$20.00

INTERCEPT PROTOTYPING SYSTEM FROM INTERSIL

Intersil
i

HARDWARE MANUAL

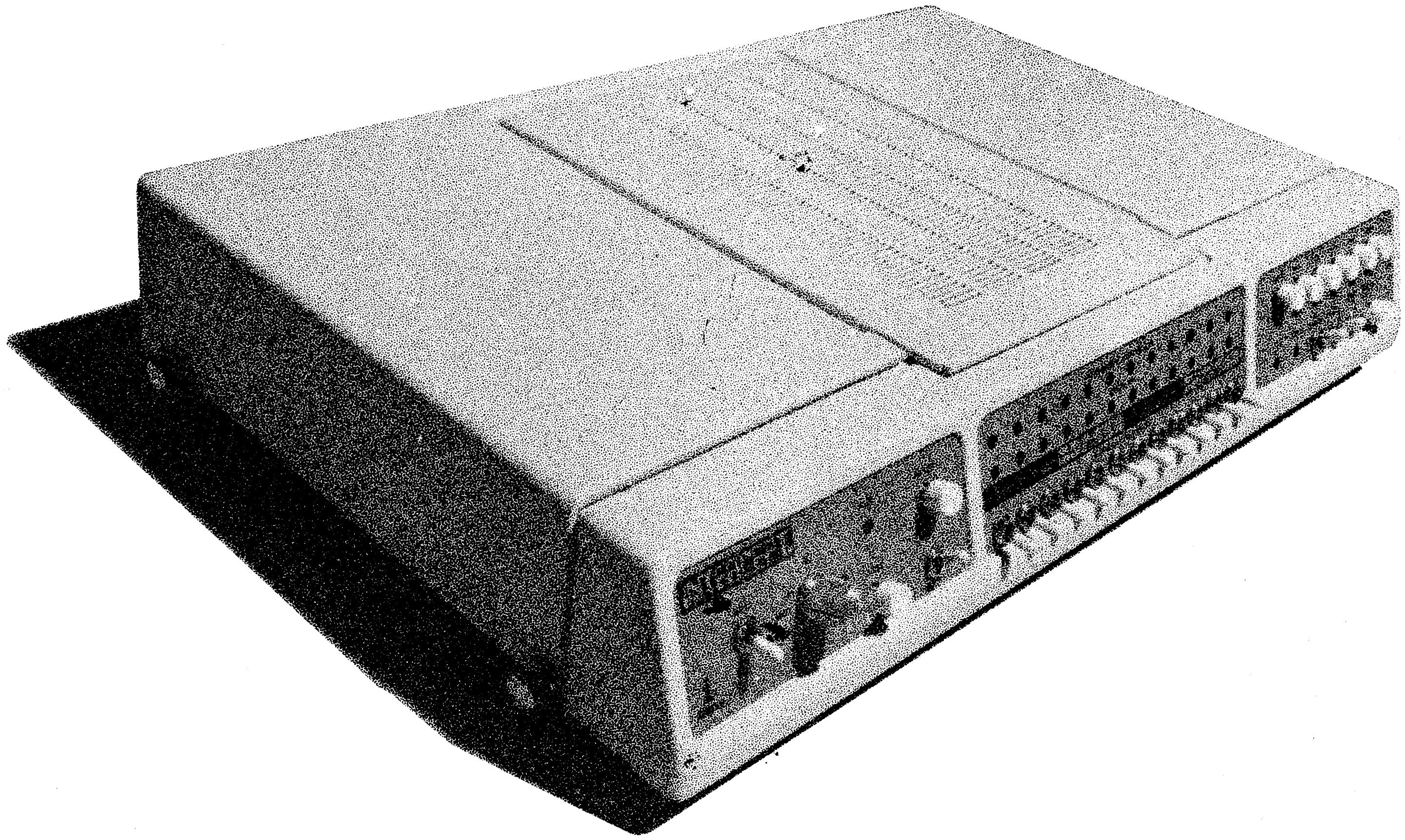


TABLE OF CONTENTS

APPENDICES

TABLES

FIGURES

CHAPTER 1
INTRODUCTION

The INTERCEPT provides the engineer with a sohpisticated design tool to develop IM6100 microprocessor based systems. The INTERCEPT is also a general purpose microcomputer, software compatible with Digital Equipment Corporation's PDP-8/E minicomputer.

This manual is organized into a series of chapters. A detailed discussion of the operation of the operator console is given in Chapter 2. Chapter 3 discusses the IM6100 Microprocessor as it is applied in the INTERCEPT System. No attempt has been made to explain the complete detailed operation of the device itself since adequate documentation is provided on the IM6100 and its family of support devices in the IM6100 Data Book. Chapter 3 also defines the INTERCEPT bus.

Chapter 4 summarizes the available software including the IFDOS operating system. Software written for PDP-8 will run properly in the INTERCEPT with few exceptions. The INTERCEPT does not provide for two PDP-8 options--the "user flag" (for time-sharing applications) and "EAE" (for hardware multiply and divide). These are the major constraints on the software compatibility between the INTERCEPT and the PDP-8. Any other incompatibility would probably result from running software on the INTERCEPT which required the faster operation speed of the PDP-8/E or attempting to use software which was written for hardware not existing in the INTERCEPT system. The PDP-8/E is approximately twice as fast as the INTERCEPT (with the IM6100 operating at 4 MHz).

The detailed hardware descriptions of the basic modules provided in the INTERCEPT system--the 4K X 12 nonvolatile memory system, the processor module with the serial interface and the operator console--are found in Chapters 5 through 7. The Extended Memory Controller option to expand the addressing capability of the INTERCEPT from 4K to 32K is discussed in Chapter 8. A summary of the other hardware options including the bus extension is given in Chapter 9.

Appendices A-K provide information on the binary format, ASCII character codes, the operator console software, Teletype modifications, ROM based subroutine calls, user interface to the INTERCEPT bus, assembly drawings, list of materials, engineering changes, the bus extension and the performance of the PDP-8 diagnostic programs on the INTERCEPT. The user is recommended to read appendices F ("User Interface on the INTERCEPT Bus") and I ("Engineering Changes") before designing any interfaces to be used with the INTERCEPT.

CHAPTER 2
INTERCEPT OPERATOR CONSOLE

The operator console (Figure 2-1) for the INTERCEPT consists of an array of console switches and indicators to facilitate computer operation and maintenance. The operator may start and stop program execution, examine and modify the contents of main memory, modify and display internal processor information, select various modes of microprocessor operation, manually load and execute short machine language programs or load and execute programs via the Teletype or a high speed reader.

Since the microprocessor register and control signals are not available externally, the modification and display of internal processor information are done by the software resident in the control panel PROMs--6903B-3C, 3D and 3E (Appendix C).

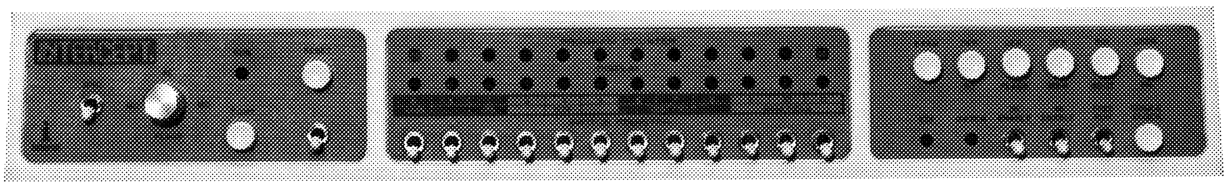


FIGURE 2-1
INTERCEPT OPERATOR CONSOLE

HARDWARE DRIVEN CONTROLS AND INDICATORS

The controls and indicators described in this section are hardware driven. They do not depend on the control panel resident software for their operation.

ON/OFF SWITCH

This switch controls the 115/230 volt AC power to the system.

RESET SWITCH

This pushbutton, when activated, grounds the RESET line of the INTERCEPT bus. On the processor module, this causes the IM6100 Accumulator (AC) and Link (L) bits to be cleared, the Program Counter (PC) to be set to 7777_8 and the processor to be halted (Chapter 3). RESET is also used for system initialization.

HLT SWITCH AND CONT PUSHBUTTON

The HLT switch, when it is down, halts the processor.

Activation of the CONT pushbutton, with the HLT switch down, causes the processor to fetch and execute the next sequential instruction, pointed to by the PC and then halt again. This mode of operation may be used to check out a program one instruction at a time.

Activation of the CONT pushbutton, with the HLT switch up, causes the processor to execute a program, starting at the location pointed to by the PC.

FREE RUN SWITCH AND SINGLE CLOCK PUSHBUTTON

This pair of switches controls the processor clock source. When the FREE RUN switch is up, the processor receives a continuous stream of clock pulses from the 4 MHz crystal oscillator on the processor module.

When the FREE RUN switch is down, the crystal oscillator is gated off, and the processor is clocked, one pulse at a time, by activating the SINGLE CLOCK pushbutton. This mode of operation is possible since the processor design is completely static. Gating is provided in the processor module to ensure integral clocking and the SINGLE CLOCK pushbutton is debounced to prevent false triggering. The single clock mode of operation is useful to 'micro examine' the operation of the processor system.

3K ENABLE AND 4K ENABLE SWITCHES

The 3K ENABLE switch, when it is down, reconfigures the 4K X 12 memory module into 1K X 12 RAM (locations 0000-1777_8) and 3K X 12 ROM (locations 2000-7777_8) to simulate a RAM-ROM system for user prototyping.

The 4K ENABLE switch, when it is down, write protects the main memory.

RUN, XTA AND IFETCH INDICATORS

These indicators continuously monitor the operation of the RUN, XTA and IFETCH lines of the INTERCEPT bus (Chapter 3). The indicators are lit when the corresponding lines are 'active'.

SOFTWARE DRIVEN CONTROLS AND INDICATORS

The controls and indicators, described in this section, depend on the control panel resident software for their operation.

30 HZ SWITCH

This switch, in the up position, activates a timer on the control module to generate control panel requests (Chapter 3) to display processor state information in 'real time'.

The timer is most useful in the single instruction mode, since after the execution of each instruction, the indicators will be 'updated'.

The timer is gated off, if the DMAGNT (Direct Memory Access Request Grant) or the INTGNT (Interrupt Request Grant) line of the bus is active. The timer service routine will adversely affect the data transfer rate of DMA devices and the operation of the priority vectoring feature of the Peripheral Interface Elements (IM6101-PIE).

The user must be careful to include the overhead of the timer service routine (approximately 200μs at 4 MHz, every 300ms - Appendix C) to the actual execution time of the user task to calculate the overall time, if timing considerations are critical.

The 30 Hz timer will also interfere with the proper execution of the ION (Interrupt ON) instruction. ION instruction (Chapter 3) defers the enabling of the processor interrupt system until the 'next' sequential instruction is executed. Assume that the timer generates a request after the ION was executed, but before the execution of the next sequential instruction from main memory. Then the interrupt system will be enabled immediately upon exiting from the timer routine, since the first instruction in the timer routine would have satisfied the one instruction 'delay'. If the INTREQ (Interrupt Request - Chapter 3) line of the bus is active the processor will not execute the instruction following ION before granting the interrupt. This may adversely affect the program structure.

It is recommended that the timer be off, in the single clock mode, since the timer routine will 'steal' the manually produced clock pulses.

The user must not activate the function switches while the processor is running with the 30 Hz. The control panel service routine scans to see if any of the function switches are active before entering the timer routine. The timer routine is executed as a 'default' option if none of the function switches are active (Appendix C).

PROGRAM COUNTER INDICATORS

The PROGRAM COUNTER indicators display the contents of the PC, if the 30 Hz timer is on. Bit 0 is the most significant bit, and bit 11, the least significant.

If the timer is off, the operator must press the EXAM pushbutton with the rotary switch set to AC, MQ or FLAGs to display the current PC.

When in the single-clock mode, the indicators display the address of the current instruction. This function is accomplished by hardware logic on the panel module and does not utilize the panel software.

ROTARY SWITCH AND DISPLAY INDICATORS

In the 30 Hz mode, the DISPLAY indicators show the contents of main memory, AC, MQ or FLAGs, depending on the position of the rotary switch.

If 30 Hz is off, the operator must press the EXAM pushbutton with the rotary switch set to the appropriate position, for the required information to be displayed.

In the single clock mode, the DISPLAY follows the DX bus and, therefore, constantly displays the states of those lines. This function is accomplished in hardware.

FLAGs indicate the status bits of various internal and external control flip-flops and their bit positions are assigned as follows:

0 : LINK

1 : Not used

2 : IRB - Interrupt Request Bus
DISPLAY bit (2) is lit if a request is active.

3 : IIB - Interrupt Inhibit Bus
DISPLAY bit (3) is lit, if the interrupt requests are disabled by the Extended Memory Control module (Chapter 8)

4 : IEFF - Interrupt Enable Flip-Flop
DISPLAY bit (4) is lit, if the CPU interrupt system is enabled

5 : Not used

6 - 8 : The currently selected Instruction Field

9 - 11 : The currently selected Data Field

Bits 3 and 6-11 must be ignored if the Extended Memory Controller is not in the system.

SWITCH REGISTER SWITCHES

The Switch Register may be read into the AC by the user program with the OSR (OR the SWITCH REGISTER) instruction (Chapter 3).

These switches are also used in conjunction with the DEP PC, DEP FLAGs, DEP MEM, BIN BOOT and USER FN pushbuttons.

FUNCTION SWITCHES

The function switches are active only if the processor is halted. They are scanned left to right, EXAM, DEP PC, DEP FLAGs, DEP MEM, BIN BOOT and USER FN, in that order, by the panel routine (Appendix C).

EXAM PUSHBUTTON

If the rotary switch is set to MD, the contents of the memory location pointed to by the Program Counter indicators are read into the Display indicators, by activating the EXAM pushbutton. The PC is then incremented by one after the information is displayed to point to the next sequential memory location. Therefore, to modify the data in an examined location, one must restore the correct address.

If the rotary switch is set to AC, MQ or FLAGs, the display will show the corresponding information when EXAM button is activated. The PC is not affected.

Note that in the memory data mode, the PC will contain the address, one greater than the address of the location whose contents are currently being displayed. In the INTERCEPT, this correspondence always exists. For example, if one is stepping through a program one instruction at a time, the PC will show the address of the next instruction to be executed and the display will show the last instruction executed except if the last instruction executed was a branch instruction. In that case, the display will show the contents of the memory location immediately preceeding the location pointed to by the PC, which may be of little interest.

DEP PC PUSHBUTTON

The DEP PC pushbutton is used to load the PC with a 12-bit address specified by the switch register. The information is displayed in the Program Counter indicators. If the rotary switch is set to MD, the display will show the contents of the memory location immediately preceeding the location pointed to by the PC.

DEP FLAGS PUSHBUTTON

This pushbutton is used to load switch register bits 6-11 into the Instruction Field (IF) 0-2, and Data Field (DF) 0-2, if the Extended Memory Controller is used.

Switch register bit 0 is loaded into the LINK.

DEP MEM PUSHBUTTON

If the operator wants to deposit data into a particular location of a specified memory field, the Instruction Field must be loaded with the memory field address (if Extended Memory Controller is used), the address must be loaded into the PC and then the data is deposited by appropriately setting the switch register and then activating DEP MEM. The PC is automatically incremented by one to set up the next sequential address. If the rotary switch was set to MD, the data that was just deposited, is shown in the Display indicators.

BIN BOOT PUSHBUTTON

The BIN BOOT button activates the bootstrap loader to read and store information contained in binary (Appendix A) coded paper tapes, using ASR-33 Teletype or a high speed reader. Refer to to the section on binary tape loading procedure.

USER FN PUSHBUTTON

This button is activated to implement user defined routines. If the user function is not implemented in the panel software, this button will decrement the PC by 1 every time it is activated (Appendix C).

POWER ON/OFF PROCEDURE FOR RETENTION OF MEMORY DATA

When power is turned on or off, the states of the microprocessor cannot be guaranteed. It may 'write' into main memory during these transition periods. It is recommended that the user follow the sequence shown below for power on/off to ensure that memory data is not disturbed:

POWER OFF:

1. 30 Hz switch down (off)
2. FREE RUN switch down (single clock)
3. SINGLE CLOCK the processor until XTA indicator is lit
4. 4K ENABLE switch down (write protect)
5. Power OFF

POWER ON:

1. 4K ENABLE switch down (write protect)
2. FREE RUN switch down (single clock)
3. 30 Hz switch down (off)
4. Power ON
5. SINGLE CLOCK until XTA indicator is lit
6. FREE RUN switch up
7. RESET
8. 30 Hz switch up
9. 4K ENABLE switch up

BINARY TAPE LOADING PROCEDURE

1. Halt the processor.
2. Place the binary tape to be loaded in the Teletype or high speed reader. The 'leader' portion of the tape must be over the read head and the reader must be 'on-line'.
3. Load the address of the field into which the program is to be loaded in the Instruction Field, if the Extended Memory Controller is used.
4. When using the Teletype, the switch register must be set to 4000_8 and for the high speed reader to 0000_8. Ensure that 4K ENABLE and 3K ENABLE switches are up.
5. Press BIN BOOT

   If the switch register option was not selected properly, the program will get 'hung up' in a loop. RESET the machine and go to step 4.

As the tape is being read in, the PC will show the address of the next location to be loaded, and the DISPLAY, independent of the rotary switch setting, will indicate the data that was just loaded into memory.

6. The processor halts at the trailer.

7. The PC will show the next location to be loaded.

   If the rotary switch was set to MD, the DISPLAY will show the last data read in.

   If the rotary switch was set to AC and the 30 Hz switch was up, the DISPLAY should be 0000_8, indicating that there was no checksum error when the tape was read.

   If the 30 Hz switch was off, press EXAM with the rotary switch set to AC, to display the contents of AC, which should be 0000_8.

STARTING A PROGRAM

1. Load the program into the memory by using the switch register and function switches or by reading in a binary tape with the Teletype or a high speed reader.

2. RESET to initialize the system, if required.

3. Put 30 Hz switch up, if 'real time' display is desired.

4. Load the address of the memory field in which the program resides into the Instruction Field and address of the memory field the program uses for data into the Data Field, if Extended Memory Controller is used.

5. Load the starting address of the program into the PC.

6. Ensure that HLT is up (RUN mode).

7. Set up the switch register, as required, if the program uses switch register settings for options.

8. Press CONT.

CHAPTER 3

INTERCEPT PROCESSOR ARCHITECTURE

SOFTWARE CONSIDERATIONS

Programming the INTERCEPT is identical to programming the PDP-8/E. This section provides an overview of INTERCEPT software considerations. For a very detailed discussion of PDP-8/E (and INTERCEPT) programming, the user is referred to DEC's "Introduction to Programming" and for details on the internal processor architecture, the user should refer to the IM6100 data book.

MEMORY ORGANIZATION

Like the PDP-8, the IM6100 has a basic addressing capacity of 4096 (4K) 12-bit words. This addressing capacity is a natural result of the 12-bit word length, and can be expanded to 32K (Chapter 8).

The memory system is organized into 4096-word blocks called "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 denote the page number 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 that may be present in the system, the 6907 Extended Memory module 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 EMC module. 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 on the EMC module.

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".

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 T-states	second T-states	third T-states	fourth 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 addresses			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	6_8		6 (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 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 one state higher than this address.
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 instructions are currently being fetched and bits 5-11 of the MAR identify the location within the Current Page (PAGE ZERO (0), by definition, denotes the first 128 words of memory, 0000_8 - 0177_8).

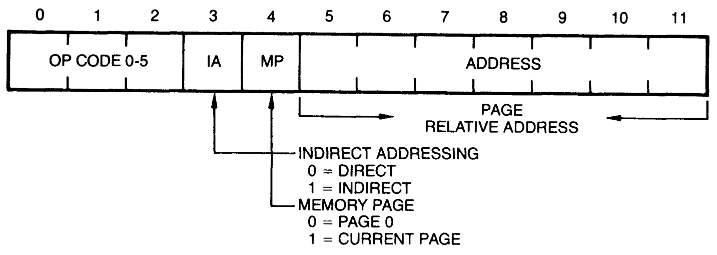


FIGURE 3-1

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. If bit 4 is a 1, the page address specified is interpreted to be on the Current Page.

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 shows 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

By this method, 256 locations may be directly addressed, 128 on PAGE 0 and 128 on the CURRENT PAGE. Other locations are addressed by utilizing bit 3. When bit 3 is a 0, the operand address is a DIRECT ADDRESS. An INDIRECT ADDRESS (pointer address) identifies the location that contains the desired address (effective address). To address a location that is not directly addressable, not in PAGE 0 or in the CURRENT PAGE, the absolute address of the desired location is stored in one of the 256 directly addressable locations (pointer address). Upon execution, the MRI will operate on the contents of the location identified by the address contained in the pointer location.

It should be noted that locations 0010_8 - 0017_8 in PAGE 0 are AUTOINDEXED. If these locations are addressed indirectly, the contents are incremented by 1 and restored before they are used as the operand address. These locations may, therefore, be used for indexing applications.

IOT INSTRUCTIONS

The IOT instructions all have an Opcode of 6_8 and are used to initiate the operation of peripheral devices, and to transfer data between peripherals and the processor. Using IOT's all device data movements are programmed data transfers; device data can also be moved to/from memory and the processor by means of interrupt initiated transfers, or by direct memory access. Programmed data transfers are the simplest way to move data to/from peripheral devices, but are also the slowest.

If 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 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 will be executed before the first interrupt is recognized.)
IOF	6002	INTERRUPTS OFF. Immediately resets the processor Interrupt Enable flip-flop, so no more interrupts will be allowed.
SRQ	6003	SKIP IF INT REQUEST. If the INTREQ line is asserted low, skip the next instruction.
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 on EMC module (1 if IIFF set) AC (4) : CPU Interrupt Enable flip-flop (1 if IEFF set) AC (6-11) : Save Field register on EMC module
RTF	6005	RETURN FLAGS. Link is set by AC (0). Interrupt Inhibit flip-flop on EMC module 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 EMC module 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 by INTERCEPT.
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 to clear the devices flags and set their interrupt enabling flip-flops.

Except for processor IOT's all IOT instructions are non-specific in that, unlike all the other instructions, the operation that they perform is not "known" by the processor. Rather, the hardware designer specifies what each of these instructions does by the logic he builds into the interface for the specific peripheral device. The IOT instructions work in conjunction with the C0, C1, C2 and SKP lines to the processor. For example: for a PDP-8 compatible Teletype interface, it is necessary that instruction 6034_8 cause the TTY keyboard data to be OR'ed into the Accumulator. Referring to Table 3-4, it is seen that in order 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. The interface logic, then, 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. He first decides what he wants a given IOT to do, then builds the necessary "interpretive" logic into his peripheral interface. If the Peripheral Interface Element (IM6101 - PIE) is used for interfacing, all control codes are preassigned.

TABLE 3-4

CONTROL LINES (C0, C1, C2) OPERATION

CONTROL LINES			OPERATION
C0	C1	C2	
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'ed 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 is loaded into Program Counter (absolute jump).

* don't care

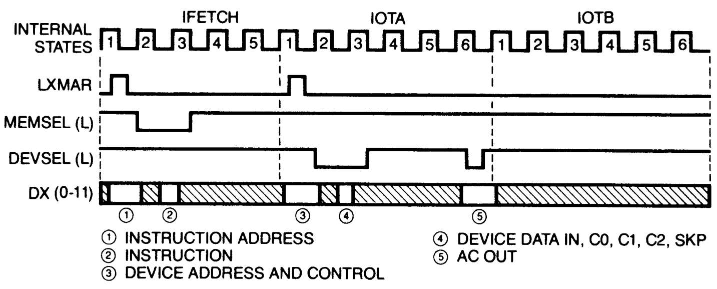


FIGURE 3-2

INPUT-OUTPUT INSTRUCTION TIMING

OPI INSTRUCTIONS

The third category of instructions are termed the Operate Instructions, all of which 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 microinstructions, since by setting or not setting given bits in the instruction word, they can be combined with one another. This cuts down 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 state of the front panel Switch Register into the processor's Accumulator. The timing diagram for this instruction is shown in Figure 3-3.

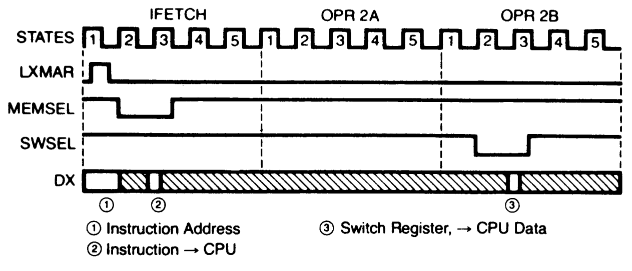


FIGURE 3-3

OSR INSTRUCTION TIMING

BASIC MACHINE TIMING

The timing for the most fundamental processor lines is illustrated in Figures 3-4 and 3-5. 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 waveform, 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.

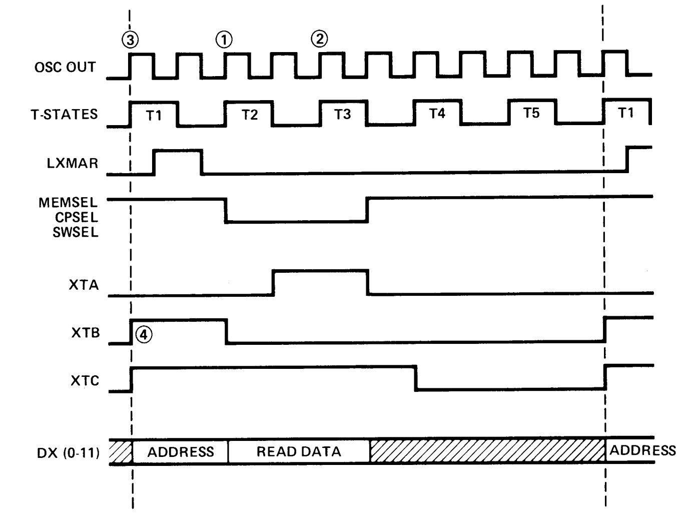
All machine cycles are composed of either five or six T-states (often referred to simply as "states"). Cycles which do not involved a "write" operation consist of five states (Figure 3-4). When a "write" is called for, the cycle is extended to six states (Figure 3-5).

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 five or six states each. Thus, a complete fetch and execution can consist of 10, 11, 15, 16, 17, 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 which looks like that shown in Figure 3-4. 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) states. Some instructions require one, and some two, execute cycles.

The six state cycle shown in Figure 3-5 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



NOTES: 1. ALL REQUEST LINES (RESET, CPREQ, DMAREQ, INTREQ AND RUN/HLT FLIP-FLOP) ARE SAMPLED IF THIS IS THE LAST CYCLE OF AN INSTRUCTION EXECUTION.

