Digital Equipment Corporation
Maynard, Massachusetts

DIFF #2

Maintenance Manual
PDP-8/I
Volume I

DEC-8I-HR1A-D

PDP-8/I

MAINTENANCE MANUAL

VOLUME I

1st Printing October 1968
2nd Printing March 1968
3rd Printing May 1969
4th Printing July 1969

Instruction times, operating speeds and the like are included in this manual for reference only; they are not to be taken as specifications.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

TABLES

TABLES (Cont)

PDP-8/I MAINTENANCE MANUAL

VOLUME I

PDP-8/I Pedestal and Cabinet Configuration

CHAPTER 1
INTRODUCTION AND DESCRIPTION

1.1 INTRODUCTION

This manual covers installation, operation, theory, and maintenance of the Programmed Data Processor - 8/I (PDP-8/I). It is the intent of this manual to provide the field service engineer or maintenance technician who is familiar with digital logic circuitry with the information he needs to install and maintain a PDP-8/I system. This manual is largely written under the assumption that the reader is conversant with DIGITAL's system of logic notation. If this is not the case, the reader should refer to applicable sections in the appendix for a description.

The PDP-8/I is a one-address, 12-bit, fixed-word-length, parallel computer using two's complement arithmetic. Normal cycle time of the 4096-word (referred to as 4K), random-access, magnetic-core memory is 1.5 μs. An additional 4K of memory with extended memory control may also be added to the system by plugging the memory modules directly into the system in spaces provided. As much as 32K of memory is available as well as other options, with their control logic internal to the basic machine. Standard features of the system include indirect addressing, facilities for instruction skipping and program interruption as functions of input/output-device conditions, and high-speed information transfers with mass-memory devices via a cycle-stealing data break.

1.2 DESCRIPTION

The integrated-circuit (PDP-8/I) is a small-scale general-purpose computer that functions as an independent information-handling facility in a large computer system, or as the control element in a complex processing system.

In terms of operating characteristics, speed, programming and available peripheral devices, the PDP-8/I is completely compatible with the PDP-8.

The PDP-8/I performs one addition in 3.0 μs (with one number in the accumulator), permitting a computation rate of 333,333 additions per second to be achieved. It performs subtraction in 6.0 μs (with the minuend in the accumulator) using two's complement addition. Multiplication takes approximately 360 μs, using a subroutine that operates on two signed 12-bit numbers to produce a 24-bit product, leaving the 12 most significant bits in the accumulator. Division of two signed 12-bit numbers takes approximately 460 μs, using a subroutine that produces a 12-bit quotient in the accumulator, and a 12-bit remainder in core memory. The optional Extended Arithmetic Element (KE8I) performs similar multiplication and division operations in 6.0 and 6.5 μs, respectively.

Flexible, high-capacity, input/output capabilities of the computer allow it to operate a variety of peripheral equipment. In addition to the Teletype keyboard/printer and perforated-tape reader/punch, equipment supplied with a basic PDP-8/I, the system can operate a number of optional devices. These options include high-speed perforated-tape reader and punch, card reader, line printer, analog-to-digital converters, cathode-ray-tube displays, magnetic drum systems, magnetic disk-file systems, and magnetic tape equipment. Instruments or equipment of special design can also be connected into the PDP-8/I system. The computer itself needs no modification for the addition of peripheral devices.

The PDP-8/I is completely self-contained and, under normal conditions, requires neither spec-

ial power sources nor rigidly controlled environmental conditions. A single source of 115 VAC, 60 Hz, single-phase power permits internal power supplies to produce all required operating voltages. M-series modules, using TTL-type integrated circuit packs, ensure reliable operation in ambient temperatures between +32 and +130°F.

1.3 PERTINENT DOCUMENTS

The following documents serve as source material and complement the information in this manual:

1. Logic Handbook, C-105 (1968 edition), printed by DIGITAL, which notes the function and specifications of the M-series modules and module accessories for the PDP-8/I.

2. Small Computer Handbook, C-800 (1968 edition), printed by DIGITAL, which contains programming and operational data and the PDP-8/I Users Handbook.

3. Technical Manual, Automatic Send and Receive Sets (ASR), Bulletin 273B (Volumes I and II). This manual covers operation and maintenance of the Teletype unit.

4. Parts, Model 33 Page Printer Set, Bulletin 1184B, contains an illustrated parts breakdown to serve as a guide to disassembly, reassembly, and ordering replacement parts for the Teletype unit.

5. Instruction List F-8I6, printed by DIGITAL. This is a shirt-pocket list of all memory-reference instructions, all augmented instructions, the most common IOT instructions, and the ASCII code used with many I/O devices.

6. Instruction manuals and MAINDEC programs for appropriate input/output devices are prepared by DIGITAL.

7. Digital Program Library Documents. Perforated program tapes and descriptive matter for the Program Assembler Language (PAL III), FORTRAN, FOCAL, utility subroutines, and the maintenance programs (MAINDEC) prepared by Digital are available to PDP-8/I users. The list of programs currently in the library and available is provided in Appendix D.

1.4 ABBREVIATIONS

Listed below are the most-commonly used abbreviations of registers, key operations, components, instructions and signal names.

AC	Accumulator
A/D	Analog-to-digital (converter or converted signal)
ADD or ADDR	Address
B	Break state
BD	Bus driver
BRK RQST	Break request
CA	Current address state
CLA	Clear accumulator (instruction or signal)
CLR	Clear
CM or COMP	Complement
CONT	Continue
CP	Central processor
D	Defer state
DCA	Deposit and clear accumulator (instruction)
DEP	Deposit
DF	Data Field Register
DIV	Divide
DLI	Data line interface
DP	Deposit
E	Execute state
EAE	Extended arithmetic element
EX or EXAM	Examine
F	Fetch state
FLG	Flag
HLT	Halt
IF	Instruction field
IFR	Instruction field register
INH	Inhibit
INST	Instruction (key)
INT	Interrupt
INT ACK	Interrupt acknowledge

I/O	Input/Output
ION	Interrupt on
IOP	Input/Output pulse
IOT	Input/Output (information) transfer
IR	Instruction register
ISZ	Increment and skip if zero
JMP	Jump (instruction)
JMS	Jump to subroutine (instruction)
KBD	Keyboard (Teletype)
L	Link
MA	Memory address register
MB	Memory buffer register
MEM	Memory register (also SENSE)
MQ	Multiplier quotient register
MUL	Multiply
OP	Operate
OPR	Operate (class of instruction)
P	Parity
PA	Pulse amplifier
PC	Program counter
PI	Program interrupt
PROG	Program
PWR CLR	Power clear
RDR	Reader
SA	Sense amplifier
SC	Step counter (EAE)
SD	Solenoid driver
SENSE	Memory register (also MEM)
SF	Start field
SING	Single (key)
SKP	Skip
ST	Start
SR	Switch register
SYNC	Synchronize
TAD	Two's complement add (instruction)
TP	Time pulse
TS	Time state
TT	Teletype
TTI	Teletype in (Teletype keyboard/ reader buffer)
TTO	Teletype out (Teletype teleprinter/ punch buffer)
WC	Word count state

CHAPTER 2
INSTALLATION

This chapter contains installation information and physical specifications of the PDP-8/I and its options.

2.1 SPACE REQUIREMENTS

Access space must be provided at the installation site to accommodate the PDP-8/I and peripheral equipment, and to allow access to all doors and panels for maintenance.

The PDP-8/I is available in either the pedestal or rack mounted configuration. The rack mounted configuration and peripherals may be purchased completely installed in DEC cabinets or may be purchased unmounted for installation in a customer cabinet. Figure 2-1 gives dimensions for the pedestal mounted PDP-8/I. Figures 2-2 through 2-5 show detailed mounting information for installing the PDP-8/I into standard DEC, BUD, and EMCOR racks.

Minimum service clearance on all standard DEC computer cabinets is 8-3/4 in. at the front and 14-7/8 in. at the back.

The standard Teletype automatic send receive set requires a floor space of approximately 22-1/4 in. wide by 18-1/2 in. deep. The Teletype signal cable requires the Teletype to be placed near the computer.

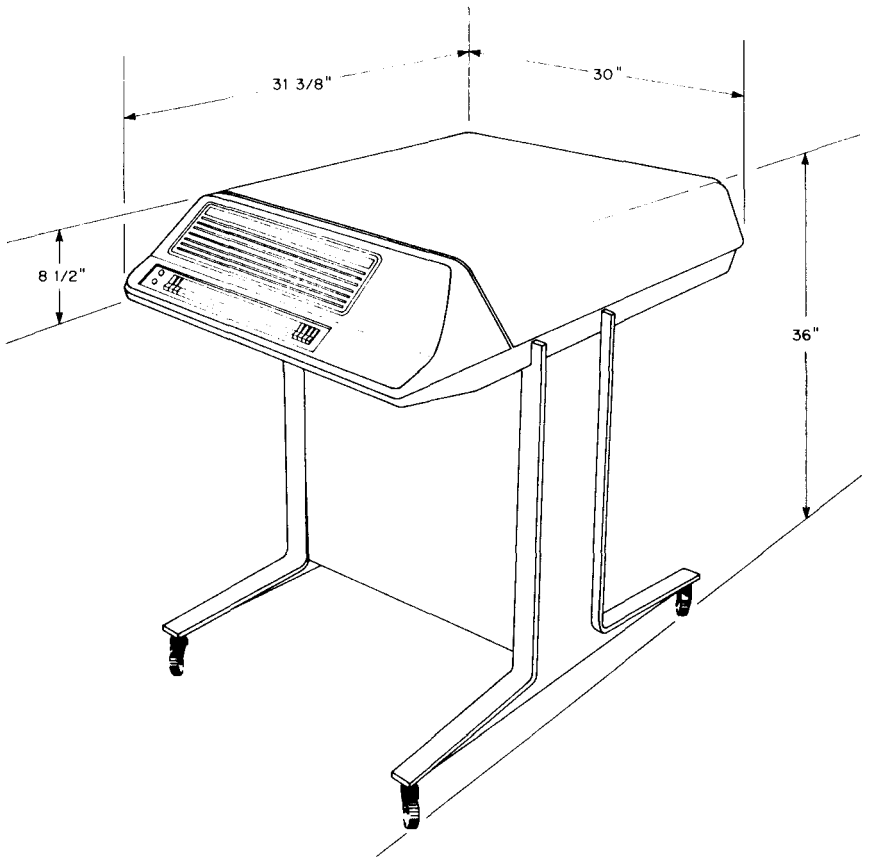


Figure 2-1 PDP-8/I Pedestal Dimensions

2.2 ENVIRONMENTAL REQUIREMENTS

Ambient temperature at the installation site can vary between 32° and 130°F (0° and 55°C). To extend the life expectancy of the system, however, it is recommended that the temperature at the installation site be maintained at between 70° and 85°F (between 20° and 30°C).

During shipping or storing of the system, the ambient temperature may vary between 32° and 130°F (between 0° and 55°C). All exposed surfaces of all DEC cabinets and hardware are treated to prevent corrosion, but exposure of systems to extreme humidity for long periods of time should be avoided.

2.3 POWER REQUIREMENTS

A source of 115V (±17V), 60 Hz (±0.5 Hz), single-phase power capable of supplying at least 15A must be provided to operate a standard PDP-8/I. To allow connection of the computer to the power cable, this source should be provided with a Hubbell 3-terminal connector plug. A rack mounted PDP-8/I has a 20A twist-lock plug; systems that draw more than 20A use a 30A twist-lock plug. All free-standing cabinets require independent 115V receptacles. However, these units may be turned on or off, or controlled from the PDP-8/I operator console.

Upon special request, a PDP-8/I can be constructed to operate from a 220V (±33V), 60 Hz (±0.5 Hz), single-phase power source or from a 100V (±15V), 50 Hz (±0.5 Hz), single-phase power source.

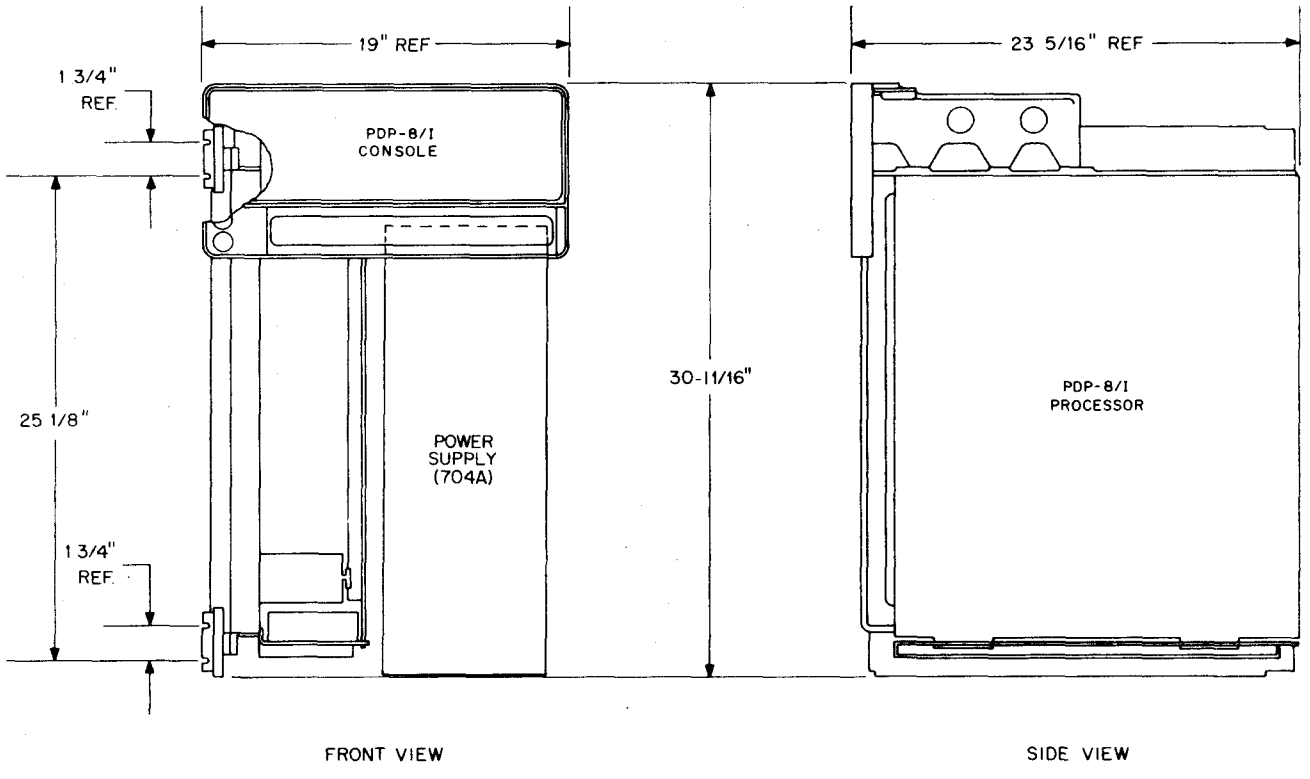


Figure 2-2 Rack Mounted PDP-8/I Dimensions

VIEW SHOWN WITHOUT FRONT DOOR (SUPPLIED BY DEC IF DESIRED)
AND WITHOUT LOWER COVER PANEL (SUPPLIED BY DEC IF DESIRED)

