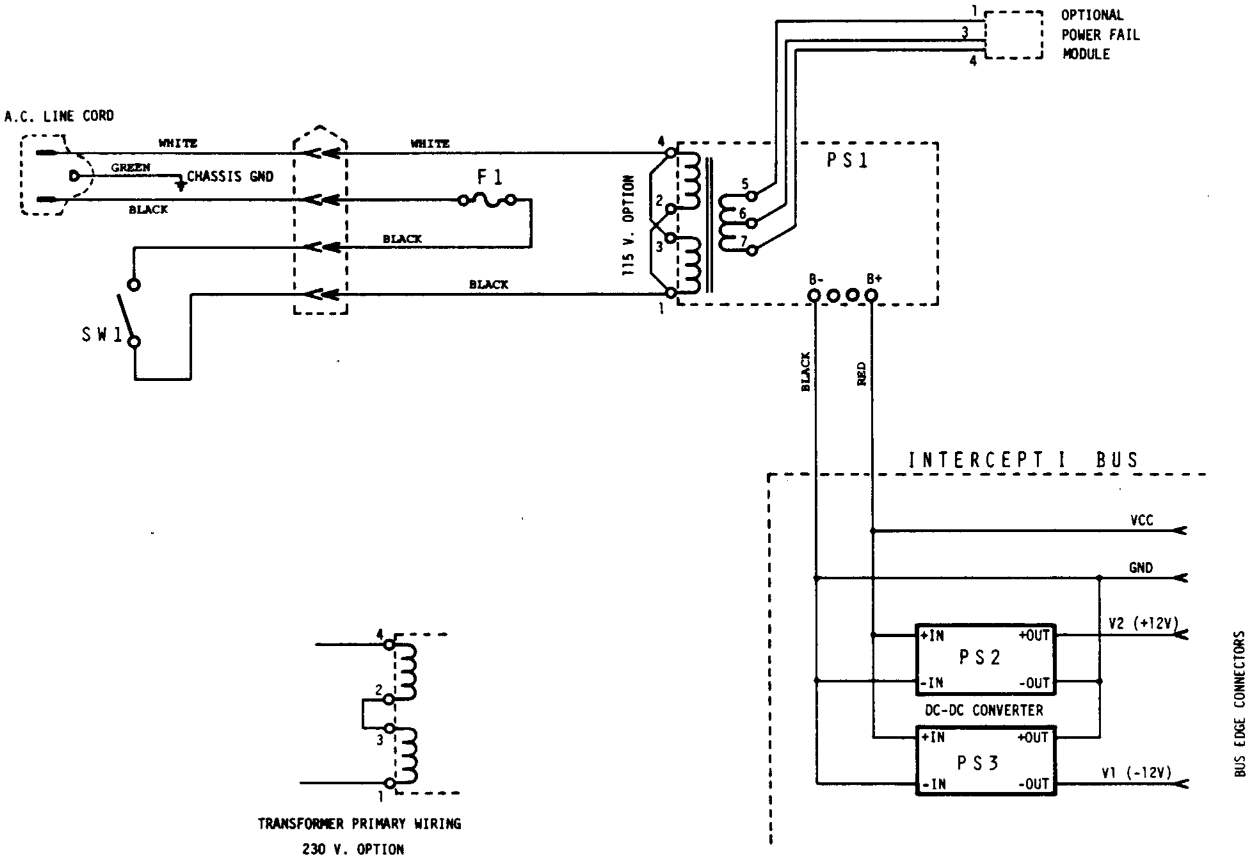
Conventional round cables may be inserted through the hole provided just under the flat cable slot (remove snap fit plug). A grommet should be used to protect the cable insulation.

## CURRENT LOOP INTERFACE AT HIGH BAUD RATES

Filter capacitor C12 must be replaced with a smaller capacitance for baud rates above 110 baud.

## POWER SUPPLIES

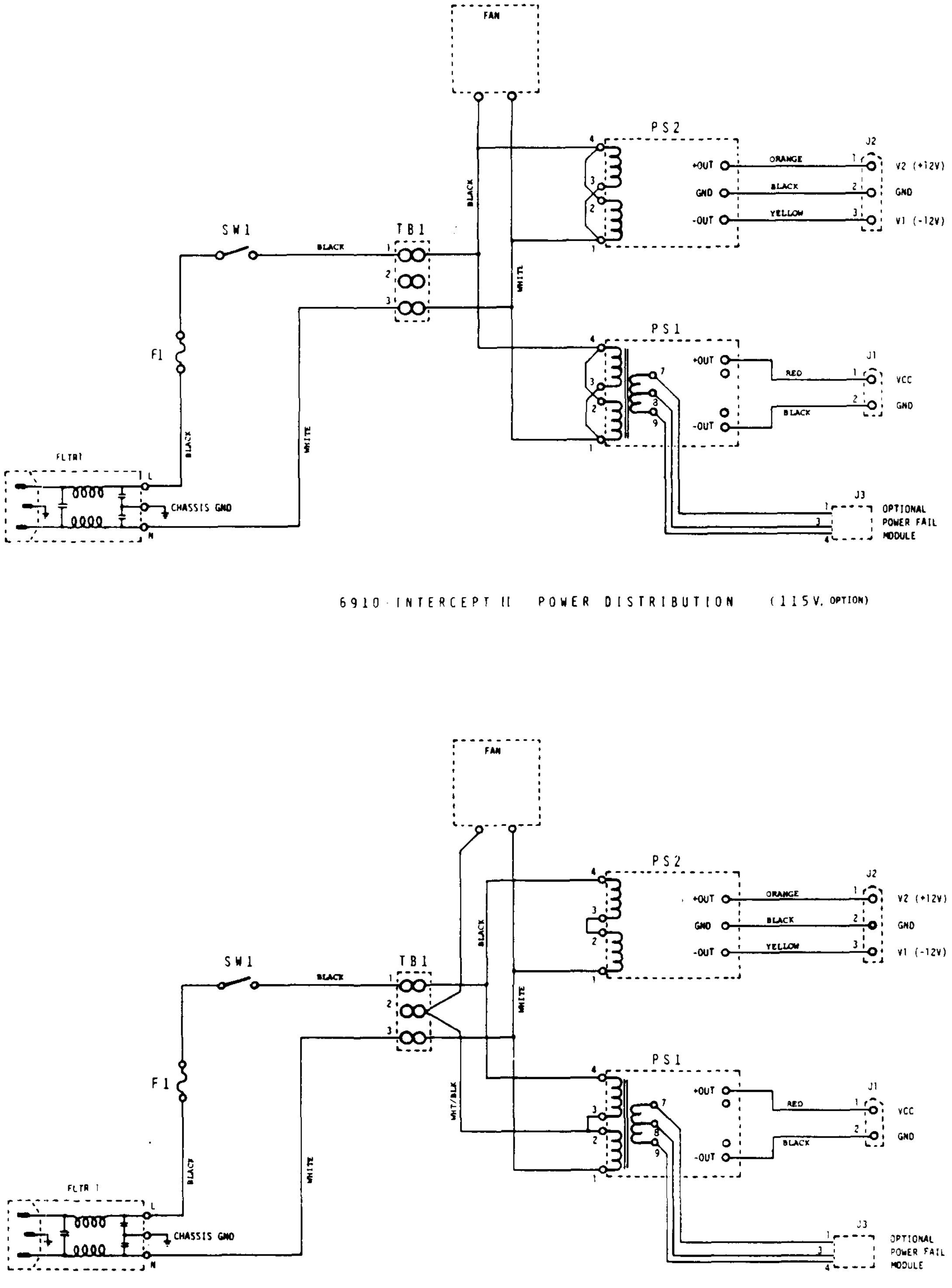
INTERCEPT I uses a single +5V supply rated at 1.5 Amps. INTERCEPT II uses two supplies; a +5V supply rated at 6 Amps and a ±12 V supply rated at 0.8 Amps. All supplies are common series regulation types using an LM723 Voltage Regulator IC (see Figures 10-8, 10-9, 10-10, 10-11).



6911-INTERCEPT I POWER DISTRIBUTION

FIGURE 10-8

POWER DISTRIBUTION - INTERCEPT I



6910 INTERCEPT II POWER DISTRIBUTION (115V. OPTION)

6910-INTERCEPT II POWER DISTRIBUTION (230V. OPTION)

FIGURE 10-9

POWER DISTRIBUTION - INTERCEPT II

POWER-ONE Inc.
531 DAWSON DR.
CAMARILLO, CALIF. 93010
(805) 484-2806

APPLICATION DATA
Including:
1. Schematic
2. Parts Lists
3. Specification
4. Outline & Mounting Dwg.
5. General User Information

MODEL

HC5-6/OVP

SPECIFICATIONS

AC Input: 115/230 VAC ± 10% 47-440 HZ (Derate Current 10% for 50 HZ operation)

Input Fusing: See Table

DC Output: 5VDC @ 6A, OVP at 6.2 ± .4VDC

Line Regulation: ± .05% for a 10% input change

Load Regulation: ± .05% for a 50% load change

Output Ripple: 1.5 mv Pk-Pk, 0.4 mv RMS

Transient Response: 30μ sec for 50% load change

Overload & Short Circuit Protection: Automatic current limit/foldback

Temperature Coefficient: ± .03%/°C maximum

Cooling: Units are full rated to 50°C in free air, must be derated or fan cooled when mounted in confined area

Temperature Rating: 0-50°C, full rated, derate linearly to 40% at 70°C

Efficiency: 47% at nominal input, full load on output

Weight: 2 lbs.

Vibration: Per Mil-Std-810B, method 514, proc.I, curve AB (to 50 HZ)

Shock: Per Mil-Std-810B, method 516, procedure V

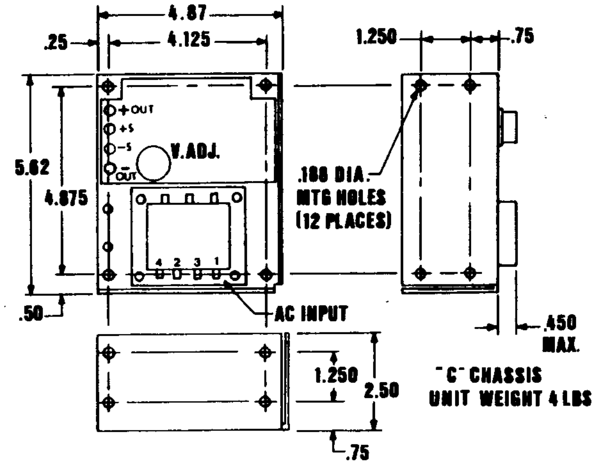
Re:ote Sensing: Provided, connection for local sense not required.

AC Connection & Fuse Table

| For Use At | Primary Fuse At | Connect | Apply Power to |
|---|---|---|---|
| 115 | 1A | 1-3, 2-4 | 1 & 4 |
| 230 | 0.5A | 2-3 | 1 & 4 |

2 YEAR GUARANTEE

POWER-ONE will repair or replace any power supply of its manufacture that does not perform to published specifications as a result of defective materials or workmanship for a period of 2 years from date of original purchase. No other obligations or liabilities are implied or expressed. Returns must be freight prepaid.



PRINTED BY PERMISSION

| REF DES | HC5-6 OVP | POWER-ONE P/N | DESCRIPTION |
|---|---|---|---|
| C1 | 1000/16 | 101-10108 | CAPACITOR, ELECT. |
| C2 | 16000/15 | 102-10096 | ELECT. |
| C3 | 220/16 | 101-10107 | ELECT |
| C4 | .0033/100 | 104-10092 | CAPACITOR, MYLAR |
| | | | |
| CR1 | AEIC | 111-10251 | DIODE, 1A, 200V |
| CR2,3 | MR750 | 111-10256 | DIODE, 6A, 50V |
| CR4 | 1N752A | 112-10006 | DIODE ZENER |
| | | | |
| SCR1 1 | S0508LS3 | 160-10013 | SCR, 8A |
| Q1 1 | 2N6551 | 172-10249 | TRANSISTOR |
| Q2 | 12505-2 | 171-10262 | TRANSISTOR |
| R5 | 3.9K | 151-10379 | RESISTOR 1/2W 5% CF |
| R2 | 2.7Ω | 151-10305 | |
| R11,12 | 6.8Ω | 151-10313 | RESISTOR, 1/2W, 5% CF |
| R3,10 | 22Ω | 151-10325 | 1/2W, 5% CF |
| R1,7,6 | 2.2K | 151-10373 | 1/2W, 5% CF |
| R9 | 2K | 152-10379 | RESISTOR, 1/2W, 2% MF |
| R4,8 | 1.5K | 155-10512 | POTENTIOMETER |
| HEATSINK THERMOTAB | 13920 | 402-13920 | HEATSINK |
| U1 | μa723 | 130-10287 | IC VOLTAGE REGULATOR |
| T1 | 12247 | 082-12247 | TRANSFORMER |
| CHASSIS | 11131 | 412-11131 | CHASSIS |
| P.C.B. | 12098 | 505-12098 | P.C. BOARD |

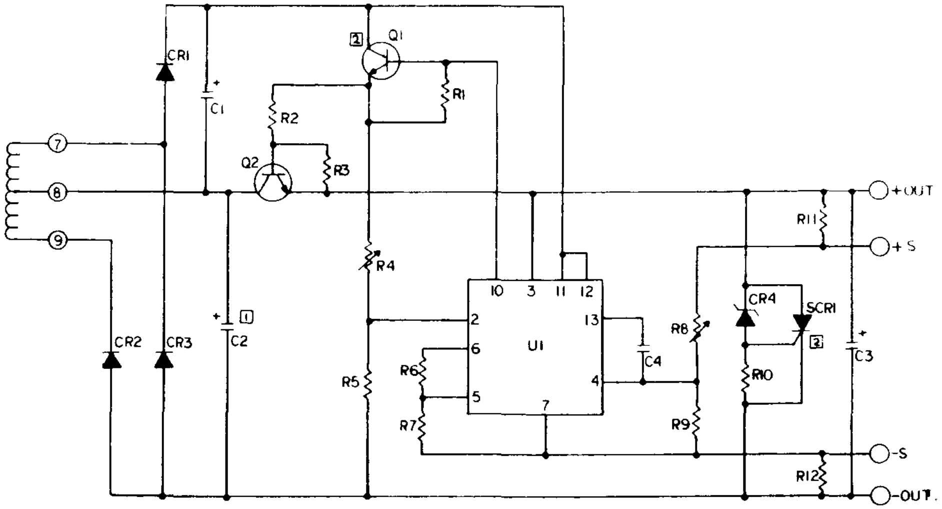


FIGURE 10-10

+5V POWER SUPPLY ATIC - INTERCEPT II

**POWER-ONE, Inc.**
531 DAWSON DRIVE
CAMARILLO, CALIF. 93010
(805) 484-2806

APPLICATION DATA
Including
1. Schematic
2 Parts Lists
3. Specification
4 Outline & Mounting Dwg
5. General User Information

MODEL
HAA15- E
HAA24- E

SPECIFICATIONS

Ac Input: 115/230 VAC ±10% 47-440 HZ (Derate Current 10% for 50 HZ operation)

Input Fusing: See Table

DC Output: See Table

Line Regulation: ±.05% for a 10% input change

Load Regulation: ±.05% for a 50% load change

Output Ripple: 2.5 mv Pk-Pk, 0.6 mv RMS

Transient Response: 30u sec for 50% load change

Overload & Short Circuit Protection: Automatic current limit/foldback

Temperature Coefficient: ±.03%/°C maximum

Cooling: Units are full rated to 50°C in free air, must be derated or fan cooled when monted in confined area

Temperature Rating: 0 - 50°C, full rated, derate linearly to 40% at 70°C

Efficiency: 55% at nominal input, full load on output

Weight: 2 lbs.