2. WAIT LINE IS SAMPLED AND IF IT IS ACTIVE LOW, THE $\overline{T_2}$ STATE IS EXTENDED.

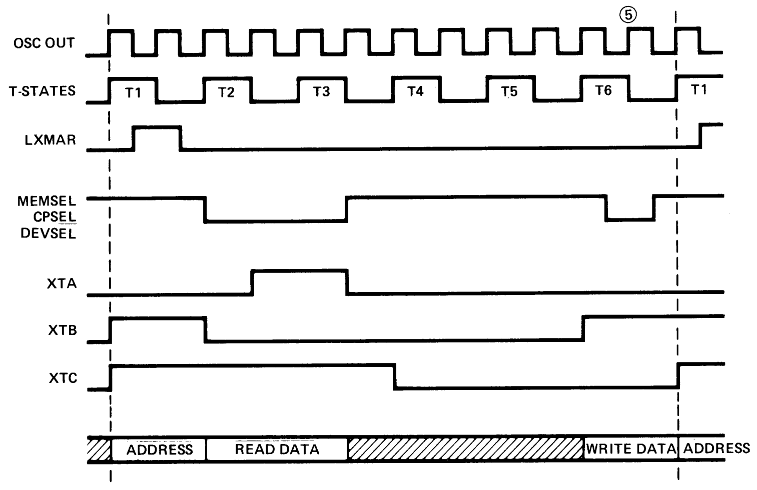
READ DATA IS SAMPLED.

3. THE STATUS SIGNALS (IFETCH, DMAGNT, INTGNT, RUN, DATAF AND LINK) BECOME VALID.

4. XTB REMAINS HIGH IF THE PREVIOUS CYCLE WAS A 6-STATE CYCLE.

FIGURE 3-4

BASIC 5-STATE CYCLE TIMING



NOTE: 5. WAIT LINE IS SAMPLED AND IF IT IS ACTIVE LOW, THE T6 STATE IS EXTENDED.

FIGURE 3-5

BASIC 6-STATE CYCLE TIMING

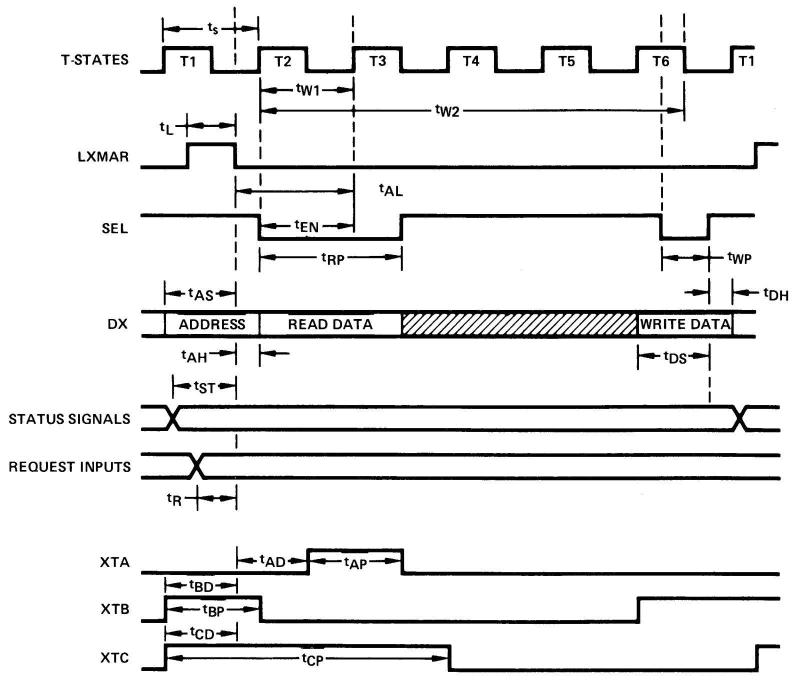
on the DX lines throughout state T6. A SELect line (MEMSEL, DEVSEL or CPSEL) is then asserted by the processor to actuate the "write" operation. The data is typically strobed into the memory or perihperal device on the trailing edge of the SELect pulse. When the SELect line is asserted by the processor, 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 3-4 and 3-5 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), he must be aware of the timing differentials that appear between the various processor signals. For this purpose he should consult the data sheet on the IM6100 device. Figure 3-6 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.

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.) INTGNT and DATAF experience similar delays before coming "true" at the beginning of the cycle.

INTERNAL PRIORITY STRUCTURE

As indicated in Figures 3-4 and 3-5, the IM6100 samples the RESET line, the request lines CPREQ, DMAREQ and INTREQ, and the state of its internal RUN/HALT flip-flop on the rising edge of T2 during the last execute cycle of each instruction, to determine what it should do next. If any of the request lines is asserted, or if the RUN/HALT flip-flop has gone to the HALT state, or if the RESET line is asserted, the processor will perform the requested operation according to the priority heirarchy listed below. If none of these actions is being requested, the processor will fetch and execute the next sequential instruction, and again sample the request lines, etc.



PARAMETER		INTERCEPT V_{CC} = 5.0V T_A = 45°C
LXMAR PULSE WIDTH	t_L	240
ADDRESS SET-UP TIME	t_{AS}	50
ADDRESS HOLD TIME	t_{AH}	150
ACCESS TIME FROM LXMAR	t_{AL}	350
OUTPUT ENABLE TIME	t_{EN}	150
READ PULSE WIDTH	t_{RP}	700
WRITE PULSE WIDTH	t_{WP}	200
DATA SET-UP TIME	t_{DS}	150
DATA HOLD TIME	t_{DH}	150
STATUS SIGNALS VALID TIME	t_{ST}	200
REQUEST INPUT SET-UP TIME	t_R	50
READ WAIT SAMPLE TIME	t_{W1}	150
WRITE WAIT SAMPLE TIME	t_{W2}	1900
XTA DELAY	t_{AD}	300
XTA PULSE WIDTH	t_{AP}	500
XTB DELAY	t_{BD}	400
XTB PULSE WIDTH	t_{BP}	500
XTC DELAY	t_{CD}	410
XTC PULSE WIDTH	t_{CP}	1500

FIGURE 3-6

INTERCEPT BUS TIMING

The priority heirarchy 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.

CPREQ - If the RESET line is not found to be asserted, but the CPREQ line is, the processor grants the control panel interrupt request at the end of the current cycle.

RUN/HALT - If neither of the foregoing lines is asserted, but the processor finds it internal RUN/HALT flip-flop in the HALT state, it enters the HALT state at the end of the last execute cycle. While halted the processor continues to generate the timing signals XTA, XTB and XTC. The RUN/HALT transition must occur at least 10 clock pulses after the RESET line has been released for it to be recognized.

DMAREQ - If none of the aforementioned 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, HALT 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 cycle.

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. For example, when the processor is executing a device interrupt routine, or is waiting for a DMA action to be completed, control panel interrupt requests are inhibited by gates on the 6903 control panel module. 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 a new data character. Using the interrupt system, the processor is free to go about execution of the "background" program until the external device indicates it is ready for a transfer by requesting an interrupt. This avoids the need to put 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, HALT command, or direct memory access request) is active when the computer finishes executing the current instruction, the machine will grant 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 INTDIS line on the bus is high.)

The timing diagram for an interrupt request/grant is shown in Figure 3-7. In the first cycle after an interrupt is granted, the processor stores the current state of the Program Counter in memory location 0000_8. (This location holds the "return address" the computer needs so that it can return to where it left the "background" program at the end of the interrupt service routine.) Then, in the next cycle, the machine fetches the first instruction in the interrupt routine from location 0001_8.

The positive going edge (effectively) of the INTGNT line is used by the EMC module to reset the extended address bits, EMA0 - EMA2, on the bus to the low state. This causes the initial instruction(s) in the interrupt routine to be drawn from memory field 0. The INTGNT line is reset to the low state midway through execution of the first IOT instruction after the interrupt. The resetting of the INTGNT line by this IOT is necessary, since the Instruction Field register on the EMC module is held in the cleared state as long as INTGNT is high. See Figure 3-8, the timing diagram for the resetting of the INTGNT line. The IM6101 PIE uses the INTGNT signal to implement a hardwired priority vectoring scheme.

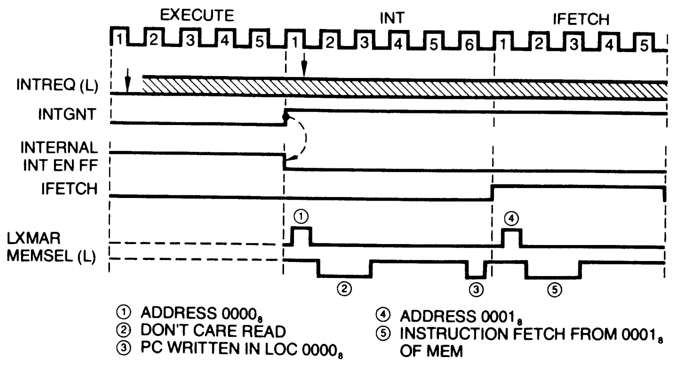


FIGURE 3-7

DEVICE INTERRUPT GRANT TIMING

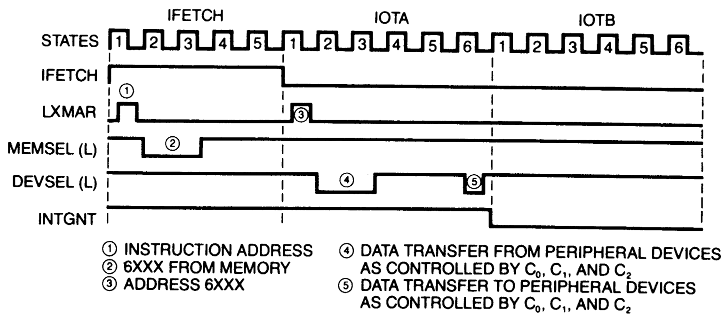

IFETCH
IOTA
IOTB
STATES
1 2 3 4 5 1 2 3 4 5 6 1 2 3 4 5 6
IFETCH
LXMAR
MEMSEL (L)
DEVSEL (L)
INTGNT
① INSTRUCTION ADDRESS
② 6XXX FROM MEMORY
③ ADDRESS 6XXX
④ DATA TRANSFER FROM PERIPHERAL DEVICES AS CONTROLLED BY C_0, C_1, AND C_2
⑤ DATA TRANSFER TO PERIPHERAL DEVICES AS CONTROLLED BY C_0, C_1, AND C_2

FIGURE 3-8

DEVICE INTERRUPT GRANT RESET 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 allowing another interrupt to be recognized. Typically, these chores involve executing a skip chain to find which device requested the interrupt, and storing the Accumulator, Link and Program Counter for restoration later. A timing diagram for setting the IEFF with an ION instruction is shown in Figure 3-9. Note that the IEFF is not actually set until the processor has fetched the next instruction after the ION; this guarantees one more instruction will be executed after an ION before the next interrupt can be recognized.

The above discussion applies to operation of the INTERCEPT with normal PDP-8 "style" software prioritized interrupts. However, the IM6101 Peripheral Interface Element (PIE) device provides for a hardware prioritized interrupt system. Operation of interrupts using this element is discussed in the IM6101 PIE data sheet.

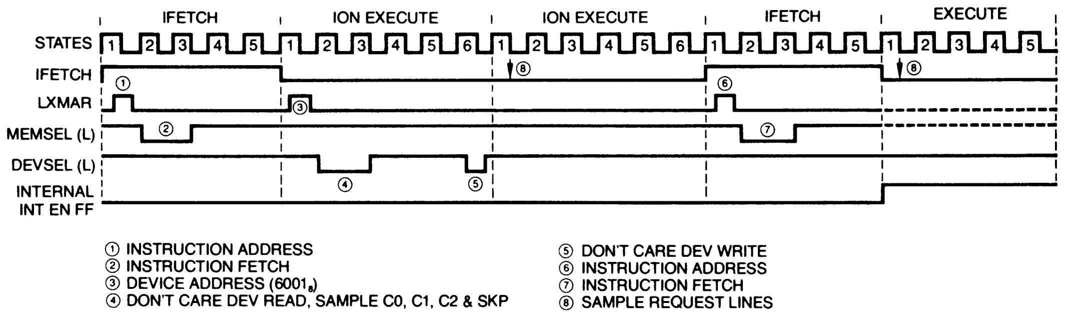


FIGURE 3-9

ION INSTRUCTION EXECUTION

CONTROL PANEL INTERRUPTS

Due to the limited number of pins available on a practical semiconductor package, the IM6100, like every microprocessor, does not provide continuous real-time access to many of its internal registers. The state of the Accumulator, Link, Program Counter, MQ register, etc., are multiplexed at various times on the processor DX lines. To find the state of one of these registers, it is necessary to temporarily suspend mainline program execution, and execute a special control panel interrupt routine to bring the required data out of the IM6100 device and latch it into external registers or indicators. Thus, it is said that the control panel is "implemented in software".

Detailed operation of the 6903 control panel logic is discussed in Chapters 2 and 7.

A control panel interrupt is requested by asserting the CPREQ line on the bus. The processor will grant the control panel interrupt request at the end of the current instruction, if a RESET action is not simultaneously being requested. A control panel interrupt request will be granted even if the machine is in the HALT state; the processor will be forced into the RUN state for the duration of the control panel interrupt routine, and then return to the HALT state.

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 is used for a special purpose, as described below.)

A control panel interrupt is granted according to the timing shown in Figure 3-10. The CPREQ line is sampled by the processor at the indicated time in the last execute cycle of every instruction (or, if the machine is halted, the line is sampled every cycle). If the line is found asserted, the processor grants the control panel request by setting its internal Control Panel Flip-Flop (CNTRL F/F). While this flip-flop is set, further control panel interrupts are ignored, as are DMA requests and device interrupt requests. Changes to the device interrupt system by the instructions ION and IOF are also inhibited while the CNTRL F/F is set.

During the CPINT cycle the state of 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 mainline program at the end of the control panel service routine. The Program Counter is then jam 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 the control panel interrupt, while the CNTRL F/F is set, the control panel memory rather than main memory, is selected for all memory reference operations. This is accomplished by the processor asserting the CPSEL line rather than the MEMSEL line. However, the control panel routine does have access to the main memory, through indirectly addressed AND, TAD, ISZ and DCA instructions. When either the TAD or DCA instruction is executed in the indirect address mode, the instruction fetch cycle and the indirect cycle empoly the CPSEL line to fetch the instruction and pointer from control panel memory, then use the MEMSEL line during the execute cycle to deposit the data into the selected main memory location with DCA, or fetch data from the main memory location with TAD. Indirectly addressed ISZ and AND instructions operate in an exactly analagous manner. (Note that, since these instructions are indirectly addressed, the DATAF line will be asserted during the execute cycle by the CPU. Thus, when the EMC module is present in the system, the control panel routine has access only to the Data Field currently specified by the Memory Extender.) Thus the control panel routine can fetch the data in any main memory location using TAD I, deposit data into any main memory location using DCA I, increment and skip on any main memory location using ISZ I, or AND the content of any main memory location into the Accumulator using AND I.

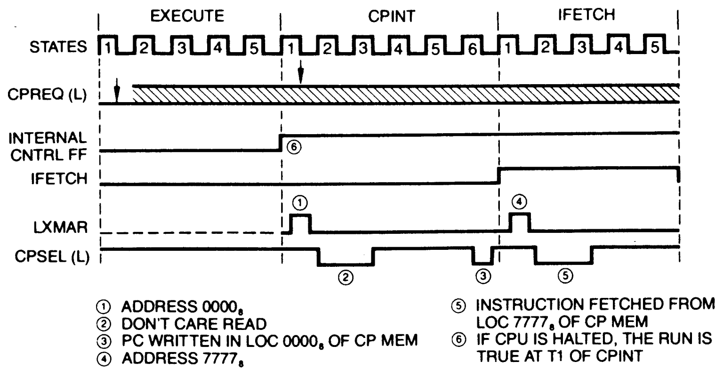
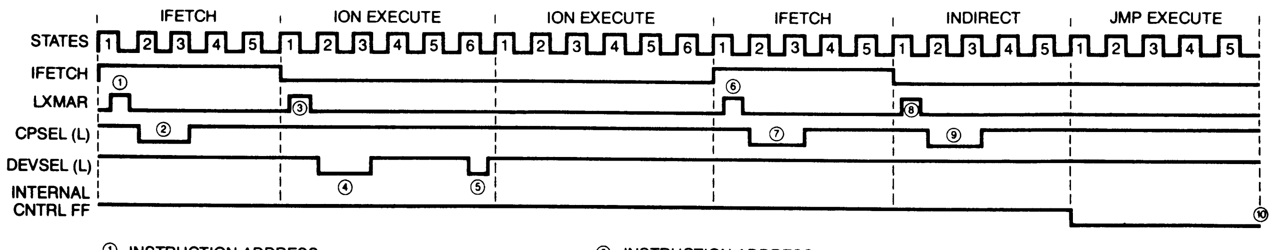

EXECUTE
CPINT
IFETCH
STATES
CPREQ (L)
INTERNAL CNTRL FF
IFETCH
LXMAR
CPSEL (L)
① ADDRESS 0000$_8$
② DON'T CARE READ
③ PC WRITTEN IN LOC 0000$_8$ OF CP MEM
④ ADDRESS 7777$_8$
⑤ INSTRUCTION FETCHED FROM LOC 7777$_8$ OF CP MEM
⑥ IF CPU IS HALTED, THE RUN IS TRUE AT T1 OF CPINT

FIGURE 3-10

CONTROL PANEL INTERRUPT GRANT TIMING

The end of the control panel routine is marked by the execution of an ION instruction, followed immediately by an indirect JMP through control panel memory location 0000$_8$. The ION instruction has no effect on the device interrupt system, since the CNTRL F/F is still set when it is executed. However, the ION causes the CNTRL F/F to be reset midway through execution of the next instruction. This next instruction is the aforementioned JMP I, which sets the processor Program Counter to the address contained in control panel memory location 0000$_8$. Normally this will be the stored return address where the processor left the mainline program to execute the control panel interrupt. However, if the control panel routine modified the content of control panel memory location 0000$_8$, the Program Counter will be set to a new starting address upon emergence from the routine. See Figure 3-11.

A forced exit from the control panel routine can be achieved by asserting the RESET line on the bus (e.g., by activating the RESET switch on the front panel). RESET can accomplish this forced exit, since it has a higher priority than CPREQ, and can override the control panel routine. Execution of an RTF (6005$_8$) instruction will reset the CNTRL F/F, and thus can also effectively end the control panel routine.



① INSTRUCTION ADDRESS
② INSTRUCTION FETCH FROM CP MEM
③ DEVICE ADDRESS (6001$_8$)
④ DON'T CARE DEVICE READ, SAMPLE C0, C1, C2 & SKP
⑤ DON'T CARE DEVICE WRITE
⑥ INSTRUCTION ADDRESS
⑦ INSTRUCTION FETCH FROM CP MEM
⑧ EFFECTIVE ADDRESS (0000$_8$)
⑨ JMP ADDRESS FROM CP MEM LOC 0000$_8$
⑩ IF CPU WAS IN THE HALT STATE, THE RUN IS FLASE AT T1

FIGURE 3-11

"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 action, control panel interrupt request, or CPU RUN/HALT flip-flop in the HALT state), 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 and memory extender modules: all DX lines, MEMSEL, XTC, LXMAR and DATAF. Also, the EMC module 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 action, the processor continues to sample the DMAREQ line on the rising edge of T2. When it finds this line to have been released by the DMA port, the processor will resume execution of the mainline program in the next cycle. Figure 3-12 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 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 memory extension is identical to that of the 6907-EMC module (Chapter 8). The IM6102 also has an on-chip crystal controlled programmable real time clock. For additional details, please refer to the data sheet on the IM6102 MEDIC.

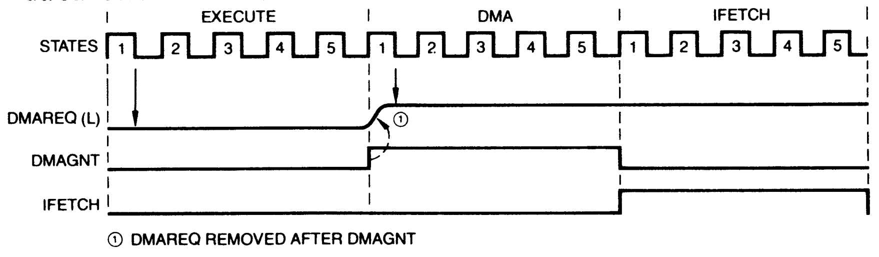


FIGURE 3-12

DMA CYCLE TIMING

DMA transfers actually entail some fairly complex cooperation between program software and DMA port logic. In conclusion 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 will simply cause 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 front panel causes the processor Accumulator and Link to be cleared, the Program Counter to be set to 7777_8, and the machine to go to the HALT state. When in the HALT state, however, the processor continues to cycle and produce the timing signals XTA, XTB and XTC. All SELect lines remain high.

The RESET switch asserts the RESET line bus low. This causes all flags in device interfaces to be cleared in addition to affecting the processor in the manner described above. Executing a CAF (6007_8) instruction will also assert the RESET line low and cause all device interface flags to be cleared. However, the machine is not forced to the HALT state by a CAF instruction, since the RESET line is not sampled by the processor until after the assertions of the RESET line by the CAF have been concluded.

CPU RUN/HALT FLIP-FLOP

The IM6100 microprocessor contains an internal RUN/HALT flip-flop. When this flip-flop is in the RUN state, the RUN line on the bus is asserted low and the RUN lamp on the front panel is lighted. The state of the RUN/HALT Flip-Flop is changed by pulsing the RUN/HALT line on the bus. This is accomplished by the CONT pushbutton on the front panel. See the detailed discussion of the actions of this switch in Chapter 7. The RUN/HALT Flip-Flop is toggled by the rising edge of a pulse on the RUN/HALT line.

INTERCEPT BUS STRUCTURE

The INTERCEPT contains 72 lines. All lines are parallel connected to every edge connector. Table 3-5 gives a summary of the signals on the bus. Pins 1 and 2 of 6900-INTERCEPT BUS and 6904-INTBUS are daisychained to implement the priority network for IM6101-PIE devices.

Some of the bus lines are heavily used in the system, and therefore are driven by high current drivers. Other lines which are used by only a few modules are driven by LS gates. A summary of typical driver capabilities is given in Table 3-6. Each plug in module usually loads each line with one standard LS load, therefore the drivers used in the system will be found to be adequate for a fully loaded bus.

The letter H or L following each signal names in Table 3-5 denotes whether the assertion level for that signal is normally a TTL logic high or low. H refers to a logic high (2.4-5.0 volts). L refers to a logic low (0.0-0.8 volts).

SUMMARY OF BUS SIGNAL FUNCTIONS

C0 L C1 L C2 L	During the execute phase of an IOT instruction these lines are asserted by the device interface logic to define the exact operation that will take place. Sources: 6902-CPUTTY Drivers: 74LS505 6907-EMC 7407
CPREQ L	This line is asserted low by the 6903 control panel logic to request a control panel interrupt. Source : 6903-CONTRL Driver : 74LS03
CPSEL L	The processor asserts this line low to read data from, or write data into, the control panel memory, switches or indicators during a control panel interrupt. Source : 6902-CPUTTY Driver : 74365
DATAF H	This line is asserted high when the processor wishes to read from, or write into, the data field. The distinction between instruction field and data field is defined by the EMC module and therefore is only meaningful in a system with more than 4K words of memory. Source : 6902-CPUTTY Driver : 74365
DEVSEL L	During an IOT execute cycle this line is asserted low by the processor to read from, or write into, the peripheral device interface. Source : 6902-CPUTTY Driver : 74365
DMAEN L	DMA transfer enable to the IM6102-MEDIC.
DMAGNT H	When the processor grants a DMA cycle request, it asserts this line high. Source : 6902-CPUTTY Driver : 74LS04
DMAREQ L	This line is asserted low by peripheral devices for DMA transfers.

DXO-DX11	These are the 12 multiplexed bi-directional lines that carry address, instruction and data between the processor, memory and device interfaces (and front panel). DXO is the most significant bit, and DX11 the least significant bit. Sources: 6901-M4KX12 Drivers: 8833 6902-CPUTTY 74365 6903-CONTRL 74366
FIELD SELECT EMAO H EMA1 H EMA2 H	When the system employs more than 4K words of memory, these lines are driven by the EMC module to develop the required 3-bit extension to the processor's Program Counter and Memory Address Register. Source : 6907-EMC Driver : 74LS257
FREERUN H	This line is controlled by the front panel FREE RUN switch. When this line is in the "up" position, the FREE RUN line is high, and the processor is driven by the crystal controlled clock oscillator. When the switch is in the "down" position, FREE RUN is low and the processor is driven by the SNGL CLK. Source : 6903-CONTRL Driver : SW
IFETCH H	This line is asserted high by the processor throughout an instruction fetch cycle. Source : 6902-CPUTTY Driver : 74365
INTREQ L	A peripheral device interface requests an interrupt by asserting this line low. Source : 6902-CPUTTY Driver : 74LS03
INTGNT H	When the CPU grants a device interrupt request, it asserts this line high. Source : 6902-CPUTTY Driver : 74365
INTDIS L	The EMC module prevents peripheral devices from being granted an interrupt by asserting this line low. Source : 6907-EMC Driver : 7407
LINK L	This line is asserted low whenever the processor Link flip-flop is set. Source : 6902-CPUTTY Driver : 74365
LXMAR H	This line is asserted high by the processor early in every cycle. The falling edge of this pulse is used by memory and device interfaces to latch addresses and instructions into each module. Source : 6902-CPUTTY Driver : 74365

MEM DIS L	Used to disable reads and writes from/to main memory
MEMSEL L	This line is asserted low by the CPU to read from, or write into, main memory. Source : 6902-CPUTTY Driver : 74365
PRIN H PROUT H	Daisy chained priority signals for the IM6101 PIE.
RESET L	When this line is asserted low by the front panel RESET switch, it forces the processor to the RESET state (i.e., halted, with Program Coutner set to 7777_8) Sources: 6902-CPUTTY Drivers: NPN 6903-CONTRL SW
RUN L	The processor asserts this line low when it is in the RUN state; when halted the line is high. Source : 6902-CPUTTY Driver : 74LS04
RUN/HALT L	This line is pulsed low by the control panel to invert the run/halt state of the processor. Source : 6903-CONTRL Driver : 74LS05
SKIP L	A device interface causes the Program Counter to be incremented by asserting this line low during an IOT instruction execution. Source : 6902-CPUTTY Driver : 74LS03
SNGL CLK H	In the SNGL CLK mode of operation, this line is asserted high by the SNGL CLK pushbutton to advance the processor by one-half T-state. Source : 6903-CONTRL Driver : 74LS00
SWSEL L	This line is asserted low by the processor during the execute phase of an OSR instruction, to read the front panel Switch Register. Source : 6902-CPUTTY Driver : 74365
XTA H XTB H XTC H	The three lines are asserted by the processor during various parts of each cycle. The respective states of the three lines indicate the instantaneous state of the processor. Source : 6902-CPUTTY Driver : XTA, XTB 74LS04 XTC 74365
UP L	DMA transfer signal generated by IM6102 MEDIC.
V1 V2	Power supply bussing for user expansion. Planned assignment: $V_1 = -12V$, $V_2 = +12V$.
WAIT L	This line is asserted low by a peripheral device interface or memory module to cause the processor to pause for an integral number of system clock cycles while the peripheral or memory module "catches up" with the processor.