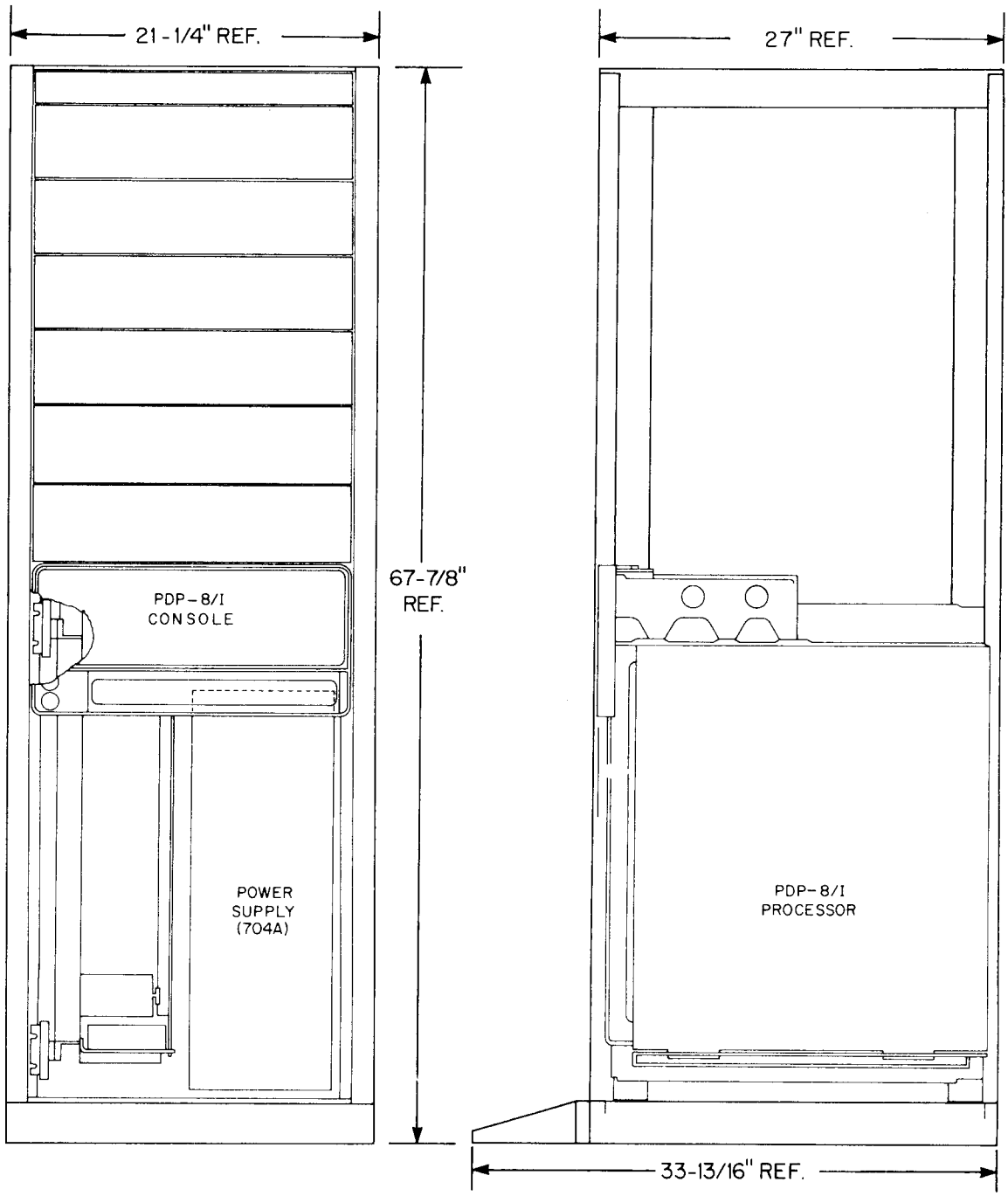


Figure 2-3 PDP-8/I Installation in Standard DEC Cabinet

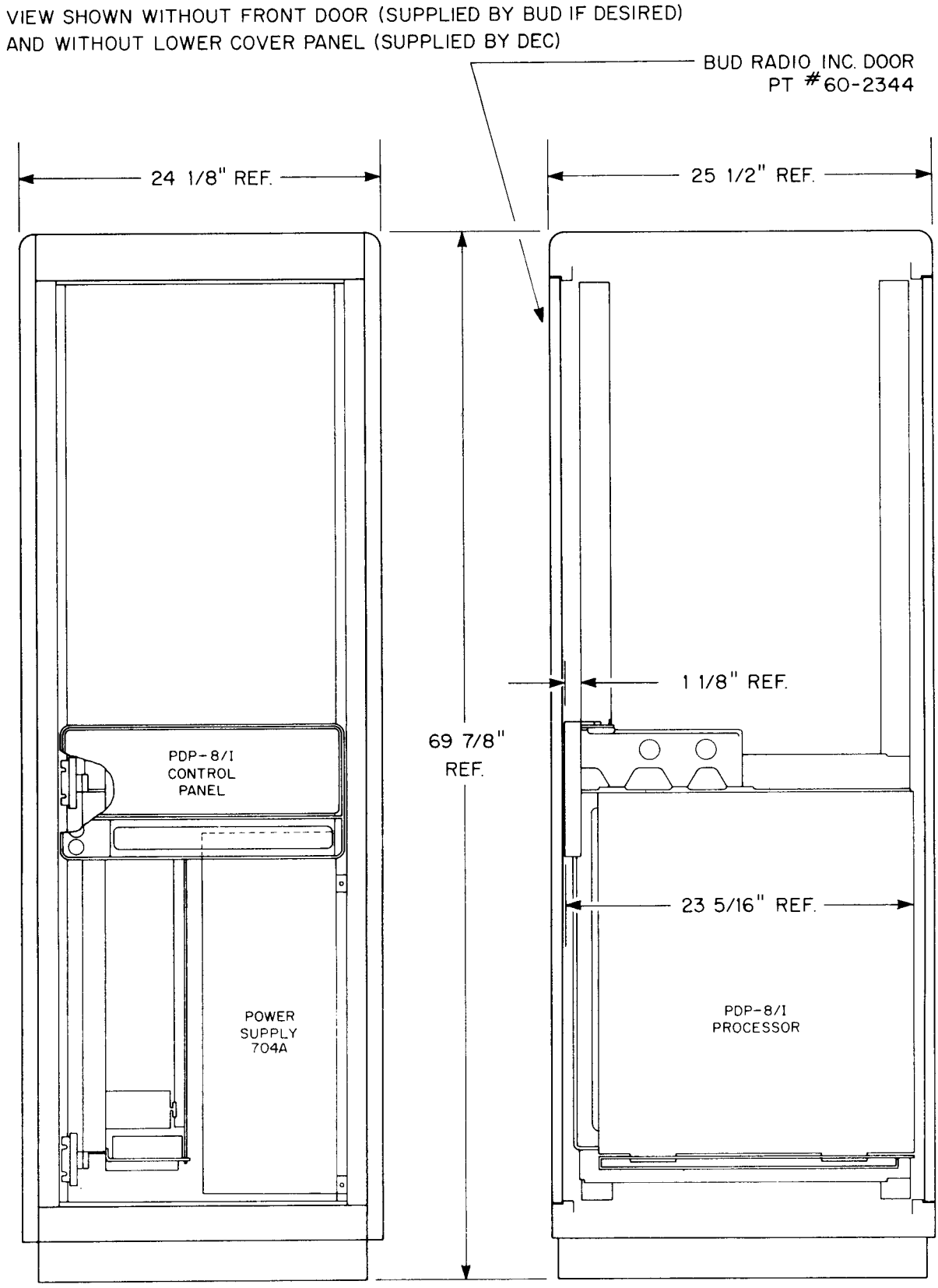


Figure 2-4 PDP-8/I Installation in Bud Cabinet

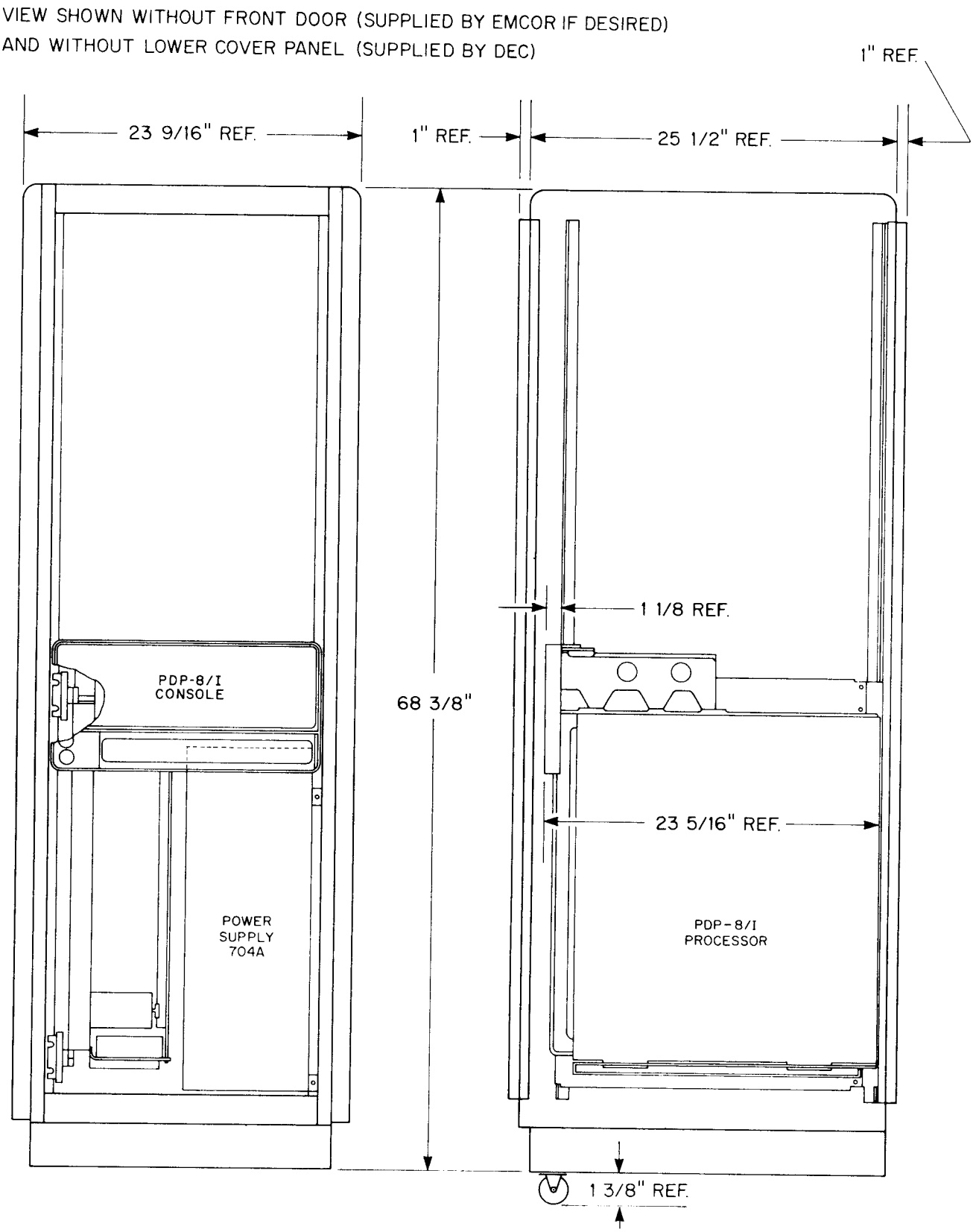


Figure 2-5 PDP-8/I Installation in Emcor Cabinet

2.4 CABLE REQUIREMENTS

Signal conductor cables with Type W011 Cable Connectors provide signal connections between the computer and the optional equipment in free-standing cabinets. These cables are connected by plugging the W011 connectors into standard module receptacles. Cables connect to cabinets through a drop panel in the bottom of cabinets. Subflooring is not necessary because casters elevate the cabinets high enough from the floor to provide sufficient cable clearance. Cable details are included in the I/O Interface paragraph.

2.5 INSTALLATION PROCEDURE

Installation of a PDP-8/I system requires no special tools or equipment. A fork-lift truck, or other pallet-handling equipment, and normal hand tools should be available for receiving and installing the equipment. To install the computer observe the following procedure:

1. Place the computer package near the final installation location. Pry the top, and side sections of the wooden shipping crate apart with a hammer, and wedge at the staple joints. Unfasten the four bolts holding the computer frame to the pallet. Using a ramp, slide the computer off the pallet to the final location. When a cabinet mounted PDP-8/I is installed, a similar procedure is used. First, remove the shipping straps with cutting shears. Remove the packing material, and the cardboard crate to disassemble the wooden corner supports.

2. Remove the plastic cover. Remove the two machine screws in the rear of the computer logic frame that hold it firm. This allows the logic frame to be serviced pulling it out on the tracks.

3. Open the Teletype carton, and remove the packing material. Remove the back cover from the stand. Remove and unwrap the copyholder, chad box, and power pack. Remove the stand from the shipping carton. Remove the Teletype console from the carton, holding it by means of the wooden pallet attached to the bottom. Snap the power pack in place at the top of the rear side of the Teletype stand. Remove the Teletype console from the pallet, and mount it on the stand. Connect the Teletype console to the power pack (a six-lead cable attached at the console is connected to the power pack by means of a white plastic Molex 1375 Female Connector which mates with a male output plug on the power pack). Pass the three-wire power cable, and the seven-conductor signal cable (which is terminated in a type W076 Cable Connector Module) through the opening at the lower left-hand corner of the Teletype stand; then replace the back cover of the stand by means of the two mounting screws.

4. Dress the Teletype cable under the PDP-8/I cabinet, through the large opening and into logic frame slot J12. It is necessary that this cable be dressed through the cable clamp at the lower rear corner of the logic frame where the power cables are secured. A second clamp may be desirable at the bottom of the PDP-8/I cabinet to assure that sufficient slack exists irrespective of the Teletype position.

5. Connect the three-prong male connector of the Teletype power cable to the connector at the rear of the computer power supply chassis.

6. Set the PANEL LOCK switch to the full counterclock wise position (OFF). Set the POWER LOCK switch to the full counterclockwise position (OFF). Set the main power switch (circuit breaker at rear of computer power supply chassis) to ON.

7. Turn the POWER switch on.

8. Install a roll of printed paper into the Teletype keyboard/printer, and install a tape in the punch as described in the Teletype technical manual.

9. Set the LINE/OFF/LOCAL switch to LINE. Press the punch ON pushbutton. Strike several keys, and note whether or not the printer and punch operate. Check the operation of the printer with the LINE/OFF/LOCAL switch set to LOCAL. After completion of these checks, set the switch to OFF.

10. Turn the power OFF.

This completes the installation of a standard PDP-8/I system. Before normal operating use, verify the operating capability of the system. Perform the power supply checks, and run the complete set of diagnostic programs (MAINDECs) as described in Chapter 5.