Vibration: Per Mil-Std-810B, method 514, proc. I, curve AB(to 50HZ)

Shock: Per Mil-Std-810B, method 516, procedure V

AC Connection & Fuse Table

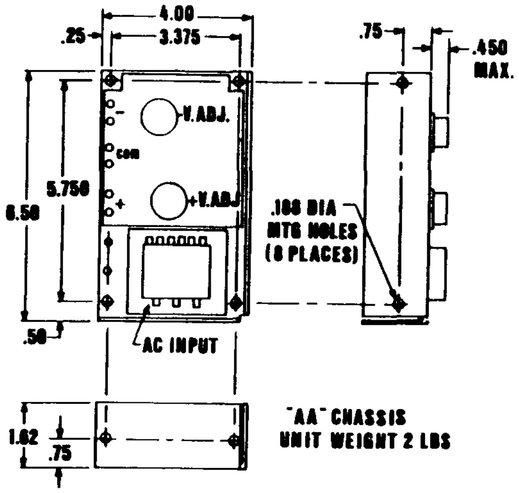
| For Use At | Primary Fuse At | Connect | Apply Power to |
|---|---|---|---|
| 115 | .5 | 1-3,2-4 | 1 & 4 |
| 230 | .25 | 2-3 | 1 & 4 |

OUTPUT RATING

| | |
|---|---|
| HAA15 | ±12V @ 1A or ±15V @ .8A |
| HAA24 | ±24V @ .6A |

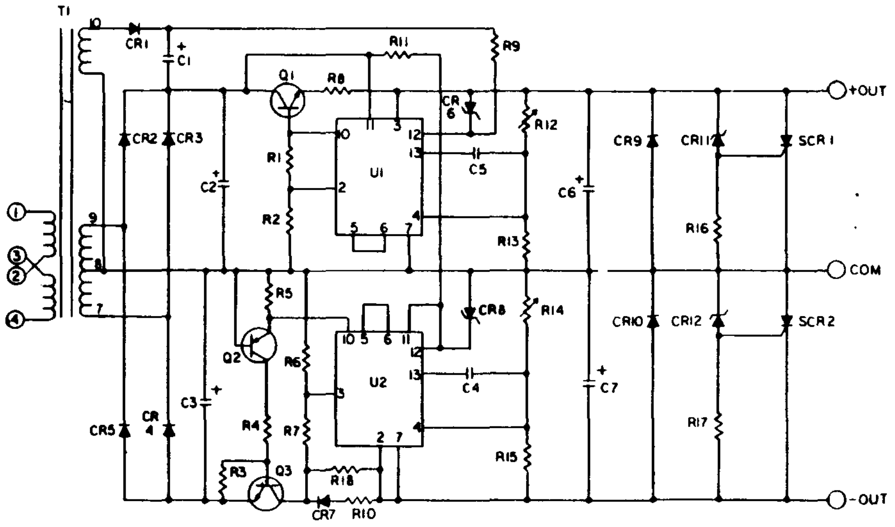
2 YEAR GUARANTEE

POWER-ONE will repair or replace any power supply of its manufacture that does not perform to published specifications as a result of defective matertials or workmanship for a period of 2 years from dat of original purchase. No other obligations or liabilities are implied or expressed. Returns must be freight prepaid.



"AA" CHASSIS
UNIT WEIGHT 2 LBS

| REF DES | HAA 15 | HAA 24 | DESCRIPTION |
|---|---|---|---|
| C1,6,7 | 100/35 | 100/35 | CAPACITOR ALUM ELECT |
| C2,3 | 2200/35 | 220/63 | ALUM ELECT |
| C4,5 | .001/100 | .001/100 | CAPACITOR MYLAR |
| | | | |
| | | | |
| CR1,2,3,4,5,7 | AEIC | AEIC | DIODE 1A 200V |
| CR9,10 | —— | —— | |
| CR11 | —— | BUSS | ZENER |
| CR6,8 | —— | 1N752A | DIODE ZENER |
| Q3 | 12500-4 | 12505-2 | |
| Q1 | 12500-4 | 12500-4 | TRANSISTOR |
| Q2 | 2N2905 | 2N2905 | TRANSISTOR |
| SCR1,2 | —— | | SCR 3A |
| | | | |
| U1,U2 | ua723 | MA723 | I.C. VOLTAGE REGULATOR |
| R18 | 3.9Ω | 6.8Ω | RESISTOR 1/2W 5% CARBON |
| R1,7 | 330Ω | 330 | RESISTOR 1/2W 5% CARBON |
| R16,9 | —— | 1.6K | |
| R4 | 270Ω | 270Ω | |
| R2,3,6 | 10K | 7.5K | |
| R8,10 | 1Ω | 1Ω | 1/2W 5% CARBON |
| R13,15 | 511 | 511 | RESISTOR 1/2 W 5% MF |
| R12,14 | 1.5K | 1.5K | POTENIOMETER 2W WW |
| R11 | 270Ω | 2.2K | |
| T1 | 12244 | | TRANSFORMER |
| | | | |
| PCB | 12222 | 12222 | PRINTED CIRCUIT BOARD |
| | | | |
| CHASSIS | 11091 | 11091 | CHASSIS ALUM |



THE INFORMATION AND DESIGN CONTAINED HEREIN ARE THE PROPERTY OF POWER ONE AND THE UNAUTHORIZED USE THEREOF IS PROHIBITED

FIGURE 10-11

±12V POWER SUPPLY SCHEMATIC - INTERCEPT II

## SERIAL I/O CONNECTOR BOARD

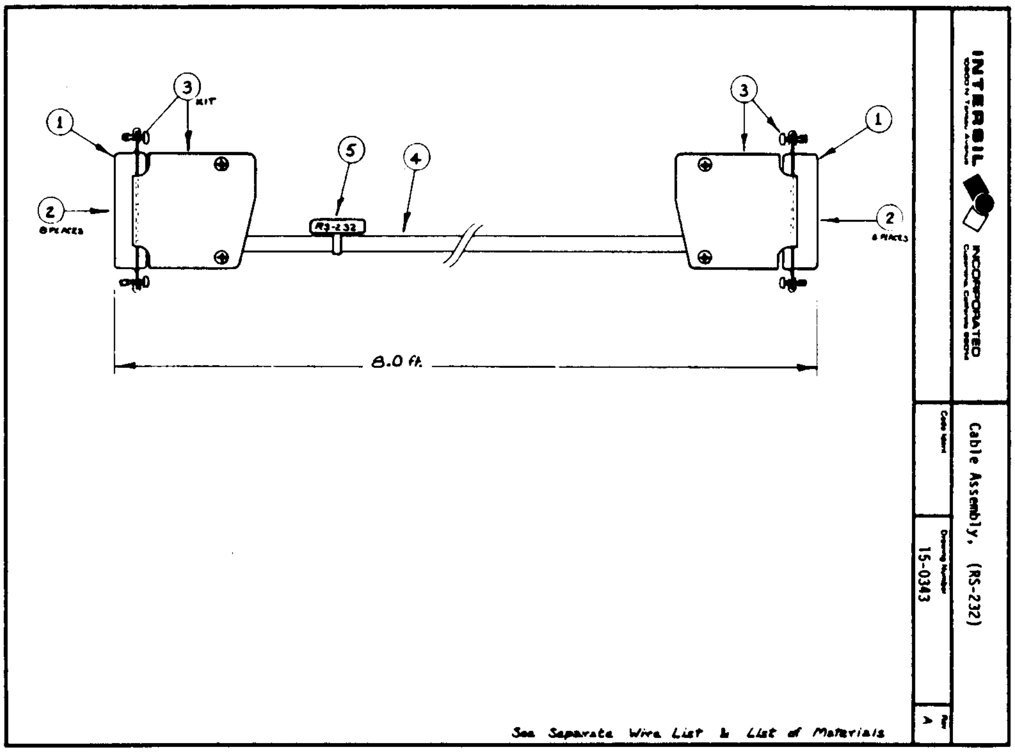
This printed circuit assembly provides breakout of the 20 conductor ribbon cable from the CPUCTRL module to the two DB-25 connectors and the Reset switch. Pin selection and jumper options on the external mating cable connector determine the electrical interface mode (current loop or RS-232). Refer to Chapter 8 "Serial Interfaces" section for circuit description, and to Table 10-2 for connector signal list. Also refer to Figures 10-12 and 10-13 for external serial interface cable information.

TABLE 10-2

SERIAL I/O PIN LIST

| RIBBON CABLE PIN NUMBER | SIGNAL NAME | PRIMARY PORT PIN NUMBER | SECONDARY PORT PIN NUMBER |
|---|---|---|---|
| - | Chassis GND | 1 | 1 |
| 1 | EIA RCV 1 | 2 | - |
| 13 | EIA RCV 2 | - | 2 |
| 3 | EIA XMIT 1 | 3 | - |
| 14 | EIA XMIT 2 | - | 3 |
| - | Resistor pull-up | 5 | - |
| 15 | CTS | - | 5 |
| 2,18 | Signal GND | 7 | 7 |
| 5 | CAR DET | - | 8 |
| 17 | SUP TRANS* | - | 11 |
| 7 | 20ma RDR | 12 | 12 |
| 6 | 20ma XMIT DR | 13 | 13 |
| 9 | 20ma RCV | 14 | 14 |
| 4 | 20ma XMIT | 17 | 17 |
| 11 | -12V | 18 | 18 |
| 12 | DTR* | - | 20 |
| 11 | -12V | 22 | 22 |
| 16 | +12V | 23 | 23 |
| 9 | 20ma RCV | 24 | 24 |
| 11 | -12V | 25 | 25 |
| 8 | Polarizing Key | | |
| 10 | +5V | | |
| 19 | POW RES | | |
| 20 | RESET | | |

*NOTE: When using terminals on the Secondary Port that do not support the RS-232C signals Supervisory Transmit (DB-25 connector pin 11) and Data Terminal Ready (DB-25 pin 20), both of these signals must be tied to +12V (DB-25 pin 23) for proper operation of the Secondary Port. This connection should be made on the user's cable. For terminals which support these signals, the jumpers must not be present.


3
KIT
1
2
8 PLACES
5
RS-232
4
3
1
2
8 PLACES
8.0 ft.
See Separate Wire List & List of Materials
INTERSIL 10900 N. Tantau Avenue
INCORPORATED Cupertino, California 95014
Cable Assembly, (RS-232)
Code Ident
Drawing Number
15-0343
Rev
A

# WIRE LIST

| Signal | From | Sta | To | Sta | | Color | Size | Comments |
|---|---|---|---|---|---|---|---|---|
| GND (chassis) | J1 | 1 | J2 | 1 | | Green | 22 AWG | |
| EIA RCV | " | 2 | " | 2 | | BRN | " | |
| EIA XMIT | " | 3 | " | 3 | | Red | " | |
| CTS | " | 5 | " | 5 | | Orange | " | |
| GND (signal) | " | 7 | " | 7 | | Black | " | |
| CAR DET | " | 8 | " | 8 | | Yellow | " | |
| SUP TRANS | " | 11 | " | 11 | | Blue | " | |
| D T R | " | 20 | " | 20 | | Violet | " | |

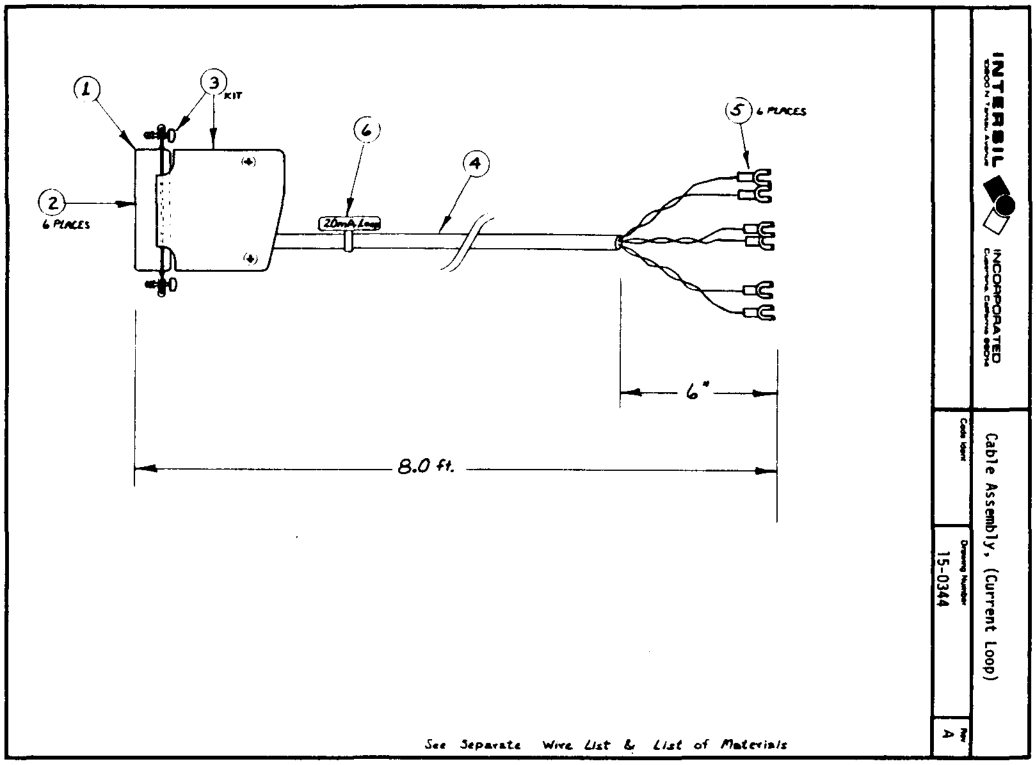
INTERSIL 10900 N. Tantau Avenue INCORPORATED Cupertino, California 95014

Title: Cable Assembly, (RS-232)