3K WRITE DIS L	This line when active will write protect the upper 3K words of the memory module. Source : 6903-CONTRL Driver : SW
4K WRITE DIS L	This line when active will write protect the memory module. Source : 6903-CONTRL Driver : SW

TABLE 3-5

EDGE CONNECTOR PIN ASSIGNMENTS

COMPONENT SIDE			REVERSE SIDE		
Pin	Description	Active Level	Pin	Description	Active Level
1	PR$_{OUT}$	H	2	PR$_{IN}$	H
3	+5V		4	V$_2$	
5	+5V		6	XTB	H
7	XTA	H	8		
9	INTREQ	L	10	SKP	L
11	DEVSEL	L	12	GND	
13	DMAGNT	H	14	RESET	L
15	CPREQ	L	16	INT DIS	L
17	C2	L	18	C$_0$	L
19	DMAREQ	L	20	C$_1$	L
21	V$_2$		22	GND	
23	V$_2$		24	+5V	
25	FIELD SELECT 2		26	RUN/HLT	L
27	XTC	H	28	WAIT	L
29	FIELD SELECT 1		30	MEMSEL	L
31	3K WRITE DIS	L	32	GND	
33	FIELD SELECT 0		34	INTGNT	H
35	4K WRITE DIS	L	36	+5V	
37	MEMORY DISABLE	L	38	LXMAR	H
39			40	GND	
41	UP	L	42	DX(8)	
43	DX(7)		44	DX(9)	
45	DX(0)		46	GND	
47	DX(10)		48	DX(1)	
49	DMA EN	L	50	DX(11)	
51	DX(6)		52	GND	
53	RUN	L	54	DX(2)	
55	CPSEL	L	56	DX(3)	
57			58	GND	
59	DX(4)		60	DATAF	H
61	IFETCH	H	62	DX(5)	
63	SINGLE CLOCK	H	64	GND	L
65			66	SWSEL	L
67	FREE RUN	H	68	LINK	
69	V$_1$		70	+5V	
71	V$_1$		72	+5V	

TABLE 3-6

BUS DRIVER CHARACTERISTICS

Driver Type	Standard TTL-Load Drive Capability	Low-power TTL-load Drive Capability
74365 (8095)/ 74366 (8096)/8833	20	100
74LS365/74LS366	10	50
7400/04	10	50
74LS00/04/257	5	20
7403/05/07	10	50
74LS03/05	5	20
NPN transistor	100	500

CHAPTER 4
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 sponsers 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 IM6100 and the PDP-8/E are software compatible. The basic 4K PDP-8/E Paper Tape Software supplied by DEC will operate properly with the IM6100. 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, and Extended Memory Control element can be used with the IM6100 to extend its addressing capacity from 4K to 32K.

For more information on DECUS Software, please refer to the application note, "DCAN000--DECUS PDP-8 Software Program Library".

DEC 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 with 4K words of memory, a Teletype and a Control Panel--the basic modules provided in the 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.

DEBUGGING PROGRAMS

Dynamic Debugging Technique (DDT) and Octal Debugging Technique (ODT) are two debugging programs for the PDP-8. These two service programs allows the user to run the user program on the computer and to use the Teletype keyboard to control program execution, examine registers, change their contents and make alterations to the program. With the DDT, the user can debug the programs, using the symbolic language of the source program with the DDT performing all translations to and from the binary representation. ODT has the same capabilities as the DDT, except that the programmer must use octal representation instead of the mnemonic symbols. Chapter 5 of Introduction to Programming discusses the features of both of these service routines.

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 LANGAUGES

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 is recommended to reference the DEC PDP-8 software catalog and the DECUS PDP-8 software catalog for a list of all the available 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.

An octal debugger which allows the user to examine, modify, and control execution of programs from the terminal.

An interactive highlevel language interpreter and on-line algebraic calculator with floppy disc input/output capabilities.

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

INTERCEPT DIAGNOSTIC SOFTWARE

Programs to test the processor, memory and the Teletype interface (IDIAG-1) are supplied with the INTERCEPT system. A PROM resident memory test program is also available to run out of the control panel.

All the option modules (6907-EMC, 6970-IFDOS, etc.) have their own diagnostic programs, which are supplied as part of the hardware.

CHAPTER 5

6901-M4KX12: NONVOLATILE CMOS MEMORY MODULE - HARDWARE DESCRIPTION

The detailed circuit diagram of the 4K X 12 nonvolatile CMOS memory system is shown in Figure 5-1. The addresses (A_0-A_{11}) need be valid only for a short duration when the LXMAR pulse makes a negative transition. The IM6508 latches the addresses internally. Since the IM6508 address strobe and chip select functions are provided on a single pin (STR), the address strobes are sent to devices through a decimal decoder (7442). All chip selects are high when the LXMAR goes high. If the memory board is selected, then the chip select to a specific row of IM6508's will be enabled low when LXMAR goes low. The timing requirements are such that the Address, Data-In and Data-Out can be time-shared on the same lines without any degradation in the memory system performance.

The system makes provision for expansion with the FIELD SELECT input. If the Extended Memory Controller is not used, 13B-6 is grounded and the module is always selected. The module is selected when FIELD SELECT is high or low depending on whether 13B-6 is connected to 13C-5 or 13C-6.

Bipolar devices are used for buffering and decoding and the battery backup is provided only for the CMOS devices as shown in Figure 5-2. When the power supply is interrupted, only the CMOS RAMs will draw power from the battery and, hence, the standby characteristic of the system is not affected by using bipolar peripheral circuits.

The PNP transistor, Q4, is on when the system is supplying power and off when memory is on battery backup. The POWER FAIL (L) signal is used to turn off the PNP transistor. This signal also provides a logic flag to the system when the memory devices are on battery backup. Read/write functions to the memory are disabled when the system is on battery backup.

With conventional bipolar devices, resistor pullups to CMOS V_{CC} are required to ensure against CMOS inputs floating when power supply to bipolar devices are interrupted. Low power Schottky devices (54/74 LS) have the interesting property that when their V_{CC} is at GND, the outputs will be at GND also. If inputs to CMOS devices are allowed to float, both P and N channel transistors of input inverters could be active simultaneously, increasing the standby power dissipation considerably. Thus, resistor pullups can be avoided if 54/74 LS devices are used for line buffering.

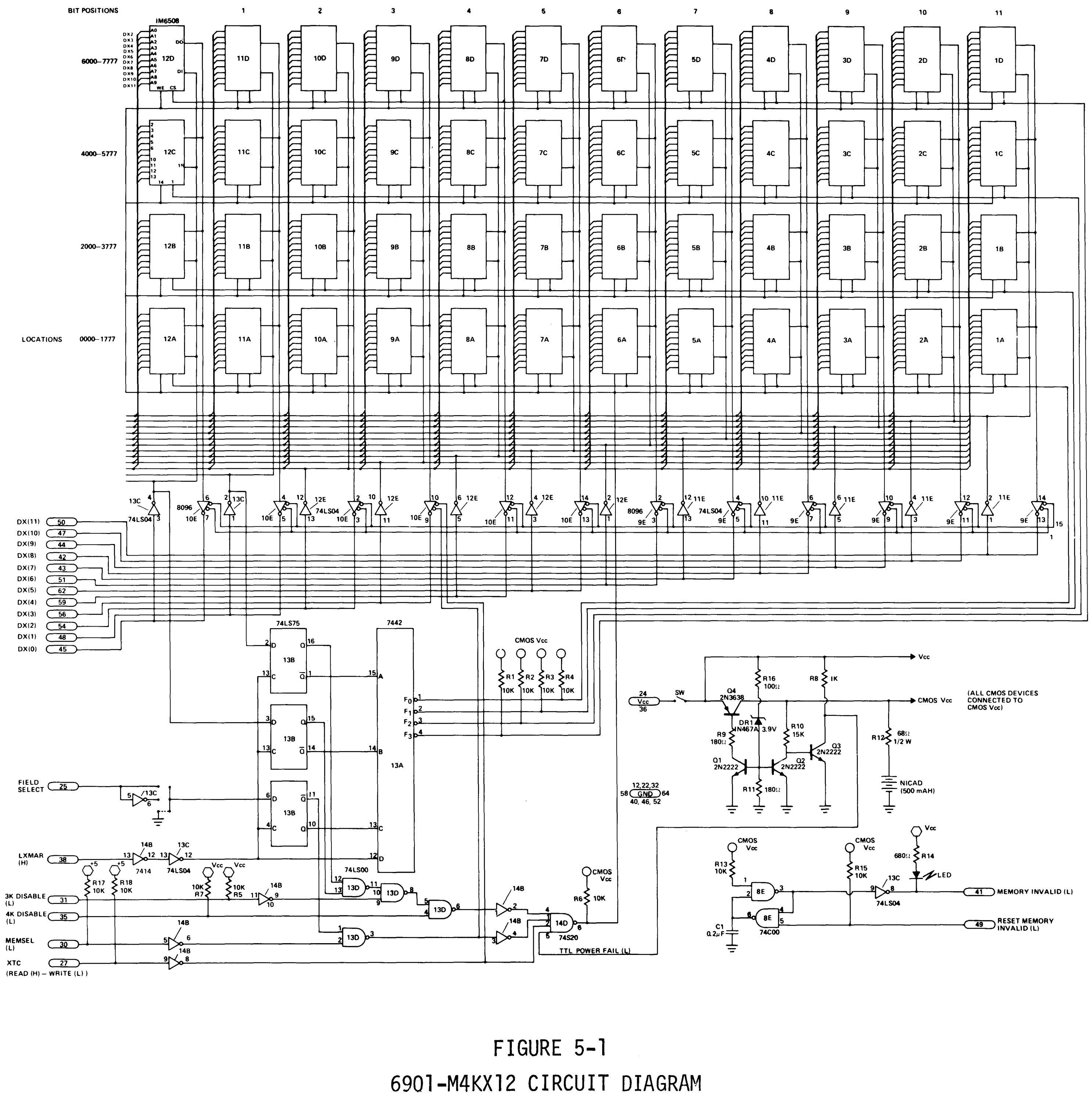


FIGURE 5-1

6901-M4KX12 CIRCUIT DIAGRAM

Certain control lines to the CMOS devices, Chip Selects and Write Enable, must be held high so that the devices are not accidentally written in when the memory devices are on battery backup. 54/75 LS devices should not be used to buffer these control lines.

A DATA INVALID flag as shown in Figure 5-3 is used to inform the system if the power to the memory devices has gone below, even momentarily, the level required to guarantee data retention in CMOS devices (typically 3 volts). The flag once set, must be reset only by a positive action from the system or the user. The capacitance loading on the output of the CMOS gates that make up the latch guarantee that the CMOS latch circuit will come up with a DATA INVALID flag if the CMOS power is disrupted even momentarily or if the CMOS VCC has dropped below the voltage to maintain the state of the CMOS latch. If the data is invalid, the LED indicator illuminates as the system power comes back up.

A switch in series with the power supply is used to protect the memory against stray contact signals when plugging or unplugging the module from a "live" system.

The entire 4K memory can be write protected with the Write Protect 4K signal. There is also an option of write protecting only the upper 3K of memory with the Write Protect 3K signal. The user can simulate 4K RAM, 4K ROM, or 1K RAM-3K ROM combinations with these two signals.

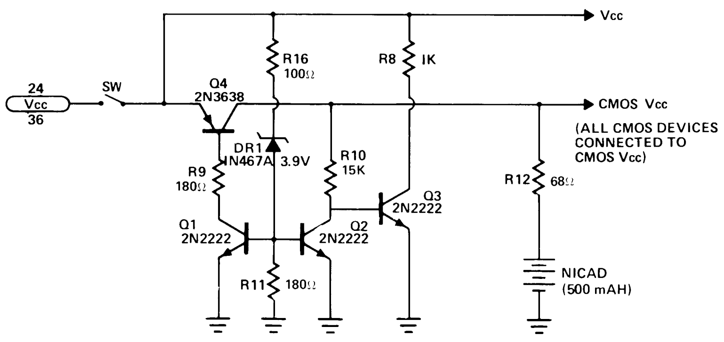


FIGURE 5-2

POWER FAIL DETECTION

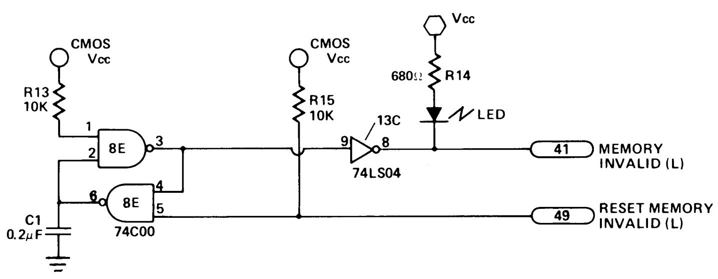


FIGURE 5-3

DATA INVALID INDICATOR

The 6901B-M4KX12 module (Figure 5-4) is a redesign of the 6901-M4KX12 module incorporating the following additional features:

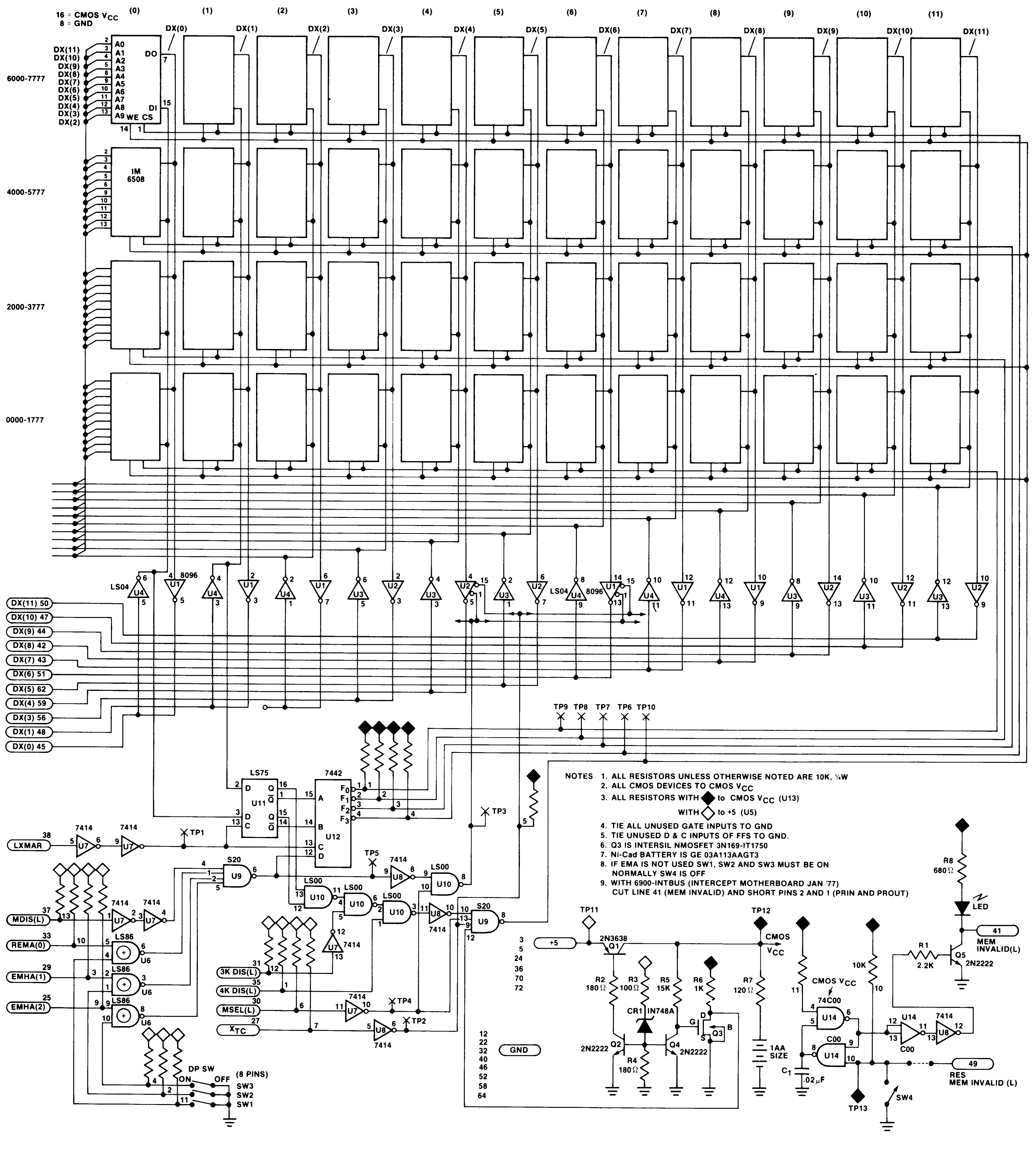
1. On-board field select logic is provided to expand the memory capacity up to 32K. A module is selected, when the DIP SW (1), (2) and (3) setting is the complement of EMA (0), (1) and (2). EMA (0) is the most significant bit of the extended address field. Resistor pull ups are provided on the EMA lines so that when 6907-EMA module is not in the system, the DIP SW setting must correspond to a field selection of 7_8 (all switches closed).

   DIP SW (4) is used to turn off the MEM DATA INVALID indicator.

FIELD SELECTION	DIPSW (1)	DIPSW (2)	DIPSW (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
EMA lines not asserted	ON	ON	ON

2. An additional control signal, MDIS (L), is provided to inhibit all data transfers to and from main memory. One can then, with additional logic in the control panel module, selectively disable main memory and force all 'indirect' references to come from the control panel memory. This is a useful feature to have for more complex control panel programs.

3. The Q2 NPN transistor (2N2222) in the 6901-M4KX12 module is replaced by the Q3 NMOS FET (3N169-IT1750). When the memory module was on battery stand-by, the base current of the Q2 transistor (about 200 μA) was significantly more than the total stand-by current drain of all memory devices (typically 10-20 μA). The NMOS FET eliminates this current drain. So for the same stand-by characteristics as the 6901-M4KX12, one can use 1/3 AA cells (GE 03A113AAGT3 3.75V 100 mAH) instead of AA cells.



NOTES
1. ALL RESISTORS UNLESS OTHERWISE NOTED ARE 10K, ¼W
2. ALL CMOS DEVICES TO CMOS V_{CC}
3. ALL RESISTORS WITH ◆ to CMOS V_{CC} (U13)
   WITH ◇ to +5 (U5)
4. TIE ALL UNUSED GATE INPUTS TO GND
5. TIE UNUSED D & C INPUTS OF FFS TO GND.
6. Q3 IS INTERSIL NMOSFET 3N169-IT1750
7. Ni-Cad BATTERY IS GE 03A113AAGT3
8. IF EMA IS NOT USED SW1, SW2 AND SW3 MUST BE ON
   NORMALLY SW4 IS OFF
9. WITH 6900-INTBUS (INTERCEPT MOTHERBOARD JAN '77)
   CUT LINE 41 (MEM INVALID) AND SHORT PINS 2 AND 1 (PRIN AND PROUT)

FIGURE 5-4
6901B-M4KX12 CIRCUIT DIAGRAM

CHAPTER 6

6902-CPUTTY: CENTRAL PROCESSOR UNIT WITH SERIAL I/O INTERFACE - HARDWARE DESCRIPTION

CENTRAL PROCESSOR UNIT

The processor module forms the nucleus for a versatile tri-state bus organized system. The module contains the IM6100 CMOS microprocessor device, buffers to interface it to the INTERCEPT bus structure, a 4.000 MHz system clock and a DEC PDP-8/E compatible Teletype interface.

All output signals from the IM6100 microprocessor are buffered. Signals which are heavily used in the system use high-current bus drivers. Signals which are used by only a few accessory modules use LS gates for bus drivers (Figure 6-1).

The bi-directional, multiplexed data lines DX0-DX11 employ type 8833 tri-state bi-directional transeiver devices (4G, 5E and 5F) to interface to the bus. During the time when bus line XTA is asserted, the microprocessor is in the "read" mode, and the 8833's are enabled to drive the bus data toward the IM6100. When bus line XTB is asserted, the IM6100 is in the "write" mode, and the 8833's are enabled to drive the microprocessor DX lines onto the bus. If neither XTA or XTB is asserted, the 8833's are tri-stated, and the CPU module leaves the DX lines on the bus to "float". This is also the case during the period of a direct memory access (DMA) operation, when the line DMAGNT is asserted high.

The microprocessor output lines XTC, MEMSEL, DATAF and LXMAR are driven onto bus through sections of an 74365 tri-state buffer (4F). During a DMA operation, 4F is tri-stated and the aforementioned bus lines are left to "float". All lines left floating by the CPU module during a DMA operation are asserted by a DMA port, which is part of a (optional) device-interface for a high-speed peripheral, such as a disk memory.

The microprocessor output lines CPSEL, LINK, DEVSEL, SWSEL, IFETCH and INTGNT are driven onto the bus through sections of another 74365 buffer (4E). These lines are never tri-stated. The output lines RUN and DMAGNT are applied to the bus through LS gates.

The 4 MHz system clock is generated by the ICM7209 crystal oscillator.

The lines C0, C1, C2, WAIT, CPREQ, DMAREQ, RESET, RUN/HALT and SKIP are all input control or request lines to the CPU. These are all asserted, at various times, by logic on the control panel and device interface modules. The assertion level of all these lines is a TTL low; they are normally held in the high state by pull-up resistors.

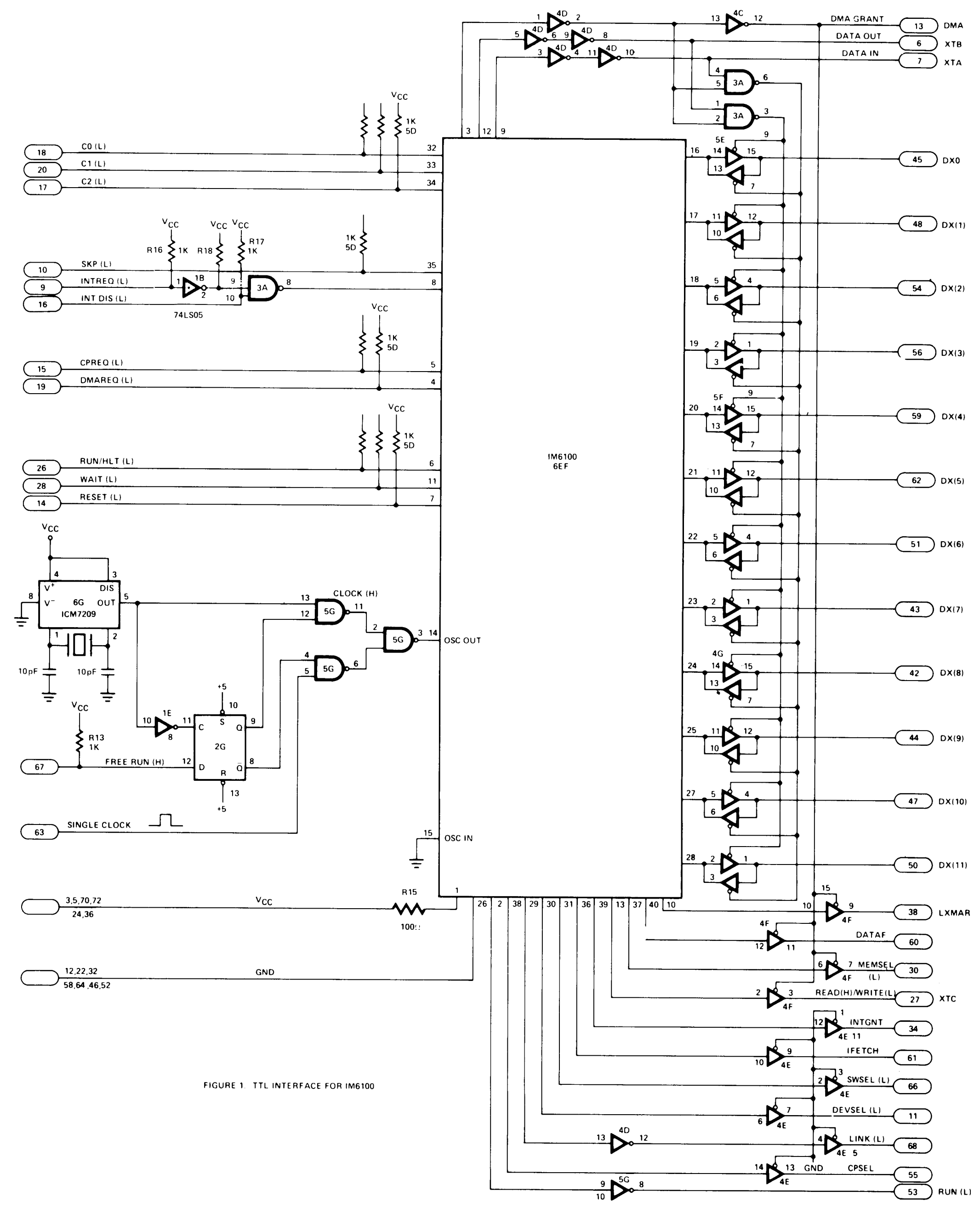


FIGURE 6-1

IM6100 WITH TRISTATE BUFFERS

Pin 8 on the IM6100 microprocessor is the peripheral-device-interrupt request line. It has an active-low assertion level. Thus, interrupt requests can be prevented by asserting the bus line INTDIS to a logic low, due to the action of gate 3A-9. This is often done for brief intervals by the Interrupt-Inhibit Flip-Flop (IIFF) on the Extended Memory Controller module, when this module is present in the system (Chapter 8). Interrupt requests by peripheral devices are routed to gate 3A-9 through the inverter in 1B-2.

The IM6100 can be clocked from the system clock or, in Single Clock mode, from a front-panel pushbutton. The former source is selected when the bus line FREERUN is high, the latter when FREERUN is low. A D flip-flop (2C) prevents clock "slivers" from reaching the IM6100 when switching from one clock source to the other.

TELETYPE INTERFACE

The Teletype allows the user to communicate directly with the INTERCEPT using a keyboard for input, and printed copy for output. Additionally, the Teletype provides a low-speed paper tape input and output capability. The Teletype operates at 110 baud, using two stop bits, and it interfaces to a 20 mA current loop.

To be compatible with the INTERCEPT software, the Teletype requires some small modifications to the standard wiring. The required changes are discussed in Appendix D.

All serial data input and output between the interface and the Teletype is made via 6B, a Universal Asynchronous Receiver/Transmitter (UART) device (IM6402), see Figures 6-2 and 6-3. Input from the terminal arrives at pin 20 of the UART. Output to the terminal comes from UART pin 25.

On the processor side of the UART, the data is in parallel format. Inputs from the bus arrive at pins 26-33. Outputs to the bus are at pins 5-12. Since the UART does not have sufficient output to drive the bus directly, devices 5B and 5C buffer the UART outputs onto the bus. These devices are enabled only during the "read" portion of an IOTA cycle, when the module has been addressed for input to the CPU. Note that the eight data bits used in asynchronous communication come from/go to the eight least significant bits in the processor's Accumulator.