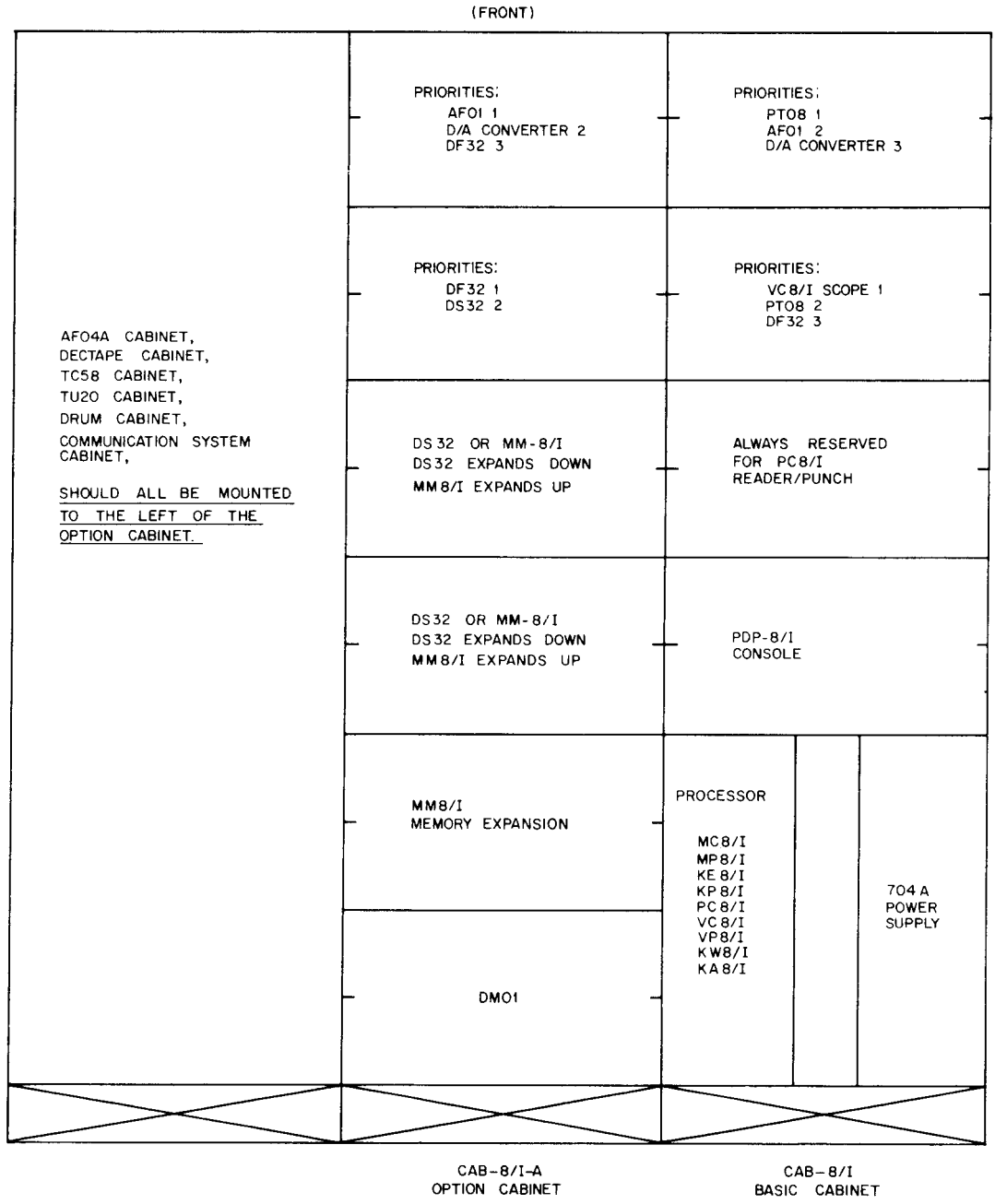


Figure 2-6 System Configurations

2.6 INTERNAL OPTION INSTALLATION

The installation of the internal options involves the addition of the logic modules in the proper locations. Turn off the computer before inserting or removing any modules in the logic frame. Refer to both sheets of the module utilization drawings MU-8I-0-17 for the locations of each module of all internal options. If an option involves an external device, dress connecting cables through the large opening from the logic frame to the option. Proper adequately ventilated mounting facilities are necessary to insure protection and safety of the equipment. Refer to the system configurations drawing (Figure 2-6) for proper location of options.

2.7 SYSTEM CONFIGURATIONS

PDP-8/I systems are mounted in standard DEC cabinets. The basic cabinet contains the processor and the power supply. The other cabinets and spaces in Figure 2-6 show the position priority assignments and the top priority when a choice exists. The priorities are assigned with considerations of ease of control, and cable lengths.

2.8 I/O INTERFACE

The following paragraphs describe the PDP-8/I I/O bus, the I/O cables, the logic level converters and their driving capabilities, and the processor location terminals of each I/O bus signal.

2.8.1 I/O Bus Signal

The PDP-8/I employs a series I/O bus system which allows interface connections to be made between all of the external devices without modifying the computer wiring. In a series I/O bus, the computer sends all I/O signals to the first device which makes use of pertinent signals and sends all of the I/O signals to the next device, as shown in Figure 2-7. Where physical location of equipment makes series connections impractical, or when cable length become excessive, additional interface connectors are usually installed near the computer.

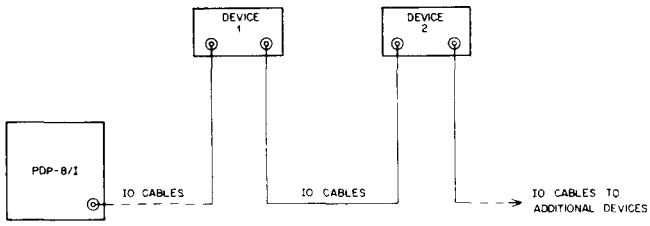


Figure 2-7 I/O Bus Configuration

2.8.2 I/O Cables

The standard PDP-8/I I/O bus uses 18-conductor 93Ω ribbon cable that terminates into W011 Cable Connectors. Terminals C,F,J,L,N,R, and U of receptacles J01 through J10 are grounded within the computer, and terminals D,E,H,J, M,P,S,T, and V carry signals.
All PDP-8/I I/O cable connections are made at the assigned module receptacle connectors J01 through J10 in the mounting frame, with the exception of the Address Extend 1,2,3 inputs, and Data Field 0,1, and 2 outputs which terminate at location H01.

2.8.3 Logic Levels and Level Converters

The PDP-8/I internal logic levels are positive levels of +3V and 0V. The present DEC external peripherals function with negative DEC standard levels or pulses. The standard levels are either ground potential (0.0V to -0.3V) designated by an open diamond (◇), or -3V (-3V to -4V) designated by a solid diamond (◆). To provide compatibility between the internal PDP-8/I levels and the external negative logic

levels, the computer inputs and outputs are converted by the M506 and M650 Level Converters (Drawing BS-8I-0-10).

Input interface signals to the computer use the M506 Negative Input Converters which shift the DEC standard negative levels of -3V and 0V to PDP-8/I positive levels of 0V and +3V. A -3V input produces an M506 output of 0V, and a 0V input generates a +3V output. In addition, the M506 contains positive-logic internal bus input-gating that allows the outputs from the Teletype logic, extended memory internally generated skip and interrupt, and AC CLEAR from the internal options to input to the major register gating network.

All I/O bus output signals to the external options except the MM8I, are first converted by the M650 Negative Output Converter and Bus Driver modules which shift the PDP-8/I levels of +3V and 0V to DEC levels of 0V and -3V. An M506 input voltage level of +3V is converted to 0V as an I/O signal, and a 0V input signal converts to -3V. The M650 can drive 20 mA at 0V or sink 20 mA at -3V.

2.8.4 Interface Signal Connections

Figure 2-8 shows the I/O bus interface signals as well as the cable and signal locations. The signal direction is also shown by the logic levels which are shown with the functions in the active states.

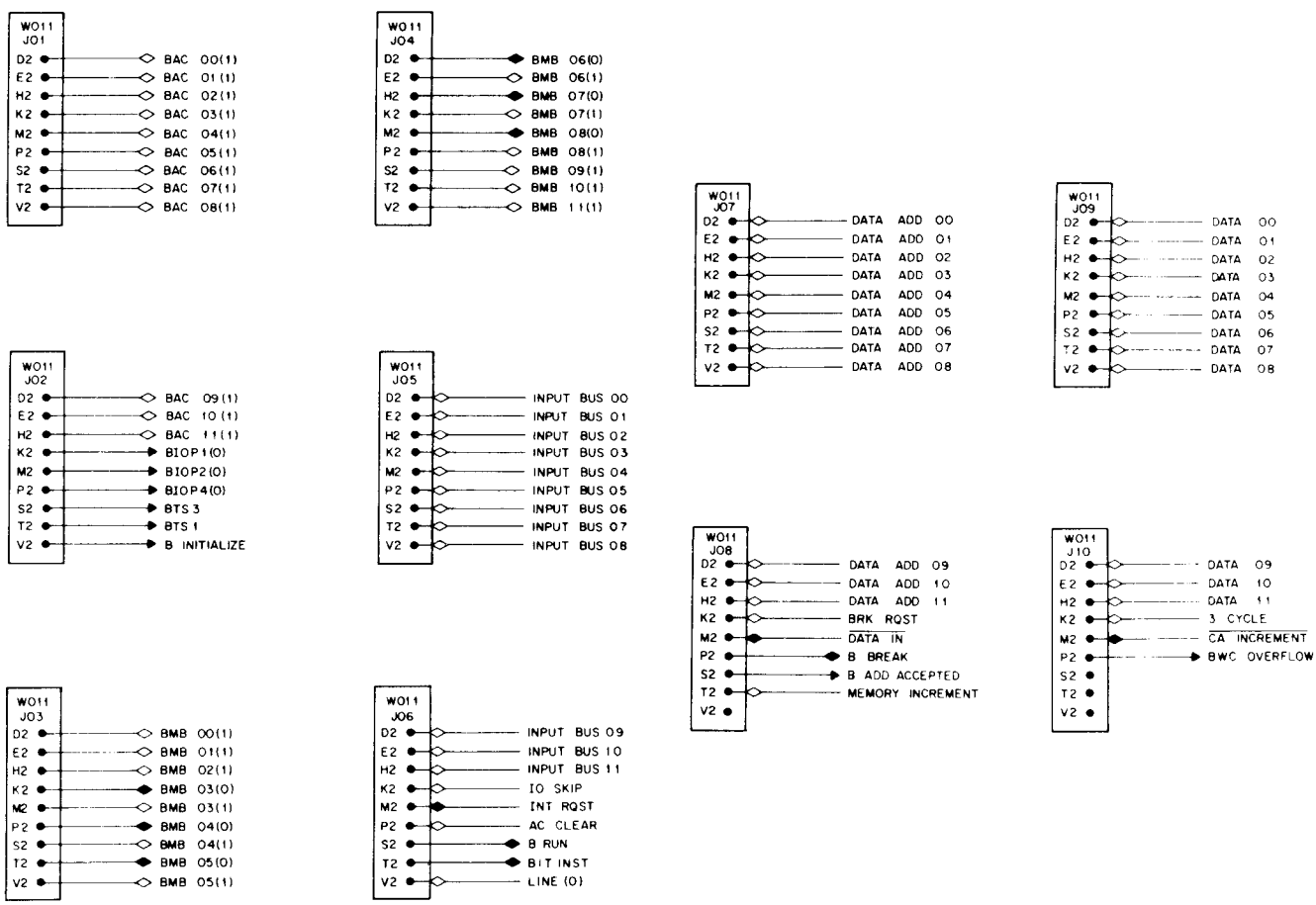


Figure 2-8 I/O Signal Connections

CHAPTER 3
OPERATION

This chapter contains operating information for the PDP-8/I and the ASR33 Teletypewriter. Operating information for the peripheral input/output devices is contained in their respective manuals.

3.1 CONTROLS AND INDICATORS

The following subparagraphs contain detailed information regarding the controls and indicators of the PDP-8/I and the ASR33 Teletypewriter.

3.1.1 Computer

Figure 3-1 shows the location of the PDP-8/I controls and indicators. Although not marked on the front panel, register bits are numbered from left to right starting with zero. Therefore the most significant (leftmost) bit in the program counter (PC) is identified as PC00, and the least (rightmost) significant bit is identified as PC11. Table 3-1 contains a listing of the PDP-8/I controls and indicators within their functions. The PDP-8/I controls (except the power and panel lock switches) are of two types: butterfly switches, and momentary-contact switches. The butterfly switches are considered to be in their zero or off-state when the top half of the butterfly is fully depressed, and are considered to be in their one or on state when the bottom half of the butterfly is depressed. The momentary-contact switches include the Start, Exam, Load Add, Cont, Dep and Stop switches. These switches (except Dep) are actuated when the bottom half is fully depressed. The Dep switch is the reverse of the above. Indicators are considered to be in their on or one state when they are lit, and in their off or zero state when not lit.