Dwg. No. 15-0343/A
Sheet 1 of 1

FIGURE 10-12

RS-232 CABLE ASSEMBLY



# WIRE LIST

| Signal | From | Sta | To | Sta | | Color | Size | Comments |
|---|---|---|---|---|---|---|---|---|
| XMIT Driver | J1 | 3 | J1 | 13 | | any | 22 AWG | jumper |
| | | | | | | | | |
| RCV | " | 2 | " | 14 | | any | " | jumper |
| | | | | | | | | |
| 20 XMIT + | " | 17 | Spade Terminal | | | Red | " | PAIR |
| 20 XMIT - | " | 18 | " | | | Black | " | |
| | | | | | | | | |
| 20 RCV + | " | 24 | " | | | Green | " | PAIR |
| 20 RCV - | " | 25 | " | | | Black | " | |
| | | | | | | | | |
| 20 RDR + | " | 12 | " | | | White | " | PAIR |
| -12V | " | 22 | " | | | Black | " | |

INTERSIL 10900 N. Tantau Avenue INCORPORATED Cupertino, California 95014

Title: Cable Assembly, "20mA Loop"

Dwg. No. 15-0344
Sheet 1 of 1

FIGURE 10-13

20ma CABLE ASSEMBLY

# APPENDIX A

## AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

### CHARACTER CODES

| 8-bit ASCII CODE | 6-bit CODE | CHARACTER REPRESENTATION | REMARKS |
|---|---|---|---|
| 240 | 40 | | space |
| 241 | 41 | ! | exclamation point |
| 242 | 42 | " | quotation marks |
| 243 | 43 | # | number sign |
| 244 | 44 | $ | dollar sign |
| 245 | 45 | % | percent |
| 246 | 46 | & | ampersand |
| 247 | 47 | ' | apostrophe |
| 250 | 50 | ( | opening parenthesis |
| 251 | 51 | ) | closing parenthesis |
| 252 | 52 | * | asterisk |
| 253 | 53 | + | plus |
| 254 | 54 | , | comma |
| 255 | 55 | - | minus sign or hyphen |
| 256 | 56 | . | period or decimal point |
| 257 | 57 | / | slash |
| 260 | 60 | 0 | |
| 261 | 61 | 1 | |
| 262 | 62 | 2 | |
| 263 | 63 | 3 | |
| 264 | 64 | 4 | |
| 265 | 65 | 5 | |
| 266 | 66 | 6 | |
| 267 | 67 | 7 | |
| 270 | 70 | 8 | |
| 271 | 71 | 9 | |
| 272 | 72 | : | colon |
| 273 | 73 | ; | semicolon |
| 274 | 74 | < | less than |
| 275 | 75 | = | equals |
| 276 | 76 | > | greater than |
| 277 | 77 | ? | question mark |
| 300 | 00 | @ | at sign[1] |
| 301 | 01 | A | |
| 302 | 02 | B | |
| 303 | 03 | C | |
| 304 | 04 | D | |

| 8-bit<br>ASCII<br>CODE | 6-bit<br><br>CODE | CHARACTER<br>REPRESENTATION | REMARKS |
|---|---|---|---|
| 305 | 05 | E | |
| 306 | 06 | F | |
| 307 | 07 | G | |
| 310 | 10 | H | |
| 311 | 11 | I | |
| 312 | 12 | J | |
| 313 | 13 | K | |
| 314 | 14 | L | |
| 315 | 15 | M | |
| 316 | 16 | N | |
| 317 | 17 | O | |
| 320 | 20 | P | |
| 321 | 21 | Q | |
| 322 | 22 | R | |
| 323 | 23 | S | |
| 324 | 24 | T | |
| 325 | 25 | U | |
| 326 | 26 | V | |
| 327 | 27 | W | |
| 330 | 30 | X | |
| 331 | 31 | Y | |
| 332 | 32 | Z | |
| 333 | 33 | [ | opening bracket, SHIFT/K |
| 334 | 34 | \ | backslash, SHIFT/L |
| 335 | 35 | ] | closing bracket, SHIFT/M |
| 336 | 36 | ↑ | up arrow |
| 337 | 37 | ← | back arrow[2] |

Footnotes:

(1) In IFDOS PAL source programs, $00_8$ represents CARRIAGE RETURN

(2) In IFDOS PAL source programs, $37_8$ represents TAB

CONTROL CODES

| 8-bit ASCII CODE | CHARACTER NAME | | REMARKS (typical operating system usage) |
|---|---|---|---|
| 000 | null | | Ignored in ASCII input |
| 200 | leader/trailer | | Leader/trailer code precedes and follows the data portion of binary paper tapes. |
| 203 | CTRL/C | (1) | IFDOS break character, forces return to Keyboard Monitor, echoed as ↑C |
| 207 | BELL | | CTRL/G |
| 211 | TAB | | CTRL/I, horizontal tabulation |
| 212 | LINE FEED | | Used as a control character by the keyboard monitor. |
| 213 | VT | | CTRL/K, vertical tabulation |
| 214 | FORM | | CTRL/L, form feed |
| 215 | RETURN | | Carriage return, generally echoed as carriage return followed by a line feed |
| 217 | CTRL/O | | Break Character, used conventionally to suppress Teletype output |
| 225 | CTRL/U | | Delete current input line |
| 232 | CTRL/Z | | End-of-File character for 8-bit ASCII files |
| 233 | ESC | | Escape replaces ALTMODE on some terminals. Considered equivalent to ALTMODE. |
| 375 | ALTMODE | | Special break character for Teletype input |
| 376 | PREFIX | | PREFIX replaces ALTMODE on some terminals. Considered equivalent to ALTMODE |
| 377 | RUBOUT | | Key is labeled DELETE on some terminals. Deletes the previous character typed. |

(1) IFDOS break character

# APPENDIX B

## BIN FORMAT

"BIN" format paper tapes may be prepared with Intersil's IFDOS operating system, Intersil's FOPAL Fortran Cross-Assembler, or any of the Digital Equipment Corporation PDP-8 assemblers. The Intercept "BIN" loader is functionally equivalent to the DEC Binary Loader described in the DEC Utility Routine Manual (DEC-81-RZPA-D) and the "Introduction to Programming" book, except for the following conditions: The Intercept loader does not allow diagnostic messages, binary input can be made from any RS232 or 20ma loop device, and no user memory space is occupied.

### EXTERNAL TAPE FORMAT

Tapes to be read by the binary loader must be in binary-coded format and have preceding leader-trailer code (any code with channel 8 punched; preferably code 200). The first two characters represent the initial address or origin. The initial character of an origin has no punch in channel 8, while channel 7 is punched. The second character designating the origin has no punches in either channel 8 or 7. Data characters have no punches in channel 8 or 7. A 12-bit binary word is represented by two 6-bit characters on the tape in channels 6 through 1, channel 6 of the initial character being the most significant bit. The data characters are loaded into sequential locations following the origin set up. If more than 4K of memory is used, the assembler outputs a "field-setting" command of the form 11 XXX 000 (channel 8-1) to indicate the memory field into which the following data is to be loaded. If for example, XXX were 101, all data following the field designator should be loaded into memory field five. Trailer tape is similar to the leader. A concluding 2-character group before the trailer represents the checksum and has no punches in channel 8. If channel 7 is punched, it is ignored.

### CHECKSUM

When any of the assemblers are used to produce a binary tape, a checksum is automatically punched at the end of the binary tape. This is the sum of all data on the tape including the origin but excluding diagnostic messages, leader/trailer code and field settings. The sum is accumulated character by character and not word by word. Carry out of the Accumulator, AC, is ignored.

If the checksum accumulated while using the binary leader does not agree with the last two characters on tape (i.e., the checksum on the tape calculated and placed there by the assembler), an error in loading has occurred.

The computer returns to command mode after loading. An error message is printed if the checksum error has occurred.

If the tape was started before the leader, a checksum error occurs. The tape must be started on the leader.

## MEMORY EXTENSION USAGE

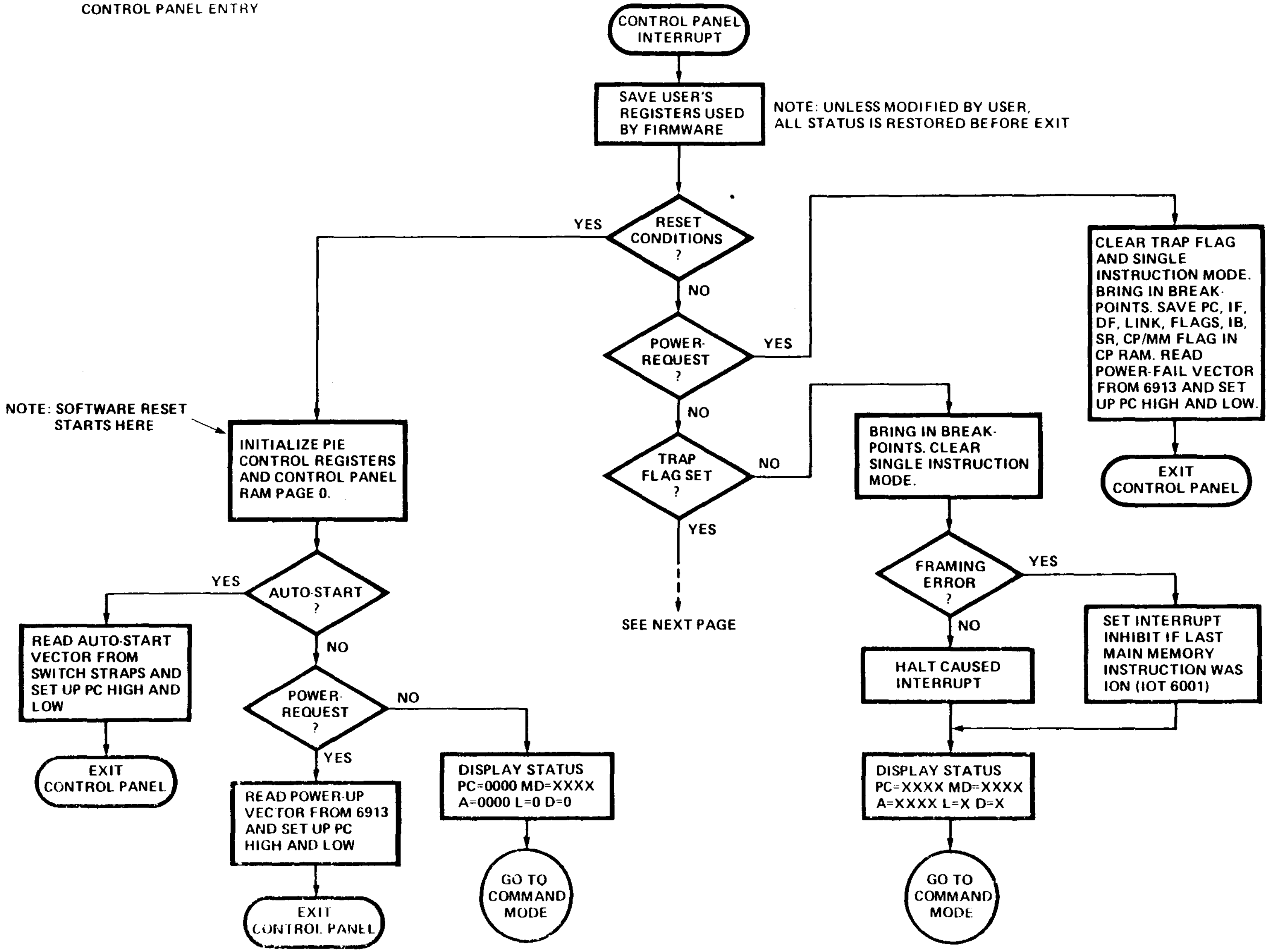
The binary loader may be used to load the binary tape into any valid memory field as described in Chapter 2.