Data communication between the terminal and the UART is as follows. On the receive side of the UART, pin 20 is always high when the keyboard and reader are inactive. This is the "marking" state; the terminal is sending a constant stream of stop bits toward the computer. When a key is struck on the keyboard, pin 20 is pulled low by the "spacing" start bit, which causes the UART to begin clocking in the eight data bits from the terminal (Figure 6-4),

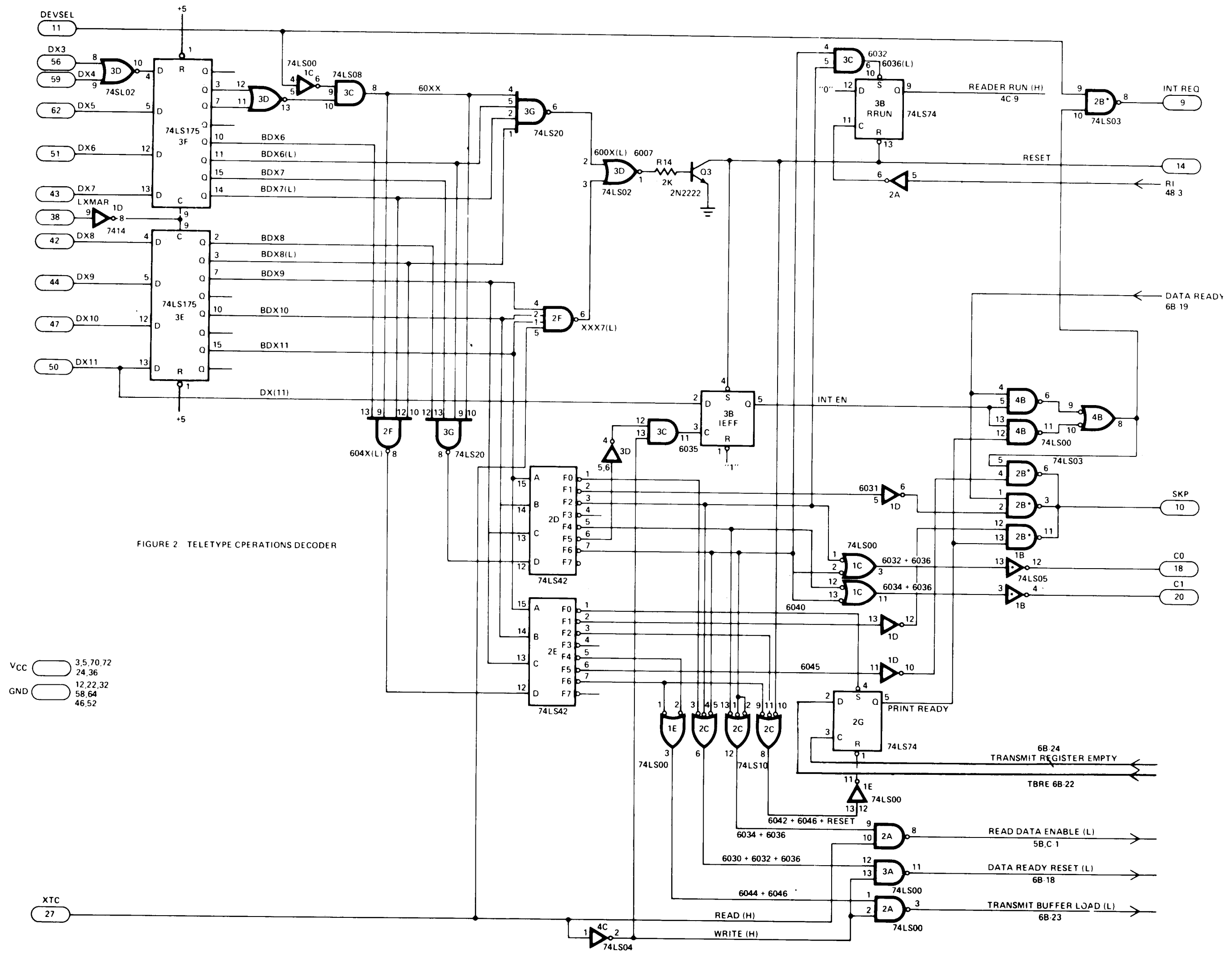


FIGURE 6-2

TELETYPE INTERFACE INSTRUCTION DECODING CIRCUIT

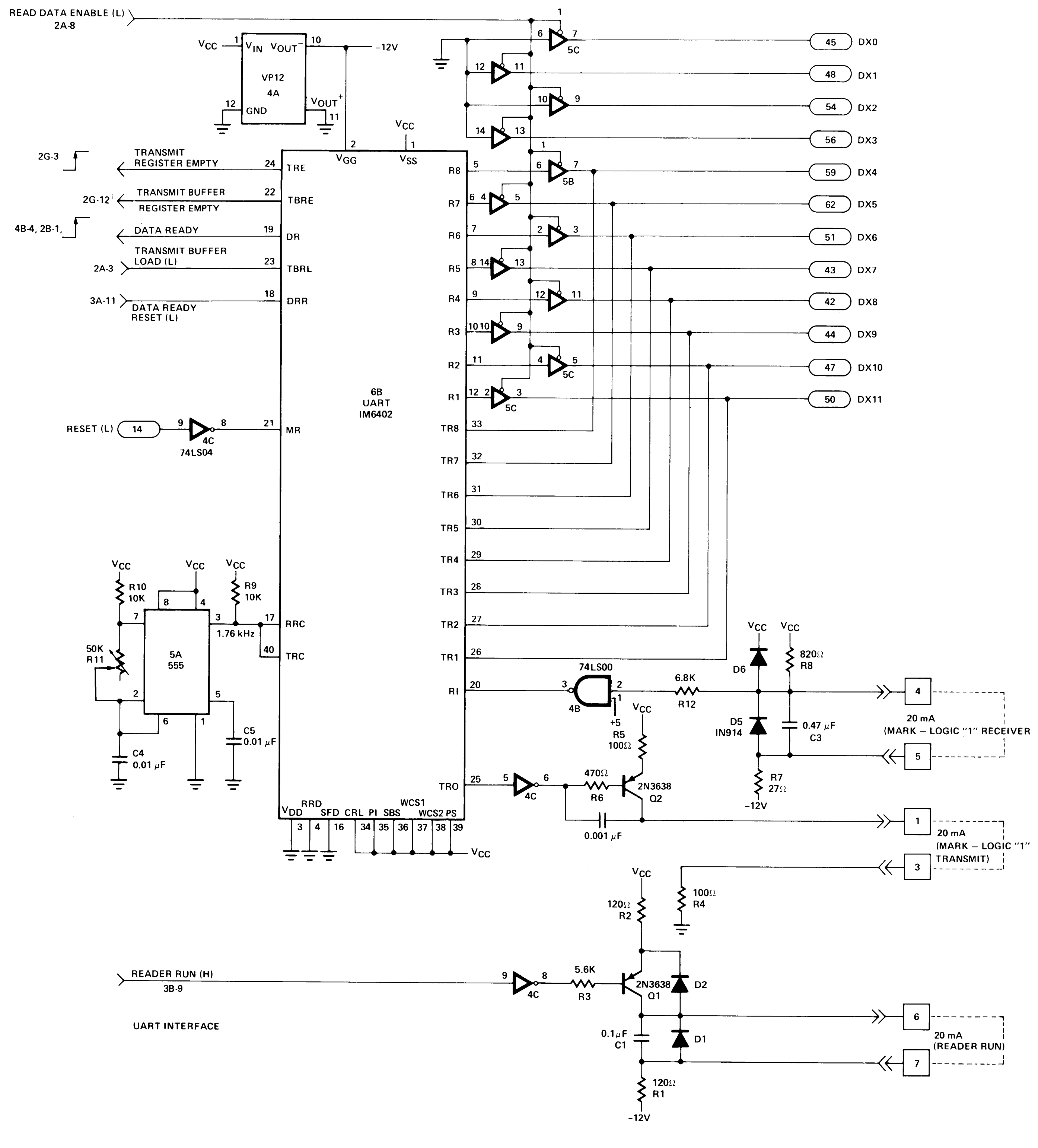


FIGURE 6-3

UART WITH TRISTATE BUFFERS

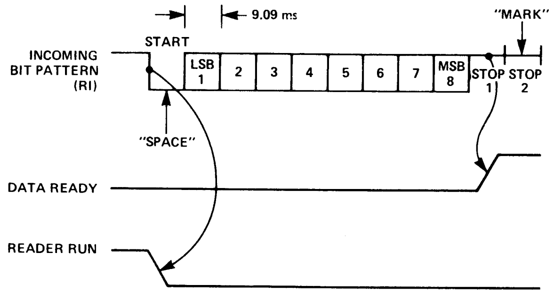
and also causes the reader relay to be disabled by clocking flip-flop 3B to the reset state. As soon as the entire character has been clocked into the UART, the UART's Data Ready flag is set, pulling pin 19 high. This causes the INTREQ line to be asserted low, if the Teletype Interrupt Enable Flip-Flop, 3B, is set, through gates 4B-6, 4B-8 and 2B-8. The program software uses instructions KSF and KRS or KRB to read the data into the processor's Accumulator.

On the transmit side, the processor sends data to the UART using instructions TSF and TPC or TLS, when it finds the Printer/Punch Ready flag is set. When the UART picks up the data from the DX lines on the bus, it begins to transmit the character to the terminal in serial format, by first sending a "spacing" start bit, then the eight data bits, then two stop bits (Figure 6-5). At the end of the character, pin 25 is in its normal rest "marking" state, where it stays until another character is ready to be transmitted to the terminal. Also at the end of the character, the UART signals the completion of transmission by pulling its pin 24 high. This sets the Printer/Punch Ready flag, so the processor can ascertain that the UART is ready to transmit another character.

Inputs and outputs to and from the terminal are made via a seven-pin connector. Pin numbers on the connectors are noted in square boxes on the schematic diagram. The input pins are numbers 4 and 5, and the corresponding pin-pair for putput is 1/3. Current for the reader control relay is at pins 6 and 7. Pin 2 is keyed. At the rear panel of the INTERCEPT, a connector is provided for connection to the Teletype.

The module uses device addresses 03_8 and 04_8, and, therefore, is fully compatible with all DEC software instructions intended for the Teletype terminal. These device numbers are embedded in the IOT instruction, at bits 3-8. The instruction set for the Teletype interface is shown in Table 6-1.

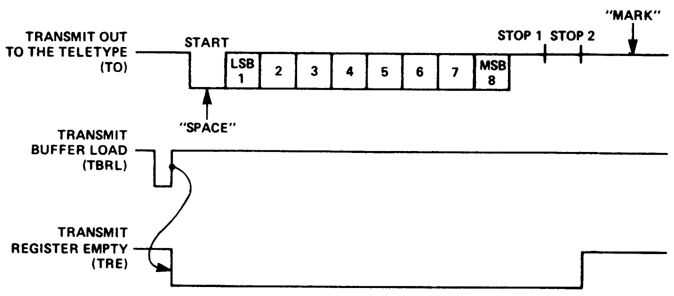
In addition to all instructions for devices 03_8 and 04_8, the TTY interface also decodes instruction 6007_8 (Clear all Flags, CAF). When executed, this instruction clears all interrupt request flags (2G-5 and the Data Ready flip-flop in the UART) on the module, and sets the Teletype Interrupt Enable Flip-Flop, 3B-9. In addition, CAF causes the RESET line on the bus to be asserted low by Q3, which clears all the similar flags in other peripheral interfaces. (Note that CAF, although asserting RESET, is only asserted during the time DEVSEL is active, and this does not include the time when the processor samples the RESET line and hence the processor is not reset.)



RECEIVER TIMING DIAGRAM

FIGURE 6-4

UART RECEIVER TIMING DIAGRAM


"MARK"
TRANSMIT OUT TO THE TELETYPE (TO)
START
STOP 1
STOP 2
LSB 1
2
3
4
5
6
7
MSB 8
"SPACE"
TRANSMIT BUFFER LOAD (TBRL)
TRANSMIT REGISTER EMPTY (TRE)

TELETYPE TRANSMITTER TIMING DIAGRAM

FIGURE 6-5

UART TRANSMITTER TIMING DIAGRAM

TABLE 6-1

MNEMONIC	OCTAL	OPERATION
KCF	6030	Reset the Keyboard Reader Data Ready flag.
KSF	6031	Skip the next instruction if the Keyboard Reader Data Ready flag is set.
KCC	6032	The reader is enabled to fetch the next character*, the Data Ready flag is reset, and the Accumulator (AC) is cleared.
KRS	6034	Tke keyboard reader data is OR'ed into the AC.
KIE	6035	AC bit 11 is loaded into the Teletype Interrupt Enable Flip-Flop (TIEFF). This sets the TIEFF if AC (11) = 1, and clears it if AC (11) = 0.
KRB	6036	The keyboard reader data is jam-loaded into the AC, the Data Ready flag is reset, and the reader is enabled to fetch the next character.
TFL	6040	Set the Printer/Punch Ready flag.
TSF	6041	Skip the next instruction if the Printer/Punch Ready flag is set.
TCF	6042	Reset the Printer/Punch Ready flag.
TPC	6044	The AC is non-destructively outputted to the printer/punch buffer, from where the character is automatically delivered to the device.
TSK	6045	Skip the next instruction if the TIEFF is set <u>and</u> either the Keyboard Reader Data Ready flag or <u>the</u> Printer/Punch flags (or both), is set.
TLS	6046	The AC is outputted to the device, as in TPC. The Printer/Punch Ready flag is reset. The flag will be set again by the interface logic when the printer/punch is ready to accept another character.
CAF	6007	Reset the Keyboard Reader Data Ready flag, and the Printer/Punch Ready flag. (Also asserts RESET bus line low.)

* The TTY reader is automatically disabled after fetching each character.

Each IOT instruction is latched into devices 3F and 3E by the trailing edge of LXMAR. (It is not necessary to latch, or decode, the most significant three bits, since these always form 6_8 for an IOT, and this Opcode and the DEVSEL pulse are redundant.

DEVSEL only appears during IOT cycles.) Decoding of the instruction takes place at 2D, 2E, 2F and 3G. Due to the action of gate 3C-8, a pulse for each decoded instruction (for devices 03_8 and 04_8) appears during DEVSEL time at the outputs of 2D and 2E. The logical action caused by each instruction is noted in Table 6-2.

For example, instruction 6030_8 resets the Keyboard Reader Data Ready flag by pulling pin 18 low on the UART, through gates 1E-3 and 3A-11. Note that this instruction writes into the interface, so the action takes place during T6, when XTC is low. Instruction 6036_8 performs this same operation, but also enables the reader to fetch the next character (by pulling pin 6 low on 3C), enables the UART to drive the DX bus lines with the "read" data during T2 and T3 when XTC is high (by pulling pin 1 low on 5B and 5C), and asserts the C0 and C1 bus lines to cause the processor to read the DX data as a jam-transfer into the Accumulator.

Similiarly, instruction 6041_8 causes the next instruction to be skipped if the Printer/Punch Ready flag (2G-5) is set, by asserting the SKP line on the bus, through 2B-11. Note that the SKP line is enabled and the INTREQ line disabled when DEVSEL is active. The SKP and INTREQ line can then be tied together, if so desired. The IM6100 samples INTREQ at T1 and SKP at DEVSEL (L) and XTC (H). Peripheral Interface Element (IM6101) and the DMA/EMC device (IM6102) time multiplex these two signals on the same pin, INT/SKP.

The UART must be clocked at 16X the bit transfer rate. The Teletype transmits/receives one bit every 9.09 ms. So, the UART must transfer a START bit, 8 DATA bits and 2 STOP bits in 99.99 ms, achieving a transfer rate of 110 bits/second (110 baud) or 10 characters/second. The UART rate is controlled by the 1.76 KHz (110 X 16) clock, generated by the 555 analog timer circuit (5A).

CHAPTER 7

6903-CONTRL: OPERATOR CONSOLE LOGIC AND DISPLAY MODULE - HARDWARE DESCRIPTION

A schematic diagram for the control panel module is given in Figure 7-1. This drawing should be reviewed in relation to the panel software routine given in Appexidx C, since most front panel operations are software driven. Reference is also made to Chapter 3, which discusses the machine's provisions for control panel interrupts.

Any of the six function switches can cause a control panel (CP) interrupt request by grounding an input on gate U13, so long as the machine is in the HALT state. The latter condition is assured by the input to pin 2 of gate 1A. This open collector gate asserts the bus line CPREQ to the low state.

When activated by its front panel switch, the timer (U6) can generate a CP interrupt request with the processor in either the RUN or HALT state. (With the machine in the RUN state, gate 1A will only assert the CPREQ bus line low if both the INTGNT and DMAGNT lines are in the low state.) The timer generates clock pulses to the FF (4A), which in turn enables gate 1A. The first pulse on the CPSEL bus line resets 4A as soon as the CP routine begins.

Note that the D input to 4A is tied to the RUN/HALT bus line. This guarantees that every activation of the CONT pushbutton will be serviced; a (asynchronous) timer generated CP request cannot block out a CONT request by forcing the machine into the RUN mode while the CONT button is active.

On every cycle of the CP routine, the trailing edge of the LXMAR pulse through inverter 4B, latches the address of instruction data into D flip-flops 4E and 4D. Note that all incoming DX data is inverted by hex inverters 5D and 5E.

During each PROM access cycle (usually an instruction fetch cycle), the PROM chips 3C, 3D and 3E are addressed. The PROM's occupy locations 7400_8-7777_8. For these high order locations DX (0) = 1, so "enable" pin 13 (CS1) on each PROM is asserted low. Then during the "read" time, while CPSEL is asserted low, the other PROM "enable" pin, pin 14 (CS2) is asserted low, and the PROM's drive the inputs of the tri-state buffers, 4C and 5C. While CPSEL is low and XTC is high, 4C and 5C are enabled to drive the PROM data to the processor on the DX lines.

Addressing of the RAM devices, 2C, 2D and 2E is accomplished similarly. The RAM's occupy locations 0000_8-0017_8. In every cycle which accesses these locations, pin 7 on the octal decoder, 3B, is asserted low when CPSEL is low. This enables the RAM's. Then during the "read" portion of the cycle, RAM data from the specific location whose address is latched into 4D and 4E is buffered out to the processor, while XTC is high and CPSEL is asserted low. But during the "write" portion (whenthe T6 state is entered) XTC is low, which puts the RAM's into write mode while CPSEL is low, and they pick up the DX data from the processor at the output of hex inverters 5D and 5E. (The RAM's invert all data stored in them.)

Note that 3B acts as an address decoder for DX lines 5-7, and also DX (0) through the positive sense "enable", pin 6. The RAM devices, Rotary Switch, Function Switches, Program Counter and Display Indicators are all addressed as low order memory locations (DX (0) low, so pin 6 on 3B high). The five outputs from 3B act to differentiate between these five functions by decoding DX (5-7). As noted above, the RAM's occupy locations 0000_8-0017_8. The Rotary Switch is addressed as location 0100_8, the Function Switches as location 0060_8, the PC Indicators as 0020_8, and the Display Indicators as 0040_8. Note that the outputs from 3B can only be asserted during the time when CPSEL is asserted low.

The Rotary Switch is only involved in "read" operations. When it is addressed, pin 12 on 3B becomes low to enable pin 8 of the open collector gate 1B, and one of the DX lines 8-11 is driven low during the "read" portion of the cycle, depending on the position of the Rotary Switch.

The Function Switches are similarly only involved in "read" operations, after a CP interrupt is generated. When they are addressed, open collector gate devices U11 and U12 are enabled by 3B at read time. The DX (0-5) are all driven low, except for the line driven by an active Function Switch. This DX line will be driven high, and read by the processor.

The PC display is only involved in "write" operations. The hex D flip-flops U1 and U3 act as a one-word "write only" memory when clocked by pin 9 on 3B through outputs 2B-8, 5B-6, 5B-3 and 2B-6. At this time they pick up the data at the outputs of inverters 5D and 5E and latch it into the PC indicators.

Similarly, the Display Indicators are only involved in "write" operations. They are selected by pin 10 on 3B. The quad-latch devices U2, U4 and U5 pick up the outputs of 5D and 5E on the trailing edge of the CPSEL pulse, and latch this data into the Display Indicators.

Whenever the panel is in the Single Clock mode, the FREERUN bus line is low, and pin 8 of 5B is always high. This causes all DX data to pass through the latch devices U2, U4 and U5 and to be shown in the Display Indicators. Thus the Display simply follows the DX lines.

Also, while FREERUN is low, the output 5B-6 remains high, so the output 5A-6 is able to clock U1 and U3 through 5B-2. Output 5A-6 is enabled during each instruction fetch cycle by 4A-5. The result is that the PC displays the address of the current instruction being clocked through the the Single Clock switch. The D flip-flop 4A is used to truncate the IFETCH signal since it overlaps into the 'next' cycle at 4 MHz.

The FREERUN toggle switch grounds the FREERUN bus line, in the Single Clock position. On the processor module this has the effect of disabling the processor drive from the system clock oscillator, and enabling the Single Clock pushbutton on the front panel. This pushbutton generates a single clock pulse to the processor each time it is activated by setting, then resetting, the flip-flop formed by 2C-6 and 2C-3.

The processor can be enabled to enter the RUN state by putting the HLT switch up. This brings the output 2A-6 high. When the CONT pushbutton is activated, the output 2A-11 will go low. Note that pin 12 on 2A is high, since the machine is still halted. This asserts the RUN/HALT bus line low. Then when the CONT pushbutton is released the RUN/HALT line returns high. This positive going edge toggles the processor's internal RUN/HALT flip-flop, putting the machine into the RUN state.

If the RUN/HALT switch is then thrown to the HALT position, the next assertion of the IFETCH bus line will assert the RUN/HALT bus line low through outputs 3A-6, 3A-4, 2A-6, 2A-8 and 1B-6. When IFETCH again goes low, the RUN/HALT line will produce a positive going edge which toggles the RUN/HALT flip-flop. The machine will halt at the conclusion of the last execute cycle of the current instruction.

With the RUN/HALT switch in the HALT position, a single instruction can be caused to be fetched and executed by activating the CONT pushbutton. With the RUN/HALT switch in the HALT position, pin 5 on 2A is low, so pin 6 is high, enabling gate 2A-10. Pin 12 on 2A is high, so when the CONT pushbutton is activated, the output 2A-11 asserts the RUN/HALT line low through gates 2A-8 and 1B-6. Then when the switch is released, a positive going edge is produced on the RUN/HALT line which puts the machine in the RUN state. But immediately the IFETCH line pulses the RUN/HALT line through gates 3A-6, 3A-4, 2A-4, 2A-8 and 1B-6, which forces the machine to halt again after the execution of a single instruction.

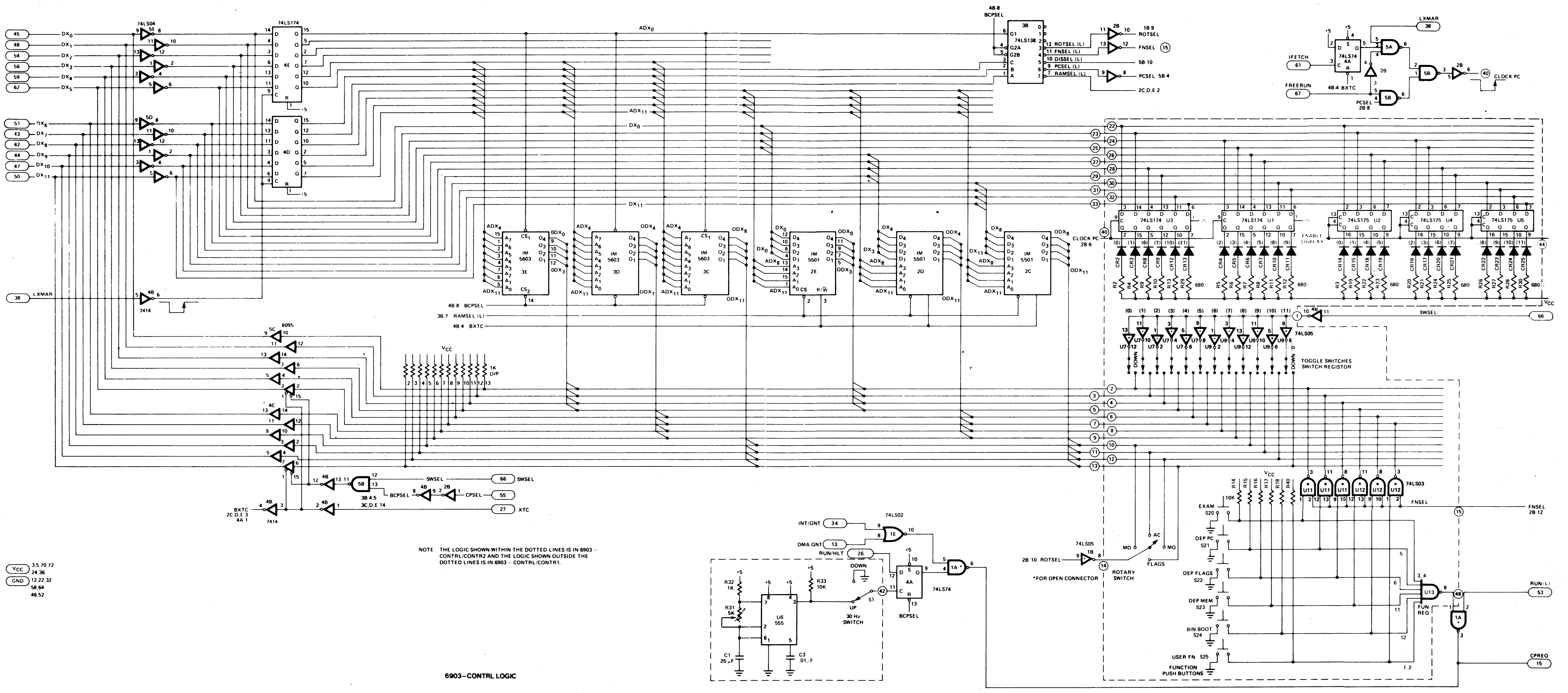
Note that when a control panel interrupt is executed with the RUN/ HALT switch in the HALT position, the RUN/HALT line is pulsed on every instruction in the control panel routine by the IFETCH line. These pulses toggle the processor's internal Run Halt Flip-Flop. However, since the CP routine has a higher priority than the HALT state, the processor stays in the RUN mode until the end of the control panel routine. Then at the end of the CP routine, the processor enters the state to which the Run Halt Flip-Flop was toggled by the IFETCH line on the last instruction in the routine. It is obviously not desirable, though, that the machine emerge from the CP routine in a different state than the one it was in when it entered the routine.

To solve this potential problem for timer generated CP interrupts, it is necessary that the portion of the CP routine which is used when the timer causes the interrupt contain an even number of instructions. Fortunately, this is easy to do since there are only four different paths through the CP routine for timer generated interrupts, one for each possible position of the Rotary Switch, Thus, the Run Halt Flip-Flop will be toggled an even number of times, producing no net change in its state at the point of emergence from the CP routine.

For CP interrupts generated by the Function Switches, the problem is solved by inserting a HLT instruction an even number of instructions from the end of the CP routine. The HLT instruction jams the Run Halt Flip-Flop to the HALT state regardless of how many pulses previously appeared on the RUN/HALT line. Then, since an even number of pulses on the RUN/HALT line follow the HLT instruction, the machine will emerge from the CP routine in the HALT state. Note that the HLT instruction is outside the loop traversed by timer generated interrupts.

When the RUN/HALT switch is in the RUN position, the CP routine does not toggle the RUN/HALT line.