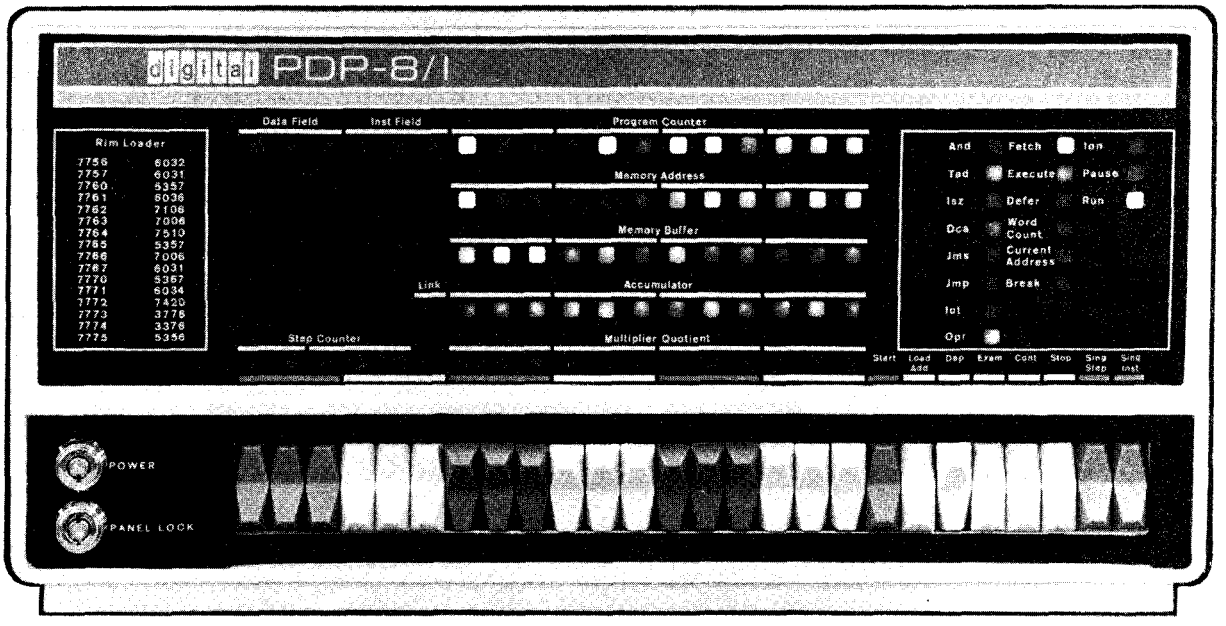


Figure 3-1 PDP-8/I Front Panel

Table 3-1
Computer Controls And Indicators

Control or Indicator	Function
Panel Lock key switch	When turned clockwise, this key-operated switch disables all controls except the Switch Register switches on the operator console. In this condition, inadvertent key operation cannot disturb the program. The program can, however, monitor the content of SR by execution of the OSR instruction.
Power key switch	This key-operated switch controls application of primary power to the computer. When this switch is turned clockwise, primary power is applied.
Start key	Starts the program by turning off the program interrupt circuits clearing the AC and L, setting the Fetch state, and starts the central processor.
Load Add key	This key transfers the content of SR into PC, the content of INST FIELD * switches into IF, the content of the DATA FIELD * switches into DF, and clears the major state flip-flops.
Dep key	This key transfers the content of SR into MB and core memory at the address specified by the current content of PC. The major state flip-flops are cleared. The contents of PC is then incremented by one to allow storing of information in sequential core memory addresses by repeated operation of the Dep key.
Exam key	This key transfers the content of core memory at the address specified by the content of PC, into the MB. The content of the PC is then incremented by one to allow examination of the contents of sequential core memory addresses by repeated operation of the Exam key. The major state flip-flop register cleared. The MA indicates the address of the data in the MB.
Cont key	This key sets the RUN flip-flop to continue the program in the state and instruction designated by the lighted console indicators, at the address currently specified by the PC if key SS is not on.
Stop key	Causes the RUN flip-flop to be cleared at the end of the instruction in progress at the time the key is pressed.
Sing Step key	This key causes the RUN flip-flop to be cleared to disable the timing circuits at the end of one cycle of operation. Thereafter, repeated operation of the Cont key steps the program one cycle at a time so that the operator can observe the contents of registers in each major state.

* Activated only on systems containing the MC8/I, Memory Extension Control option.

Table 3-1
Operator Console Controls And Indicators (Cont)

Control or Indicator	Function
Sing Inst key	This key allows execution of one instruction. When the computer is started by pressing the Start or Cont key, the Sing Inst key causes the RUN flip-flop to be cleared at the end of the last cycle of the current instruction. Thereafter, repeated operation of the Cont key steps the program one instruction at a time.
Switch Register switches	Provide a means of manually setting a 12-bit word into the machine. Load the content of this register into PC by pressing the Load Add key or load the content into the MB and core memory by the Dep key. Under program control, the OSR and LAS instructions can set the content of SR into AC.
Data Field indicators and switches *	The indicators denote the content of the data field register (DF), and the switches serve as an extension of SR to load DF by means of the Load Add key. DF determines the core memory field of data storage and retrieval.
Inst Field indicators and switches *	The indicators denote the content of the instruction field register (IF), and the switches serve as an extension of SR to load the IF by means of the Load Add key. IF determines the core memory field from which instructions are to be taken.
Register Indicators	
Program Counter indicators	The PC contains the location of the next instruction to be performed.
Memory Address indicators	Indicate the content of MA. Usually, the contents of MA denote the core memory address of the word currently or previously read or written. After operation either the Dep or Exam key, the contents of MA indicate the core memory address just examined or deposited into.
Memory Buffer indicators	Indicates the content of MB. Usually, the contents of MB designate the word just written at the core memory address in MA.
Accumulator	Indicates the content of AC.
Link	Indicates the content of L.
Multiplier Quotient	Indicates the content of the multiplier quotient (MQ). MQ holds the multiplier at the beginning of a multiplication and holds the least-significant half of the product at the conclusion. It holds the least-significant half of the dividend at the start of division and holds the quotient at the conclusion.

*Activated only on systems containing the MC8/I, Memory Extension Control option.

Table 3-1
Operator Console Controls And Indicators (Cont)

Control or Indicator	Function
Major State Indicators	
Fetch	Indicates that the processor is currently performing or has performed a Fetch cycle.
Execute	Indicates that the processor is currently performing or has performed an Execute cycle.
Defer	Indicates that the processor is currently performing or has performed a Defer cycle.
Word Count	Indicates that the processor is currently performing or has performed a Word Count cycle.
Current Address	Indicates that the processor is currently performing or has performed a Current Address cycle.
Break	Indicates that the processor is currently performing or has performed a Break cycle.
Miscellaneous Indicators	
Ion	Indicates the 1 status of the INT.ENABLE flip-flop. When lit, the interrupt control is enabled for information exchange with an I/O device.
Pause	Indicates the 1 status of the PAUSE flip-flop when lit. The PAUSE flip-flop is set for 2.75 μs by any IOT instruction that requires generation of IOP pulses or by any EAE instruction ** that require shifting of information.
Run	Indicates the 1 status of the RUN flip-flop. When lit, the internal timing circuits are enabled and the machine performs instructions.
Instruction Indicators	
And	Indicates that the processor is currently performing or has performed an And instruction.
Tad	Indicates that the processor is currently performing or has performed a Tad instruction.

** Activated only on systems containing the KE8I, Extended Arithmetic Element option.

Table 3-1
Operator Console Controls And Indicators (Cont)

Control or Indicator	Function
Instruction Indicators	
Isz	Indicates that the processor is currently performing or has performed an Isz instruction.
Dca	Indicates that the processor is currently performing or has performed a Dca instruction.
Jms	Indicates that the processor is currently performing or has performed a Jms instruction.
Jmp	Indicates that the processor is currently performing or has performed a Jmp instruction.
Iot	Indicates that the processor is currently performing or has performed an Iot instruction.
Opr	Indicates that the processor is currently performing or has performed an Opr instruction.

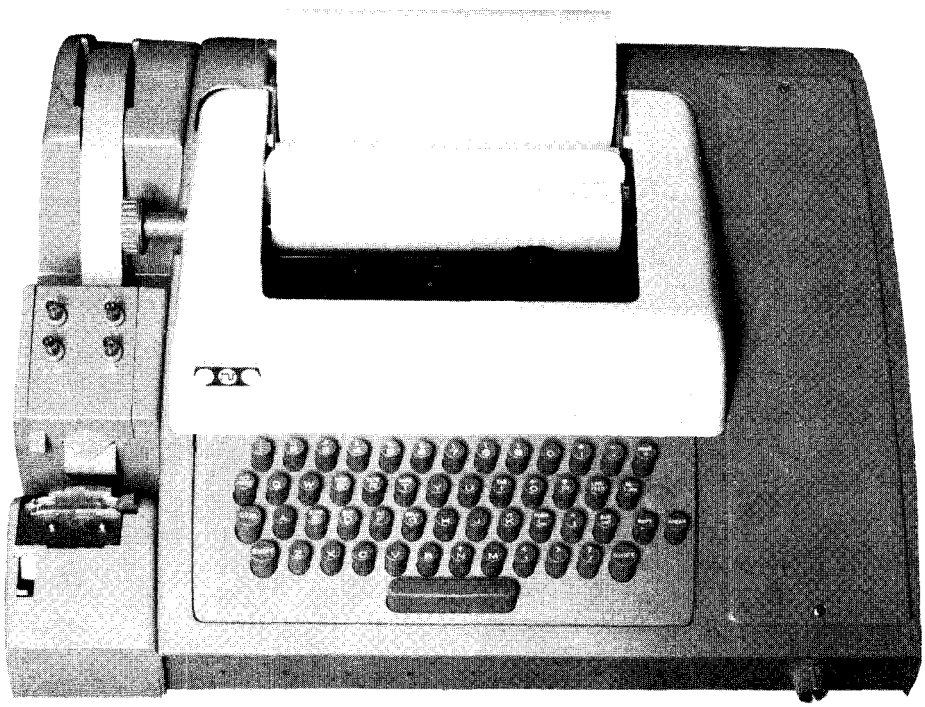


Figure 3-2 Teletype Model ASR33 Console

3.1.2 Teletype

Figure 3-2 shows the location of the ASR33 Teletypewriter controls and indicators. Table 3-2 contains a listing of the ASR33 controls and indicators with a description of their functions.

3.2 OPERATING PROCEDURES

Many means are available for loading and unloading PDP-8/I information. The means used depend upon the form of the information, time limitations, and the peripheral equipment connected to the computer. The following procedures are basic to any use of the PDP-8/I. Although these procedures are used infrequently as the programming and use of the computer become more sophisticated, they are valuable in preparing the initial programs and learning the function of machine input and output transfers.

3.2.1 Common Procedures

All of the following procedures require that the PANEL LOCK switch be rotated fully counterclockwise (off), and that the Power switch be rotated fully clockwise(on).

3.2.2 Manual Loading Procedures

Programs and data can be stored or modified manually by means of the facilities on the operator console. The chief use of the manual data storage facility is to load the Readin Mode Loader program into the computer core memory. The Readin Mode Loader (RIM) is a program used for loading into the PDP-8/I other programs that have been assembled on perforated tape in RIM format. This program and the RIM tape format are described in the PDP-8/I Users Handbook (see Small Computer Handbook, C-800, 1968 edition) and in Digital Program Library descriptions. The RIM program is also listed in Table 3-3 for rapid reference and can be used as an exercise in manual data storage. To store data manually in the PDP-8/I core memory proceed as follows:

1. Set the bit switches of the Switch Register (SR) to correspond with the address bits of the first word to be stored. Press the Load Add key and observe that the address specified by the SR is held in the PC, as designated by lighted Program Counter indicators corresponding to switches in the 1 position and unlighted indicators corresponding to switches in the 0 position.

2. Set the SR to correspond with the data or instruction word to be stored at the address just set into the PC. Press the Dep key and observe that the MB, and hence the core memory, hold the word set by the SR.

3. Observe that the contents of the PC have been incremented by 1 so that additional data can be stored at sequential addresses by repeated SR setting and Dep key operation.

To check the contents of an address in core memory, set the address into the PC as in step 2; then press the Exam key. The Memory Buffer indicates the contents of the Address. The contents of the PC are incremented by 1 with the operation of the Exam key, so that the contents of consecutive addresses can be examined by repeated operation of the Exam key after the original (or starting) address is loaded. Any address can be modified by repeating steps 2 and 3.

3.2.3 Teletype Loading Procedures

Information can be stored or modified in the computer under program control. For example, having the RIM Loader stored in core memory allows RIM format tapes to be loaded as follows.

1. Set the Teletype LINE/OFF/LOCAL switch to the LINE POSITION.

2. Load the tape in the Teletype reader by setting the START/STOP/FREE switch to the FREE position, releasing the cover guard by means of the latch at the right, loading the tape so that the sprocket wheel teeth engage

the feed holes in the tape, closing the cover guard, and setting the switch to the STOP position. Load the tape in the back of the reader so that it moves toward the front as it is read. Proper positioning of the tape in the reader finds three channels being sensed to the left of the sprocket wheel and five channels being sensed to the right of the sprocket wheel.

3. Load the starting address of the RIM Loader program (7756_8) into the PC using the SR and the Load Add key.

4. Press the computer Start key and set the 3-position Teletype reader switch to the START position. The tape is read into memory by program control.

The RIM Loader program loads the Binary Loader (BIN) program as previously described. With the BIN Loader stored in core memory, program tapes assembled in Program Assembly Language (PAL III) binary format can be stored as described in the previous procedure, except that the starting address of the BIN Loader (7777_8) is used in step 4. After storing a program in this manner, the computer stops; the AC should contain all 0's if the program is stored properly. If the computer stops with a number other than 0 in the AC, a checksum error has been detected; therefore, the program has been stored incorrectly, and the storage procedure should be repeated. When the program has been stored correctly, initiate it by loading the program starting address (usually designated on the leader of the tape) into the PC using the SR and Load Add key. Then press the Start key.

Table 3-2 Teletype Controls and Indicators

Control or Indicator	Function
REL. pushbutton	Disengages the tape in the punch to allow tape removal or tape loading.
B. SP. pushbutton	Backspaces the tape in the punch by one space, allowing manual correction or rubout of the character just punched.
OFF and ON pushbuttons	Control use of the tape punch with operation of the Teletype keyboard/printer.
START/STOP/FREE switch	Controls use of the tape reader with operation of the Teletype. In the lower FREE position, the reader is disengaged and can be loaded or unloaded. In the center STOP position, the reader mechanism is engaged but de-energized. In the upper START position, the reader is engaged and operated under program control.
Keyboard	Provides a means of printing on paper in use as a typewriter and punching tape when the operator presses the punch ON pushbutton. The keyboard also supplies input data to the computer when the LINE/OFF/LOCAL switch is in the LINE position.
LINE/OFF/LOCAL switch	Controls application of primary power in the Teletype and controls data connection to the processor. In the LINE position, the Teletype is energized and connected as an I/O device of the computer. In the OFF position, the Teletype is de-energized. In the LOCAL position, the Teletype is energized for off-line operation, and signal connections to the processor are broken. Only line use of the Teletype requires that the computer be energized through the POWER switch if primary power for the Teletype is supplied from a source other than the outlet at the back of the computer.

Table 3-3
Readin Mode Loader Program

Address	Octal Content	Tag	Mnemonic	Comments
7756,	6032	BEG,	KCC	/CLEAR AC AND FLAG
7757,	6031		KSF	/SKIP IF FLAG = 1
7760,	5357		JMP .-1	/LOOKING FOR CHARACTER
7761,	6036		KRB	/READ BUFFER
7762,	7106		CLL RTL	
7763,	7006		RTL	/CHANNEL 8 IN AC0
7764,	7510		SPA	/CHECKING FOR LEADER
7765,	5357		JMP BEG+1	/FOUND LEADER
7766,	7006		RTL	/OK, CHANNEL 7 IN LINK
7767,	6031		KSF	
7770	5367		JMP .-1	
7771,	6034		KRS	/READ, DO NOT CLEAR
7772,	7420		SNL	/CHECKING FOR ADDRESS
7773,	3776		DCA I TEMP	/STORE CONTENTS
7774,	3376		DCA TEMP	/STORE ADDRESS
7775,	5356		JMP BEG	/NEXT WORD
7776,	0	TEMP,	0	/TEMP STORAGE

3.2.4 Off-Line Teletype Procedures

The Teletype can operate separately from the PDP-8/I for typing, punching tape, or duplicating tapes. To use the Teletype in this manner follow the procedure described below.

1. Assure that primary Teletype power is ON.

2. Set the Teletype LINE/OFF/LOCAL switch to the LOCAL position.

3. Load the punch as follows. Raise the cover and manually feed the tape from the top of the roll into the guide at the back of the punch. Advance the tape through the punch by manually turning the friction wheel; then close the cover.

4. Energize the punch by pressing the ON pushbutton, and produce about 2 ft of leader. The leader-trailer can be either 200_8 or 377_8 code. To produce the 200_8 code leader, simultaneously press and hold the CTRL and SHIFT keys with the left hand; press and hold the REPT key; press and release the P key. When the required amount of leader has been punched, release the REPT key, then CTRL and SHIFT keys. To produce the 337_8 code leader, simultaneously press and hold both the REPT and RUB OUT keys until a sufficient amount of leader has been punched.