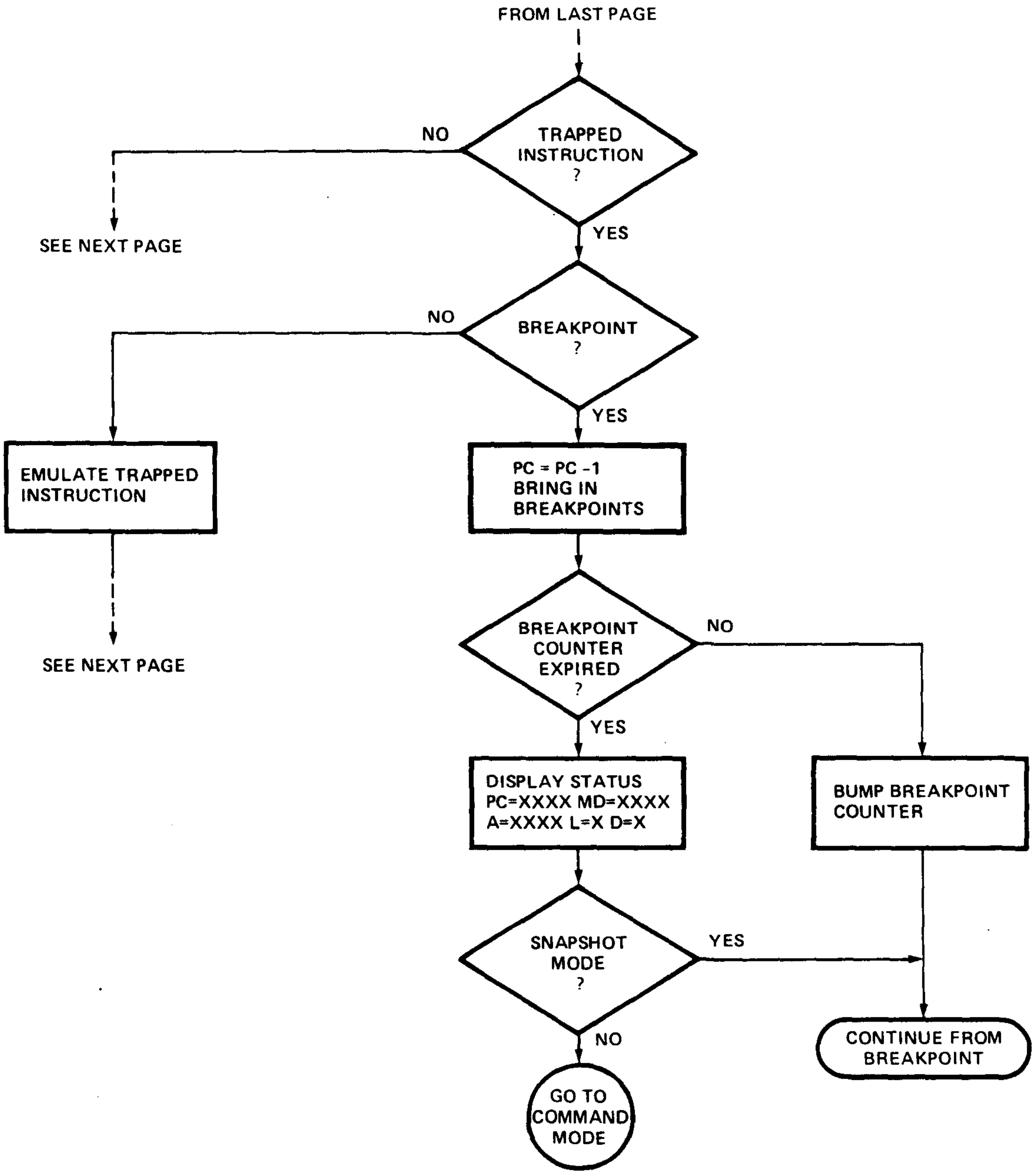
## BINARY LOADER ROUTINE

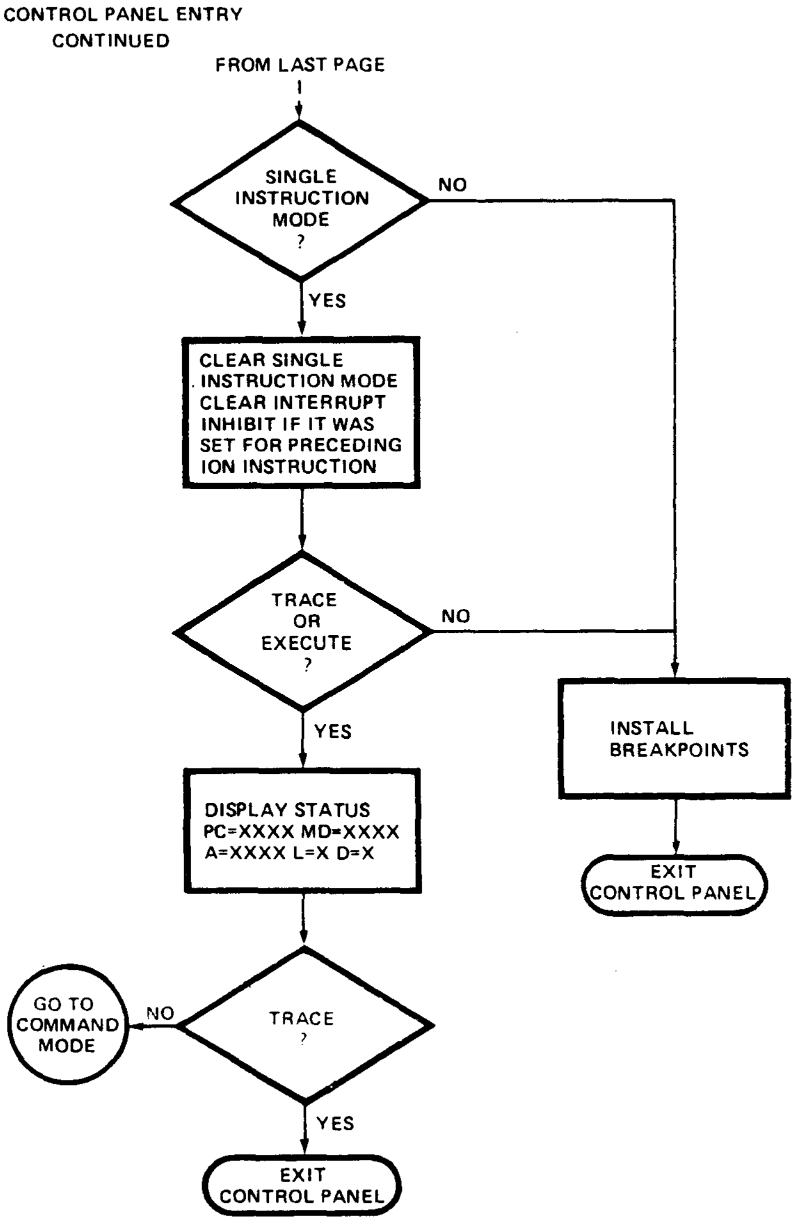
The Program proceeds as follows: The incoming character is tested to see if it is leader or field setting. Leader information is ignored. The "change data field" routine is executed if the character is in the field format.

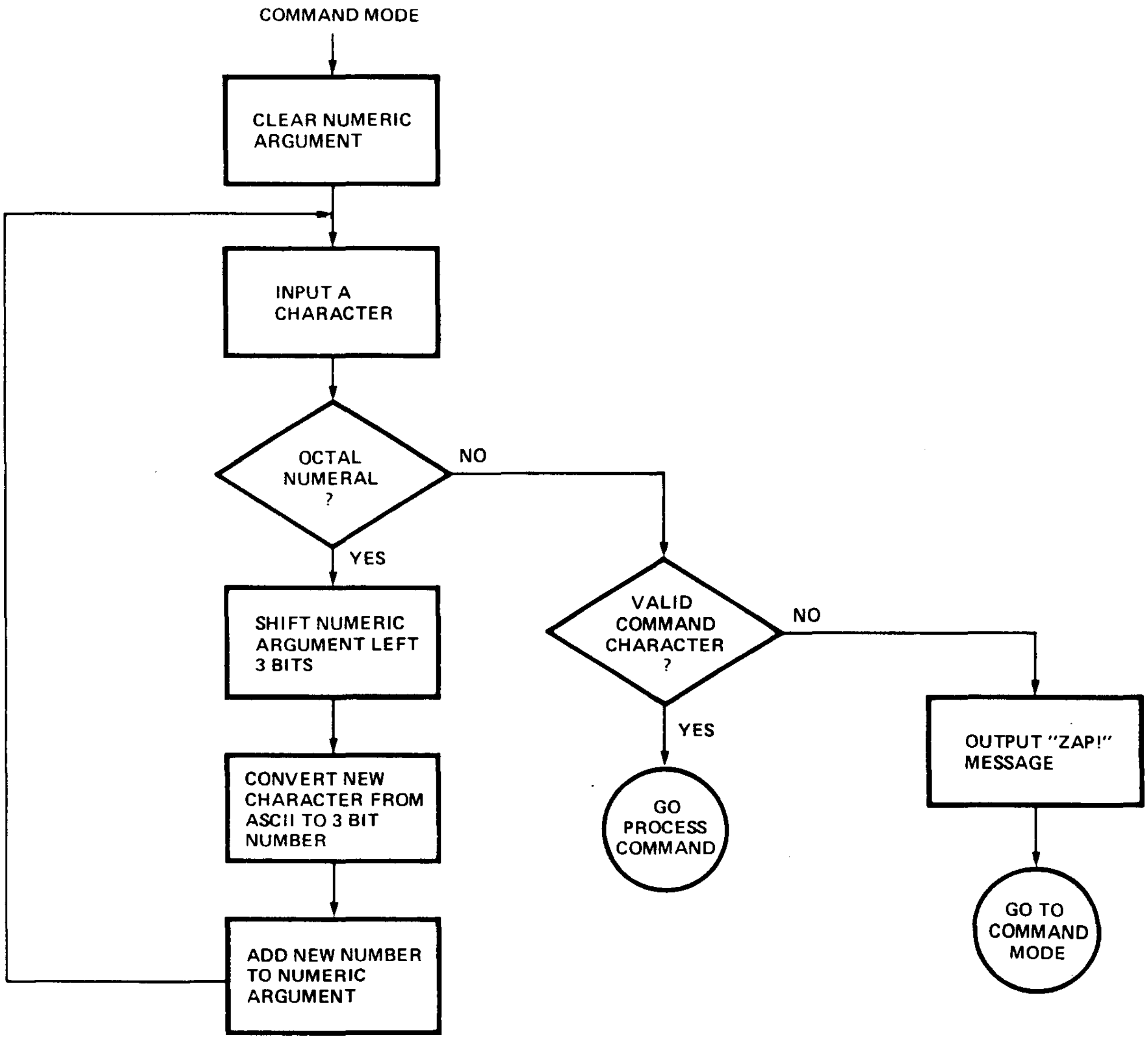
If the character is not part of the leader or field setting, then it is part of the origin address, contains part of the data word and is part of the checksum and the appropriate course is followed. The routine always "looks ahead" by one character to see if trailer follows the character just read. If it does, then the two characters read before the trailer compose the checksum.

## CONTROL PANEL ENTRY

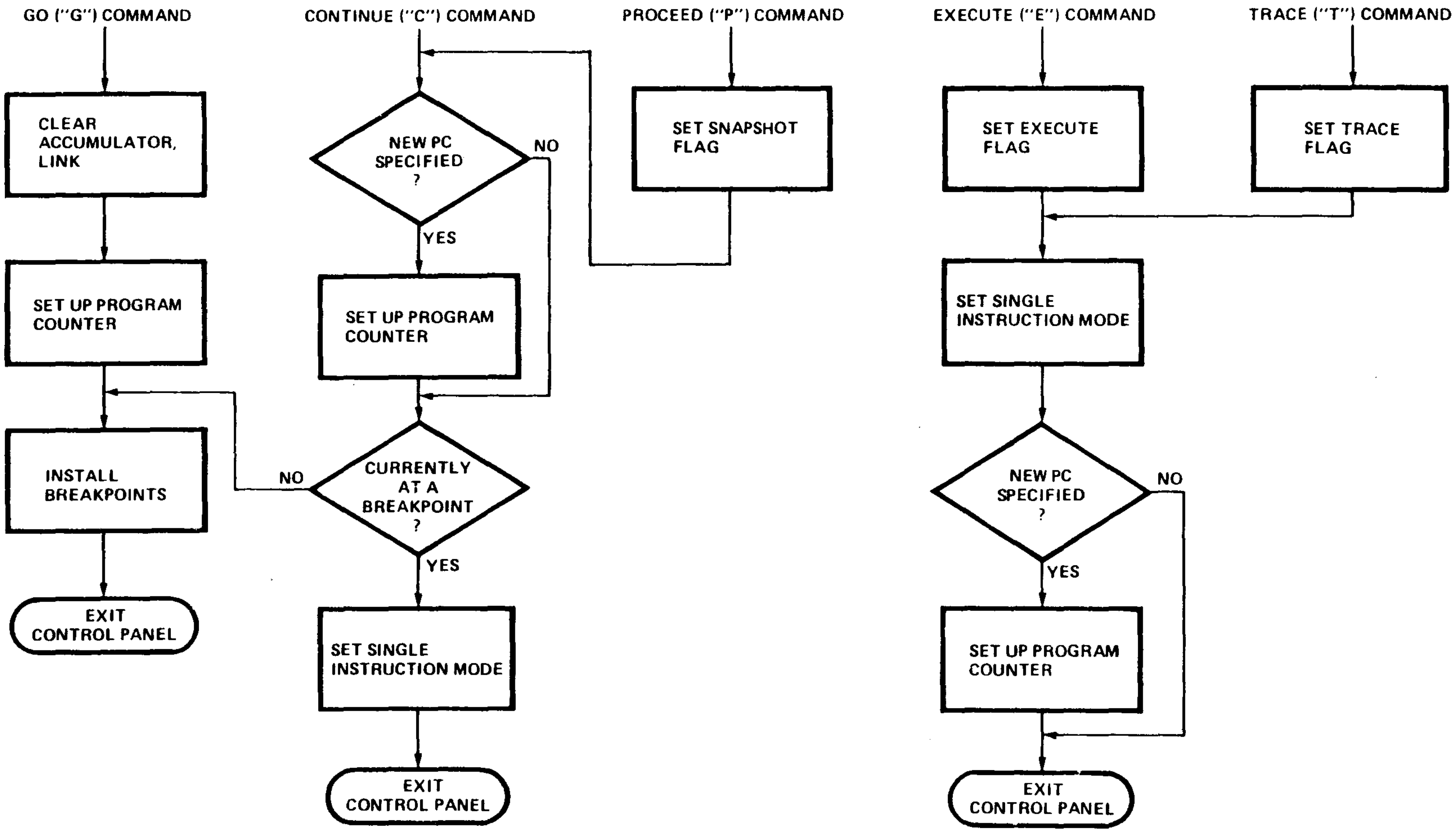

CONTROL PANEL INTERRUPT
SAVE USER'S REGISTERS USED BY FIRMWARE
NOTE: UNLESS MODIFIED BY USER, ALL STATUS IS RESTORED BEFORE EXIT
RESET CONDITIONS ?
YES
NO
POWER-REQUEST ?
YES
NO
CLEAR TRAP FLAG AND SINGLE INSTRUCTION MODE. BRING IN BREAK-POINTS. SAVE PC, IF, DF, LINK, FLAGS, IB, SR, CP/MM FLAG IN CP RAM. READ POWER-FAIL VECTOR FROM 6913 AND SET UP PC HIGH AND LOW.
EXIT CONTROL PANEL
TRAP FLAG SET ?
NO
YES
SEE NEXT PAGE
BRING IN BREAK-POINTS. CLEAR SINGLE INSTRUCTION MODE.
FRAMING ERROR ?
YES
NO
SET INTERRUPT INHIBIT IF LAST MAIN MEMORY INSTRUCTION WAS ION (IOT 6001)
HALT CAUSED INTERRUPT
DISPLAY STATUS PC=XXXX MD=XXXX A=XXXX L=X D=X
GO TO COMMAND MODE
NOTE: SOFTWARE RESET STARTS HERE
INITIALIZE PIE CONTROL REGISTERS AND CONTROL PANEL RAM PAGE 0.
AUTO-START ?
YES
NO
READ AUTO-START VECTOR FROM SWITCH STRAPS AND SET UP PC HIGH AND LOW
EXIT CONTROL PANEL
POWER-REQUEST ?
NO
YES
READ POWER-UP VECTOR FROM 6913 AND SET UP PC HIGH AND LOW
EXIT CONTROL PANEL
DISPLAY STATUS PC=0000 MD=XXXX A=0000 L=0 D=0
GO TO COMMAND MODE


FROM LAST PAGE
TRAPPED INSTRUCTION ?
NO
SEE NEXT PAGE
YES
BREAKPOINT ?
NO
EMULATE TRAPPED INSTRUCTION
SEE NEXT PAGE
YES
PC = PC -1
BRING IN BREAKPOINTS
BREAKPOINT COUNTER EXPIRED ?
NO
BUMP BREAKPOINT COUNTER
YES
DISPLAY STATUS
PC=XXXX MD=XXXX
A=XXXX L=X D=X
SNAPSHOT MODE ?
YES
CONTINUE FROM BREAKPOINT
NO
GO TO COMMAND MODE


CONTROL PANEL ENTRY
CONTINUED
FROM LAST PAGE
SINGLE INSTRUCTION MODE ?
NO
YES
CLEAR SINGLE INSTRUCTION MODE CLEAR INTERRUPT INHIBIT IF IT WAS SET FOR PRECEDING ION INSTRUCTION
TRACE OR EXECUTE ?
NO
YES
INSTALL BREAKPOINTS
DISPLAY STATUS
PC=XXXX MD=XXXX
A=XXXX L=X D=X
EXIT CONTROL PANEL
GO TO COMMAND MODE
NO
TRACE ?
YES
EXIT CONTROL PANEL


COMMAND MODE
CLEAR NUMERIC ARGUMENT
INPUT A CHARACTER
OCTAL NUMERAL ?
NO
YES
SHIFT NUMERIC ARGUMENT LEFT 3 BITS
CONVERT NEW CHARACTER FROM ASCII TO 3 BIT NUMBER
ADD NEW NUMBER TO NUMERIC ARGUMENT
VALID COMMAND CHARACTER ?
NO
YES
GO PROCESS COMMAND
OUTPUT "ZAP!" MESSAGE
GO TO COMMAND MODE

PROGRAM CONTROL COMMANDS



# APPENDIX D

# ROM BASED SUBROUTINE CALLS WITH THE IM6100

Frequently the same sequence of instructions must be executed in different parts of a program. There are obvious advantages to writing a program in which the identical code segment is written only once and invoked when necessary. The code segment is called a "subroutine" since it is a subsidiary part of a larger routine or program. After the subroutine has been executed, a transfer of control is made back to the instruction following the transfer to the subroutine. This immediately poses the problem of how the subroutine knows which location to return to since many different parts of the main program "call" the same subroutine.