The "even number" instruction restriction can be eliminated by changing the 2A-6 gate into a 3-input gate and feeding MEMSEL (H) signal to the third input. Then, the RUN/HLT line will be pulsed only if the instruction fetch is from main memory.

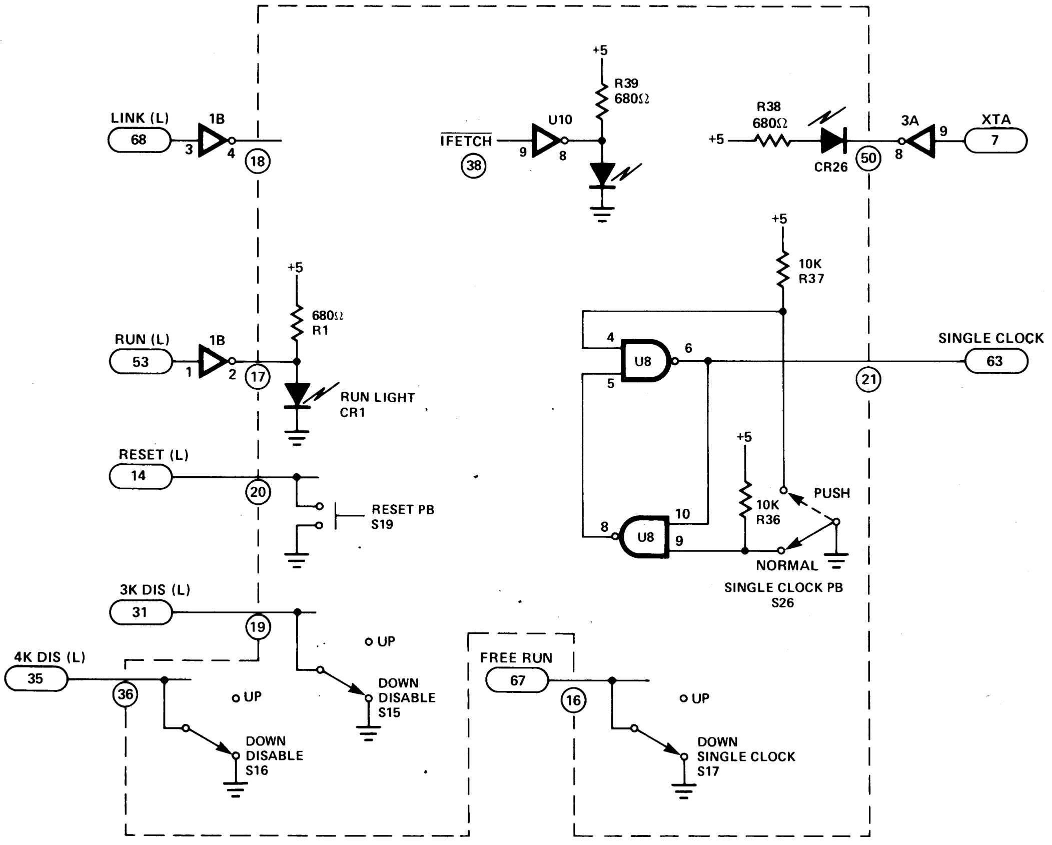


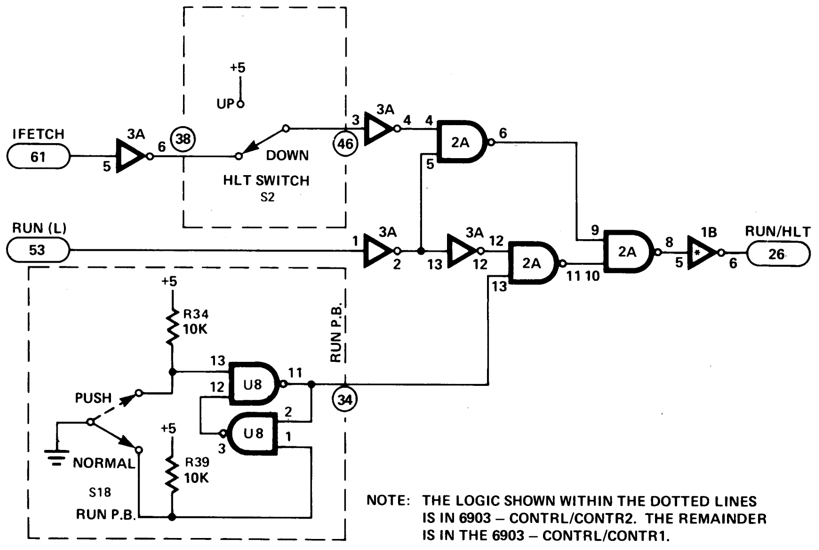
NOTE: THE LOGIC SHOWN WITHIN THE DOTTED LINES IS IN 6903 – CONTRL/CONTR2 AND THE LOGIC SHOWN OUTSIDE THE DOTTED LINES IS IN 6903 – CONTRL/CONTR1.

ADDRESSES

```
1XX   XX......   ...   PROM – 7400 – 7777
0XX   XX0   00.   ...   RAM – 0000 – 0017
0XX   XX0   01X   XXX   PC – 0020
0XX   XX0   10X   XXX   DISPLAY – 0040
0XX   XX0   11X   XXX   FN – 0060
0XX   XX1   00X   XXX   ROT – 0100
```

FIGURE 7-1 (a)
OPERATOR CONSOLE LOGIC DIAGRAM





NOTE: THE LOGIC SHOWN WITHIN THE DOTTED LINES IS IN 6903 – CONTRL/CONTR2. THE REMAINDER IS IN THE 6903 – CONTRL/CONTR1.

6903–CONTRL LOGIC

FIGURE 7-1 (b)

OPERATOR CONSOLE LOGIC DIAGRAM

TABLE 7-1

RIBBON CABLE PIN ASSIGNMENTS
6903-CONTRL/LOGIC to 6903-CONTRL/DISPLAY

PIN #	SIGNAL	ORIGIN/DESTINATION	PIN #	SIGNAL	ORIGIN/DESTINATION
1	SWSEL	(from 4B-10)	26	DX(4) (L)	(from 5E-4)
2	DATA(0)	(to 5C-10)	27	DX(5) (L)	(from 5E-6)
3	DATA(1)	(to 5D-12)	28	DX(6) (L)	(from 5D-8)
4	DATA(2)	(to 5C-14)	29	DX(7) (L)	(from 5D-10)
5	DATA(3)	(to 5C-6)	30	DX(8) (L)	(from 5D-12)
6	DATA(4)	(to 5C-4)	31	DX(9) (L)	(from 5D-2)
7	DATA(5)	(to 5C-2)	32	DX(10) (L)	(from 5D-4)
8	DATA(6)	(to 4C-14)	33	DX(11) (L)	(from 5D-6)
9	DATA(7)	(to 4C-12)	34	CONTINUE	(to 2A-13)
10	DATA(8)	(to 4C-10)	35	GND	
11	DATA(9)	(to 4C-2)	36	4KDIS (L)	(edge #35)
12	DATA(10)	(to 4C-4)	37	GND	
13	DATA(11)	(to 4C-6)	38	IFETCH (L)	(from 3A-6)
14	ROTSEL (L)	(from 1B-8)	39	GND	
15	FNSEL	(from 2B-12)	40	CLOCK PC	(from 2B-12)
16	FREE RUN	(edge #67)	41	GND	
17	RUN	(from 1B-2)	42	30HZ	(to 4A-11)
18	LINK	(from 1B-4)	43	+5V	
19	3KDIS (L)	(edge #19)	44	ENABLE DISPLAY	(from 5B-8)
20	RESET (L)	(edge #14)	45	+5V	
21	SINGLE CLOCK	(edge #63)	46	HLT	(to 3A-3)
22	DX(0) (L)	(from 5E-8)	47	+5V	
23	DX(1) (L)	(from 5E-10)	48	PANEL CPREQ	(to 1A-1)
24	DX(2) (L)	(from 5E-12)	49	+5V	
25	DX(3) (L)	(from 5E-2)	50	XTA (L)	(from 3A-8)

CHAPTER 8

6907-EMC EXTENDED MEMORY CONTROLLER

INTRODUCTION

The 6907-EMC is a factory designed interface for the INTERCEPT prototyping system. The purpose of the 6907-EMC is to extend the effective 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.

MEMORY EXTENSION CONTROLLER

A simplified block diagram of the Memory Extension Controller is represented in Figure 8-1. The diagram shows two 3-bit field registers: the Instruction Field, which acts as an extension to the instruction and directly obtained operand addresses and the Data Field, which augments indirectly obtained operand addresses. The program 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.

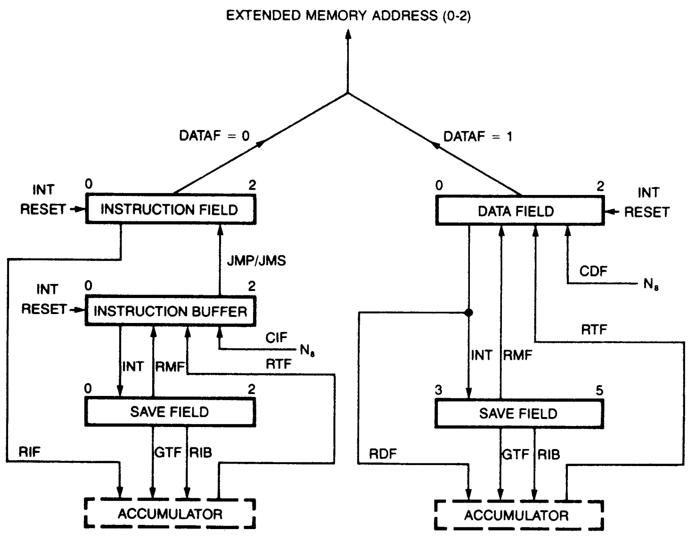


FIGURE 8-1

EXTENDED MEMORY CONTROLLER BLOCK DIAGRAM

INSTRUCTION FIELD REGISTER (IF)

The IF is a 3-bit register that serves as an extension of 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 indirect pointer for all indirectly addressed memory reference instructions reside in the Instruction Field.

DATA FIELD REGISTER (DF)

The DF is a 3-bit register which determines the memory field from which operands are fetched in indirectly addressed AND, TAD, ISZ or DCA instructions. However, 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 takes place. If instructions could change the IF directly, it would be impossible to execute the "next" sequential instruction, followed by 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 provide the ability to load the IF register from the IB register. This allows the control panel routines to be executed transparently while the IB and IF differ and also yields a method for 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 must be "gated" by the Interrupt Inhibit Flip-Flop so that, when the Instruction Field is changed under program control, all 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, this inhibition of the INTREQ's ensures that the program sequence resumes 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 is generated when the IF and IB bits are different. The Interrupt Inhibit FF guarantees the structural integrity of the program segment.

INSTRUCTION REGISTER

Although not shown in Figure 8-1, the instruction resigter is important. This register latches the necessary IM6100 instruction information as it is fetched from any program segment so that it may be combined with timing signals to create the actual control signals to the other elements and back to the IM6100 via C0 and C1. Additionally, since some of the 6907 instructions have embedded data, this register serves as a data source for the IB and DF registers.

OPERAND FETCHING

Instructions are accessed from the currently assigned Instruction Field. For indirect AND, TAD, ISZ or DCA instruction, the operand address refers first to the Instruction Field to obtain an Effective Address which in turn refers to a location in the currently addressed 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 (2) ← INTREQ Line AC (3) ← INT INHIBIT FF AC (4) ← INT ENABLE FF AC (6-11) ← SF (0-5)
		Description:	LINK, INTREQ and INT ENABLE FF are internal to the CPU. The INT INHIBIT FF and SR 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 (5) 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.
CDF	$62N1_8$	CHANGE DATA FIELD	
		Operation:	DF ← N_8
		Description:	Change DF register to N (0_8-7_8).

CIF $62N2_8$ **CHANGE INSTRUCTION FIELD**

Operation: $IB \leftarrow N_8$

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.

CDF, CIF $62N3_8$ **CHANGE DF, IF**

Operation: $DF \leftarrow N_8$
$IB \leftarrow N_8$

Description: Combination of CDF and CIF.

RDF 6214_8 **READ DATA FIELD**

Operation: AC (6-8) $\leftarrow$ AC (6-8) V DF

Description: OR's the contents of DF into bits 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) $\leftarrow$ 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**
READ SAVE FIELD

Operation: AC (6-11) $\leftarrow$ 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

CIRCUIT DESCRIPTION

A good portion of the circuitry of the 6907-EMC is comprised of latches, used for the register functions and buffers used to gate these registers onto the Intercept DX lines. The remaining are used for generating load and clear controls to the registers, enable signals for the buffers and to form the discrete Interrupt Inhibit Flip-Flop (Figure 8-2).

Referring to the circuit schematic note that the two primary registers, the IF and DF are contained in devices 4D and 5B, respectively. Package 5D, a quad 2-input multiplexer, uses the DATAF line to select which register is used for the extended address lines EM0, EM1 and EM2. Additionally, the EM lines are forced to a high impedance state when DMAGNT is high.

The buffers to enable the IF and DF onto the Intercept bus are packages 6D and 6B, respectively. The Instruction Buffer register is device 5C. It feeds to the IF register and, along with the DF, it also goes to device 5A which is the SAVE FIELD register. Devices 4C and 6F are the instruction register. Gate 3F pin 11 produces a load pulse to this register at the time instructions are fetched from memory. Besides pertinent IOT instruction information, which could be recovered using LXMAR, this register also holds encoded information representing any main memory JMP or JMS instructions on 6F pin 5 and main memory directly referenced JMP or JMS instructions on 6F pin 15.

The Instruction Register bits IR3 through IR11 are either used directly or encoded somewhat to produce addresses to two bipolar programmable read only memories which do the bulk of the decoding necessary to produce the various control signals for the buffers and other registers. The PROM in position 4A, an IM5600, is used primarily to enable buffers for read operations, therefore, its chip select is active at XTC • $\overline{\text{DEVSEL}}$ time of the IM6100 timing scheme. This PROM has a 32 X 8 organization wherein 6 outputs are used for actual functional operations. The other PROM positioned at 4B is an IM5603 used for functions requiring an active signal during either the read or write cycle or for functions that benefit from its larger organization, which is 256 X 4. Notice here that $\overline{\text{XTC}}$ from 4F pin 4 is used as an address and that CS2 of the IM5603 is the ungated DEVSEL from the Intercept bus. Both of the PROMs are open collector types necessitating the resistor pullups in position 3A. All of the PROM outputs are considered "active-low". All outputs are kept glitch free by allowing only the chip selects to create an active output while maintaining stable addresses.

For the IM5600, output 01 enables device 6E which forces the lines DX0 through DX5 for all instructions that cause a read function. This line also forces the reads by pulling C1 low via device 5E. The data forced will be all zeros except when output 02 signals a GTF instruction which causes DX3 to reflect the state of the Interrupt Inhibit Flip-Flop. Output 02 also causes the GTF instruction to result in a "jam" type read by forcing C0 low via the output of 5E pin 12. Output 03 is used to clock the IF and clear the Inhibit Flip-Flop for the LIF instruction. It does this by forcing a positive on output 3E pin 12. The directly referenced JMP/JMS also causes this event at fetch time via the input at 3E pin 13. The circuitry associated with 3D pins 2 through 5 causes the "any" JMP/JMS value to be delayed until T1 (XTB • XTC of IM6100) of the second state following the fetch so that indirectly referenced JMP/JMS operations may obtain their indirect operands from the "old" Instruction Field but the actual "object" location of the JMS will be in the "new" Instruction Field.

A cross-coupled pair of NAND gates from 3F and 3E form the Interrupt Inhibit Flip-Flop which is active when the IB and IF differ. It disables Interrupt requests to the IM6100 by driving gate 5E pin 5 which pulls INTDIS LOW.

IM5600 output 04 enables the IF onto DX6-DX8 forcing DX9-DX11 low for the RIF instruction. Output 06 does the same function for the DF during a RDF instruction. Output 05 enables the SAVE FIELD onto the same DX lines for two different instructions. One is for the read portion of a GTF instruction. The other is for the execution of a RMF instruction wherein no C-line is forced, but the fields information is loaded back into the IB and DF by clocking them at the appropriate time.

These clocks are generated by the IM5603 at outputs 02 and 03. Not only do they occur together for the RMF instruction, but also for the RTF and the combined CIF CDF instruction. The RTF differs in that the operation takes place when XTC is low and the IM6100 is forcing the DX lines via its TTL buffers. The IB and DF are clocked independently on a CIF or a CDF, respectively. Again the IM5603 does the decoding and DEVSEL does the final timing via CS2.

For the CIF, CDF and their combination, which changes both registers, the data is embedded within the instruction. The IM5603 output 04 comes into use by gating IR6 to IR8 onto the proper DX lines, at "read time", for subsequent loading into the registers.

The final output 01 of the IM5603 is used to set the Inhibit Flip-Flop when an instruction causes the IB to differ from the IF.

One half of flip-flop pair 3D is used to generate signals for fixing the register values during the first cycle of an IM6100 Interrupt. The first edge of INTGNT clocks this flip-flop whose output on pin 9 causes the SF to be loaded from the IB and DF. Output $\overline{Q}$ from pin 8 concurrently causes the three registers IF, IB and DF to be cleared. The finite delay of this clear operation and the minimal hold time required by the SF assure the proper timing relationship and guarantees that the new IF value for the Interrupt cycle will be set to FIELD 0 by the time that a memory module evaluates the EM lines at the falling edge of LXMAR. The flip-flop is reset again with the first IFETCH to allow immediate programming of the registers if necessary.

CONCLUSION

The memory extension controller that we have discussed in this bulletin shows three important design considerations involved in extending memory addressing space. The first is the concept of having separate instruction and data fields for program flexibility. The second is the importance of double buffering the instruction field register to maintain structural integrity of programs and the third is the provision for saving the current field status upon interrupts and disabling interrupts until a change of instruction field has been completely executed.

The length of the field registers is limited only by the instructions that manipulate the contents of the field registers. For example, the instruction CDF-62N1$_8$ (Change Data Field to N$_8$) provides for only 3 bits (N$_8$) for the Data Field. The user, of course, has the option to have any subset or superset of these features for specific implementations.

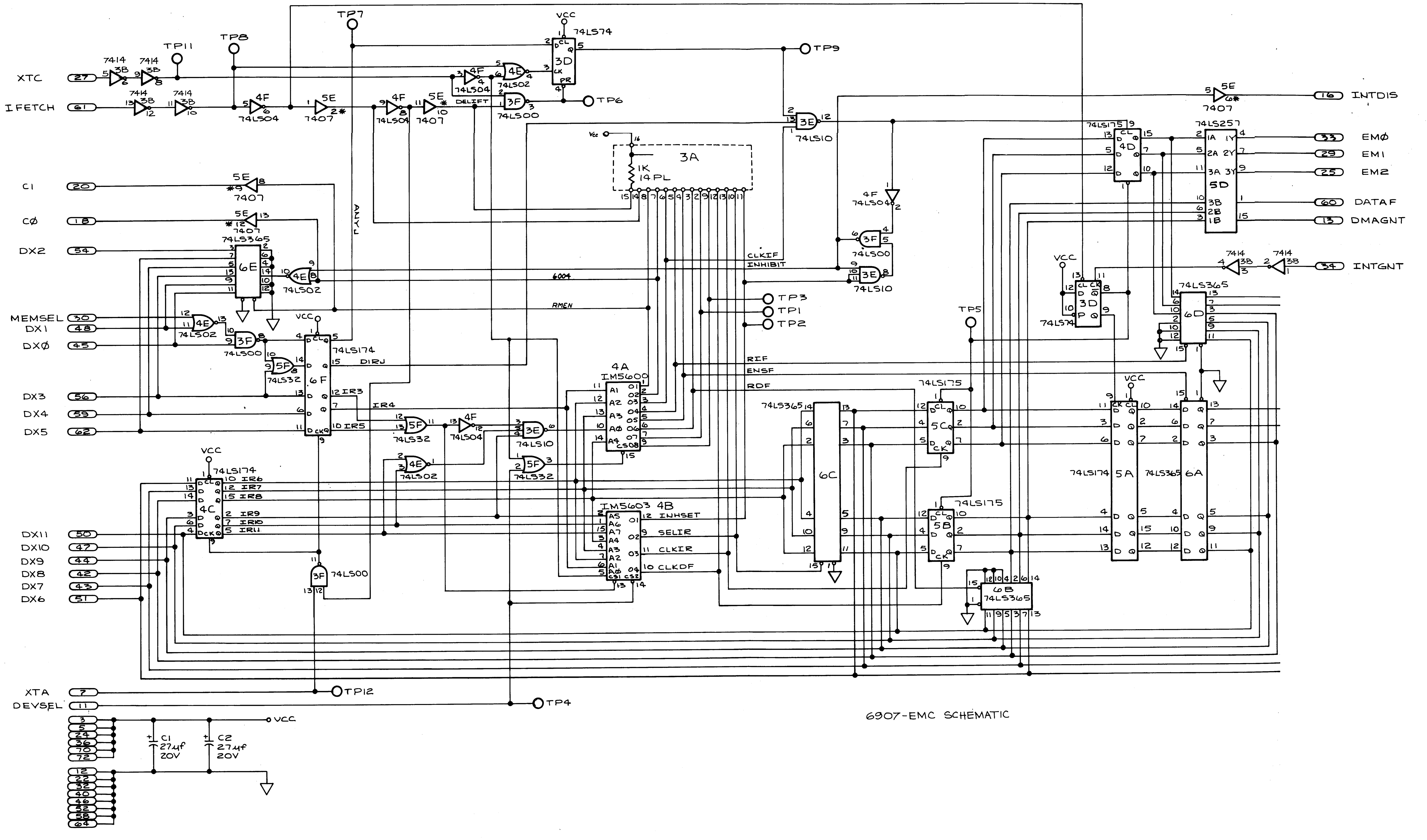


FIGURE 8-2
EXTENDED MEMORY CONTROLLER SCHEMATIC

CHAPTER 9
HARDWARE OPTIONS

The following additional hardware modules are available from Intersil.

6904-INTBUS

The 6904-INTBUS has the same structure as the INTERCEPT bus. The bus provides for eleven 72-pin 36-position connectors with 1-1/4" connector to connector spacing. The INTBUS is attached to the INTERCEPT with a 'paddle' card and flat cables. The power supply for the bus is user supplied (Appendix F).

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 6900-series card for servicing, testing and debugging.

6909-RRELAY

This module provides a means for remote control of the Teletype paper tape reader (Appendix F).

6970-IFDOS

The floppy disc operating system, designated 6970-IFDOS, together with the 4096 words of memory provided with INTERCEPT and an ASCII terminal (Teletype ASR33, or equivalent) enable the user to rapidly develop software for the 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 is rack mountable or can be 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 which provide 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 is 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 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 (A0-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 (DX0 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

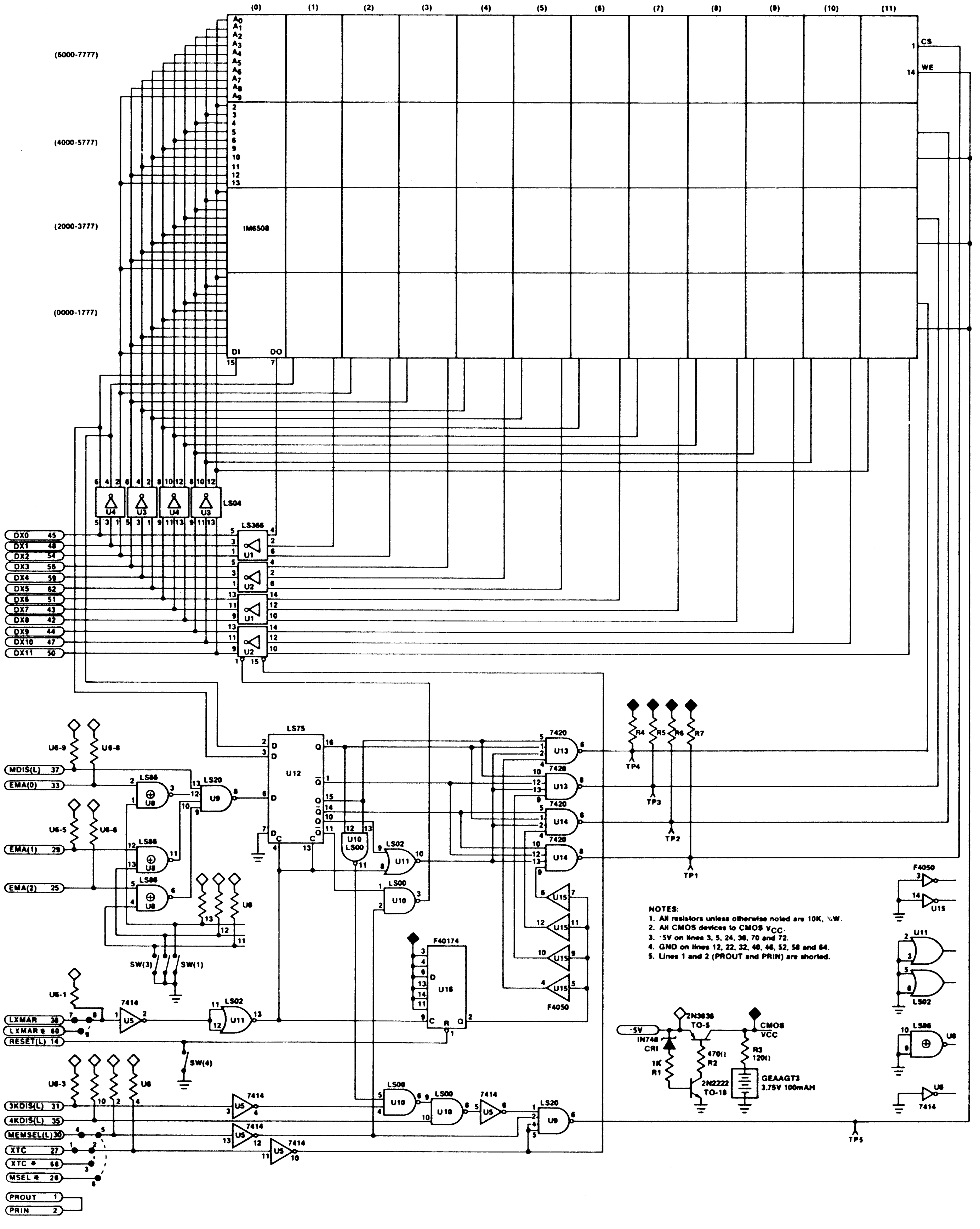
The maximum stand-by current drain of the 4K X 12 memory module at 25°C and 3.6 volts is 50 μA. With the 100 mAH Ni-Cad batteries used in 6901-M4KX12C module, the stand-by period is approximately 2000 hours (80 days). Since the battery storage capacity decreases and the device leakage current increases 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 MIN (ns)	TA = 25°C MAX (ns)
Access time from LXMAR (TAL)		390
Address set-up time (TAS)	50	
Address hold time (TAH)	115	

	VCC = 5.0V MIN (ns)	TA = 25^oC MAX (ns)
LXMAR pulse width (TL)	235	
Write pulse width (TWP)	200	
Write data set-up time (TDS)	150	
Write data hold time (TDH)	150	
Output enable time (TEN)	10	50
Output disable time (TDIS)	10	50
ICC at 250 KHz (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

APPENDIX A
BIN FORMAT

The BIN BOOT accepts tapes prepared with Digital Equipment Corporation PAL III, PAL D, PAL 8 or MACRO-8 assemblers and Intersil's FORTRAN/PAL III Cross Assembler and IFDOS PAL assembler. Diagnostic messages may be included on tapes. The BIN BOOT program resident in the INTERCEPT control panel is functionally identical to the DEC BINARY LOADER described in the DEC Utility Routine Manual, DEC-81-RZPA-D, and the "Introduction to Programming" handbook. However, unlike the DEC BIN LOADER, the BIN BOOT does not use any locations in the main memory and hence all of main memory is available for user programs.