If an incorrect key is struck while punching a tape, the tape can be corrected as follows. If the error is noticed after typing and punching N characters, press the punch B.SP. (backspace) pushbutton N + 1 times and strike the keyboard RUB OUT key N + 1 times. Then continue typing and punching with the character which was in error.

To duplicate and obtain a listing of an existing tape; load the tape to be duplicated in the paper tape reader. Set the LOCAL/LINE switch to LOCAL, turn the punch on, and turn the paper tape reader on.

CHAPTER 4
THEORY

This chapter is divided into four sections and covers the theory of operation of the PDP-8/I Computer. Section I contains a discussion of the theory at a block diagram level; Section II contains a discussion in terms of general theory of operation; Sections III and IV cover detailed memory theory and detailed processor theory, respectively.

SECTION I
BLOCK DIAGRAM DISCUSSION

The following paragraphs discuss the major functional elements of the PDP-8/I as shown on the simplified system block diagram (Figure 4-1).

4.1 REGISTERS

4.1.1 Accumulator (AC)

This 12-bit AC serves as an input/output register for programmed information transfers between core memory and peripheral equipment, and as a transfer register through which arithmetic and logic operations are performed.

4.1.2 Link (L)

This 1-bit register extends the arithmetic facilities of the accumulator and serves as the carry register for two's complement arithmetic.

4.1.3 Program Counter (PC)

This 12-bit register contains the address of the core-memory location from which the next instruction will be taken.

4.1.4 Memory Address Register (MA)

This 12-bit register contains the address in core memory that is currently selected for reading or writing. This address is decoded by the memory selection matrix to permit addressing of all 4096 words of core memory.

4.1.5 Memory Buffer Register (MB)

All data to be written into core memory is loaded first into the 12-bit MB. Through the facilities provided by the major-register gating network, the MB accepts data from any of the major registers in the processor and, during a high-speed data-break transfer, from mass-memory devices. Its only output capability, other than its direct access to core memory, is through the processor interface to optional peripheral equipment.

4.1.6 Sense Register (SENSE or MEM)

All data read from core memory is strobed first into this 12-bit register. It accepts data only from the core memory and transfers data directly to the Instruction Register (IR) and, through the major register gating network, to other registers in the processor.

4.1.7 Instruction Register (IR)

This 3-bit register contains the operation code of the instruction currently being performed by the computer. The three most-significant bits of the current instruction load into the IR from SENSE during a Fetch cycle. The contents of the IR are decoded to produce logic signals for each of the eight basic instructions.

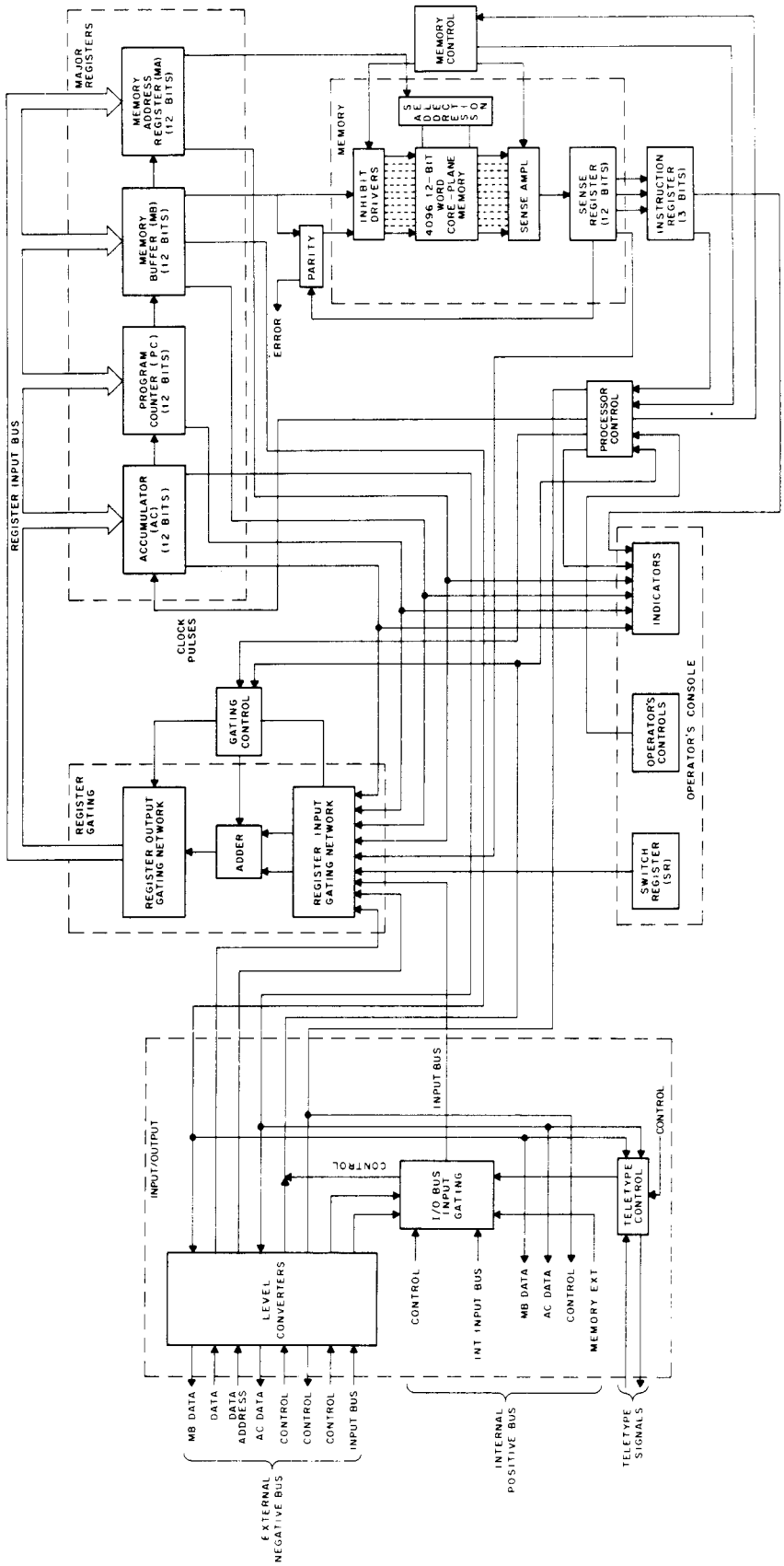


Figure 4-1 System Block Diagram

4.1.8 Switch Register (SR)

The 12-bit SR performs a dual function in that it permits the manual loading of either a discrete core-memory address into the PC or a 12-bit data or control word into core memory. The SR is loaded by 12 toggle switches located on the operator's console. Actuation of either the Load Address or Deposit keys then causes the stored information to be loaded into the PC or MB, respectively.

4.2 MAJOR REGISTER GATING NETWORK

All internal data transfers occuring in the PDP-8/I, except those performed through hardwired facilities, such as MEM→IR and MB→MEMORY, are implemented through the major-register gating network. The network contains a separate gate structure and a common register input bus for the 12 bits. Transfers between registers and into and out of memory, as well as the implementing of logical and two's complement arithmetic functions occur through the network. Data and address information received through the I/O interface also pass through this network.

4.3 TIMING AND CONTROL ELEMENTS

4.3.1 Timing Elements

The processor and memory control circuits in the standard PDP-8/I used fixed and variable delay lines in place of timing clocks. Interleaving of fixed delay sequences provide asynchronous control between the processor and the memory. The overall cycle time of approximately 1.5 μs is determined by the memory timing. For applications involving real time, the KW8/I Real Time Clock option is added to the system.

4.3.2 Control Elements

Circuits in the PDP-8/I control program advance and instruction skipping. These circuits, which operate in response to conditions established in either the processor or peripheral equipment, control the flow of information between registers. In addition, they initiate program interrupt operations during which subroutines enable the servicing of peripheral equipment.

4.4 INPUT/OUTPUT

The PDP-8/I has two bus systems to input/output (I/O) equipment: an internal positive bus; and an external negative bus.

Typical of the available internal bus peripherals are paper-tape reader and punch, punched-card reader, incremental plotter, and CRT display equipment. The control logic elements for these options are located in the PDP-8/I processor logic rack. The control logic interfaces with the internal portion of the I/O bus and does not operate through the I/O level converters.

The data-break peripherals, also optionally available, are represented by mass-memory devices such as magnetic tape, magnetic drum, and disk file systems. These peripherals employ the negative external I/O but to communicate with the PDP-8/I. All of the signals to and from these options pass through the I/O level converters.

A Teletype ASR33 Automatic Send-Receive Set is provided as standard equipment with the PDP-8/I. In addition to a manual keyboard and hard-copy printout facilities, the ASR33 contains an 8-level paper-tape punch and a paper-tape reader, all of which are interfaced with the processor through the Teletype control logic. This logic interfaces with the processor in a similar manner as the internal options.

The PDP-8/I integrated circuits operate on logic levels of 0 and +3V. At the present time, most peripherals contain discrete-component control

elements operating on logic levels of 0 and -3V. To permit proper interfacing with these peripherals, the PDP-8/I contains I/O level converters which produce appropriate changes in the levels of both input and output signals. The output circuits, in addition, provide the necessary line-drive capability to operate over interconnecting cables of reasonable lengths. Refer to the Logic Level and Level Converter discussions in Chapter 2, Section 2.8.3.

4.5 MEMORY

The standard memory supplied with the PDP-8/I is a random access, coincident current, magnetic core memory with a storage capacity of 4096 12-bit words. The core planes and diode matrices that make up the core array are mounted on printed circuit cards. These cards plug directly into the PDP-8/I logic rack receptacles. The Extended Memory Control (MC8I) allows an additional 4K of memory with control for 32K of memory to be installed directly into the logic rack as an option. The additional memory fields above 8K are external as the MM8I option. The major functional elements of the core memory are described in the following paragraphs.

4.5.1 Core Array

The ferrite-core array consists of 12 64 x 64 core planes. This provides a total of 4096 12-bit words of data and program storage. A thirteenth core plane is optionally available to permit a parity bit for each word in memory.

4.5.2 Memory Control

Memory control circuits determine the sequence of operations of the complete read/write memory cycle, starting and stopping each function as required.

4.5.3 Address Selection

The Memory Address register (MA) contains the 12-bit address of the currently selected core-memory location. This address is decoded through the selection switches and the diode matrix to enable passage of read/write currents through specific X and Y drive lines of the memory. The coincidence of these currents select the specific 12-bit core-memory location desired.

4.5.4 Inhibit Drivers

The PDP-8/I memory is so configured that, unless prohibited, all bit locations of the addressed memory cell would be switched to a logical 1 during the write portion of the memory cycle. Inhibit drivers, therefore, are used to ensure that the logic 0 levels stored in the MB will be retained in the corresponding bit locations of the addressed memory cell.

4.5.5 Sense Amplifiers

During the read portion of the memory cycle, sense amplifiers detect analog signals induced in the sense windings of the core array. These signals are amplified and used in conjunction with $\overline{\text{STROBE}}$ to set corresponding bits of the SENSE register.

SECTION II
GENERAL THEORY

The following paragraphs discuss the major functional elements of the PDP-8/I in terms of their operational dynamics. These dynamics will be discussed in greater detail in Sections III and IV.

4.6 TIME STATES/TIME PULSES

Each computer cycle consists of four basic time divisions, T1, T2, T3, and T4, as denoted on the system flow diagrams. Each time division consists of a time state (TS) and its associated time pulse (TP). The time states each extend

throughout their particular time division (TS1, TS2, TS3, TS4) and end with a time pulse (TP1, TP2, TP3, TP4).

In general, the time states generate enabling levels associated with register outputs. Time pulses are used to strobe data into registers.

4.7 MAJOR STATES

The PDP-8/I contains six major-state flip-flops. These are: Fetch, Defer, Execute, Word Count, Current Address, and Break. The outputs of these flip-flops generate enabling levels used within the control elements of the processor to implement particular machine functions.

The first three major states (Fetch, Defer, and Execute) are sufficient to perform most machine functions in the areas of logical operations, arithmetic functions, memory read/write operations, and data transfers through the processor I/O bus. The last three major states (Word Count, Current Address, and Break) are used only for high-speed data transfers through the Data-Break facility.

The processor determines, near the end of each computer cycle, which major state will be needed for the activities to be performed in the next computer cycle. At the very end of the cycle (TP4) the new major-state will be entered by the setting of that particular flip-flop.

4.8 INTERNAL DATA FLOW

The simplified system block diagram shown in Figure 4-1 depicts the flow of data through the major elements of the PDP-8/I. Note that all data transfers into the four major registers (AC, PC, MB, and MA) occur through a register gating network and a common register bus. The outputs of these four registers, plus the SENSE and SR, and the data input from the interface are all connected to the input gates of the major-register gating network.

This permits incoming data to be strobed into any desired major register, or the contents of any register to be complemented, incremented, or transferred into any other major register. The complementing function is implemented by transferring the 0 output of the desired register through the gating network and back into the same register. The incrementing function is performed by transferring the 1 output of the register through the gating network while inserting a carry into the low-order bit of the word. The data is then transferred back into the desired register.

4.9 INSTRUCTIONS

Instruction words are of two types: memory reference and augmented. Memory reference instructions store or retrieve data from core memory, while augmented instructions do not. All instructions utilize bits 0 through 2 to specify the operation code. Operation codes of 0_8 through 5_8 specify memory reference instructions, and codes of 6_8 and 7_8 specify augmented instructions. Memory reference instruction execution times are multiples of the 1.5 μs memory cycle. Indirect addressing increases the execution time of a memory reference instruction by 1.5 μs. The augmented instructions, input-output transfer, and operate, are performed in 4.25 and 1.5 μs respectively. (All computer times are ±20%.)

4.9.1 Memory Reference Instructions

Since the PDP-8/I system contains a 4096-word core memory, 12 bits are required to address all locations. To simplify addressing, the core memory is divided into blocks, or pages, of 128 words (200_8 addresses). Pages are numbered 0_8 through 37_8; each field of 4096-words of core memory uses 32 pages. The seven address bits (bits 5 through 11) of a memory reference instruction can address any location in the page on which the current instruction is located by placing a 1 in bit 4 of the instruction. By placing a 0 in bit 4 of the instruction, any location in page 0 can be

addressed directly from any page of core memory. All other core memory locations can be addressed indirectly by placing a 1 in bit 3 and placing a 7-bit effective address in bits 5 through 11 of the instruction to specify the location in the current page or page 0 which contains the full 12-bit absolute address of the operand.

Word format of memory reference instructions is shown in Figure 4.2 and the instructions perform as follows:

Logical AND (AND Y)

Octal Code: 0
Indicators: AND, FETCH, EXECUTE
Execution Time: 3.0 μs with direct addressing, 4.5 μs with indirect addressing.
Operation: The AND operation is performed between the content of memory location Y and the content of the AC. The result is left in the AC, the original content of the AC is lost, and the content of Y is restored. Corresponding bits of the AC and Y are operated upon independently. This instruction, often called extract or mask, can be considered as a bit-by-bit multiplication.