## IM6100 SUBROUTINE CALL

In the IM6100, the JMS, Jump to Subroutine, instruction eliminates the need for writing a set of instructions each time a repetitive task is performed, such as finding a square root or typing a character on the Teletype. Since the IM6100 is software compatible with the DEC PDP-8/E, it uses the same subroutine linkage. It stores the "return" address in the first location of the called subroutine, and after the subroutine has been executed, a return transfer is made by jumping "indirectly" through the first location of the subroutine. This is a simple means of returning to the correct location of the program upon completion of the task.

This convention, though extremely simple and straightforward, has two drawbacks. First, when the user program is stored in read-only memory, ROM, the JMS instruction cannot be used to call a ROM based subroutine since one cannot write into a read-only location to establish the return link. The second drawback is associated with "recursive" subroutine calls. There are instances when a subroutine may call itself over and over, recursively. Obviously, the simple linkage mechanism does not work since a call to itself destroys the return address associated with the call immediately preceding it. Although it is possible to design around recursive techniques, recursion is important, in some cases, since it permits a better structured program with less memory when compared with iterative designs.

Note that this problem is not the same as one subroutine calling another.

## LINKAGE THROUGH RAM

If one is not interested in recursion, which is true in most instances, ROM based subroutines may be called by providing a RAM entry point for each subroutine. For example, a subroutine in ROM location $6600_8$ may be called from location $5013_8$ with the linkage mechanism, shown below:

```
                        /CALLING A SUBROUTINE BY LINKING THRU RAM
                        /SUBROUTINE IN LOCATION 6600 (ROM)
                        /EXAMPLE CALL IN LOCATION 5013

       *5013            *5013

05013  4170             JMS     SUBR             /CALL SUBROUTINE
                        /CONTROL RETURNS HERE AFTER SUBROUTINE

       *0170            *170                     /COULD BE ANYWHERE IN RAM

00170  0000   SUBR,     0000                     /SUBROUTINE RETURN ADDRESS
00171  5572             JMP I   .+1              /GO TO SUBROUTINE
00172  6600             XSUBR                    /THRU HERE

       *6600            *6600                    /COULD BE ANYWHERE IN ROM

06600  7000   XSUBR,    NOP                      /DO NOTHING
06601  5570             JMP I   SUBR             /RETURN TO MAINLINE PROGRAM
```

Memory overhead for each subroutine in the program:

3 RAM locations in Page Zero, two of which must be initialized at power-on.
6 ROM locations to initialize the two locations in RAM.

## RETURN STACK

ROM based subroutines, as well as recursion, can be handled through the medium of a pushdown stack or LIFO (Last-in-first-out). Most of the currently available microprocessors put subroutine return addresses into a stack memory which may be part of the CPU chip or part of the external memory.

When return addresses are stored in an on-chip pushdown stack, there is a natural limit to the number of dynamic subroutines active at any given time. For example, if there are eight stack positions, then, generally, only seven subroutine calls may be active at one time since the real used stack size must be kept smaller to allow some stack depth for interrupt service routines, if any. This, of course, assumes that no processor state information other than the Program Counter need be saved when calling subroutines. If the Accumulator or other status information must be saved, the number of subroutines that may be "simultaneously" active is significantly reduced. The on-chip stack does allow for faster subroutine calls since external memory accesses are kept to a minimum.

Another approach is to maintain a stack pointer in the CPU and to store return addresses in the external read-write memory. When a subroutine is called, the return address is pushed into the RAM stack and the pointer is updated. Stacks in RAM are of potentially huge depth and this allows certain kinds of algorithms to be easily programmed. If the on-chip stack is accessible to the programmer, the depth of the stack can be extended by software. Most on-chip stack manipulations are cumbersome and time consuming, and this imposes a rigid limit on the allowed depth of the subroutine calls. Since most microprocessor applications involve some amount of external RAM, the external RAM stack solution is widely accepted. The microprocessor chip area is also reduced by providing the stack memory externally.

## SOFTWARE STACK

The IM6100 architecture provides for the simulation of a stack in software. The following is one example of a software stack subroutine linkage mechanism.

### PROGRAM DESCRIPTION

A subroutine is "called" by invoking a supervisory routine, CALL, followed by the entry address of the subroutine. Call leaves the Program Counter, PC, on a stack, starting at a user defined base. A return from the subroutine is executed with another supervisory routine, RETURN, which links back to the main program. The "entry address" which follows CALL is skipped over when returning from the subroutine.

AC, LINK and MQ are not affected. The supervisory routines do not check for stack overflow or underflow. The program makes no provision for interrupt service routines using the stack since the locations used for temporary variables by a subroutine call or return may be overwritten by the higher priority interrupt service call. The program is easily modified to save AC or any other processor state information on the stack and since the stack pointer itself is maintained in memory, one can also check for overflow and underflow conditions.

The supervisory routines may be assembled any place in the user program. For illustration purposes, we have assigned arbitrary locations. The user memory is expected to be organized as RAM in the lower pages and ROM in the higher pages. The CALL and RETURN routines use six locations in page zero. Since page zero is directly accessible from any other page, the supervisory routines may be called from any location in memory.

Four of the page zero locations used by the supervisory routines must be initialized when power is turned on. The IM6100 Program Counter is set to $7777_8$ when the RESET line is active. The power-on routine, starting at $7777_8$, is expected to initialize the user system.

Fixed memory overhead for CALL and RETURN:

> 6 RAM locations in Page Zero, four of which must be initialized at power-on.
> 16 ROM locations for routines
> 12 ROM locations for power-on initialization.

Memory overhead for each active call:

1 RAM location for the stack to grow.

PAL convention:

The symbols CALL and RETURN must be defined in the user program, as shown below:

CALL = JMS CALLX
RETURN = JMP I RETX

```
/IM6100 SOFTWARE STACK ROUTINES  IFDOS PAL 2A        01-MAY-78  PAGE 1

                  /IM6100 SOFTWARE STACK ROUTINES

                  /PAGE ZERO RAM LOCATIONS
      *0160               *160                      /DEPENDS UPON DESIRED STACK DEPTH

00160  0000  CALLX,  0000                          /SUBROUTINE RETURN ADDRESS
00161  5562          JMP I    .+1                  /GO TO SUBROUTINE CALL ROUTINE
00162  7400          CALLY                         /THRU THIS POINTER

00163  7411  RETX,   RETY                          /SUBROUTINE RETURNS THRU HERE
00164  0000  AC,     0000                          /TEMPORARY LOCATION FOR AC

00165  0165  STACK,  .                             /STACK POINTER

             /NOTE: THE LOCATIONS CALLX, CALLX+1, RETX, AND STACK MUST
             /BE INITIALIZED BY USER SOFTWARE AT POWER-ON.

             /ROM LOCATIONS
      *7400          *7400                         /COULD BE ANYWHERE

07400  3164  CALLY,  DCA      AC                   /SAVE AC TEMPORARILY
07401  2165          ISZ      STACK                /BUMP STACK POINTER
07402  1160          TAD      CALLX                /GET RETURN ADDRESS
07403  7001          IAC                           /INCREMENT TO SKIP OVER
                                                   /USER SUBROUTINE ADDRESS
07404  3565          DCA I    STACK                /AND SAVE ON STACK
07405  1560          TAD I    CALLX                /GET USER SUBROUTINE ADDRESS
07406  3160          DCA      CALLX                /SAVE IT
07407  1164  CALOUT, TAD      AC                   /RESTORE AC
07410  5560          JMP I    CALLX                /GO TO USER SUBROUTINE

07411  3164  RETY,   DCA      AC                   /SAVE AC TEMPORARILY
07412  1565          TAD I    STACK                /"POP" RETURN ADDRESS
07413  3160          DCA      CALLX                /SAVE IT
07414  7060          CMA CML                       /AC=7777, COMPLEMENT LINK
07415  1165          TAD      STACK                /RESTORE LINK AND
07416  3165          DCA      STACK                /UPDATE STACK POINTER
07417  5207          JMP      CALOUT               /RESTORE AC AND RETURN TO MAINLINE

             /EXAMPLE OF USER PROGRAM CALLING A SUBROUTINE
      *1234          *1234                         /COULD BE ANYWHERE
       4160          CALL=    JMS      CALLX       /DEFINE 'CALL' INSTRUCTION
       5563          RETURN=  JMP I    RETX        /DEFINE 'RETURN' INSTRUCTION

01234  4160          CALL                          /CALL 'DUMMY'
01235  2345          DUMMY                         /SUBROUTINE
                     /CONTROL RETURNS HERE AFTER SUBROUTINE

      *2345          *2345                         /COULD BE ANYWHERE

02345  7000  DUMMY,  NOP                           /DO NOTHING
02346  5563          RETURN                        /AND RETURN
```

# APPENDIX E

## CONTROL PANEL PROGRAMMING

### INTRODUCTION

Control panel programming for the INTERCEPT is more difficult than main memory programming for many reasons. The most important is that no debugging facilities exist for control panel programs. Therefore, control panel programs should be written and debugged as thoroughly as possible in main memory.

### PROGRAMMING CONSIDERATIONS

The control panel memory space of the INTERCEPT consists of 2048 (2K) words of Read Only Memory (ROM) and 256 words of read/write Random Access Memory (RAM). ROM addresses range from $4000_8$ to $7777_8$. RAM addresses range from $0000_8$ to $0377_8$. The RAM is volatile--its contents are lost when power goes off. Thus, RAM constants and pointers used by ROM programs must be initialized at power-up.

Control panel interrupts are caused by:

1. any form of HALT executed in main memory
2. UART Framing Error from either I/O port (e.g.,Break key)
3. AC power low sensed by 6913 module, if present
4. power-up
5. any form of OSR executed in main memory
6. any IOT in the range $6007_8$-$6046_8$ executed in main memory
7. hardware single instruction

The last three types of control panel interrupt are also called "traps". Most of the above are self-explanitory. A UART Framing Error occurs when the BREAK key is typed on the keyboard of a terminal connected to either I/O port. The hardware single instruction occurs when a control panel program resets PIE B Control Register A bit 6 (WP1). This allows exactly one main memory instruction to execute followed immediately by a hardware single instruction control panel interrupt.

Control panel interrupts cause the microprocessor to store the PC in control panel location $0000_8$ and fetch the instruction in control panel location $7777_8$. A "JMP I 7776" is usually used in control panel location $7777_8$ to transfer control to the desired address via $7776_8$.

If control panel "transparency" is required, all system status altered by the control panel program must be saved for later restoration. This might include the following items, although this list is not all-inclusive: AC, Link, Data Field, MQ, Interrupt Inhibit Flip-Flop, terminal flags, etc. It is advantageous to modify as little status as possible to minimize control panel memory usage and execution time.

When power is applied to the system, all four PIE Control Registers and the volatile control panel RAM must be initialized. It is usually undesirable to initialize at any time other than power-up, so the control panel program must distinguish between this and other forms of control panel entry. The following conditions indicate a power-up entry: PC=7777, Instruction Field=0, AC=0000, Link=0, Interrupt Enable=0, Interrupt Inhibit=0, Interrupt Request=0. In most cases, only the first three or four conditions are sufficient to determine power-up with a reasonable probability.

After checking for power-up, the control panel program can use the SNPRQ, SKPTRP, SKPOSR, and SKPFRE instructions to determine the cause of control panel entry (see Chapter 5).

After control panel functions are completed, the control panel program should restore system status as necessary and exit with the sequence:

```
        ION        /EXIT PANEL
        JMP I 0    /VIA SAVED PC
```

PRECAUTIONS AND RESTRICTIONS

1. With few execeptions, the condition that created control panel entry should be cleared before exiting.

2. A dummy IOT must be executed in control panel before other IOTs function correctly. IOF is a good choice. The dummy IOT destroys the AC.

3. Never execute IOT $6047_8$ in control panel.

4. Trapped IOTs are NOPs in control panel. All trapped IOTs executed in main memory cause the same entry conditions. The control panel program must determine which IOT caused the trap and perform the correct function according to the user's needs.

5. OSRs in control panel read either zeros or the auto-

start straps as per PIE B CRA FL3. Control panel OSRs cause a pulse on PIE B sense 3, which should generally be cleared (with IOT $6072_8$) before leaving panel.

6. Resetting PIE B WP1 to 0 causes control panel entry after exactly one main memory instruction execution (hardware single instruction). This toggles the RUN/HLT Flip-Flop, so IOT $6061_8$ should generally be used to restore the flip-flop before leaving control panel. A hardware single instruction trap sets PIE B sense 1 so that control panel programs may sense the entry condition.

7. If a 6913 Power-fail Module is used and it indicates a power transition is in progress, Framing Error control panel interrupts are inhibited.

8. Main memory HLT instructions cause control panel entry. The IM6100 does not actually halt, but the RUN/HLT Flip-Flop is toggled and generally should be restored before leaving control panel.

9. PIE control registers, sense inputs, and control panel RAM is arbitrary at power-up until initialized by the control panel program.

10. The CLRTRP and RESET instructions (see Chapter 5) affect the AC in an undefined manner and should generally be followed by a CLA instruction.

11. If a main memory ION instruction is immediately followed by a control panel interrupt (e.g.,Framing error or hardware single instruction), the device interrupt system will be enabled upon leaving control panel. This could cause the main memory program to malfunction because normally the device interrupt system is not enabled until after the instruction following the ION is executed. One solution is for the control panel program to check the main memory location pointed to by PC-1 whenever the control panel interrupt is caused by the above conditions. If the instruction was ION, the control panel program can set the Interrupt Inhibit (if not already set) and the hardware single instruction bit before leaving control panel. Upon the next control panel entry, the Interrupt Inhibit must be cleared (if set by the control panel program) as well as the hardware single instruction bit.

12. If the control panel program outputs data to the serial I/O ports, it should first check the "flag" to see if the port is already transmitting data. If the flag indicates the port is not ready, the program should time-out the flag for one "character time" (the length depends on the baud rate) to prevent destruction of data

transmitted by the main memory program over the same port. Also, the flag should generally be restored before leaving control panel to ensure "transparency". This includes waiting for control panel output data to be fully transmitted before leaving panel.