EXTERNAL TAPE FORMAT

Tapes to be read by the BIN BOOT must be in binary-coded format and have about one foot of 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 the 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 BIN BOOT 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 microprocessor will halt after the tape has been loaded and the AC will be unequal to zero if the checksum error has occurred.

If the tape was started before the leader, the microprocessor will halt at the leader with AC equal to 7600 or 0000, depending on the number of blank characters read before the microprocessor halts.

MEMORY EXTENSION USAGE

The BIN BOOT may be used to load the binary tape into any valid memory field. If the memory extension controller is not used, the extended memory field instructions of the BIN BOOT are treated as "don't cares".

BIN BOOT PROGRAM

Refer to Appendix C for the listing of the BIN BOOT program.

The Program proceeds as follows: The incoming character is tested to see if it is a "rub-out" (all eight tape channels punched). If this is the case, all subsequent information coming from the reader is ignored until another rub-out is detected. This is the mechanism by which the assembler diagnostic messages are detected. They are preceded and followed by a single rub-out character. Within the diagnostic message any character is valid except, of course, a single rub-out character which would prematurely conclude the diagnostic message. Note that two consecutive rub-outs within the diagnostic message would, in effect, be ignored.

Next the 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 diagnostic message, 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 BIN BOOT 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 is the checksum.

APPENDIX B
ASCII CHARACTER CODES

CHARACTER CODES

8-bit ASCII CODE	6-bit CODE	CHARACTER REPRESENTATION	REMARKS
240	40		space (non-printing)
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 or acute accent
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

8-bit ASCII CODE	6-bit CODE	CHARACTER REPRESENTATION	REMARKS
300	00	@	at sign[1]
301	01	A	
302	02	B	
303	03	C	
304	04	D	
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 code, 00_8 represents CARRIAGE RETURN

(2) In IFDOS code, 37_8 represents TAB

CONTROL CODES

8-bit ASCII CODE	CHARACTER NAME		REMARKS
000	null		Ignored in ASCII input
200	leader/trailer		Leader/trailer code precedes and follows the data portion of binary files
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 Command Decoder and ODT
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, echoed as ↑O
225	CTRL/U		Delete current input line, echoes as ↑U
232	CTRL/Z		End-of-File character for all ASCII and binary files (in relocatable binary files CTRL/Z is not a terminator if it occurs before the trailer code)
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 C

6903-CONTRL PROGRAM LISTING

```
/INTERCEPT 6903B-3C,3D,3E CNTRL        IFDOS PAL 1A 04-JAN-77  PAGE 1

          /INTERCEPT 6903B-3C,3D,3E CNTRL
                /PROM LOCATIONS 7400-7777.  THE PROM ADDRESS IS
                      /COMPLIMENTED-  SEE LOGIC DIAGRAM
                /RAM LOCATIONS 0000-0017
                /PC DISPLAY IN 0020
                /DISPLAY IN 0040
                /FUNCTION SWITCHES IN 0060
                      /EXAM-DX0 DEP PC -DX1 DEP FLAGS- DX2
                      /DEP MEM- DX3 BINBOOT- DX4 USER -DX5
                /ROTARY SWITCH IN 0100
                      /MD-DX8 AC-DX9 MQ-DX10 FLAGS-DX11

                /RAM 0004-0017 MAY BE USED FOR USER FN

                /PROM LOCATIONS 7557-7624 ARE AVAILABLE
                /TO THE USER.
                 /IMPLEMENT USER FUNCTION. CURRENTLY
                 /THE  "USER FN " DECREMENTS PC BY 1 TO
                 /RESTORE PC FOR A "EXAM AND MODIFY"
                 /MEMORY FUNCTION.

                / THE TIME REQUIRED TO SERVICE A 30HZ
                 /REQUEST IS 200 MICROSECONDS AT 4MHZ.
                 /FOR CP ROUTINES TO FUNCTION PROPERLY
                 /WHEN THE HLT SW IS DOWN, AN EVEN NUMBER
                 /OF INSTRUCTIONS MUST BE EXECUTED IN
                 /THE CONTROL PANEL PROGRAM. "NOP" IS
                 /USED IN CERTAIN ROUTINES TO ENSURE THIS.

                /THE PROGRAM DEBOUNCES A FN SW CLOSURE AND
                 /RELEASE BY 63.5 MS AT 4 MHZ



       0020                PCLEDS=0020
       0040              DISLEDS=0040
       0060                  FNSW=0060
       0100                 ROTSW=0100


     *0000                *0000
                /RAM LOCATIONS

00000  0000  PC,     0000    /PC SAVED HERE BY CP REQ GRANT
00001  0000  AC,     0000    /SAVE AC
00002  0000  FLAGS,  0000    /L,0,IREQ,IIF,IENFF,0,<IF>,<DF>

00003  0000  EXEC,   0000    /SUBROUTINE ENTRY. ALSO TEMP
00004  0000          0000    /INIT TO CDF OR CIF
00005  0000          0000    /INIT TO JMP I EXEC

00006  0000  RDRSEL, 0000    /0000 FOR HS RDR: BINBOOT
```

```
/INTERCEPT 6903B-3C,3D,3E CNTRL        IFDOS PAL 1A 04-JAN-77  PAGE 1-1

00007  0000  BEGSW,  0000
00010  0000  RUBSW,  0000    /BINBOOT SWITCHES
00011  0000  RDRSW,  0000
00012  0000  CHAR,   0000    /BINBOOT SAVE LOCATIONS
00013  0000  WORD1,  0000
00014  0000  WORD2,  0000
00015  0000  CHKSUM, 0000    /BIN CHECK SUM ACCUMULATED
00016  0000          0000    /UNUSED
00017  0000          0000    /0003-0017 ARE AV TO USER


                /PROM PROGRAM

     *7777                *7777
07777  5776     JMP I 7776           /CPREQ ENTRY
     *7776                *7776
07776  7400     START                /CP ROUTINE ENTRY

     *7400                *7400

                        /SAVE AC AND FLAGS. SINCE GTF GETS SF
                        /AND NOT IF AND DF, FORM A COMPOSITE
                        /BY GTF,RDF AND RIF INSTRUCTIONS

07400  3001  START,  DCA AC  /SAVE AC
07401  6004          GTF     /L,0,IREQ,IIF,IENFF,0,<SF>
07402  0257          AND K7700       /MASK SF
07403  3002          DCA FLAGS
07404  7100          CLL
07405  6214          RDF     /OR IN DF INTO AC6-8
07406  7010          RAR
07407  7012          RTR     /DF IN AC 9-11
07410  6224          RIF     /OR IN IF INTO AC 6-8
07411  1002          TAD FLAGS       /L,0,IREQ,IIF,IENFF,0,IF0-2,DF0-2
07412  3002          DCA FLAGS

                        /ALL INDIRECT REFERENCES COME FROM DF FOR
                        /AND,TAD,DCA,ISZ INSTRUCTIONS. SINCE ONE
                        /IS INTERESTED IN THE IF LOCATIONS
                        /CHANGE DF INTO IF AND RESTORE DF ON EXIT

07413  6224          RIF
07414  1353          TAD K6201       /GET IF AND FORM CDF INSTR
07415  3004          DCA EXEC+1
07416  1355          TAD RET         /JMP I EXEC CONSTANT
07417  3005          DCA EXEC+2
07420  4003          JMS EXEC        /EXECUTE CDF INSTRUCTION

07421  3003          DCA EXEC        /INIT TO 0000 TO COUNT
                                     /DEBOUNCE DELAY
```

```
/INTERCEPT 6903B-3C,3D,3E CNTRL        IFDOS PAL 1A 04-JAN-77  PAGE 1-2

07422  1060          TAD FNSW
07423  0257          AND K7700
07424  7450          SNA
07425  5246          JMP HZ30                /CPREQ GENERATED BY
                                             /TIMER

07426  2003          ISZ EXEC        /63.5 MS DELAY
07427  5226          JMP .-1

                                             /ACTIVE FNSW WILL BE 1

07430  7004          RAL
07431  7430          SZL
07432  5313          JMP EXAM
07433  7510          SPA
07434  5323          JMP DEPPC

07435  7006          RTL
07436  7430          SZL
07437  5331          JMP DEPFLAGS
07440  7510          SPA
07441  5326          JMP DEPMEM

07442  7006          RTL
07443  7630          SZL CLA
07444  5756          JMP I BINBOOT
07445  5357          JMP USER



07446  1000  HZ30,   TAD PC
07447  3020          DCA PCLEDS

07450  1100          TAD ROTSW
07451  7012          RTR
07452  7500          SMA
07453  5304          JMP FLDIS
07454  7420          SNL
07455  5307          JMP MQDIS

07456  7012          RTR
07457  7700  K7700,  SMA CLA
07460  5311          JMP ACDIS

07461  7000  MDDIS,  NOP     /FOR EVEN INSTRUCTION SYNC
07462  7240          CLA CMA
07463  1000          TAD PC
07464  3003          DCA EXEC
```

```
/INTERCEPT 6903B-3C,3D,3E CNTRL        IFDOS PAL 1A 04-JAN-77  PAGE 1-3

07465  1403          TAD I EXEC

07466  3040  EXIT,   DCA DISLEDS     /INSTR THAT WAS JUST EXECUTED
                             /OR DATA THAT WAS JUST DEPOSITED


                             /RESTORE DF
07467  1002          TAD FLAGS
07470  7004          RAL
07471  7006          RTL
07472  0352          AND K0070
07473  1353          TAD K6201
07474  3004          DCA EXEC+1
07475  4003          JMS EXEC        /EXECUTE CDF TO RESTORE DF
07476  1002          TAD FLAGS
07477  7004          RAL                     /RESTORE LINK
07500  7200          CLA
07501  1001          TAD AC          /RESTORE AC
07502  6001          ION             /RESET CP INT FF
07503  5400          JMP I 0000      /EXIT
```

```
/INTERCEPT 6903B-3C,3D,3E CNTRL       IFDOS PAL 1A 04-JAN-77  PAGE 2

07504  7300  FLDIS,  CLA CLL
07505  1002          TAD FLAGS
07506  5266          JMP EXIT

07507  7701  MQDIS,  CLA MQA
07510  5266          JMP EXIT

07511  1001  ACDIS,  TAD AC
07512  5266          JMP EXIT


07513  7300  EXAM,   CLA CLL
07514  1100          TAD FOTSW

07515  7012          RTR
07516  7012          RTR
07517  7620          SNL CLA          /SKP IF NOT MD
07520  2000  ISZPC,  ISZ PC                    /MD EXAM, INCREMENT PC
07521  5341          JMP DEB
07522  5341          JMP DEB

07523  7604  DEPPC,  CLA OSR
07524  3000          DCA PC
07525  5341          JMP DEB

07526  7604  DEPMEM, CLA OSR
07527  3400          DCA I PC
07530  5320          JMP ISZPC



                          /RTF CANNOT BE USED TO RESTORE FLAGS
                          /SINCE IT WILL CAUSE CPINTFF TO BE RESET
07531  7604  DEPFLA, CLA OSR
07532  3002          DCA FLAGS
07533  1002          TAD FLAGS
07534  0352          AND K0070
07535  1354          TAD K6202
07536  3004          DCA EXEC+1
07537  4003          JMS EXEC         /EXECUTE CIF

                                      /THE EMA LOGIC INHIBITS
                          /IB TO IF TRANSFER IN CP MEM. THIS
                          /MUST BE DONE BY EXECUTING LIF (6254)
                          /WHICH LOADS IF DIRECTLY FROM IB
       6254               LIF=6254
07540  6254          LIF

/INTERCEPT 6903B-3C,3D,3E CNTRL       IFDOS PAL 1A 04-JAN-77  PAGE 2-1


07541  1060  DEB,    TAD FNSW
07542  0257          AND K7700
07543  7640          SZA CLA          /STAY IN LOOP UNTIL SW
07544  5341          JMP .-3          /IS RELEASED

07545  3003          DCA EXEC
07546  2003          ISZ EXEC
07547  5346          JMP .-1                   /DEBOUNCE

07550  7402          HLT      /EVEN NUMBER OF INSTRUCTIONS FROM HERE
07551  5246          JMP HZ30

07552  0070  K0070,  0070
07553  6201  K6201,  6201
07554  6202  K6202,  6202
07555  5403  RET,    JMP I EXEC
07556  7631  BINBOOT, BEGIN

07557  0000  USER,   0000     /USER LOCATIONS START HERE
```

```
/INTERCEPT 6903B-3C,3D,3E CNTRL       IFDOS PAL 1A 04-JAN-77  PAGE 3

       *7625                *7625
07625  7240                 CLA CMA         /DEFAULT USER FN. DECREMENT PC
07626  1000                 TAD PC          /BY 1
07627  3000                 DCA PC
07630  5770                 JMP I XDEB
                            /INTERCEPT ABSOLUTE BIN LOADER COMPATIBLE
                            /WITH DEC BIN



                            /INTERSIL ABSOLUTE BIN LOADER. PROGRAM COMPATIBLE
                            /WITH DEC BIN
                            /IF THE USER TAPE IS PROPRLY LOADED CPU WILL STOP
                            /WITH AC=0000.IF THE USER TAPE IS STARTED BEFORE
                            /LEADER THEN THE PROGRAM WILL STOP AT THE LEADER
                            /WITH AC=0000 OR 7600.


                            /SW(0)=1 FOR TTY ENTRY
                            /SW(0)=0 FOR HS RDR

                            /DEC PDP-8/E COMPATIBLE TTY AND HS RDR
                            /MNEMONICS
                             /KCC    -SET TTY RDR RUN
                             /RFC    -SET HS RDR RUN
                             /KSF    -SKP IF TTY CHAR RDY
                             /RSF    -SKP IF HSRDR CHAR RDY
                             /KRB    -AC(4-11) GETS TTY CHAR
                                              /SET TTY RDR RUN
                             /RRB RFC-AC(4-11) GETS HSRDR CHAR
                                              /SET HSRDR RUN



07631  6032  BEGIN,         KCC              /INIT TTY READER
07632  6014                 RFC              /INIT HS READER
07633  6224                 RIF      /GET IF
07634  1372                 TAD X6201
07635  3004                 DCA EXEC+1      /FORM CDF INSTR. DEFAULT

/INTERCEPT 6903B-3C,3D,3E CNTRL       IFDOS PAL 1A 04-JAN-77  PAGE 3-1

                                  /IS THE CURRENT IF
07636  7604                 CLA OSR          /READ SW REG
07637  3006                 DCA RDRSEL       /READER SELECT

                                  /THE "BEGG" ROUTINE MAY BE ENTERED
                                  /FROM "BEGIN" OR "GO" LOOP. BEGSW=7777
                                  /IF FROM BEGIN AND =0000 IF FROM GO.

07640  7040                 CMA
07641  3007                 DCA BEGSW

07642  3010  BEGG,          DCA RUBSW       /RUBSW=7777 FOR DIAGNOSTIC
                                             /MESSAGES PUNCHEDN BIN TAPE.



                                  /THE "READ" MAY BE ENTERED FROM "BEGG"
                                  /OR "GO". RDRSW=7777 IF FROM BEGG ELSE
                                  /IT IS = 0000.

07643  7040                 CMA
07644  3011                 DCA RDRSW

07645  1006  READ,          TAD RDRSEL
07646  7710                 SPA CLA
07647  5254                 JMP LO          /SW(0)=1 FOR TTY RDR

                                  /HS RDR ENTRY
07650  6011  HI,            RSF
07651  5250                 JMP .-1
07652  6016                 RRB RFC
07653  5257                 JMP SAV

                                  /TTY RDR ENTRY
07654  6031  LO,            KSF
07655  5254                 JMP .-1
07656  6036                 KRB

                                  /SAVE CHARACTER
07657  3012  SAV,           DCA CHAR
07660  1012                 TAD CHAR

                                  /CHECK SW FOR PROPER EXIT
07661  2011                 ISZ RDRSW
07662  5323                 JMP GO+5        /SW=0000 RETURN TO GO LOOP



                                  /CONTINUE BEGG LOOP
                                  /CHECK FOR RUB OUT
07663  1371                 TAD M376
07664  7750                 SPA SNA CLA     /AC=0001 FOR RUBOUT; SKIP
07665  5271                 JMP NORUB
```

```
/INTERCEPT 69Ø3B-3C,3D,3E CNTRL       IFDOS PAL 1A Ø4-JAN-77  PAGE 3-2

                        /RUB OUT ENTRY
Ø7666  2Ø1Ø  RUB,    ISZ RUBSW       /FIRST OR SECOND RUB
Ø7667  7Ø4Ø          CMA             /FIRST
Ø767Ø  5242          JMP BEGG        /SET RUBSW AND FETCH NEXT CHAR
                        /DATA ENTRY
Ø7671  1Ø1Ø  NORUB,  TAD RUBSW
Ø7672  764Ø          SZA CLA         /IGNORE DATA IF SW=7777
Ø7673  5243          JMP BEGG+1      /LEAVE RUBSW SET AND LOOK
                                     /FOR NEXT RUBOUT
                        /VALID DATA ENTRY POINT
Ø7674  1Ø12          TAD CHAR
Ø7675  Ø374          AND XØ3ØØ       /CH 7&8
Ø7676  1342          TAD M2ØØ        /AC<Ø IF DATA OR ORIGIN
Ø7677  751Ø          SPA             /SKIP IF L/T OR FIELD
Ø77ØØ  5313          JMP DAORG
Ø77Ø1  775Ø  MASKX,  SPA SNA CLA     /SKP IF FIELD. IF L/T AC=ØØØØ
Ø77Ø2  531Ø          JMP LT

                        /FIELD ENTRY POINT
Ø77Ø3  1Ø12          TAD CHAR
Ø77Ø4  Ø375          AND XØØ7Ø
Ø77Ø5  1372          TAD X62Ø1
Ø77Ø6  3ØØ3          DCA EXEC        /UPDATE EXEC SUBROINE
Ø77Ø7  5243          JMP BEGG+1      /FETCH NEXT CHARACTER

                        /LT EXIT
Ø771Ø  2ØØ7  LT,     ISZ BEGSW
Ø7711  5353          JMP END         /BEG ENTERED FROM GO AND HENCE
                                     /TRAILER. IBIN EXIT
Ø7712  524Ø          JMP BEGG-2      /BEGG ENTERED FROM BEGIN AND
                                     /HENCE LEADER. GO FETCH NEXT CHAR
                        /DATA OR ORIGIN EXIT
Ø7713  72ØØ  DAORG,  CLA
Ø7714  2ØØ7          ISZ BEGSW
Ø7715  5327          JMP GO+11       /ENTERED FROM GO ; RETURN

                        /CONTINUE BEGIN ENTRY


/INTERCEPT 69Ø3B-3C,3D,3E CNTRL       IFDOS PAL 1A Ø4-JAN-77  PAGE 4

Ø7716  3Ø15  GO,     DCA CHKSUM      /CHECK SUM CLEARED IF INITIAL
                                     /ENTRY.
Ø7717  1Ø12          TAD CHAR
Ø772Ø  3Ø13          DCA WORD1       /SAVE CHAR IN WORD1

Ø7721  3Ø11          DCA RDRSW       /SET UP RDR SW FOR ENTRY
                                     /FROM GO
Ø7722  5245          JMP READ        /PSEUDO JMS TO READ
                        /RETURN FROM READ
Ø7723  3Ø14          DCA WORD2

Ø7724  3ØØ7          DCA BEGSW       /SET UP BEGG SW FOR
                                     /ENTRY FROM GO

Ø7725  4ØØ3          JMS EXEC        /EXECUTE CDF

Ø7726  5242          JMP BEGG        /PSEUDO JMS TO BEGG
                        /CHARACTER LOOK AHEAD RETURN
                        /FROM BEGG IF NEXT CHAR IS NOT
                        /TRAILER.
Ø7727  1Ø13          TAD WORD1
Ø773Ø  71Ø6          CLL RTL
Ø7731  7ØØ6          RTL
Ø7732  7ØØ6          RTL
Ø7733  1Ø14          TAD WORD2
Ø7734  743Ø          SZL             /L=Ø IF DATA;=1 FOR ORIGIN
Ø7735  5344          JMP ORIGIN

                        /DATA ENTRY
Ø7736  34ØØ          DCA I ØØØØ      /DEPOSIT DATA IN MEM


                        /DIPLAY CODE

Ø7737  14ØØ          TAD I ØØØØ
Ø774Ø  3Ø4Ø          DCA DISLEDS     /DISPLAY DATA THAT
                                     /WAS JUST DEPOSITED



Ø7741  2ØØØ          ISZ ØØØØ        /UPDATE POINTER
Ø7742  76ØØ  M2ØØ,   76ØØ            /GROUP 2 CLA ALSO -2ØØ CONSTANT
Ø7743  741Ø          SKP
```

```
/INTERCEPT 69Ø3B-3C,3D,3E CNTRL       IFDOS PAL 1A Ø4-JAN-77  PAGE 4-1

                           /ORIGIN ENTRY
Ø7744  3ØØØ  ORIGIN, DCA ØØØØ          /UPDATE ORIGIN- NEW

                           /DISPLAY CODE
Ø7745  1ØØØ          TAD ØØØØ
Ø7746  3Ø2Ø          DCA PCLEDS        /DISPLAY PC

                           /CHECKSUM CALCULATION
Ø7747  1Ø13  CHEX,   TAD WORD1
Ø775Ø  1Ø14          TAD WORD2
Ø7751  1Ø15          TAD CHKSUM
Ø7752  5316          JMP GO  /UPDATE CHECK SUM AND CONTINUE
                                       /ON GO LOOP


                           /RETURN FROM BEGG IF NEXT CHARACTER
                           /ON LOOK AHEAD IS TRAILER
Ø7753  1Ø13  END,    TAD WORD1
Ø7754  7ØØ2          BSW
Ø7755  Ø3Ø1          AND MASKX         /SOMETIMES PAL PUNCHES CH 7
                                       /FOR PARITY . MASK OUT  BIT 11
                                       /AFTER BSW
Ø7756  1Ø14          TAD WORD2
Ø7757  7Ø41          CIA               /FETCH AND NEGATE CHECKSUM
                                       /FROM TAPE
Ø776Ø  1Ø15          TAD CHKSUM        /AND ADD TO CALCULATED
                                       /CHECKSUM. AC=ØØØØ IF OK


                           /BINBOOT EXIT
Ø7761  3ØØ1          DCA AC

                           /CHANGE IF TO DF
Ø7762  6214          RDF
Ø7763  1373          TAD X62Ø2
Ø7764  3ØØ4          DCA EXEC+1
Ø7765  4ØØ3          JMS EXEC
Ø7766  6254          LIF               /IE TO IF TRANSFER

Ø7767  577Ø          JMP I .+1
Ø777Ø  7541  XDEB,   DEB

Ø7771  74Ø2  M376,   HLT
Ø7772  62Ø1  X62Ø1,  62Ø1
Ø7773  62Ø2  X62Ø2,  62Ø2
Ø7774  Ø3ØØ  XØ3ØØ,  Ø3ØØ
Ø7775  ØØ7Ø  XØØ7Ø,  ØØ7Ø

                           /AUG 12 1976 THTH
/INTERCEPT 69Ø3B-3C,3D,3E CNTRL       IFDOS PAL 1A Ø4-JAN-77  PAGE 5

AC     ØØØ1    XØØ7Ø  7775
ACDIS  7511    XØ3ØØ  7774
BEGG   7642    X62Ø1  7772
BEGIN  7631    X62Ø2  7773
BEGSW  ØØØ7
BINBOO 7556
CHAR   ØØ12
CHEX   7747
CHKSUM ØØ15
DAORG  7713
DEB    7541
DEPFLA 7531
DEPMEM 7526
DEPPC  7523
DISLED ØØ4Ø
END    7753
EXAM   7513
EXEC   ØØØ3
EXIT   7466
FLAGS  ØØØ2
FLDIS  75Ø4
FNSW   ØØ6Ø
GO     7716
HI     765Ø
HZ3Ø   7446
ISZPC  752Ø
KØØ7Ø  7552
K62Ø1  7553
K62Ø2  7554
K77ØØ  7457
LIF    6254
LO     7654
LT     771Ø
MASKX  77Ø1
MDDIS  7461
MQDIS  75Ø7
M2ØØ   7742
M376   7771
NORUB  7671
ORIGIN 7744
PC     ØØØØ
PCLEDS ØØ2Ø
RDRSEL ØØØ6
RDRSW  ØØ11
READ   7645
RET    7555
ROTSW  Ø1ØØ
RUB    7666
RUBSW  ØØ1Ø
SAV    7657
START  74ØØ
USER   7557
WORD1  ØØ13
WORD2  ØØ14
XDEB   777Ø
```

```
/INTERCEPT 6903B-3C,3D,3E CNTRL       IFDOS PAL 1A 04-JAN-77  PAGE 6

NO ERRORS DETECTED

NO LINKS GENERATED

59 SYMBOLS

4K MEMORY UTILIZED
```

APPENDIX D

TELETYPE MODIFICATIONS FOR THE INTERCEPT SYSTEM

The Intersil INTERCEPT systems have been deisgned to 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 D-1 to locate the various parts located below.

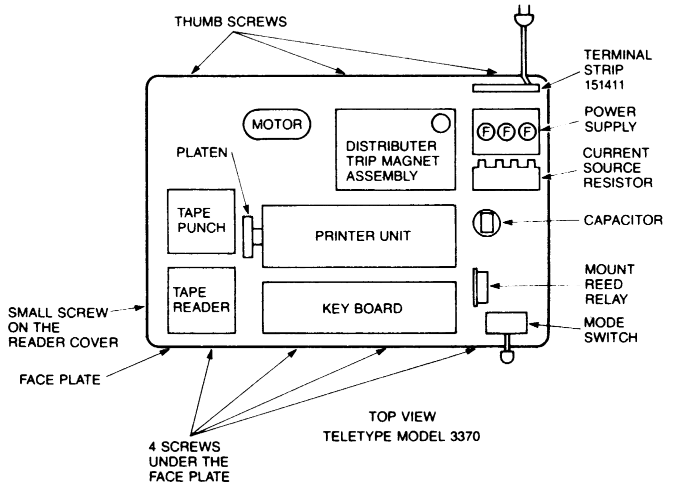


FIGURE D-1

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 accomplished by moving the BLUE wire from Terminal #3 to Terminal #4 of the large power resistor shown in Figure D-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 D-3 with Terminal #1 at 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 D-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 D-6. Locate the BROWN wire coming from the Distributor Trip Magnet which is connected to terminal J4 - Pin 11 as shown in Figure D-5. Cut this 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 D-6. A preassembled Reader Relay Card is available from Intersil Inc., Model # 6909-RRELAY.