Example:

Original ACj	Yj	Final ACj
0	0	0
0	1	0
1	0	0
1	1	1

Symbol: ACj ∧ Yj = > ACj

Two's Complement Add (TAD Y)

Octal Code: 1
Indicators: TAD, FETCH, EXECUTE
Execution Time: 3.0 μs with direct addressing, 4.5 μs with indirect addressing.
Operation: The content of memory location Y is added to the content of the AC in two's complement arithmetic. The result of this addition is held in the AC, the original content of the AC is lost, and the content of Y is restored. If there is a carry from AC0, the link is complemented. This feature is useful in multiple precision arithmetic.
Symbol: AC0 - 11 + Y0 - 11 = > AC0 - 11

Increment and Skip if Zero (ISZ Y)

Octal Code: 2
Indicators: ISZ, FETCH, EXECUTE
Execution Time: 3.0 μs with direct addressing, 4.5 μs with indirect addressing.
Operation: The content of memory location Y is incremented by one. If the resultant content of Y equals zero, the content of the PC is incremented by one and the next instruction is skipped. If the resultant content of Y does not equal zero, the program proceeds to the next instruction. The incremented content of Y is restored to memory. The content of the AC is not affected by this instruction.
Symbol: Y + 1 = > Y
If resultant Y0 - 11 = 0, then PC + 1 = > MA

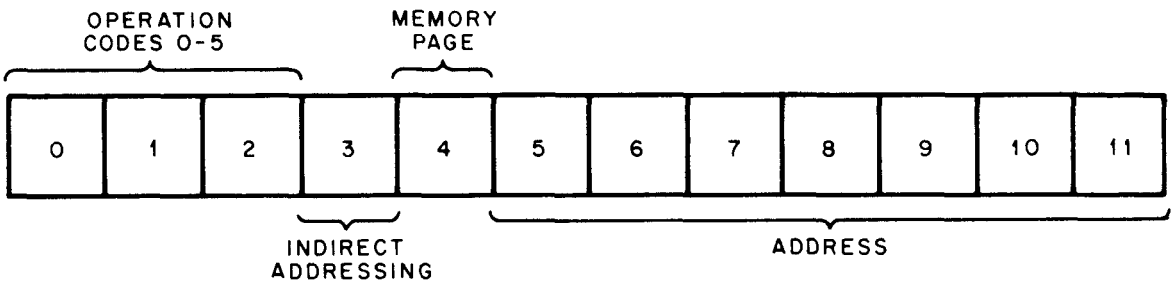


Figure 4-2 Memory Reference Instruction Bit Assignments

Deposit and Clear AC (DCA Y)

Octal Code: 3
Indicators: DCA, FETCH, EXECUTE
Execution Time: 3.0 μs with direct addressing, 4.5 μs with indirect addressing.
Operation: The content of the AC is deposited in core memory at address Y and the AC is cleared. The previous content of memory location Y is lost.
Symbol: AC = > Y
then 0 = > AC

Jump to Subroutine (JMS Y)

Octal Code: 4
Indicators: JMS, FETCH, EXECUTE
Execution Time: 3.0 μs with direct addressing, 4.5 μs with indirect addressing.
Operation: The content of the PC is deposited in core memory location Y and the next instruction is taken from core memory location Y + 1. The content of the AC is not affected by this instruction.
Symbol: PC = > Y
MA + 1 = > PC

Jump to Y (JMP Y)

Octal Code: 5
Indicators: JMP, FETCH
Execution Time: 1.5 μs with direct addressing, 3.0 μs with indirect addressing.
Operation: Address Y is set into the PC so that the next instruction is taken from core memory address Y. The original content of the PC is lost. The content of the AC is not affected by this instruction.
Symbol: Y = > PC

4.9.2 Augmented Instructions

There are two augmented instructions which do not reference core memory. They are the input-output transfer, which has an operation code of 6, and the operate which has an operation code of 7. Bits 3 through 11 within these instructions function as an extension of the operation code and can be microprogrammed to perform several operations within one instruction. Augmented instructions are one-cycle (Fetch) instructions that initiate various operations as a function of bit microprogramming.

4.9.2.1 Input/Output Transfer Instruction - Microinstructions of the input-output transfer (IOT) initiate operation of peripheral equipment and effect information transfers between the processor and an I/O device. Specifically, upon recognition of the operation code 6 as an IOT instruction, the computer enters a 4.25 μs expanded computer FETCH cycle by setting the PAUSE flip-flop and enabling the IOP generator to produce IOP 1, IOP 2 and IOP 4 pulses as a function of the three least significant bits of the instruction (bits 9 through 11). These pulses occur at 1 μs intervals designated as event times 3, 2 and 1 as follows.

Instruction Bit	IOP Pulse	IOT Pulse	Event Time
11	IOP 1	IOT 1	1
10	IOP 2	IOT 2	2
9	IOP 4	IOT 4	3

The IOP pulses are gated in the device selector of the program-selected equipment to produce IOT pulses that enact a data transfer or initiate a control operation. Selection of an equipment is accomplished by bits 3 through 8 of an IOT instruction. These bits form a 6-bit code that enables the device selector in a given device.

The format of the IOT instruction is shown in Figure 4-3.

4.9.2.2 Operate Instruction - With operate instructions, the programmer can consider logical sequences occurring during one computer FETCH cycle. These sequences provide a logical method of forming microinstructions.

The operate instruction consists of two groups of microinstructions. Group 1 (OPR 1) is principally for clear, complement, rotate, and increment operations and is designated by the presence of a 0 in bit 3. Group 2 (OPR 2) is used principally in checking the content of the accumulator and link and continuing to, or skipping, the next instruction based on the check. A 1 in bit 3 designates an OPR 2 microinstruction.

4.9.2.2.1 Group 1 Microinstructions - The Group 1 operate microinstruction format is shown in Figure 4-4 and the microinstructions are explained in the succeeding paragraphs. Any logical combination of bits within this group can be combined into one microinstruction. For example, it is possible to assign ones to bits 5, 6, and 11; although it is not logical to assign ones to bits 8 and 9 simultaneously since they specify conflicting operations.

No Operation (NOP)

Octal Code: 7000
Sequence: None
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: This command causes a 1-cycle delay in the program and then the next sequential instruction is initiated. This command is used to add execution time to a program, such as to synchronize subroutine or loop timing with peripheral equipment timing.
Symbol: None

Increment Accumulator (IAC)

Octal Code: 7001
Sequence: 3
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the AC is incremented

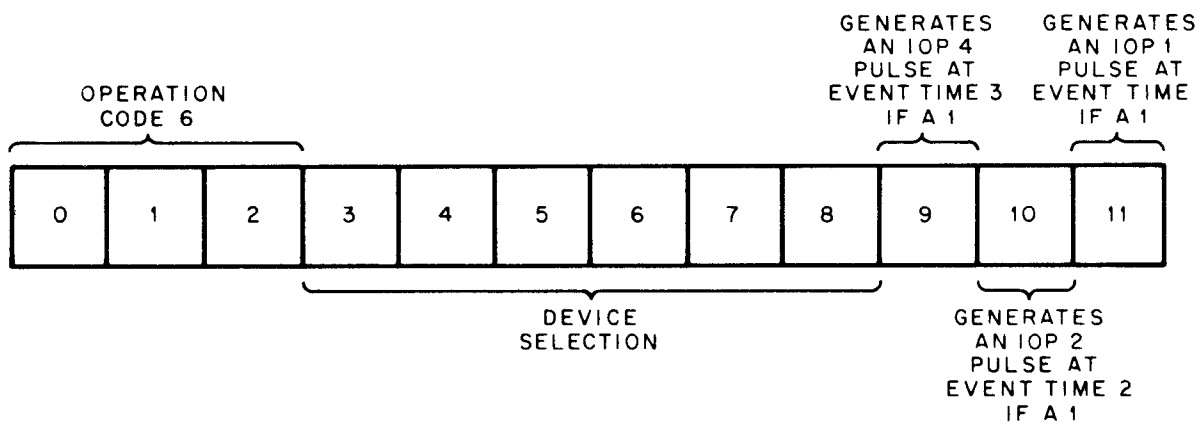


Figure 4-3 IOT Instructions Bit Assignments

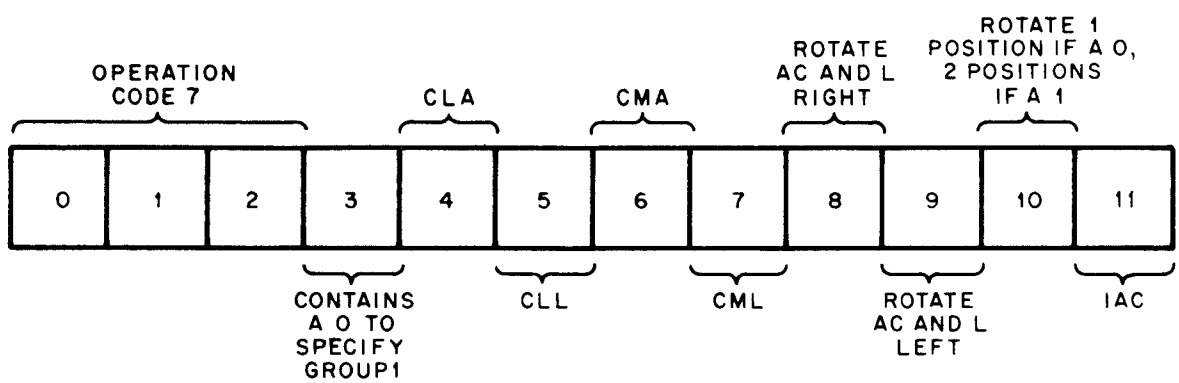


Figure 4-4 Group 1 Operate Instruction Bit Assignments

by one in two's complement arithmetic.
Symbol: AC + 1 = >AC

Rotate Accumulator Left (RAL)

Octal Code: 7004
Sequence: 4
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the AC is rotated one binary position to the left with the content of the link. The content of bits AC1 - 11 are shifted to the next greater significant bit, the content of AC0 is shifted into the L, and the content of the L is shifted into AC11.
Symbol: ACj = > ACj - 1
AC0 = > L
L = > AC11

Rotate Two Left (RTL)

Octal Code: 7006
Sequence: 4
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the AC is rotated two binary positions to the left with the content of the link. This instruction is logically equal to two successive RAL operations.
Symbol: ACj = > ACj - 2
AC1 = > L
AC0 = > AC11
L = > AC10

Rotate Accumulator Right (RAR)

Octal Code: 7010
Sequence: 4
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the AC is rotated one binary position to the right with the content of the link. The content of bits AC0 - 10 are shifted to the next less significant bit, the content of AC11 is shifted into the L, and the content of the L is shifted into AC0.
Symbol: ACj = > ACj + 1
AC11 = > L
L = > AC0

Rotate Two Right (RTR)

Octal Code: 7012
Sequence: 4
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the AC is rotated two binary positions to the right with the content of the link. This instruction is logically equal to two successive RAR operations.
Symbol: ACj = > ACj + 2
AC10 = L
AC11 = AC0
L = > AC1

Complement Link (CML)

Octal Code: 7020
Sequence: 2
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the L is complemented.
Symbol: $\overline{L}$ = > L

Complement Accumulator (CMA)

Octal Code: 7040
Sequence: 2
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the AC is set to the one's complement of the current content of the AC. The content of each bit of the AC is complemented individually.
Symbol: $\overline{ACj}$ = > ACj

Complement and Increment Accumulator (CIA)

Octal Code: 7041
Sequence: 2, 3
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operations: The content of the AC is converted

from a binary value to its equivalent two's complement number. This conversion is accomplished by combining the CMA and IAC commands, thus the content of the AC is complemented during sequence 2 and is incremented by one during sequence 3.
Symbol: $\overline{AC_j} => AC_j$,
then AC + 1 = > AC

Clear Link (CLL)

Octal Code: 7100
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the L is cleared to contain a 0.
Symbol: 0 = > L

Set Link (STL)

Octal Code: 7120
Sequence: 1, 2
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The L is set to contain a binary 1. This instruction is logically equal to combining the CLL and CML commands
Symbol: 1 = > L.

Clear Accumulator (CLA)

Octal Code: 7200
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of each bit of the AC is cleared to contain a binary 0.
Symbol: 0 = > AC

Set Accumulators (STA)

Octal Code: 7240
Sequence: 1, 2
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: Each bit of the AC is set to contain a binary 1. This operation is logically equal to combining the CLA and CMA commands.
Symbol: 1 = > ACj

Group 2 Microinstruction

The Group 2 operate microinstruction format is shown in Figure 4-5 and the primary microinstructions are explained in the following paragraphs. Any logical combination of bits within this group can be composed into one microinstruction.

If skips are combined in a single instruction, the inclusive OR of the condition determines the skip when bit 8 is a 0; and the AND of the inverse of the conditions determines the skip when bit 8 is a 1. For example, if ones are designated in bits 6 and 7 (SZA and SNL), the next instruction is skipped if either the contents of the AC=0, or the content of L = 1. If ones are contained in bits 5, 7, and 8, the next instruction is skipped if the AC contains a positive number and the L contains a 0.