# APPENDIX F

## GENERAL SYSTEM RESTRICTIONS

1). User designed modules must input buffer the following critical signals with 7414 (Hex Schmitt inverter) or 74132 (Quad 2-input Schmitt NAND gate) for noise immunity:

DMAGNT
XTA
XTB
XTC
IFETCH
UP

2). The INTERCEPT power supplies are rated to power the 6912 - CPUCTRL with one 6901 - M4KX12 and also leave the following currents available for factory or user designed interfaces:

| | |
|---|---|
| INTERCEPT I | 5V @ 1.25A |
| INTERCEPT II | 5V @ 5A |
| | +12V @ .5A |
| | -12V @ .5A |

3). Pins 1 and 2 of the INTERCEPT bus are daisy chained for priority vectoring. The priority is established by the position of the interface on the bus, as follows:

| | |
|---|---|
| INTERCEPT I | LEFT TOP (highest priority) |
| | LEFT BOTTOM |
| | RIGHT TOP |
| | RIGHT BOTTOM (lowest priority) |
| INTERCEPT II | Leftmost bus position has highest priority |

If a module does not use priority vectoring, pins 1 and 2 on the module should be shorted.

4). Recommended I/O device address usage:

| | |
|---|---|
| 600X | IM6100 internal IOT's |
| 601x, 602x | Control panel traps |
| 603x, 604x | DEC compatible terminal keyboard and printer |
| 605x | Reserved |

| | |
|---|---|
| 606x thru 611x | INTERCEPT CPU PIE IOT's |
| 612x, 613x | IM6102-MEDIC Real Time Clock |
| 614x, 615x | 6913-Power-fail module |
| 616x, 617x | Reserved for INTERCEPT option 1 |
| 620x thru 627x | IM6102-MEDIC EMC/DMA |
| 630x thru 637x | IM6103-PIP |
| 640x thru 667x | Available for user interfaces |
| 670x thru 673x | Reserved for INTERCEPT options 2 and 3 |
| 674x, 675x | INTERCEPT FLOPPY DISK system |
| 676x, 677x | Reserved for INTERCEPT option 4 |

5). User module designs should not present so much load as to exceed the drive capability of the 6912 drivers. For the DX lines, the output buffers of each module must drive the input load of all modules. In most cases, inputs do not need to be buffered to present a single unit load due to the high drive ability of the 6912 output buffers, but each line should be checked to assure proper operation.

6). User designed modules should use the 5 volt power supply provided with the system. Designing modules which cause the system to require more current than provided may cause thermal problems. The most important electrical considerations are decoupling and ground inductance. For decoupling, a .01 to .1 microfarad disc ceramic capacitor placed close to the VCC and ground pins of each IC is adequate. Additionally, a large electrolytic capacitor (25 to 100 microfarad) should be provided on each module to act as storage for the 5V supply. Ground inductance is minimized by using all ground pins on the bus and connecting all ground pins of ICs together with wire or printed circuit traces to form an x-y matrix.

7). Although the INTERCEPT contains many CMOS devices, the system employs a strictly TTL compatible bus. This means that any interface to MOS devices must ensure proper input signal levels and output loading for these devices. Special level shifters are not required since a TTL gate with an additional resistor pull-up provides good input levels for most MOS devices, and these devices usually can drive at least one standard TTL input. When using MOS devices with bare (open) drain outputs external pull-ups should be used in addition to normal TTL loading. The resistor should be a value which causes near maximum rated current for the MOS output. This improves the rise time and noise immunity of the associated signal. Here again, direct connection to the system bus is impossible since the system load exceeds the output drive specification of most MOS devices.

## APPENDIX G

## TELETYPE MODIFICATIONS FOR THE INTERCEPT SYSTEM

The following should be performed by qualified Teletype service personnel.

The Intersil INTERCEPT systems can be used. in conjunction with a Model ASR-33 Teletype. Before attempting to use your system inspect your Teletype for the following modifications and additions. If they have not yet been performed, you must complete them before using INTERCEPT.

To check for, or make, these modifications remove the cover of the Teletype. Loosen the three thumb screws in the back and remove the Platen that holds the roll of paper, the Mode Switch knob and the Face Plate. Remove the small screw on the Reader cover and the four screws under the Face Plate. You should now be able to lift the cover off. Use Figure G-1 to locate the various parts located below.

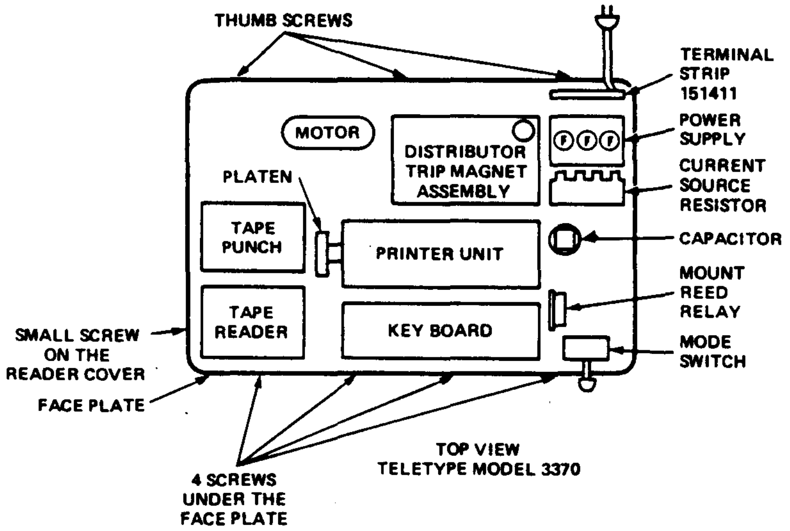


FIGURE G-1

TOP VIEW OF TELETYPE MODEL 3370

The modifications are:

CURRENT LOOPS CHANGED FROM 60 TO 20 MILLIAMPS

The Current Source Resistor must be changed from 750 ohms to 1450 ohms. This is done by moving the BLUE wire from Terminal #3 to Terminal #4 of the large power resistor shown in Figure G-2. The receiver current level is changed by moving the PURPLE wire of Terminal #8 on Terminal Strip 151411 to Terminal #9 on the same strip. Terminal Strip 151411 is shown in Figure G-3 with Terminal #1 on the far left.

## TELETYPE WIRED FOR FULL DUPLEX OPERATION

The half duplex wiring must be changed by moving the BROWN/YELLOW wire from Terminal #3 to Terminal #5 and the WHITE/BLUE wire from Terminal #4 to Terminal #5 on Terminal Strip 151411.

## THE READER RUN RELAY ADDED

The Reader circuit should have a 12 volt relay inserted to allow program control of the reader. This relay is shown along with the mode switch in Figure G-4. Mount the relay with two 6-32 screws on the available bracket. A schematic diagram for the relay and its connections is shown in Figure G-6. Locate the BROWN wire coming from the Distributor Trip Magnet which is connected to J4 pin 11 as shown in Figure G-5. Cut the BROWN wire and connect to the wire marked BROWN on the relay circuit (note that this leaves J4 pin 11 with no connection). Connect the wire marked LINE to terminal L1 and the wire marked LOCAL to terminal N of the mode switch as in Figure G-6. A preassembled Reader Relay Card (6909-RRELAY) is available from Intersil, Inc.

## LEVEL 8 OPTION WIRED TO 'ALWAYS MARK'

The level 8 option must be changed from parity to 'ALWAYS MARK'. This causes the keyboard to always output a 1 for the 8th bit, and the Reader to read the 8th bit as it was written. Locate the Left Contact Block and the Right Contact Block as shown in Figure G-7. It may be necessary to remove a clear plastic shield to gain access to the Left Contact Block. On the Left Contact Block remove the RED/GREEN wire from the upper left contact, leave the RED/GREEN wire open and connect the GREEN wire to the upper left contact. On the Right Contact Block connect the GREEN wire to the upper left contact. For a detailed reference see TELETYPE keyboard schematic 9334WD.

## CONNECT INTERCEPT CABLE TO THE TELETYPE

The cable is connected to Terminal Strip 151411 and the relay as shown in Figures G-6 and G-8.

## "ANSWERBACK" DISABLED

Between the keyboard and the printer unit below the carriage movement area, there is a piece of metal with several dozen notches which is used to select various options. Directly below some notches is a "function lever", and a spring can be connected between each notch and each lever to select a function. Removing the spring disables the corresponding function.

The second function lever from the right enables "Answerback" if the corresponding spring is present. Every time the computer sends a "CONTROL/E" (ASCII code 205) to the Teletype, the Teletype "answers back" as if the "HERE IS" key was pressed, if the function is enabled.

The "answerback" function must be disabled by removing the second spring on the right from its corresponding function lever to ensure correct system operation.