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 D-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 CPUTTY OUTPUTS TO THE TELETYPE

The TTY outputs of the CPUTTY board are connected to Terminal Strip 151411 and the relay as shown in Figures D-6 and D-8.

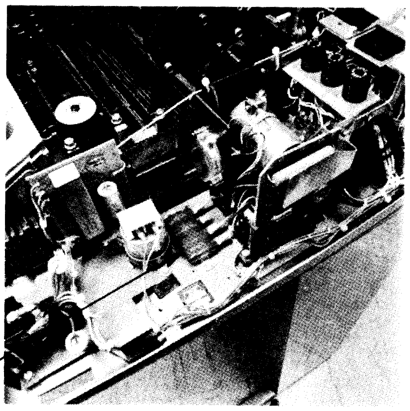


FIGURE D-2
CURRENT LOOP RESISTOR

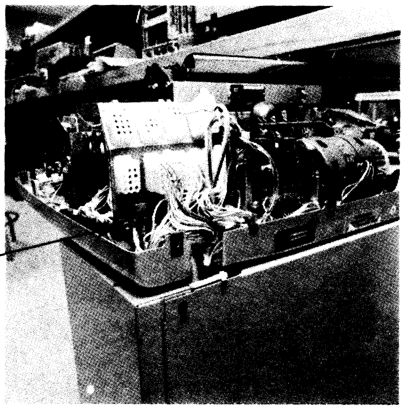


FIGURE D-3
TERMINAL STRIP

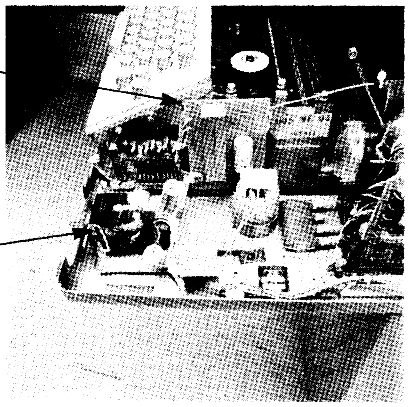


FIGURE D-4
RELAY CARD

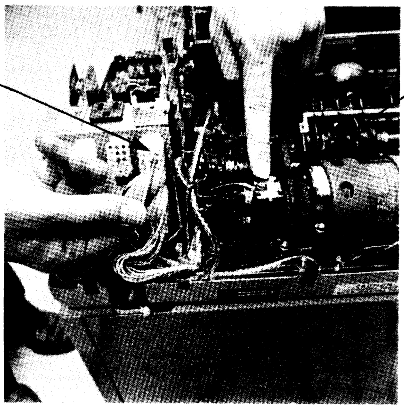


FIGURE D-5
DISTRIBUTOR TRIP MAGNET

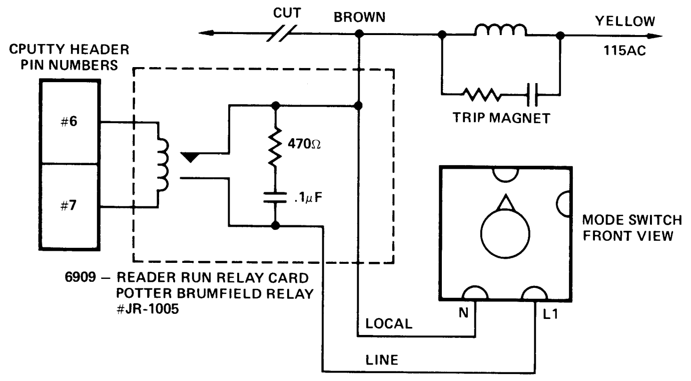

CUT
BROWN
YELLOW
115AC
CPUTTY HEADER PIN NUMBERS
#6
#7
470Ω
.1μF
TRIP MAGNET
MODE SWITCH FRONT VIEW
6909 – READER RUN RELAY CARD POTTER BRUMFIELD RELAY #JR-1005
LOCAL
N
L1
LINE

FIGURE D-6

READER RELAY CIRCUIT

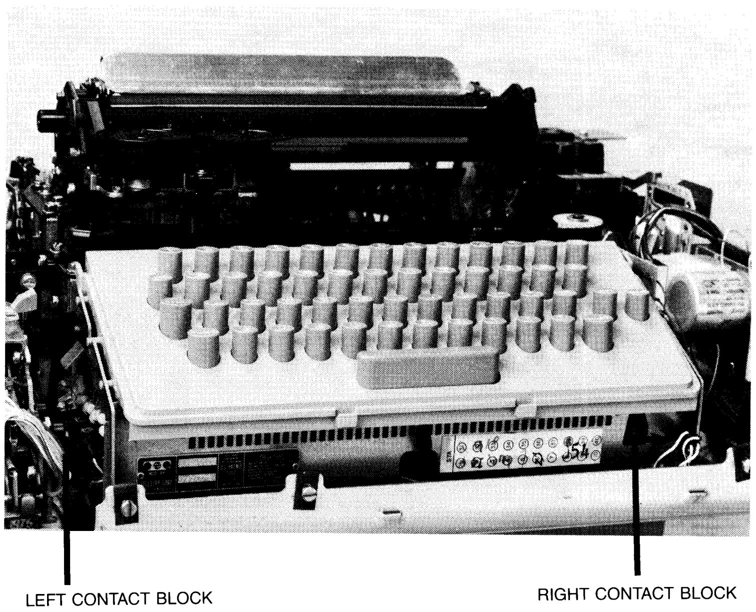


FIGURE D-7

LEFT AND RIGHT CONTACT BLOCKS

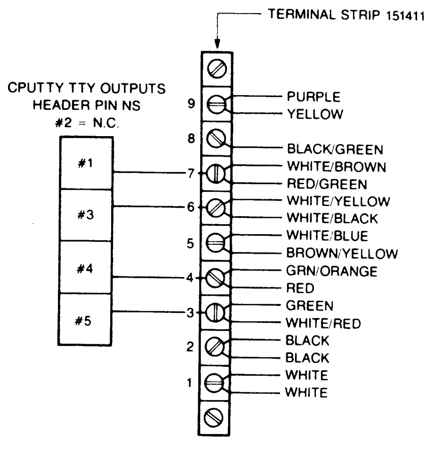


FIGURE D-8

TELETYPE CONNECTION DIAGRAM

APPENDIX E
ROM BASED SUBROUTINE CALLS WITH THE IM6100

Frequently the same or similar sequence of instructions must be executed in different parts of a program. There are obvious advantages to writing a program in which the identical piece of code is written only once and each time it is used in the main part of the program, the program flow is changed to execute the code. The piece of code 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 make "calls" to the same subroutine.

IM6100 SUBROUTINE CALL

In the IM6100, the JMS, Jump to Subroutine, instruction is used to eliminate the need for writing the complete set of instructions each time an intermediate task must be performed, be it finding a square root or typing a character on the Teletype. Since the IM6100 is designed to be program compatible with the DEC PDP-8/E it uses the same convention as the PDP-8 for subroutine linkage which is to store the "return" address in the first location of the called subroutine. After the subroutine code has been executed, a return transfer is made by jumping back "indirectly" through the first location of the subroutine. Thus, the programmer has a simple means of exiting and for returning to the correct location of the program upon completion of the task.

This convention, though extremely simple and straightforward, has two drawbacks, the first drawback being 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. It is quite possible that one subroutine may call another. The IM6100 linkage mechanism is applicable in this case. However, there are instances, when a subroutine may call itself over and over, recursively. Obviously, the simple linkage mechanism will not work since a call to itself will destroy 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.

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 OF BEING CALLED FROM 5013

                        *5013
5013    4170            JMS 0170
                        *0170
0170    0000            0000            /RETURN ADDRESS
0171    5572            JMP I .+1       /ENTER SUBROUTINE THRU
0172    6600            6600            /RAM LOCATION 0170

                             /LOCATIONS 171 & 172 MUST
                             /BE INITIALISED AT POWER ON

                        /EXIT FROM SUBROUTINE
                        *6676           /LAST INSTRUCTION
6676    5570            JMP I 0170      /RETURN VIA 0170
```

Execution times:

CALL 13μs at 4 MHz
RETURN 7.5μs at 4 MHz

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 the subroutine return addresses into a stack memory which may be part of the CPU chip or part of the external memory.

When the 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. In view of the fact that most microprocessor applications involve some amount of external RAM, the external RAM stack solution is achieving wider acceptance. 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. In the following section we discuss a specific software implementation of a stack oriented 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 implements the linkage 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.

Execution times:

CALL 70μs at 4 MHz
RETURN 54μs at 4 MHz

Fixed memory overhead for CALL and RETURN:

6 RAM locations in Page Zero, four of which must be initialized at power-on.
29 ROM locations, 17 for routines and 12 for power-on initializing.

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
```

PROGRAM LISTING:

```
                        /SOFTWARE STACK ROUTINES FOR IM6100

                        /RAM LOCATIONS IN PAGE ZERO

                        *162

0162  0000  CALLX,      0000            /ENTRY POINT FOR "CALL" ROUTINE
0163  5564              JMP I .+1       /GO TO "CALL" IN ROM
0164  7400              CALLY           /START OF "CALL" IN ROM

0165  7411  RETX,       RETY            /POINTER TO "RETURN" ROUTINE IN ROM

0166  0170  STACK,      .+2             /CURRENT STACK POINTER. INIT TO
                                        /0170 BY POWER-ON ROUTINE
0167  0000  AC,         0000            /TEMPORARY LOC FOR AC

                        /THE LOCATIONS CALLX+1,CALLX+2,RETX AND
                        /STACK MUST BE INITIALISED AT POWER-ON.



                        /ROM LOCATIONS

                        *7400

7400  3167  CALLY,      DCA AC          /SAVE AC
7401  2166              ISZ STACK       /UPDATE STACK POINTER

7402  1162              TAD CALLX       /CALLX HAS RETURN ADDRESS
7403  7001              IAC             /INCREMENT BY 1 TO SKIP OVER
7404  3566              DCA I STACK     /ENTRY ADDRESS OF USER SUBROTINE
                                        /AND SAVE ON STACK
7405  1562              TAD I CALLX     /GET USER ROUTINE ENTRY ADDRESS
7406  3162              DCA CALLX       /AND PUT IT IN CALLX

7407  1167              TAD AC          /RESTORE AC
7410  5562              JMP I CALLX     /GO TO USER  SUBROUTINE



7411  3167  RETY,       DCA AC  /SAVE AC
7412  1566              TAD I STACK     /GET RETURN ADDRESS FROM STACK
7413  3162              DCA CALLX       /AND PUT IT IN CALLX

7414  7060              CMA CML         /AC=7777; COMPLEMENT LINK
7415  1166              TAD STACK       /STACK POINTER-1; RESTORE LINK
7416  3166              DCA STACK       /UPDATE STACK POINTER

7417  1167              TAD AC          /RESTORE AC
7420  5562              JMP I CALLX     /RETURN



                        *7600
7600  1372  INIT,       TAD JMPI
7601  3163              DCA CALLX+1
7602  1373              TAD KCALLY
7603  3164              DCA CALLX+2
7604  1374              TAD KRETY
7605  3165              DCA RETX
7606  1375              TAD BASE
7607  3166              DCA STACK
                                /CONTINUE WITH REST OF SYSTEM POWER-ON
                                /INITIALISE
                        *7772
7772  5564  JMPI,       JMP I CALLX+2
7773  7400  KCALLY,     CALLY
7774  7411  KRETY,      RETY
7775  0170  BASE,       STACK+2

                        *7776
7776  7600              7600            /START OF INIT ROUTINES
7777  5776              JMP I 7776      /RESET STARTING



                        /EXAMPLE OF USER PROGRAM CALLING A SUBROUTINE
                        /IN LOCATION 6600 FROM LOCATION 5013

                        CALL= JMS CALLX
                        *5013

5013  4162              CALL
5014  6600              6600            /SUBROUTINE STARTS AT 6600

                        /EXAMPLE OF A SUBROUTINE EXIT AT LOCATION 6676

                        RETURN= JMP I RETX
                        *6676

6676  5565              RETURN
```

CONCLUSION

The two different approaches for ROM based subroutine calls are summarized in Table E-1.

TABLE E-1

	Fixed Overhead		Overhead for Each Active Call	Overhead for Each Subroutine in the Program		Execution Time at 4 MHz	
	RAM	ROM	RAM	RAM	ROM	CALL	RETURN
ALL RAM SYSTEM	0	0	0	1	0	5.5/8.0*	7.5
LINKAGE THRU RAM	0	0	0	3	6	13.0	7.5
SOFTWARE STACK	6	29	1	0	0	70.0	54.0

* 8.0μs if the subroutine is not in the Current Page

If the program has more than four subroutines, the memory overhead requirements for the RAM linkage technique exceeds the fixed overhead for the software stack. However, directly linking through RAM is six times faster than what could be achieved with the software stack, and it is only slightly slower than the optimum. The software stack is completely general purpose and the memory overhead is small. The performance penalty is not significant if subtask execution times exceed 1 ms which is the typical IM6100 execution time for a software multiply or divide at 4 MHz. The user must, of course, choose the appropriate method, depending on the speed and memory requirements for a specific task.

APPENDIX F

USER INTERFACES ON THE INTERCEPT BUS

1. The user interface must input buffer the following critical signals with 7414 (Hex Schmitt inverter) or 74132 (Quad 2-input Schmitt NAND gate) for better noise immunity.

   DMAGNT
   XTA
   XTB
   XTC
   IFETCH
   INTGNT
   UP
   DATAF

2. When using the 6904-INTBUS, the Intercept Power Supply must be disconnected. The Intercept power supply has a rated capacity of 2 amps at 5 volts. The 6901-M4KX12, 6902-CPUTTY and 6903-CONTRL modules use 1.5 amps at 5 volt. The user must supply power to these modules with an external power supply when the Intercept bus is extended.

3. Pins 1 and 2 of the Intercept bus are daisy chained to be used for priority vectoring. The priority is established by the position of the interface on the bus. The priority is as follows, with the highest priority first:

Intercept:	Left top	(6902-CPUTTY)
	Left bottom	(6903-CONTRL)
	Right top	(normally 6901-M4KX12)
	Right bottom	(normally 6908-IFDOS Interface/6904-INTBUS paddle card)

   On the 6904-INTBUS, the connector closest to the 3M connectors has the highest priority.

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

   This note applies only to the Intercept motherboard labelled 6900-INTBUS (January 1977).

4. Pins 8, 39, 57 and 65 are spares.

   The following eight lines are left open on the paddle card since they are involved only in the CPU-control panel communications. The user may connect them up by the jumpers provided on the paddle card.

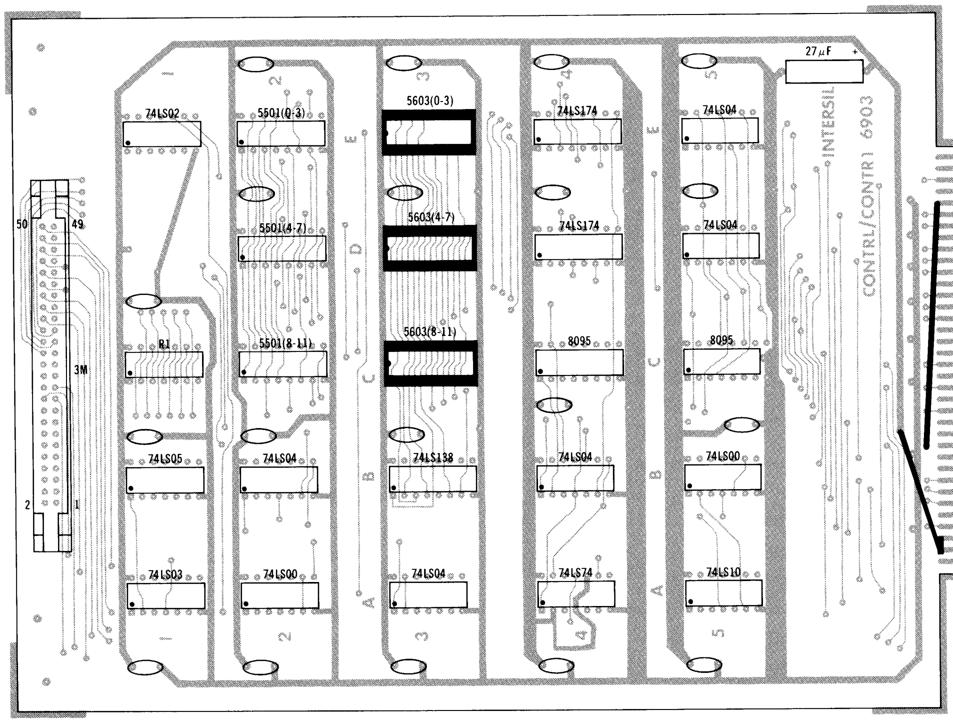
Pin 15	CPREQ (L)
Pin 26	RUN/HLT (L)
Pin 53	RUN (L)
Pin 55	CPSEL (L)
Pin 63	SINGLE CLOCK (H)
Pin 66	SWSEL (L)
Pin 67	FREE RUN (H)
Pin 68	LINK (L)

5. Recommended address assignments for the IM6101-PIE (Peripheral Interface Element) are as follows:

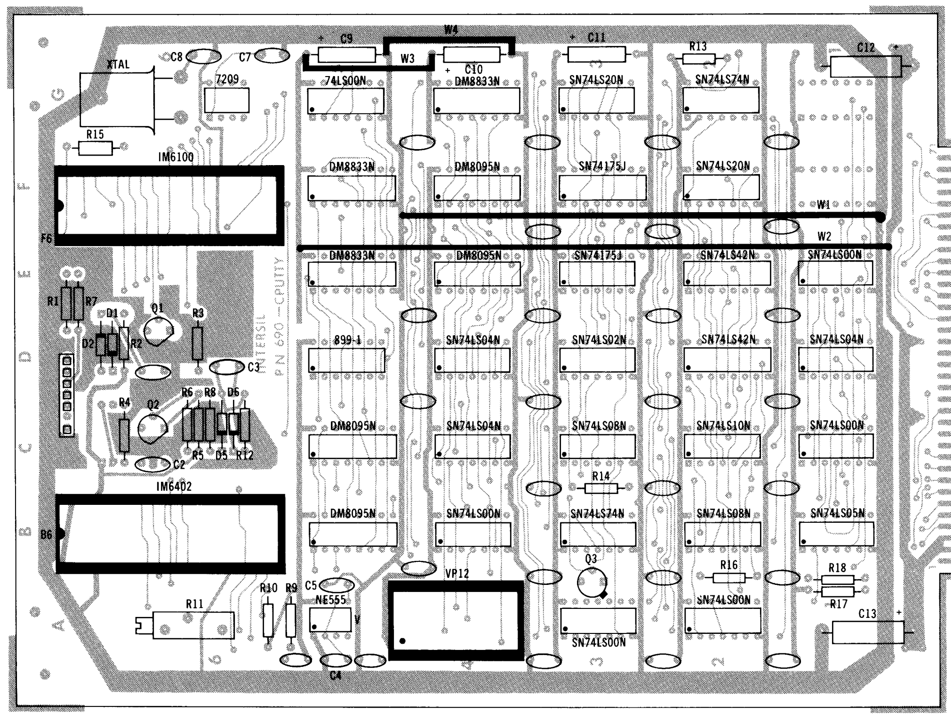
000	00	Internal IOT (600X) and DEC HS RDR (601X)
000	01	DEC HS PUNCH (602X) and DEC TTY Keyboard (603X)
000	10	DEC TTY PRINTER (604X)
000	11	INTERCEPT PIE-UART Serial Interface
001	00	INTERCEPT PIE-UART PRINTER Interface
001	01	IM6102-MEDIC REAL TIME CLOCK
001	10	Reserved for Intercept Option -1
001	11	Reserved for Intercept Option -2
010	00	IM6102-MEDIC EMC/DMA
010	01	IM6102-MEDIC EMC/DMA
010	10	IM6102-MEDIC EMC/DMA
010	11	IM6102-MEDIC EMC/DMA
011	00	IM6103-PIP
011	01	IM6103-PIP
011	10	IM6103-PIP
011	11	IM6103-PIP
100	00	USER
100	01	USER
100	10	USER
100	11	USER
101	00	USER
101	01	USER
101	10	USER
101	11	USER

110	00	USER
110	01	USER
110	10	USER
110	11	USER
111	00	Reserved for Intercept Option -5
111	01	Reserved for Intercept Option -4
111	10	Intercept FLOPPY DISK System (675X)
111	11	Reserved for Intercept Option -3

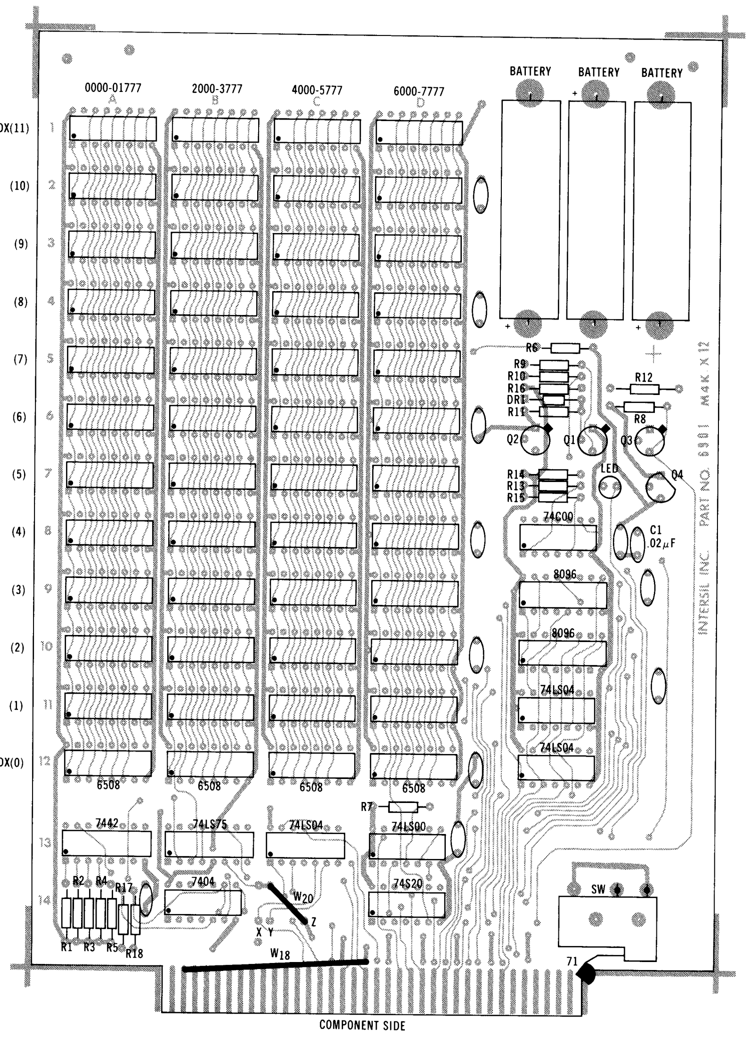
APPENDIX G
ASSEMBLY DRAWINGS



6903-CONTRL LOGIC
Rev. B



6902-CPUTTY
Rev. B



6901-M4KX12
Rev. C

APPENDIX H
LIST OF MATERIALS

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Req.	Qty Iss'd	Qty Short	Remarks
1	15-6901	6901-M4KX12 Assy Rev. F		1				
2	15-6902	6902-CPUTTY Assy Rev. C		1				
3	15-6903	6903-CONTRL LOGIC Assy Rev. C		1				
4	15-0191	INTERCEPT CONTRL Assy Rev. B		1				
5	15-0192	INTERCEPT BUS Assy Rev. C		1				
6	15-0193	INTERCEPT Power Supply Assy Rev. A		1				Deltron 8835X (Q5-3)
7	15-0135	Base Rev. B		1				
8	15-0136	Bracket, Support Rev. A		2				
9	15-0137	Bracket, Chassis Rev. B		4				
10	15-0138	Air Vent Rev. A		1				Made from perforated blank 15-0139
11	15-0132	Enclosure, Fiberglass Rev. A		1				
12	15-0155	Switch, Rocker C&K 7101-J51-2-Q-Black		1				
13	15-0148	Feet H.H.Smith #2135		4				
14	15-0149	Card guides Waldom E650		8				
15	15-0146	Line Cord Belden 17239		1				Pacific Electricord 2112-008-BL
16	15-0147	Strain Relief H.H.Smith 939		1				
17	15-0158	Screw, Thumb H.H.Smith 2366		4				
18	15-0159	Screw, Machine 4-40 x 1/4 Pan		18				

No. 6900 Descr. INTERCEPT System
Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Req.	Qty Iss'd	Qty Short	Remarks
19	15-0160	Screw, Machine 4-40 x 3/8 Pan		8				
20	15-0161	" " 4-40 x 1/4 Flat Head		4				
21	15-0166	Screw, " 8-32 x 3/4 R.H.		4				
22	15-0164	" " 8-32 x 1/4 R.H.		2				
23	15-0163	" " 6-32 x 3/8 Pan		4				
24	15-0150	Washer, Nylon # 4 Flat		8				H.H. Smith 2671
25	15-0169	Nut, 'KEP' 4-40 x 1/4		2				
26	15-0170	Standoff, 8-32 x 1/2 M/F		2				H.H.Smith 8282
27	15-0195	TTY Cable Assy Internal Rev. A		1				
28	15-0194	TTY Cable Assy External Rev. A		1				
29	15-0151	Graphic Overlay Pnl left		1				
30	15-0152	Graphic Overlay Pnl center		1				
31	15-0153	Graphic Overlay Pnl right		1				
32	15-0154	Label, Serial/Model		1				G.M. Nameplate 35538
33	15-0125	Connector, 25 pin Male AMP 205208-1		2				
34	15-0173	Contact, Socket AMP 1-66506-0		50				
35	15-0172	Connector, 25 pin Female AMP 205207-1		2				
36	15-0174	Contact, Pins AMP 1-66504-0		50				

No. 6900 Descr. INTERCEPT System
Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
37	15-0176	Screwlock Kit AMP 205 817-1		3				
38	15-0182	Terminal, Spade Waldom T-2717		2				
39	15-0181	Connector, Parallel Crimp		1				Waldom N5105S
40	15-0177	Connector, Receptacle Molex 03-06-1032		1				
41	15-0179	Contact, Sockets Molex 02-06-1103		2				
42	15-0190	Cable Tie Clamp		2				