Halt (HLT)

Octal Code: 7402
Sequence: 3
Indicators: OPR, not RUN
Execution Time: 1.5 μs
Operation: Clears the RUN flip-flop at Sequence 3, so that the program stops at the conclusion of the current machine cycle. This command can be combined with others in the OPR 2 group that are executed during either sequence 1, 2, and so are performed before the program stops.
Symbol: 0 = > RUN

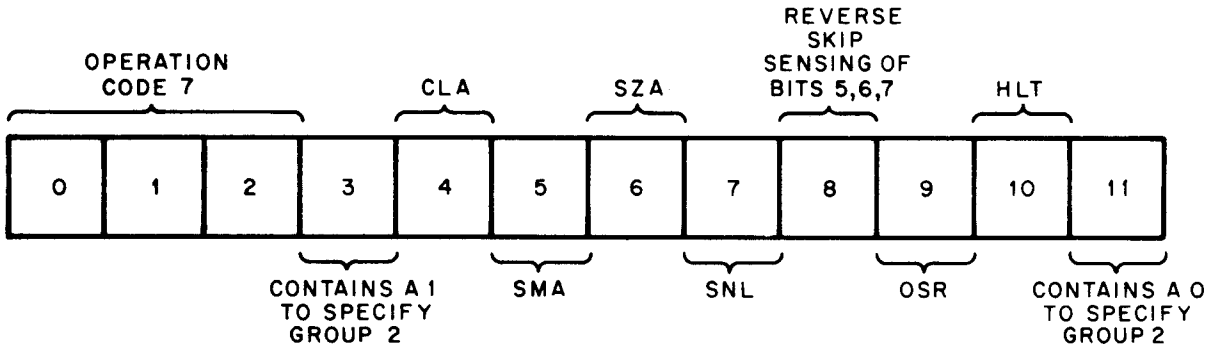


Figure 4-5 Group 2 Operate Instruction Bit Assignments

OR with Switch Register (OSR)

Octal Code: 7404
Sequence: 3
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The inclusive OR operation is performed between the content of the AC and the content of the SR. The result is left in the AC, the original content of the AC is lost, and the content of SR is unaffected by this command. When combined with the CLA command, the OSR performs a transfer of the content of the SR into the AC.
Symbol: ACj V SRj = > ACj

Skip, unconditional (SKP)

Octal Code: 7410
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the PC is incremented by one so that the next sequential instruction is skipped.
Symbol: PC + 1 = > MA

Skip on Non-Zero Link (SNL)

Octal Code: 7420
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the L is sampled, and if it contains a 1 the content of the PC is incremented by one so that the next sequential instruction is skipped. If the L contains a 0, no operation occurs and the next sequential instruction is initiated.
Symbol: If L = 1, then PC + 1 = > MA

Skip on Zero Link (SZL)

Octal Code: 7430
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the L is sampled, and if it contains a 0 the content of the PC is incremented by one so that the next sequential instruction is skipped. If the L contains a 1, no operation occurs and the next sequential instruction is initiated.
Symbol: If L = 0, then PC + 1 = > MA

Skip on Zero Accumulator (SZA)

Octal Code: 7440
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of each bit of the AC is sampled, and if any bit contains a 0 the content of the PC is incremented by one so that the next sequential instruction is skipped. If all bits of the AC contain a 0, no operation occurs and the next sequential instruction is initiated.
Symbol: If AC0 - 11 = 0, then PC + 1 = > MA

Skip on Non-Zero Accumulator (SNA)

Octal Code: 7450
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of each bit of the AC is sampled, and if any bit contains a 1 the content of the PC is incremented by one so that the next sequential instruction is skipped. If all bits of the AC contain a 0, no operation occurs and the next sequential instruction is initiated.
Symbol: If AC0 - 11 $\neq$ 0, then PC + 1 = > MA

Skip on Minus Accumulator (SMA)

Octal Code: 7500
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the most significant bit of the AC is sampled, and if it contains a 1, indicating the AC contains a negative two's complement number, the content of the PC is incremented by one so that the next sequential instruction is skipped. If the AC contains a positive number no operation occurs and program control advances to the next sequential instruction.
Symbol: If AC0 = 1, then PC + 1 = > MA

Skip on Positive Accumulator (SPA)

Octal Code: 7510
Sequence: 1
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: The content of the most significant bit of the AC is sampled, and if it contains a 0, indicating a positive (or zero) two's complement number, the content of the PC is incremented by one so that the next sequential instruction is skipped. If the AC contains a negative number, no operation occurs and program control advances to the next sequential instruction.
Symbol: If AC0 = 0, then PC + 1 = > MA

Clear Accumulator (CLA)

Octal Code: 7600
Sequence: 2
Indicators: OPR, FETCH
Execution Time: 1.5 μs
Operation: Each bit of the AC is cleared to contain a binary 0.
Symbol: 0 = > AC

4.10 PROGRAM INTERRUPT

Some of the I/O devices used with the PDP-8/I require several instructions to complete a data transfer or perform a specified operation. Others are so slow in operation, relative to the computer, that it would consume a prohibitive amount of time to have the computer wait in a skip loop for their operation to be completed. These devices, therefore, employ the program interrupt facility.

When the program enables the program interrupt facility, the computer senses interrupt requests from peripheral devices. The interrupt may also be initiated in response to a programmed IOT instruction.

An interrupt is allowed to occur only on completion of the instruction currently in process and takes effect at the beginning of the following fetch cycle.

A program interrupt is similar, in effect, to a JMS to memory address 0000. The content of PC is saved in location 0000 and the next instruction is taken from location 0001_8. The instruction stored at this location is usually a JMP to a peripheral-servicing subroutine.

After identifying the interrupting device and servicing it, the program enables the interrupt system and then performs a JMP I 0000 (jump to perform the address specified by the content of location 0000) to return to the point at which the program was interrupted.

4.10.1 Instructions

The two instructions associated with the program interrupt synchronization element are IOT microinstructions that do not use the IOP generator. These instructions are as follows.

Interrupt Turn On (ION)

Octal Code: 6001
Event Time: Not applicable
Indicators: IOT, FETCH, ION
Execution Time: 1.5 μs
Operation: This command enables the computer to respond to a program interrupt request. If the interrupt is disabled when this instruction is given, the computer executes the next instruction, then enables the interrupt. The additional instruction allows exit from the interrupt subroutine before allowing another interrupt to occur. This instruction has no effect upon the condition of the interrupt circuits if it is given when the interrupt is enabled.
Symbol: 1 => INT.ENABLE

Interrupt Turn Off (IOF)

Octal Code: 6002
Event Time: Not applicable
Indicators: IOT, FETCH
Execution Time: 1.5 μs
Operation: This command disables the program interrupt synchronization element to prevent interruption of the current program.
Symbol: 0 => INT.ENABLE, INT.DELAY

SECTION III
DETAILED MEMORY THEORY

The following paragraphs discuss memory theory at a detailed level.

4.11 OVERALL MEMORY THEORY

The basic PDP-8/I contains a single 4096-word, 12-bit core memory which performs all normal functions of data storage and retrieval. All necessary control elements for the memory are contained within the basic PDP-8/I.

The memory capacity of the computer can be increased, in increments of 4096 words, to a maximum of 32,768 words with or without parity. If the parity option is selected, the parity bit is carried as the thirteenth bit in each word. Any increase in the size of the memory from the basic 4K configuration, however, necessitates the addition of the Type MC8I Memory Extension Control with the first 4K of extended memory. The first 4K of extended memory is plugged directly into the processor main frame in reserved connector locations. The balance of the extended memory (MM8I options) must be located external to the processor.

Figure 4-6 is a block diagram showing the interrelationship of the major elements of the PDP-8/I memory and its control elements.

4.12 MEMORY OPERATION

PDP-8/I memory operation involves five major functions: address selection, read, inhibit, write, and sense. The memory control provides the timing and initiation of the read, inhibit, write, and sense functions. Address selection is performed by the contents of the MA register, applied through the address selection switches and the X and Y diode selection matrices.

4.12.1 Memory Control

The memory control consists of several series-connected delay lines with associated logic gates and control flip-flops (Drawing BS-8I-0-13). An initiating signal, MEM START, from the central processor proceeds through the delay lines alternately setting, and clearing the various control flip-flops, and returns to the central processor as the MEM DONE signal. The timing for this cycle is fixed by prewired taps on the delay lines. The control flip-flops enable the read/write and inhibit currents in the selected memory field. A variable delay is provided for the STROBE FIELD signal to the sense amplifiers, and for the STROBE signal to the central processor.

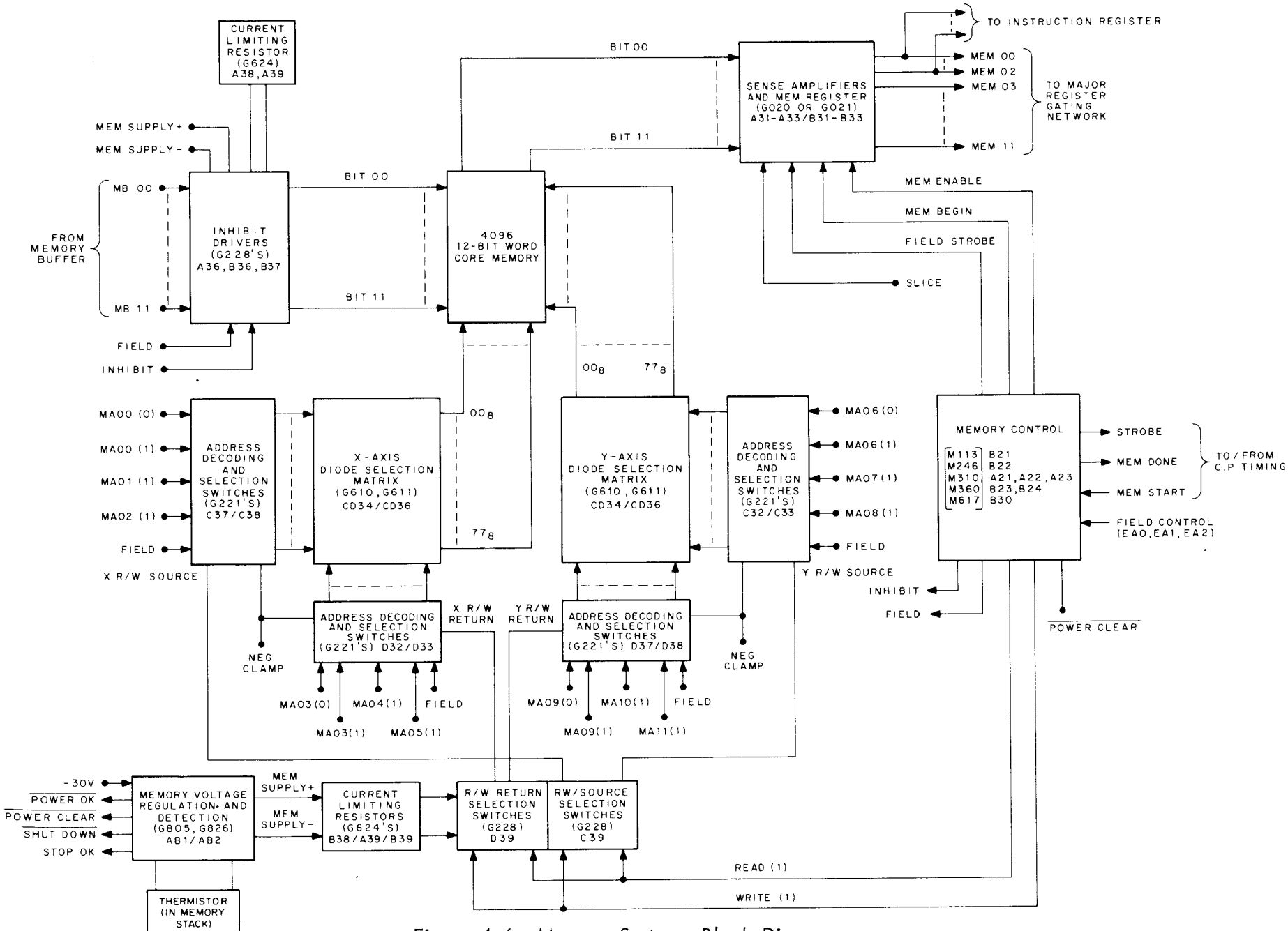


Figure 4-6 Memory System, Block Diagram

The MEM START signal to the memory control delay lines is gated by the field selection signals EA0 and EA1 (for the internal memory fields). MEM START also sets the MEM ENABLE flip-flop when the field selection levels specify either FIELD 0 or FIELD 1. The buffered MEM ENABLE output enables the contents of the appropriate SENSE register to the major register bus. MEM START clocks the FIELD flip-flop which sets when EA2(1) is active (select FIELD 1). The FIELD flip-flop is cleared for FIELD 0 (basic memory) operation.

Each memory control can service two 4K memory stacks. The FIELD flip-flop selects the specific one for a given cycle. The buffered outputs of the FIELD flip-flop enable the address selection current drivers, the Inhibit Drivers, and the variable strobe delay associated with the selected field. After passing through a delay line, MEM START, now a pulse, sets the READ flip-flop. This flip-flop enables the read/write selection switches to provide the read currents to the memory stack. Another tap of the delay line provides the input to the variable delay line enabled by the FIELD flip-flop. The occurrence of both the STROBE FIELD signal to the sense amplifiers and the STROBE signal to the central processor are controlled by this selected adjustable delay.

Further progression of the pulse along the delay line chain clears the READ flip-flop, and sets the INHIBIT flip-flop. The buffered INHIBIT output gates all the inhibit drivers associated with the memory control. Further enabling is effected by the buffered FIELD flip-flops and the specific MB output for a given bit. The WRITE flip-flop is set approximately 50 ns after the setting of the INHIBIT flip-flop. Both of these control flip-flops are cleared by the MEM FINISH signal, which inverts to MEM DONE. While set, the WRITE flip-flop enables the proper read/write switches to provide the write currents to the memory stack.

$\overline{\text{MEM DONE}}$ indicates the end of the memory cycle by setting the MEM IDLE flip-flop in the processor control. This permits the processor/memory cycle to be reinitiated.

The waveshapes of the memory control signals are shown in Figure 4-7. The transition and duration times are approximate due to the delays through pulse amplifiers (about 50 ns), and gates (about 20 ns).

4.12.2 Read/Write

The ferrite core memory consists of 12 planes (13 if the parity option is selected), each containing 4096 ferrite cores arranged in a 64 x 64-core array. Each core assumes a stable magnetic state corresponding either to a binary 1 or a binary 0.

Selection and switching of the cores is provided by four windings traversing each core in the memory in a standard 3D selection technique. An X-axis read/write winding passes through all cores in each of 64 horizontal rows; a Y-axis read/write winding passes through all cores in each of 64 vertical rows; and a sense and an inhibit winding pass through all cores of each of 12 (or 13) planes. Through the use of selection circuits controlling the input of the X and Y read/write windings, any one of the 4096 12-bit word locations can be addressed for writing data into, or reading data out of memory.

The level of the read, write, and inhibit currents passing through these windings is such that no single winding produces a magnetic field strong enough to cause a core to change its magnetic state. This current level is known as the half-select value. Only the reinforcing magnetic field caused by the coincident current of both an X and Y read/write winding can cause the core located at the point of coincidence to change state. It is this principle that allows the relatively simple winding arrangement to select one and only one memory word out of a possible 4096 in each array.

Figure 4-8 shows a simple 4 x 4 core array. The winding scheme shown on this array is identical to that used in the planes of the PDP-8/I memory.