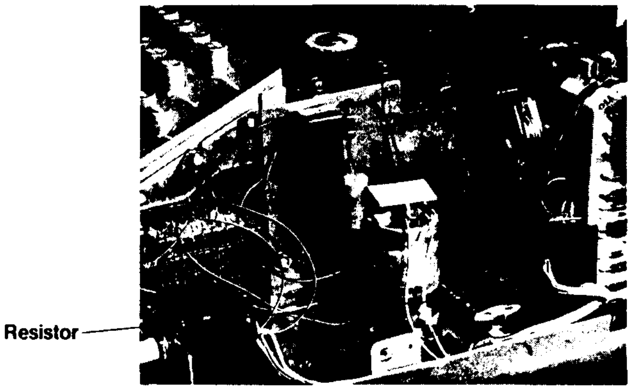


Figure G-2. Current Loop Resistor

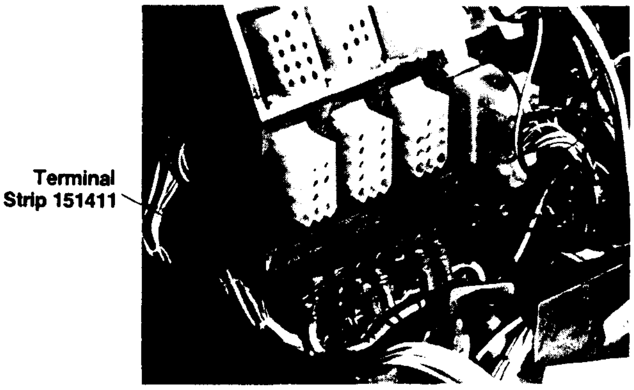


Figure G-3. Terminal Strip

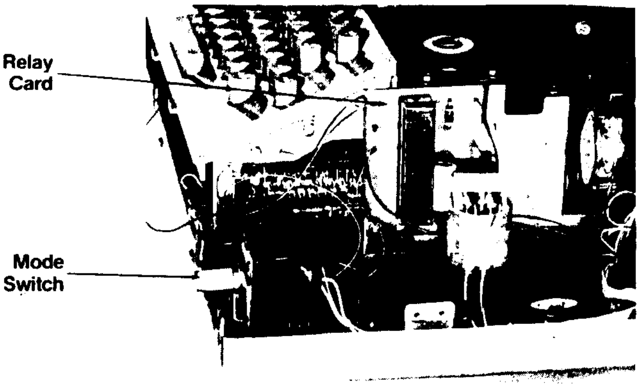


Figure G-4. Relay Card

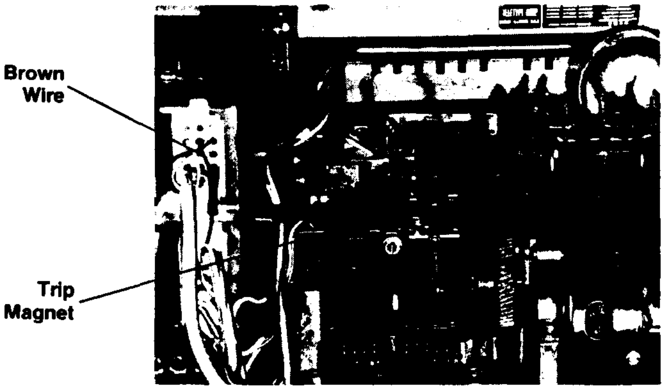


Figure G-5. Distributor Trip Magnet

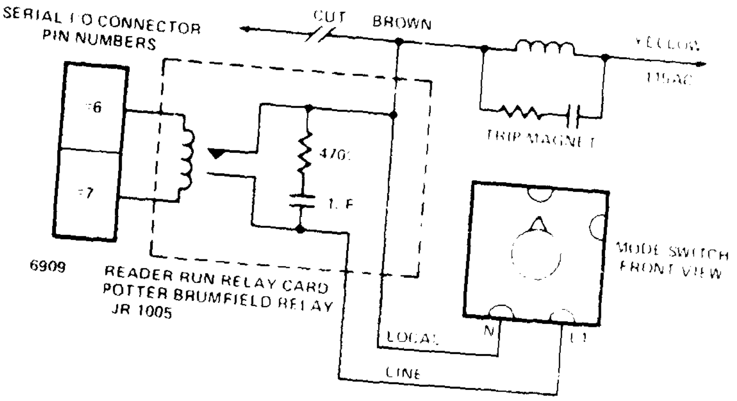


Figure G-6. Reader Relay Circuit

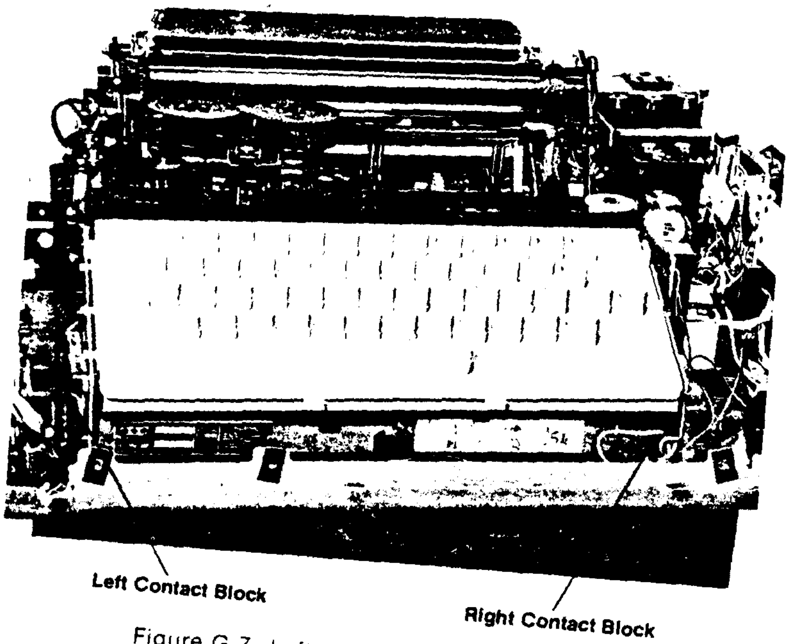


Figure G-7. Left and Right Contact Blocks

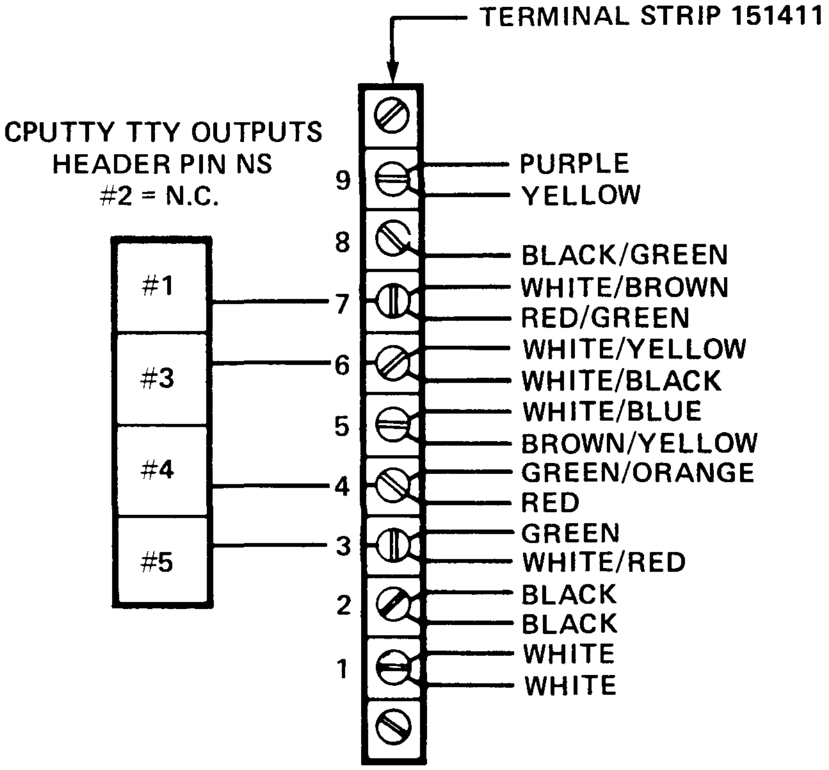


FIGURE G-8

TELETYPE CONNECTION DIAGRAM

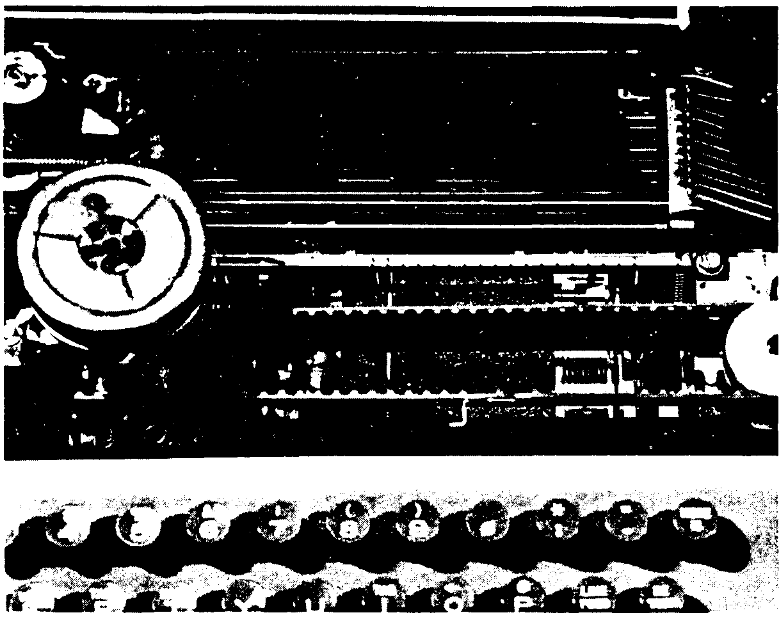


FIGURE G-9

ANSWERBACK DISABLED

# APPENDIX H

## CONTROL PANEL/DEBUGGER COMMAND SUMMARY

### MEMORY EXAMINATION /MODIFICATION

| | |
|---|---|
| nnnnn/ | Open and display location nnnnn. |
| <CR> | Close currently open location. |
| <LF> | Close current location and open next. |
| nnnnn | Display memory from nnnnn to end of memory page. |
| SHIFT/N | Close current location, use contents as memory reference to open next. |
| SHIFT/O | Close current location, use contents as indirect to open next. |
| SHIFT/P | Return to previous sequence. |
| nnnn+ | Close current location, open current +nnnn. |
| nnnn- | Close current location, open current -nnnn |
| ; | Close current location, open next without display. |

### INTERNAL REGISTER EXAMINATION/MODIFICATION

| | |
|---|---|
| A | Open and display user's Accumulator. |
| L | Open and display user's Link. (Only bit 11 significant). |
| D | Open and display user's Data Field. (Only bits 6-8 significant). |
| S | Open and display switch register. |
| Q | Open and display user's MQ. |
| M | Open and display search mask. |
| H | Open and display high limit. <LF> opens low limit. |
| F | Open and display W, X, CTRL/W, I, O extended address. (Only bits 6-8 significant). |
| R | Open and display replacement register for CTRL/W. |
| K | Open and display default breakpoint iteration counter. |
| N | Open and display breakpoint identification register. |
| J | Open and display Primary Port transmit time-out. <LF> opens nulls count. |

### PROGRAM CONTROL

| | |
|---|---|
| nnnnnG | Clear AC, and Link, set DF equal to IF, |

| | |
|---|---|
| | and go at nnnnn. |
| nnnnnC | Continue at nnnnn. |
| nnnnnE | Execute one instruction at nnnnn. |
| nnnnnT | Trace instructions beginning at nnnnn. |
| nnnnnP | Proceed in snapshot mode from nnnnn. |
| nnnnCTRL/W | Go at nnnn in control panel memory. |
| xxxx. | Set trace counter to xxxx. |
| xxxx: | Set K to xxxx. |

BREAKPOINT

| | |
|---|---|
| b, | Set N to b. |
| nnnnnB | Set breakpoint at location nnnnn. |
| Y | Yank all breakpoints. |
| # | Display addresses of breakpoints set. |

MEMORY SEARCH/SEARCH AND REPLACE

| | |
|---|---|
| mmmmW | Display all locations between H and L meeting search conditions. |
| mmmmX | Display all locations between H and L not meeting search conditions. |
| mmmmCTRL/W | Replace all memory between H and L meeting search conditions with R. |

INPUT/OUTPUT

| | |
|---|---|
| I | Read BIN format input. |
| O | Produce BIN format output. |
| 1Z | Bootstrap IFDOS. |

MISCELLANEOUS

| | |
|---|---|
| nnnnnU | Set User Display Address to nnnnn. |
| U | Disable User Display Address feature. |
| V | Display user interrupt enable, interrupt inhibit, and interrupt request. |
| ? | Repeat status display. |
| 0<ESC> | Reference main memory with control panel/ debugger commands. |
| 1<ESC> | Accept control panel/debugger commands for Primary Port. |
| 2<ESC> | Accept control panel/debugger commands from either Port. |
| 3<ESC> | Reference control panel memory with control panel/debugger commands. |
| 4<ESC> | ROM 1 error check. |
| 5<ESC> | ROM 2 error check. |

| | |
|---|---|
| 7<ESC> | Software reset. |
| xxxx" | Set Secondary Port baud rate as per Table 2-1 |

# APPENDIX I

## GETTING STARTED WITH INTERCEPT

1.) After unpacking system, save all packaging materials for future use. All warranties are null and void unless the system is returned in the original packaging.

2.) Use the checklist on shipping carton exterior to verify all contents were delivered. Report discrepencies to the distributor.

3.) Visually inspect all system components for shipping damage. Report damage to the distributor.

DO NOT CONNECT AC POWER AT THIS TIME!!

4.) If an RS-232 terminal is used, connect one of the cables marked RS-232 between the connector on the rear of the INTERCEPT marked "PRIMARY" and the RS-232 connector on the terminal. For 20ma current loop (i.e., TELETYPE) operation, connect the cable marked "CURRENT LOOP" to the "PRIMARY" connector on the INTERCEPT and refer to Appendix D, "Instructions for Teletype Connections and Modifications".

5.) The baud rate for the INTERCEPT primary port is set at the factory for 9600 baud. If the desired terminal does not support 9600 baud operation, refer to Chapter 10, "Changing Primary Port Baud Rate" section.

6.) Compare the AC power requirements of the INTERCEPT (stamped on the bottom or rear of the enclosure) with the power available to ensure nominal voltage and frequency are within acceptable limits.

7.) Connect INTERCEPT AC cable between the INTERCEPT and the intended AC power source. Connect and switch "ON" AC power to the terminal. Using the power switch on the side or rear of the INTERCEPT enclosure, switch "ON" the INTERCEPT.

8.) Place the INTERCEPT reset switch in the "RESET" position if it is not already there. Place the reset switch in the "NORMAL" position, and the following is displayed on the terminal:

```
*PC7777  MDxxxx  A0000  LO  DO
```

("xxxx" means don't care).

9.) Do not block air flow vents on the enclosure.

If successful, complete and mail the warranty card in this manual IMMEDIATELY. Failure to do so prevents INTERSIL from supporting your system with updates and "bug fixes".

READ THIS MANUAL BEFORE USING INTERCEPT!

If the start-up procedure is unsuccessful, verify steps 1 through 7 and repeat step 8. If necessary, check terminal cable connections, AC connections, power supply fuses, terminal LINE-LOCAL switch, terminal baud rate, and other obvious problems.

DO NOT OPEN THE INTERCEPT ENCLOSURE!

If necessary, refer to Chapter 10, "INTERCEPT Service and Maintenance Information".

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Req. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 1 | 15-0328 | Enclosure, Optima | | 1 | | | | |
| 2 | 15-0337 | Front Cover | | 1 | | | | |
| 3 | 15-0340 | Assembly, Card File & Bus | | 1 | | | | |
| 4 | 15-0341 | " Rear Panel | | 1 | | | | |
| 5 | 15-0342 | " Serial I/O Connectors | | 1 | | | | |
| 6 | 15-6912 | " CPUCTRL | | 1 | | | | |
| 7 | 15-6901 | " M4KX12 | | 1 | | | | |
| 8 | 15-0343 | Cable Assembly (RS-232) | | 2 | | | | |
| 9 | 15-0344 | " " (Current Loop) | | 1 | | | | |
| 10 | 15-0339 | Power Cord, SVT Vinyl | | 1 | | | | Belden 17250 |
| 11 | 15-0365 | Screw, #10-24x3/8 Truss Hd. | | 4 | | | | |
| 12 | 15-0163 | " #6-32x3/8 Pan Hd. | | 4 | | | | |
| 13 | 15-0367 | Speed nut, #10-24 | | 4 | | | | Tinnerman |
| 14 | | | | | | | | |
| 15 | - | Foam Strip 1/2" Thk x 3/4" w (self-adhesive) | | 4 ft. | | | | |
| 16 | | | | | | | | |
| 17 | - | Shipping Carton | | 1 | | | | |
| | | | | | | | | |

Board No. 6910 Descr. Intercept II

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Page 1 of 1

Rev. A

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Req. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 1 | 15-6912 | Assembly, CPUCTRL | | 1 | | | | |
| 2 | 15-6901 | " MIKx12 | | 1 | | | | |
| 3 | 15-0192 | " BUS | | 1 | | | | |
| 4 | 15-0193 | " Power Supply | | 1 | | | | |
| 5 | 15-0342 | " Serial I/O Connector | | 1 | | | | |
| 6 | 15-0135 | Base, Intercept I | | 1 | | | | |
| 7 | 15-0136 | Bracket, Support | | 2 | | | | |
| 8 | 15-0137 | " Chassis | | 4 | | | | |
| 9 | 15-0132 | Enclosure, Fiberglass | | 1 | | | | |
| 10 | 15-0343 | Cable Assembly (RS-232) | | 2 | | | | |
| 11 | 15-0344 | Cable Assembly (Current Loop) | | 1 | | | | |
| 12 | 15-0155 | Switch, Rocker w/Bezel | | 1 | | | | |
| 13 | 15-0148 | Feet | | 4 | | | | |
| 14 | 15-0149 | Card guides | | 8 | | | | |
| 15 | 15-0146 | Power cord | | 1 | | | | |
| 16 | 15-0147 | Strain Relief | | 1 | | | | |
| 17 | 15-0158 | Screw, thumb #6-32 | | 4 | | | | |
| 18 | 15-0159 | " Machine #4-40 x 1/4 pan | | 31 | | | | |

Board No. 6911 Descr. Intercept I

Page 1 of 2

Rev. A

INTERSIL INCORPORATED
10900 N. Tantau Avenue
Cupertino, California 95014

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Req. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 19 | 15-0161 | Screw, Machine #4-40x1/4 Flat hd. | | 4 | | | | |
| 20 | 15-0163 | " #6x32x3/8 Pan | | 4 | | | | |
| 21 | 15-0164 | " #8-32x1/4 RH | | 2 | | | | |
| 22 | 15-0166 | " #8-32x3/4 RH | | 4 | | | | |
| 23 | 15-0150 | Washer, Nylon #4 Flat | | 8 | | | | |
| 24 | 15-0169 | Kep nut, #4-40x1/4 | | 2 | | | | |
| 25 | 15-0170 | Standoff, 8-32x1/2 M/F | | 2 | | | | |
| 26 | 15-0151 | Graphic Overlay Panel Left | | 1 | | | | |
| 27 | 15-0152 | " " Center | | 1 | | | | |
| 28 | 15-0153 | " " Right | | 1 | | | | |
| 29 | 15-0154 | Label, Serial/Model | | 1 | | | | |
| 30 | 15-0355 | Connector, 4 cond. Receptacle | | 1 | | | | |
| 31 | 15-0357 | Contact, socket | | 4 | | | | |
| 32 | 15-0210 | Terminal, Ring tongue #8 | | 1 | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | - - | Wire, Twisted Pair Black and White UL#1015 18AWG | | 2 ft. | | | | |

Board No. 6911 Descr. Intercept I

Page 2 of 2

INTERSIL INCORPORATED
10900 N. Tantau Avenue
Cupertino, California 95014

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Reg. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 1 | 15-0326 | P. C. Board | - | 1 | | | | |
| 2 | IM6100 I | IC, Microprocessor | 1C | 1 | Socketed | | | (Third Assembly) |
| 3 | IM6102 I | " MEDIC Controller | 2C | 1 | " " | | | (Third Assembly) |
| 4 | IM6101 I | " PIE | 1E, 2E | 2 | | | | |
| 5 | IM6402 I | " UART | 6F, 7F | 2 | | | | |
| 6 | IM6312 I | " 1Kx12 ROM (63S006) | 3E | 1 | Socketed | | | (Third Assembly) |
| 7 | IM6312 I | " " " (63S007) | 4E | 1 | Socketed | | | (Third Assembly) |
| 8 | IM6561 I | " 256x4 RAM | 3F, 4F, 5F | 3 | | | | |
| 9 | IM5624 C | " 512x4 PROM | 6C | 1 | | | | |
| 10 | ICM 7209 | " Oscillator - Driver | 1AB | 1 | | | | |
| 11 | LM 149 | " Quad Op-Amp | 2H | 1 | | | | |
| 12 | 15-0200 | " 4702 Baud Rate Gen. | 3D, 4D | 2 | | | | |
| 13 | 15-0227 | " 4011 Quad NAND | 5E | 1 | | | | |
| 14 | 15-0008 | " 8095/74365 T. S. Buffer | 2B, 3B, 3C<br>7C, 8C | 5 | | | | |
| 15 | 15-0259 | " 74LS257 2 input MUX | 6D, 7D, 8D | 3 | | | | |
| 16 | 15-0015 | " 74LS74 Dual D F/F | 6A | 1 | | | | |
| 17 | 15-0369 | " 74LS76 Dual JK F/F | 7A | 1 | | | | |
| 18 | 15-0371 | " DS1489 Quad Receiver | 1H | 1 | | | | |