No. 6900 Descr. INTERCEPT System

Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
1	15-0086	Printed Circuit Board 6901-		1				
2	15-0070	Socket, 16 pin DIP	1A-12D	48				
3	15-0081	Switch, Snap Cherry		1				E-61-00K
4	15-0001	Battery, 475 mAH Ni-Cad AA size		3 ✓				
5	15-0014	I.C. 7442 Decimal Decoder	13A	1				Must be "TTL"
6	15-0016	" 74LS75 Quad DFF	13B	1				
7	15-0024	" 74LS04 Hex Inverter	13C, 11E,12E	3				11E and 12E must be "LS" devices
8	15-0021	" 74LS00 Quad 2-in NAND	13D	1				
9	15-0012	" 74S20 Dual 4-in NAND	14D	1				Must be "S"
10	15-0011	" 74C00 Quad 2-in NAND CMOS	8E	1				Must be CMOS
11	15-0009	" DM8096 Tristate Hex Inverter	9E,10E	2				National
12		" IM6508 1K X 1 CMOS RAM	1A-12D	48				Intersil
13	15-0046	Resistor, 10K Ohm 10% 1/4W	R1-R7,R13, R15,R17,R18	11				
14	15-0043	" 1K Ohm 10% 1/4W	R8	1				
15	15-0039	" 180 Ohm 10% 1/4W	R9,R11	2				
16	15-0047	" 15K Ohm 10% 1/4W	R10	1				
17	15-0036	" 68 Ohm 10% 1/4W	R12	1				
18	15-0041	" 680 Ohm 10% 1/4W	R14	1				

No. 15-6901A Descr. M4KX12 CMOS MEMORY

Rev. D

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
19	15-0037	Resistor, 100 Ohm 10% 1/4W	R16	1				
20	15-0005	Diode, Zener 3.9V IN748A	DR1	1				
21	15-0003	Transistor, NPN 2N2222	Q1,Q2,Q3	3				
22	15-0004	" PNP 2N3638	Q4	1				
23	15-0002	L.E.D., T-I		1				H-P 5082-4484 T-I Discrete LED
24	15-0050	Capacitor, 0.2μfd	C1	1				
25	15-0049	" 0.1μfd		13				Bypass
26	15-0286	I.C. 7414	14B	1				
27	15-0053	Capacitor 27μfd 20V		1				
28	15-0110	Screw #2-56 x 1/2		2				
29	15-0111	Nut #2-56		2				
30	15-0113	Washer, Lock #4 Int		2				
31	15-0150	Washer, Nylon #4 Flat		2				

No. 15-6901 A Descr. M4KX12 CMOS MEMORY

INTERSIL 10900 N. Tantau Avenue INCORPORATED Cupertino, California 95014

Rev. D

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
1	15-0086	P. C. Board 6901B		1				
2	15-0070	Socket, 16 pin DIP		48				
3	15-0204	Switch, Multi Section DIP	Sw 1	1				CTS 206-4
4	15-0205	Battery, 3.75 v. NiCad AA size		1				GE 03A1/3AA-GT3
5	3N169	MOSFET 3N169-IT1750	Q3	1				Intersil
6	15-0003	Transistor, 2N2222 NPN	Q2,Q4,Q5	3				
7	15-0004	Transistor, 2N3638 PNP	Q1	1				
8	15-0005	Diode, Zener 3.9v. 1N748A	DR1	1				
9	15-0002	LED, T1 descrete	CR2	1				HP-5082-4484
10	IM6508	RAM, 1Kx1 CMOS		48				Intersil
11	15-0014	IC 7442	U12	1				Do not substitute "LS"
12	15-0016	IC 74LS75	U11	1				
13	15-0021	IC 74LS00	U10	1				
14	15-0024	IC 74LS04	U3, U4,	2				
15	15-0012	IC 74S20	U9	1				Do not substitute "LS"
16	15-0011	IC 74C00	U14	1				
17	15-0272	IC 74LS366	U1,U2	2				
18	15-0199	IC 74LS86	U6	1				

Board No. 6901 B Descr. M4KX12 CMOS MEMORY

INTERSIL 10900 N. Tantau Avenue INCORPORATED Cupertino, California 95014

LIST OF MATERIALS

REV. F

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
19	15-0037	Resistor, 100 Ohm 10% 1/4W	R3	1				
20	15-0219	Resistor, 130 Ohm 10% 1/4W	R7	1				
21	15-0039	Resistor, 180 Ohm 10% 1/4W	R2, R4	2				
22	15-0041	Resistor, 680 Ohm 10% 1/4W	R8	1				
23	15-0043	Resistor, 1 K Ohm 10% 1/4W	R6	1				
24	15-0248	Resistor, 2.4K Ohm 10% 1/4W	R1	1				
25	15-0047	Resistor, 15K Ohm 10% 1/4W	R5	1				
26	15-0031	Res. Network, 10K DIP	U13, U5	2				Beckman 899-1-R10K
27	15-0049	Capacitor, .1 ufd	C4-C46	43				
28	15-0050	Capacitor, .2 ufd	C1	1				
29	15-0053	Capacitor, 27 ufd 20v.	C2,C3,C47	3				
30	15-0237	Terminal, Swage Turret	TP1-TP13	13				Keystone #1526
31	15-0286	I.C. 7414	U7, U8	2				

Board No. 6901B Descr. M4KX12 CMOS MEMORY

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

REV. F

LIST OF MATERIALS

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
1	15-0035	Resistor, 27 Ohm 1/4W 10%	R7	1				
2	15-0037	" 100 Ohm	R4,R5,R15	3				
3	15-0038	" 120 Ohm	R1,R2	2				
4	15-0040	" 470 Ohm	R6	1				
5	15-0042	" 820 Ohm	R8	1				
6	15-0043	" 1.0 K	,R13, R16,R17	3				
7	15-0092	" 2.2 K	R14	1				
8	15-0044	" 5.6 K	R3	1				
9	15-0045	" 6.8 K	R12	1				
10	15-0046	" 10 K	R9,R10,R18	3				
11								
12	15-0030	Resistor, Network Beckman 899-1-R1.0K	5D	1				
13	15-0033	Resistor, Variable 50K	R11	1				Dale 784-50K
14	15-0083	Capacitor, .001μfd	C2	1				
15	15-0048	" .01μfd	,C5	1				Bypass
16	15-0049	" .1μfd	C1	29				Bypass
17	15-0051	" .47μfd	C3	1				
18	15-0053	" 27μfd	C9-C13	5				

No. 15-6902 Descr. CPUTTY

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Rev. C

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
19								
20	15-0131	Crystal 4.00 MHz	XTAL	1				15-0130 3.33 MHz
21	15-0142	Capacitor, .01µfd 15% TC	C4	1				
22	15-0004	Transistor, 2N3638	Q1,Q2	2				
23	15-0003	" 2N2222	Q3	1				
24	15-0007	Diode, 1N914	D1,D2,D5 D6	4				
25	15-0143	Capacitor, 10pfd	C7,C8	2				
26	15-0021	I.C. 74LS00	1C,1E,2A, 3A,4B,5G	6				
27	15-0022	" 74LS02	3D	1				
28	15-0023	" 74LS03	2B	1				
29	15-0024	" 74LS04	4C,4D	2				
30	15-0025	" 74LS05	1B	1				
31	15-0026	" 74LS08	3C	1				
32	15-0027	" 74LS10	2C	1				
33	15-0028	" 74LS20	2F,3G	2				
34	15-0013	" 74LS42	2D,2E	2				
35	15-0015	" 74LS74	2G,3B	2				
36	15-0018	" 74LS175	3E,3F	2				

No. 15-6902 Descr. CPUTTY

Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
37	15-0008	I.C. DM8095	4E,4F 5B,5C	4				National
38	15-0010	" DS8833	4G,5E,5F	3				National
39		" NE555	5A	1				Intersil
40		" ICM7209 CMOS Crystal Oscill.	6G	1				Intersil
41		" IM6402 UART	6B	1				Intersil
42		" IM6100	6F	1				Intersil
43	15-0020	VP12 DC-DC Converter +5 to -12	4A	1				
44	15-0286	I.C. 7414	1D	1				
45	15-0089	Printed Circuit Board 6902		1				
46	15-0069	Socket, 24 pin DIP	x4A	1				
47	15-0068	Socket, 40 pin DIP	x6B,x6F	2				
48	15-0134	Connector, Rt. Angle 7 pin Molex 22-12-2071		1				

No. 15-6902 Descr. CPUTTY

Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
1	15-0090	Printed Circuit Board 6903		1				
2	15-0129	Connector, 3M-3433-2002		1				
3	15-0030	Resistor, Network Beckman 899-1-R1.0K	1C	1				
4	15-0049	Capacitor, .1μfd		19				Bypass
5	15-0053	" 27μfd	C1	1				
6	15-0021	I.C. 74LS00	2A,5B	2				
7	15-0022	" 74LS02	1E	1				
8	15-0023	" 74LS03	1A	1				
9	15-0024	" 74LS04	2B,3A, 5D,5E	4				
10	15-0025	" 74LS05	1B	1				
11	15-0027	" 74LS10	5A	1				
12	15-0015	" 74LS74	4A	1				
13	15-0087	" 74LS138		1				
14	15-0017	" 74LS174	4D,4E	2				
15	15-0008	" DM8095	4C,5C	2				National
16		" IM5501	2C,2D,2E	3				Intersil 16 x 4 RAM
17		" IM5603	3C,3D,3E	3				Intersil 256 x 4 PROM 6903-3C,6903-3D,6903-3E
18	15-0070	Socket, 16 pin DIP	x3C, x3D,x3E	3				

No. 15- 6903 Descr. CONTROL LOGIC

Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
19	15-0286	I.C. 7414	4B	1				
20	15-6903	Assembly service						

No. 15-6903 Descr. CONTROL LOGIC

Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Req.	Qty Iss'd	Qty Short	Remarks
1	15-0141	Printed Circuit Board		1				
2	15-0056	Connector, 3M 3426-0000T		1				
3	15-0057	Connector, 3M 3425-0000		1				
4	15-0058	Cable, Flat Ribbon 50 cond.						3M 33265-50
5	15-0041	Resistor, 680 1/4W 10%		27				
6	15-0043	Resistor, 1K		1				
7	15-0046	Resistor, 10K		11				
8	15-0032	Resistor, Variable 5K	R31	1				Dale 784-5K
9	15-0048	Capacitor,.01μfd		1				
10	15-0049	Capacitor, .1μfd		4				
11	15-0053	Capacitor, 27μfd 20 VDC	C1,C2	2				
12	15-0002	L.E.D., Discrete T1		27				H.P. 5082-4484
13	15-0021	I.C. 74LS00		1				
14	15-0023	I.C. 74LS03		2				
15	15-0025	I.C. 74LS05		3				
16	15-0029	I.C. 74LS30		1				
17	15-0016	I.C. 74LS75		3				
18	15-0017	I.C. 74LS174		2				

No. 15-0191 Descr. INTERCEPT CONTRL

Rev. B

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Req.	Qty Iss'd	Qty Short	Remarks
19		I.C. NE555		1				
20								
21	15-0064	Switch, Toggle w/PC tails		17				C&K 7101-S-P-Y-C-Q-E
22	15-0062	Switch, Momentary w/PC tails		9				C&K 8121-C
23	15-0071	Switch, Rotary 4 position		1				Grayhill 50-CDP-90-01-1A-JN
24								
25								
26	15-0082	Knob Alco KN500A		1				
27	15-0157	Button, C&K 7527-1 White		9				
28	15-0065	Sleeve, C&K 7062-10 White		17				
29	15-0156	Nut, Knurled C&K D7028		26				

No. 15-0191 Descr. INTERCEPT CONTRL

Rev. B

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
1	15-0140	Printed Circuit Board		1				
2	15-0073	Socket, 72 pin P.C.		4				
3	15-0049	Capacitor, .1 ufd		4				
4	15-0053	Capacitor, 27 ufd		1				

No. 15-0191 Descr. INTERCEPT BUS

Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Reg.	Qty Iss'd	Qty Short	Remarks
1	15-0145	Power Supply, Deltron Q5-3.0		1				
2	15-0183	Bracket, Power Supply		1				
3	15-0184	Bracket, Fuse		1				
4	15-0185	Fuse Holder Littlefuse 342004A		1				
5	15-0186	Fuse 3AG-0.6A		1				
6	15-0178	Connector, Plug Molex 03-06-2032		1				
7	15-0180	Contact Pins 03-06-2103		2				
8	15-0171	Standoff 6-32 x 1/2 H.H.Smith 2122		4				
9	15-0162	Screw, Machine 6-32 x 1/4 RH		4				
10	15-0168	Screw, Pan head Sheet Metal #6 x 1/2Z		2				H.H.Smith 1392
11		Wire, 22GA 600V P.V.C.		8"				

No. 15-0193 Descr. INTERCEPT Power Supply Assy

Rev. A

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty. Req.	Qty Iss'd	Qty Short	Remarks
1	15-0075	Connector Housing AMP 1-87175-5		1				
2	15-0076	Key plug AMP 87179-1		1				
3	15-0078	Contacts AMP 87165-2		6				
4		Wire 24 AWG P.V.C.		5'				
5	15-0189	Cable Tie		2				Panduit SST-1M
6	15-0124	Connector Body AMP 205203-1		1				
7	15-0128	Contact AMP 1-66505-0		6				

No. 15-0195 Descr. TTY Cable Assy (Internal)

Rev. A

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Req.	Qty Iss'd	Qty Short	Remarks
1	15-0123	Connector, 9 pin Male AMP 205204-1		1				
2	15-0127	Contact, Pin AMP 1-66507-0		6				
3	15-0126	Hood AMP 206478-1		1				
4	15-0188	Wire Cable, 3 pair twisted Belden #9745		10 ft				
5	15-0182	Terminal, Spade Waldon T-2717		6				

No. 15-0194 Descr. TTY Cable Assy (External)

Rev. A

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

Item	Part number	Description	Ref. Desig.	Qty/ Assy	Qty Req.	Qty Iss'd	Qty Short	Remarks
1	15-0261	PC Board 6907		1				
2		IC 6907-4A	4A	1				Intersil PROM 5600
3		IC 6907-4B	4B	1				Intersil PROM 5603
4	15-0021	IC 74LS00N quad 2 input NAND	3F	1				
5	15-0022	IC 74LS02N quad 2 input NOR	4E	1				
6	15-0024	IC 74LS04N hex inverter	4F	1				
7	15-0017	IC 74LS174N hex D F/F	4C,6F,5A	3				
8	15-0018	IC 74LS175N quad D F/F	4D,5B,5C	3				
9	15-0027	IC 74LS10N triple 3 input NAND	3E	1				
10	15-0015	IC 74LS74N dual D F/F	3D	1				
11	15-0270	IC 74LS365 hex buffer	6A,6B,6C, 6D,6E,	5				
12	15-0257	IC 7407 hex buffer	5E	1				
13	15-0053	Capacitor 27µfd 20V electrolytic	C1	1				board decoupling
14	15-0258	IC 74LS32 quad OR 2 input	5F	1				
15	15-0259	IC 74LS257 quad 2 to 1 data select	5D	1				
16	15-0260	Resistor Pack (1K) Beckman 898-1-R1K	3A	1				
17	15-0286	I.C. 7414N	3B	1				
18	15-0049	Capacitor 0.1 micro F	3A-6F	24				bypass

Board No. 15-6907 Descr. INTERCEPT 6907 EXTENDED MEMORY CONTROL

Rev. B

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
1	15-0288	P.C. Board, 6904-INTBUS		1				
2	15-0289	P.C. Board, 6904-01 (cable conn. board)		1				
3	15-0073	Connector, 72 pin P.C. edge		11				
4	15-0129	Connector, 50 pin straight header		2				
5	12-0290	Assembly, Flat Cable 50 cond w/conn. & GND plane		2				
6	15-0291	Polarizing key		2				
7	15-0038	Resistor, 120 ohm 1/4W		5				
8	15-0083	Capacitor, .001 ufd		5				
9	15-0049	Capacitor, .1 ufd		11				
10	15-0053	Capacitor, 27 ufd		1				

No. 15-6904 Descr. Assembly, INTBUS

Rev. C

INTERSIL INCORPORATED
10900 N. Tantau Avenue Cupertino, California 95014

APPENDIX I
ENGINEERING CHANGES

IM6100 MICROPROCESSOR

The IM6100 microprocessor with the letter code D differ from the processors without the code D in the following respects.

If the IM6100 (without the code D) was reset while the processor was in the process of writing into a control panel RAM location, the CPSEL signal was truncated and MEMSEL became active and the corresponding main memory location was disturbed. In the Intercept, if the RESET pushbutton is activated with the 30 Hz timer on, main memory locations $0000\text{-}0020_8$ may be disturbed. The revised IM6100, will not truncate the CPSEL and hence main memory locations will not be disturbed on reset exit from the control panel mode.

The IM6100 (without the code D) puts out an LXMAR pulse every cycle. In cycles which involve no external references, the DX lines are tristated and hence no valid addresses are present. CMOS RAM devices must have valid logic levels on their address lines when they are strobed and if the address lines are floating, memory data may be disturbed. The revised IM6100 generates LXMAR pulses only if valid addresses are present on the DX lines.

INTERCEPT BUS

The following pin assignments have changed:

Pin Number	Old Assignment	New Assignment
1	V3 (+12V)	PR_{OUT}
2	V2 (-5V)	PR_{IN}
4	V2 (-5V)	V2 (+12V)
21	V3 (+12V)	V2 (+12V)
40	V3 (+12V)	V2 (+12V)
41	MEM DATA INVALID	UP
49	RESET MEM INVALID	DMAEN

Pins 1 and 2 are daisy chained. The MEM DATA INVALID (41) and RESET MEM INVALID (49) signals on the 6901-M4KX12 must be isolated from the bus if IM6102 MEDIC DMA features are used.

6901-M4KX12 4K X 12 CMOS MEMORY MODULE

Tie pins 12 and 13 of the unused gate of device 8E (74L00) to CMOS VCC.

Tie edge connector pins 3 and 5 to VCC.

Tie edge connector pins 70 and 72 to VCC.

Tie pin 11 of 13C to pin 13 of 14B.

Tie pin 12 of 14B to pin 13 of 13C. Cut trace on the component side between 13C pin 13 and pin 10.

Replace device 14B (74LS04 or 7404) with a 7414 (Hex Schmitt trigger).

Cut traces on edge connector fingers 41 (MEM DATA INVALID) and 49 (RESET MEM INVALID) if IM6102 MEDIC DMA features are used.

Tie pins 1 and 2 together if 6900-INTBUS or 6904-INTBUS is used.

6902-CPUTTY CPU MODULE WITH TTY INTERFACE

Isolate VCC and GND to devices 5E, 5F and 5G (DM8833's).

Connect VCC (pin 4) and DIS (pin 3) of 6G (ICM7209) to +5.

Connect set (pin 10) and reset (pin 13) of 2G (74LS74) to VCC.

Connect XTC to pin 5 of device 2F (74LS20). Pin 5 was connected to +5V. This trace must be cut.

Connect TRE (6B-24) to pin 3 of 2G (74LS74).

Connect pins 4 and 5 of 2A (74LS00) to RI (4B-3).

Connect pin 6 of 2A (74LS00) to pin 11 of 3B (74LS74).

Connect TBRE (6B-22) to 2G-2 (74LS74). Pin 2G-2 was connected to VCC and the trace must be cut on the component side.

Change 1D (74LS04) to 7414 (Hex Schmitt trigger).

Isolate LXMAR printed circuit trace with a jumper to the edge connector.

Tie pin 1 to VCC if 6900-INTBUS is used.

XTB is connected to pin 8. Cut the trace and connect it to pin 6. (Pin 8 is assigned SPARE.)

6903-CONTRL

Tie pin 1 of devices 4C and 5C (DM8095).

Tie edge connector pins 70 and 72 to VCC.

Tie pin 12 of device 4A (74LS74) to RUN/HLT line. Pin 12 was connected to VCC on the component side and this trace must be cut.

Tie edge connector pins 3 and 5 to VCC.

Tie 3M connector pin 21 (SINGLE CLOCK) to edge connector pin 63.

Tie pin 7 of U10 (74LS05) on the display module to GND.

Tie the center contact of the 30 Hz switch on the display module to 3M connector pin 42. This change applies only to boards with no silk screen on them.

Change 4B (74LS04) to 7414 (Hex Schmitt trigger).

Change the PROMs to 6903B-3C,3D and 3E.

Tie pins 1 and 2 together on the logic module if the 6900-INTBUS is used.

6907-EMC EXTENDED MEMORY CONTROLLER

Tie 4F-4 (74LS04) to 5F-1 (74LS32).

Double buffer XTC, IFETCH and INTGNT signals with a 7414 Schmitt trigger in the unused IC socket 3B.

Tie pins 1 and 2 together if 6900-INTBUS or 6904-INTBUS is used.

6901B-M4KX12 4K X 12 CMOS MEMORY MODULE

Replace devices U7 and U8 (74LS04) with 7414 (Hex Schmitt trigger).

Cut traces on pin 41 if IM6102-MEDIC DMA features are used.

Tie pins 1 and 2 together if 6900-INTBUS or 6904-INTBUS is used.

APPENDIX J

3M CABLE ASSIGNMENTS FOR 6900-INTERCEPT BUS TO 6904-INTBUS INTERCONNECTIONS

3M CONNECTOR A

3M PIN	INTBUS PIN	SIGNAL	
1	(Note 1)	GND	
2		GND	
3	(Note 2)		Daisychained PRIN-PROUT
4		GND	
5	(Note 3)	V2	
6		V2	
7	(Note 4)	+5V	
8		+5V	
9	6		XTB
10		GND	
11	8		SPARE
12	7		XTA
13	10		SKP
14	9		INTREQ
15		GND	
16	11		DEVSEL
17		GND	
18	13		DMAGNT
19		GND	
20	14		RESET
21	16		INTDIS
22	15 (Note 5)		CPREQ
23	18		C0
24	17		C2
25	20		C1
26	19		DMAREQ
27		GND	
28		GND	
29		V2	
30		V2	
31		+5V	
32		+5V	
33	25		FIELDSEL (EMA) (2)
34		GND	
35	26 (Note 5)		RUN/HLT
36		GND	
37			XTC
38		GND	
39	29		FIELDSEL (EMA) (1)
40	28		WAIT
41		GND	
42	30		MEMSEL
43		GND	
44	31		3K WRITE DIS
45	KEY		
46	33		FIELDSEL (EMA) (0)
47		GND	
48	34		INTGNT
49		GND	
50	35		4K WRITE DIS

3M CONNECTOR B

3M PIN	INTBUS PIN	SIGNAL	
1	(Note 1)	GND	
2		GND	
3	(Note 4)	+5V	
4		+5V	
5	37		MEM DISABLE
6		GND	
7	38		LXMAR
8		GND	
9	41		UP
10		GND	
11	43		DX (7)
12	42		DX (8)
13	45		DX (0)
14	44		DX (9)
15		GND	
16		GND	
17	48		DX (1)
18	47		DX (10)
19	50		DX (11)
20	49		DMA ENABLE
21	51		DX (6)
22		GND	
23	53 (Note 5)		RUN
24	54		DX (2)
25		GND	
26	55 (Note 5)		CPSEL
27		GND	
28	56		DX (3)
29	57		SPARE
30		GND	
31	59		DX (4)
32		GND	
33	60		DATAF
34		GND	
35	61		IFETCH
36		GND	
37	62		DX (5)
38	63 (Note 5)		SINGLE CLOCK
39	65		SPARE
40		GND	
41	66 (Note 5)		SWSEL
42		GND	
43	67 (Note 5)		FREE RUN
44	68 (Note 5)		LINK
45		GND	
46	KEY		
47		+5V	
48		+5V	
49	(Note 6)	V1	
50		V1	

Note 1	6900-INTERCEPT BUS/6904-INTBUS GND pins are 12, 22, 32, 40, 46, 52, 58 and 64.
Note 2	Edge connector pins 1 and 2 are shorted together on the paddle cards, bussed out to the 6904-INTBUS through pin 3 of the 3M connector (A), and then connected to pin 2 (PRIN) of the edge connector closest to the 3M connector. Pins 1 and 2 are daisychained on the INTBUS.
Note 3	V2 pins are 4, 21 and 23. (Planned assignment +12V)
Note 4	+5V pins are 3, 5, 24, 36, 70 and 72.
Note 5	CPREQ (15), RUN/HLT (26), RUN (53), CPSEL (55), SINGLE CLOCK (63), SWSEL (66), FREE RUN (67) and LINK (68) lines are left open in the paddle card. The user may jumper them.
Note 6	V1 pins are 69 and 71. (Planned assignment -12V)

APPENDIX K
PERFORMANCE OF PDP-8/E DIAGNOSTIC PROGRAMS ON INTERCEPT

PROCESSOR AND MEMORY TESTS (ZF002-RB)

1. PDP-8/E Instruction Test Part 1 (MAINDEC-08-DHKAF-A-PB)

   There is no exception.

2. PDP-8/E Instruction Test Part 2 (MAINDEC-08-DHKAG-A-PB)

   The 30Hz switch on the control panel must be off since the test checks for the instruction sequence ION; IOF and if the 30Hz switch is on, a control panel interrupt may occur between ION and IOF. Note that if an IOF follows an ION immediately, the sequence ION; IOF is a no-operation.

   There is no other exception.

3. 8-E Adder Test (MAINDEC-08-DHKAA-B-PB)

   There is no exception.

4. 2K to 32K PDP-8/A Processor Exerciser (MAINDEC-08-DJEXB-A-PB)

   This checks to see if autoindex locations $10\text{-}17_8$ can be addressed as current page locations when the program is in page 0. Note that when the program is in page 0, page 0 is also the current page. IM6100 requires that autoindex locations must always be addressed as page 0 locations.

5. Memory Checkerboard Test (MDEC-8E-DIAB-D-PB)

   There is no exception

6. Memory Address Test (MDEC-8E-DIEC-D-PB)

   There is no exception.

Note that 6985-IDIAG-1 replaces ZF002-RB.

MEMORY EXTENSION TEST

1. PDP-8/E KM8-E Memory Extender (MAINDEC-08-DHMCA-A-PB)

A. In the PDP-8/E an INTREQ line can be read by the CPU during a GTF or SRQ instruction whether or not the interrupts are being inhibited by the KM8-E memory extender.

Since the interrupt inhibit logic is external to the IM6100, the microprocessor will not receive an INTREQ if the IIF is set.

B. The PDP-8/E does not expect the AC (3) to be set during a GTF instruction, even if the IIF is set. This is contrary to the specification in the PDP-8/E Small Computer Handbook.

In the INTERCEPT, AC (3) is set if a GTF is executed with the IIF set.

C. PDP-8/E expects non-existing memory fields to be all zeros.

INTERCEPT does not guarantee the contents of non-existing memory fields.

D. The INTERCEPT does not make any provision for the time-share option.

The following changes must be made for the KM8-E test to run on the INTERCEPT.

(i)

Location	Old contents	New contents
0115_8	5200_8	4600_8
0016_8	1200_8	0600_8
2257_8	1411_8	5266_8

(ii) Switch register bit (0) must be a 1.

(iii) 30Hz must be off.

Note that the 6985-IDIAG-2 diagnostic program supplied with the 6907-EMC module replaces MAINDEC-08-DHMCA-A-PB.

FLOPPY DISK DIAGNOSTICS

1. RX8/RX01 Diagnostic Program (MAINDEC1-08-DIRXA-B-D)

Test 27 fails since the test is speed dependent.

There is no other exception.

Note that 6985-IDIAG-3 diagnostic program supplied with the INTERCEPT D10 Diskette System replaces MAINDEC1-08-DIRXA-B-D.