Board No. 6912 Descr. CPUCTRL

Page 1 of 4

Rev. B1

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Reg. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 19 | 15-0370 | IC, 74123 Dual One-shot | 7B | 1 | | | | |
| 20 | 15-0021 | " 74LS00 Quad NAND | 5A | 1 | | | | |
| 21 | 15-0022 | " 74LS02 Quad NOR | 3A | 1 | | | | |
| 22 | 15-0024 | " 74LS04 Hex Inverter | 4A | 1 | | | | |
| 23 | 15-0025 | " 74LS05 O. C. Inverter | 5C | 1 | | | | |
| 24 | 15-0368 | " 74LS09 Quad AND O. C. | 5B | 1 | | | | |
| 25 | 15-0027 | " 74LS10 Triple NAND | 1AA | 1 | | | | |
| 26 | 15-0263 | " 74LS27 Triple NOR | 2A | 1 | | | | |
| 27 | 15-0258 | " 74LS32 Quad OR | 6B | 1 | | | | |
| 28 | 15-0030 | Res. Network 1Kx13 | 4C | 1 | | | | |
| 29 | 15-0372 | " " 2.7Kx13 | 4B | 1 | | | | |
| 30 | 15-0373 | " " 4.7Kx13 | 1B | 1 | | | | |
| 31 | 15-0004 | Transistor, 2N3638 | Q1, Q2, Q3 | 3 | | | | |
| 32 | 15-0007 | Diode, IN914 | CR1, 2, 3 | 3 | | | | |
| 33 | 15-0201 | Crystal, 2.4576 MHz | Y2 | 1 | | | | |
| 34 | 15-0131 | " 4.000 MHz | Y1 | 1 | | | | |
| 35 | 15-0084 | " 3.26 MHz | Y3 | 1 | | | | |
| 36 | 15-0203 | Switch, 6 pos. DIP | 5D | 1 | | | | (Second Assembly) |

Board No. 6912 Descr. CPUCTRL

Page 2 of 4

Rev. B.

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Reg. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 37 | 15-0374 | Resistor, 200 ohm 1/4 w 5% | R12, R15 | 2 | | | | |
| 38 | 15-0375 | " 240 " " | R13 | 1 | | | | |
| 39 | 15-0043 | " 1K " " | R5, R9 | 2 | | | | |
| 40 | 15-0248 | " 2.4K ohm " | R11, R17 | 2 | | | | |
| 41 | 15-0376 | " 2.7K ohm " | R7, R8, R10 | 3 | | | | |
| 42 | 15-0046 | " 10K ohm " | R1, 4, 14, 18 | 4 | | | | |
| 43 | 15-0298 | " 22K " " | R16 | 1 | | | | |
| 44 | 15-0300 | " 51K " " | R6 | 1 | | | | |
| 45 | 15-0202 | " 10M " " | R2 | 1 | | | | |
| 46 | | | | | | | | |
| 47 | 15-0054 | Capacitor, Trimmer 5-40pf | C5 | 1 | | | | |
| 48 | 15-0273 | " 30pf Silver Mica | C4, 6, 7, 8 | 4 | | | | |
| 49 | 15-0275 | " 56pf Silver Mica | C1, 2, 3 | 3 | | | | |
| 50 | 15-0377 | " 68pf NPO ceramic | C10, 11 | 2 | | | | |
| 51 | 15-0305 | " .0015 ufd ceramic | C14 | 1 | | | | |
| 52 | 15-0049 | " .1 ufd ceramic | C13, 16<br>C20-C 49 | 32 | | | | |
| 53 | 15-0051 | " .47 ufd ceramic | C12 | 1 | | | | |
| 54 | 15-0378 | " 4.7 ufd 10v. Tantalum | C9 | 1 | | | | |

Board No. 6912 Descr. CPUCTRL

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Page 3 of 4 Rev. B1

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Reg. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 55 | 15-0220 | Capacitor, 68 ufd 15V. Tantalum | C15, 17, 19 | 3 | | | | |
| 56 | | | | | | | | |
| 57 | 15-0068 | Socket, 40 pin | 1C, 2C | 2 | | | | |
| 58 | 15-0241 | Socket, 18 pin | 3E, 4E | 2 | | | | |
| 59 | 15-0274 | Connector, 20 pin RT. header | J1 | 1 | | | | Clip Pin # 8 for polarizing key |
| 60 | 15-0330 | Card Ejector | - | 2 | | | | (Second Assembly) |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Board No. 6912 Descr. CPUCTRL

Page 4 of 4 Rev. B1

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Reg. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 1 | 15-0086/C | PC Board, 6901 | | 1 | | | | |
| 2 | 15-0028 | IC, 74LS20 | U9 | 1 | | | | do not substitute "S" |
| 3 | 15-0230 | IC, CMOS 4050 | U15 | 1 | | | | |
| 4 | 15-0315 | IC, CMOS 40174 | U16 | 1 | | | | |
| 5 | 15-0021 | IC, 74LS00 | U10 | 1 | | | | |
| 6 | 15-0022 | IC, 74LS02 | U11 | 1 | | | | |
| 7 | 15-0024 | IC, 74LS04 | U3,U4 | 2 | | | | |
| 8 | 15-0286 | IC, 7414 | U5 | 1 | | | | |
| 9 | 15-0012 | IC, 74S20 | U13, U14 | 3 | | | | do not substitute "LS" |
| 10 | 15-0016 | IC, 74LS75 | U12 | 1 | | | | |
| 11 | 15-0199 | IC, 74LS86 | U8 | 1 | | | | |
| 12 | 15-0272 | IC, 74LS366 | U1,U2 | 2 | | | | |
| 13 | 15-0003 | Transistor, 2N2222 NPN | Q2 | 1 | | | | |
| 14 | 15-0004 | Transistor, 2N3638 PNP | Q1 | 1 | | | | |
| 15 | 15-0005 | Diode, Zener 3.9 V 1N748A | CR1 | 1 | | | | |
| 16 | 15-0031 | Resistor Network, 10K DIP | U6 | 1 | | | | Beckman 899-1-R10K<br>National RA13-10KN |
| 17 | 15-0219 | Resistor, 130 ohm 1/4W | R3 | 1 | | | | |
| 18 | 15-0040 | Resistor, 470 ohm 1/4W | R2 | 1 | | | | |

ASSY No. 15-6901 Descr. M4KX12 CMOS Memory

Rev. G.

Page 1 of 2

INTERSIL INCORPORATED
10900 N. Tantau Avenue
Cupertino, California 95014

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Reg. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 19 | 15-0043 | Resistor, 1K 1/4W | R1 | 1 | | | | |
| 20 | 15-0046 | Resistor, 10K 1/4 W | R4-8 | 4 | | | | |
| 21 | 15-0049 | Capacitor, .1 ufd | C4-C48 | 45 | | | | |
| 22 | 15-0053 | Capacitor, 27 ufd 20v | C1-C3 | 3 | | | | |
| 23 | 15-0070 | Socket, 16 pin DIP | | 48 | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | SECOND ASSEMBLY | (heat & solvent sensitive parts) | | | | | |
| 28 | 15-0204 | Switch, multi-section DIP | U7 | 1 | | | | |
| 29 | 15-0330 | Card Ejector | | 1 | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | THIRD ASSEMBLY | (monitor current leakage) | | | | | |
| 33 | 15-0205 | Battery, 3.75V NiCad AA size | | 1 | | | | GE 03A1/3AA-GT3 |
| 34 | IM6508-1I | IC, CMOS RAM 1KX12 | socketed | 48 | | | | |
| | | | | | | | | |
| | | | | | | | | |

ASSY No. 15-6901 Descr. M4Kx12 CMOS Memory

Rev. G,

Page 2 of 2

INTERSIL INCORPORATED
10900 N. Tantau Avenue
Cupertino, California 95014

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Reg. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 1 | 15-0336 | Rear Panel | | 1 | | | | |
| 2 | 15-0331 | Power Supply 5v 6A | PS 1 | 1 | | | | |
| 3 | 15-0332 | " " ±12v .8A | PS 2 | 1 | | | | |
| 4 | 15-0335 | Filter, EMI Power Line 3A | FLTR 1 | 1 | | | | |
| 5 | 15-0333 | Fan, Ultra Quiet 4 1/2" | | 1 | | | | |
| 6 | 15-0334 | Fan Finger Guard | | 1 | | | | |
| 7 | 15-0347 | Fuse holder, Shock-safe | F 1 | 1 | | | | |
| 8 | 15-0155 | Switch, Rocker w/Bezel | SW 1 | 1 | | | | |
| 9 | 15-0349 | Terminal Block, 3 Term. | TB 1 | 1 | | | | |
| 10 | | | | | | | | |
| 11 | 15-0338 | Strain Relief Assembly | | 1 | | | | |
| 12 | 15-0359 | Hole Plug, 1/2" Dia. | | 1 | | | | |
| 13 | 15-0350 | Terminal, Ring Tongue #6 | | 8 | | | | |
| 14 | 15-0120 | " Locking Lug #4 | | 1 | | | | |
| 15 | 15-0360 | Screw, #4x3/8 type B Pan Hd. | | 2 | | | | |
| 16 | 15-0361 | " #4-40x 1/2 Pan Hd. | | 4 | | | | |
| 17 | 15-0159 | " #4-40x1/4 Pan Hd. | | 3 | | | | |
| 18 | 15-0363 | " #6-32x3/4 Pan Hd. | | 4 | | | | |

Board No. 15-0341 Descr. Rear Panel Assembly

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Page 1 of 2

Rev. A

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Reg. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 19 | 15-0364 | Screw, #8-32 x 3/8 Pan hd. | | 6 | | | | |
| 20 | 15-0115 | Nut, #4-40x1/4 | | 2 | | | | |
| 21 | 15-0169 | Kep Nut #4-40x1/4 | | 3 | | | | |
| 22 | 15-0240 | " " #6-32 | | 4 | | | | |
| 23 | 15-0366 | " " #8-32 | | 6 | | | | |
| 24 | 15-0353 | Connector, 2 cond Receptical | | 1 | | | | |
| 25 | 15-0357 | Contact, Socket | | 2 | | | | |
| 26 | 15-0177 | Connector, 3 cond Receptical | | 1 | | | | |
| 27 | 15-0179 | Contact, Socket | | 3 | | | | |
| 28 | 15-0379 | Connector, 4 pos. Housing | | 1 | | | | |
| 29 | 15-0078 | Contact, Socket | | 3 | | | | |
| 30 | 15-0076 | Polarizing Key | | 1 | | | | |
| 31 | - | Cable, 3 cond | | 18 in. | | | | |
| 32 | - | Wire, UL#1015 Black 18 AWG | | 2 ft. | | | | |
| 33 | - | " UL#1015 White " | | 2 ft. | | | | |
| 34 | - | " UL#1015 Green " | | 3 in. | | | | |
| 35 | - | Twisted Pair Red & Black<br>" UL#1007 18 AWG | | 6 in. | | | | |
| 36 | - | Twisted 3 wire Black, Yellow, Orange<br>" UL#1007 18 AWG | | 6 in. | | | | |

Board No. 15-0341 Descr. Rear Panel Assembly

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Page 2 of 2

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Req. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 1 | 15-0325/B | P.C. Board, Serial I/O | | 1 | | | | |
| 2 | 15-0351 | Connector, 25 cond Receptacle | | 2 | | | | |
| 3 | 15-0348 | Switch, DPDT toggle | | 1 | | | | |
| 4 | 15-0345-8"<br>15-0346-18" | Cable Assembly, 20 cond | | 1 | | | | |
| 5 | 15-0358 | Standoff, #4-40 x 7/16 | | 3 | | | | H. H. Smith<br>9230 |
| 6 | 15-0159 | Screw, #4-40x1/4 Pan hd. | | 3 | | | | |
| 7 | 15-0043 | Resistor, 1K ohm 1/4w | | 1 | | | | |
| 8 | 15-0049 | Capacitor, .1 ufd | | 1 | | | | |
| 9 | 15-0176 | Screwlock Kit | | 2 | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Board No. 15-0342 Descr. Serial I/O Connector Assembly

Page 1 of 1 Rev. B

INTERSIL INCORPORATED
10800 N. Tantau Avenue
Cupertino, California 95014

LIST OF MATERIALS

| Item | Part number | Description | Ref. Desig. | Qty/ Assy | Qty Req. | Qty Iss'd | Qty Short | Remarks |
|---|---|---|---|---|---|---|---|---|
| 1 | 15-0288 /A | P.C. Board | | 1 | | | | |
| 2 | 15-0073 | Connector, 72 pin P.C. Edge | | 11 | | | | |
| 3 | 15-0352 | " 2 cond. plug | | 1 | | | | |
| 4 | 15-0356 | Contact, Pin | | 2 | | | | |
| 5 | 15-0178 | Connector, 3 cond. plug | | 1 | | | | |
| 6 | 15-0180 | Contact, Pin | | 3 | | | | |
| 7 | 15-0038 | Resistor, 120 ohm 1/4w | | 5 | | | | |
| 8 | 15-0083 | Capacitor, .001 ufd | | 5 | | | | |
| 9 | 15-0049 | " .1 ufd | | 11 | | | | |
| 10 | | | | | | | | |
| 11 | | | | | | | | |
| 12 | 15-0329 | Card File, SAE | | 1 | | | | |
| 13 | 15-0118 | Screw, #4-40 x 1/2 RH | | 22 | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | - | Polarizing key, for P.C. edge connector | | 11 | | | | |
| | - | Twisted Pair Red & Black<br>Wire, UL#1007 18 AWG | | 6 in. | | | | |
| | - | Twisted 3 wire Black, Yellow, Orange<br>Wire, UL #1007 18 AWG | | 12in. | | | | |

Board No. 15-0340 Descr. Card File & Bus Assembly

Page 1 of 1 Rev. A

INTERSIL INCORPORATED
10800 N. Tantau Avenue
Cupertino, California 95014

LIST OF MATERIALS