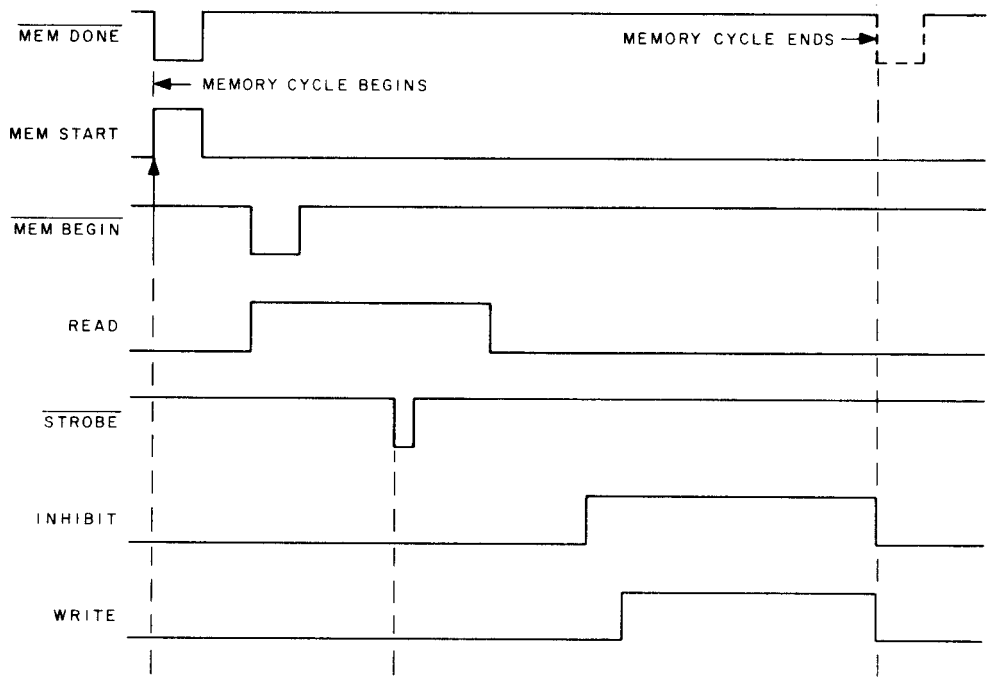


Figure 4-7 Memory Operations Timing Diagram

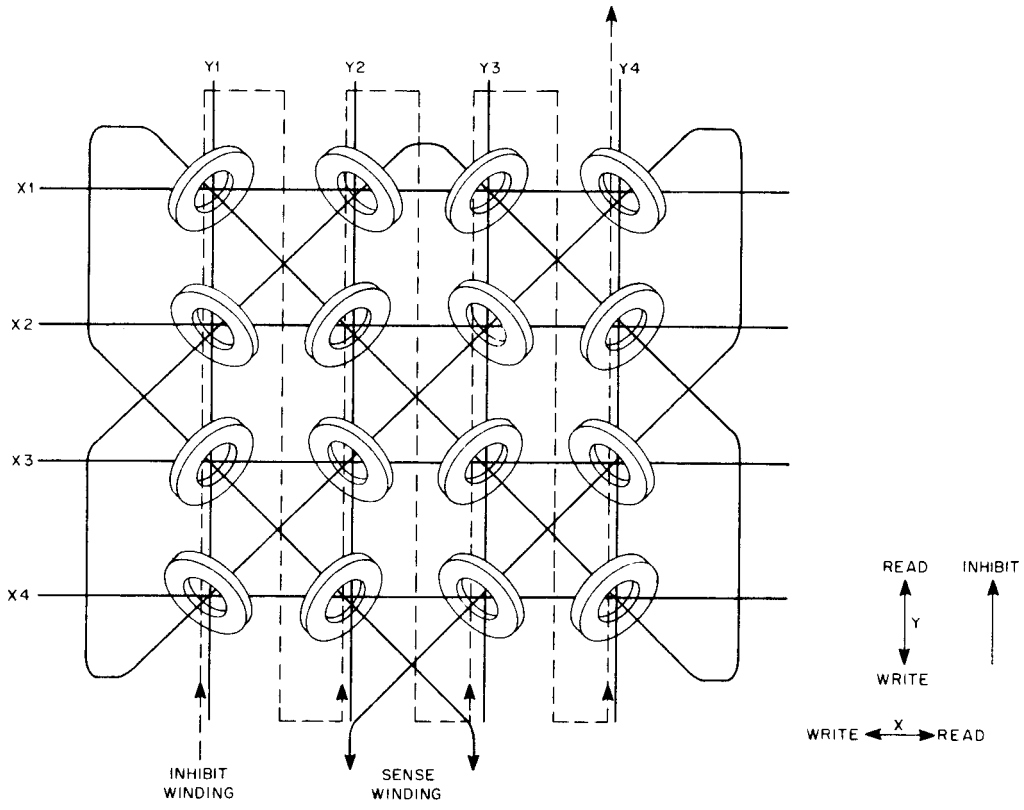


Figure 4-8 Simple Core Memory Plane

A half-select current passing through the X2 winding from right to left (write direction) produces a magnetic field that tends to change all the cores in that horizontal row from the 0 to the 1 state. The flux produced by this current is, however, insufficient to complete the state transition in any core. Simultaneously passing a half-select current through the Y3 winding from top to bottom (write direction) tends to produce the same effect on all cores in that particular vertical row. Note, however, that both currents pass through one core, located at the intersection of the X2 and Y3 windings. This then becomes the selected core.

The X and Y windings are so configured that, when half-select value write currents are passed through each, their resultant magnetic fields add in the core at their point of intersection. Their combined (full-select) current then ensures that the selected core is left in the 1 state.

In the PDP-8/I core memory, the X2 windings of all 12 planes are connected in series, as are the Y3 windings. When X2Y3 half-select write currents flow, therefore, the X2Y3 core on each plane changes to or remains in, the 1 state. This makes each of these cores equivalent to one bit of a 12-bit storage cell.

It should be noted that passing a half-select value write current through a particular pair of X and Y windings produces the 1 state in all 12-bit positions of the selected core-memory storage cell. To store usable information, however, it is also necessary to produce the 0 state during this write cycle in any or all bit location of the selected storage cell. This is performed by the inhibit windings, and the prior occurrence of a read cycle which puts all the cores in the 0 state.

Each inhibit winding, shown as a broken line on Figure 4-8, passes through all cores on a particular plane. Unlike the X and Y windings, in which the read and write currents flow in opposite directions, the half-select value current in the inhibit windings always flows in the same direction. The magnetic field generated by the inhibit current is of a value and polarity which effectively cancels the field generated by the Y-axis half-select write current. This prevents the setting to the 1 state of any core through which inhibit current is flowing.

Each of the 12 inhibit windings is connected to the output of an inhibit driver circuit. Each driver, in turn, is controlled by the output of one bit of the memory buffer register, which contains the data to be stored in memory. When the write operation commences, each inhibit driver connected to an MB bit containing a 0 is activated. The resulting half-current value output of the driver then prevents the writing of a logic 1 in its assigned core plane. The MB bits containing logic 1's disable their respective inhibit drivers. This allows the cores pertaining to these bits to have 1's written into them. The content of the MB is therefore written intact into the selected core-memory storage cell.

To read out information contained in the 12-bit X2Y3 memory cell, half-select read currents are passed through both the X2 and Y3 windings. Since read current flows in the opposite direction of write current, all cores in the X2Y3 cell previously set to the 1 state are switched to the 0 state. Cores already in the 0 state are, of course, unaffected.

A sense amplifier circuit is provided for each of the 12 core planes in the memory. The input to each amplifier is a sense winding which passes through every core on the associated plane. If, during the read operation, the addressed core in a plane makes the 1 to 0 state transition, the flux change induces a current in the sense winding of that plane. This current develops a 40-50 mV voltage pulse at the input to the sense amplifier. This input is amplified, shaped, and after threshold detection, is used to set a SENSE flip-flop connected to the output of the sense amplifier when STROBE is generated.

Addressed cores which were already in the 0 state, when saturated by the full-select read flux, will

induce a limited amount of noise into their sense winding. The voltage level produced by this noise (in the order of 5 mV) will be insufficient to activate the sense amplifier associated with that plane. The SENSE flip-flop for that bit will therefore remain clear, indicating a logic 0 in that location.

Since this type of readout destroys the content of the addressed cell (by switching all cores to 0's), the data stored in the SENSE register will be transferred to the MB for restoration to its original location during the write portion of the memory cycle.

4.12.3 Address Selection

The memory selector switches decode the address specified by the MA and select the proper source and return lines for both the X- and Y-axes. These selection circuits are shown on engineering drawings BS-81-0-15 (X-axis selection) and

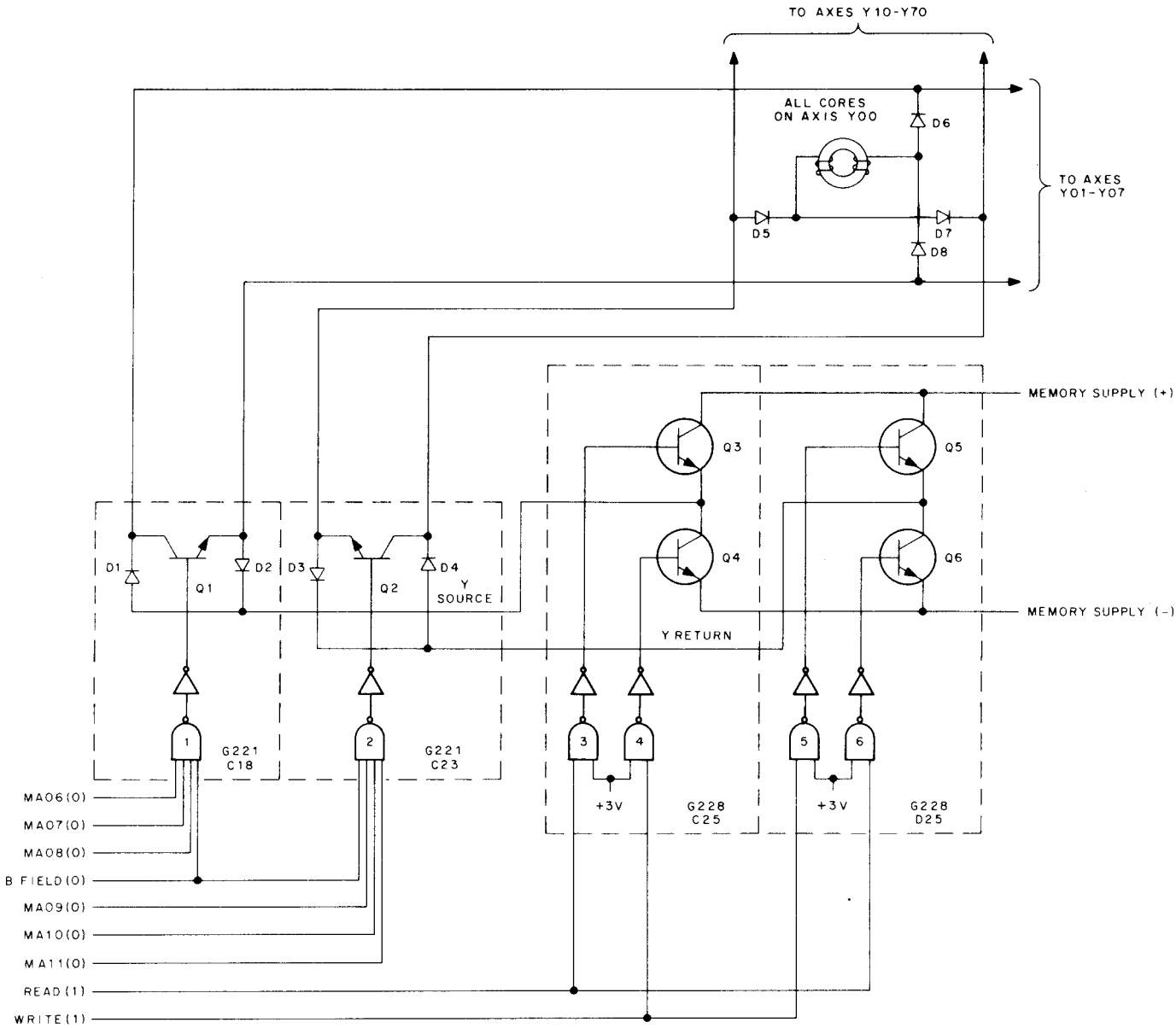


Figure 4-9 Memory Address Selector and Read/Write Current Control

BS-8I-0-16 (Y-axis selection) in Chapter 8 of this document. The polarity of the magnetic field applied to the cores of the addressed cell is determined by the direction of current flow through the core read/write windings. It differs between the read and write cycles. The read/write current selection circuits are shown on engineering drawing BS-8I-0-13 (Memory Control) in Chapter 7 of this document.

Figure 4-9 provides a combined and simplified version of these two drawings showing the selection of a Y-axis memory cell, and the method of determination of read/write current direction. The drawing assumes a content of 0 in bits 6 through 11 of the MA, addressing axis 00 on BS-8I-0-16.

NOTE

The component designation numbers used in Figure 4-9 have been arbitrarily assigned to assist this discussion and do not relate to actual designations.

With a read operation in process, the READ (1) line will be affirmed enabling gates 3 and 6. The outputs of these gates turn on both Q3 and Q6. This establishes a positive source and negative return for the windings of all Y-axis cores serviced by address-selection gates 1 and 2. This is the proper polarity and current direction for a read operation.

With MA06-11=0 gates 1 and 2 have been enabled, and both Q1 and Q2 will conduct. Current then flows through D1 and Q1 to D8. The read current then passes through D8 and the winding of cores on the Y-axis of 00 attempting to switch the cores to the 0 state. There are 12 x 64 cores along this Y-axis; one for each bit and each X-axis. It then passes through D7, Q2, and returns to the Memory Supply (-) through Q6. The current does not pass through the cores on the 01-07, and 10-77Y-axis because of back biased diodes within the matrix and on their address selection switch (BS-8I-0-16).

In a write operation, Q4 and Q5 would be turned on by gates 4 and 5, reversing the direction of current flow. The address selection path now becomes D4, Q2, D5, the cores of the Y00 axis, D6, Q1 and D2. This new current direction attempts to set the cores to the 1 state. As has been previously stated, each core through which write current passes will be set to the 1 state unless the inhibit driver associated with that core has been activated by the sensing of a logic 0 in that particular bit position of the MB.

SECTION IV
DETAILED PROCESSOR THEORY

The functional operation of the PDP-8/I Processor is summarized on the flow diagram FD-8I-0-1. The information on the flow diagram is not detailed enough, however, to facilitate trouble isolation to the level of a module. The object of this section, therefore, is to educate the service technician and to provide him with a means of transition from the flow diagram to the block schematic.

4.13 FLOW DIAGRAM INTERPRETATION

The flow diagram 8I-0-1, illustrates the sequence of events that occur during the manual, stored-program, and automatic operations of the PDP-8/I.

Sheet 1 of the flow diagram, which describes all stored-program and automatic functions, contains six vertical columns, each of which is associated with a particular major state of the computer. The first three columns, Fetch, Defer, and Execute, correspond to the three major states in programmed operation requiring access to the memory or execution of certain logic functions. The last three columns, titled Word Count, Current Address, and Break, correspond to automatic operations during the transfer of information to or from a mass-memory device (data break).

Sheet 2 defines the events that occur during manual operation of the computer. It contains three vertical columns, each of which contains a single key, or group of key operations and their resultant functions.

Horizontal rows on the flow diagram represent *time states*, with time progressing from top to bottom. The upper row represents time one (T1) during programmed and automatic operation and, on Sheet 2, Manual Function Time State 0 (MFTS0) during manual operation.

Events appear on the diagram as rectangular boxes joined by vertical flow lines. Operations in a sequence not specifically designated by a key name or instruction mnemonic are assumed to be common to all sequences (e.g., STROBE, in T1, and MEM DONE, in T4).

Branching of a common sequence into several operation chains, each associated with a specific instruction or key operation, appears as a vertical line terminated by an arrowhead on a horizontal line. Thus, in the Fetch cycle, MEM →IR (contents of the currently addressed memory location - bits 0-2- transferred to bits 0-2 of the Instruction Register) is common to all operations, after which a branch occurs.

In this branch, the decoded output of the IR determines which of the eight basic instructions is in process, permitting selection of the proper branch for the next operation.

The convergence of several sequences into a further common sequence appears as a number of vertical lines with arrowheads that meet a common horizontal line. A single vertical line, descending from the horizontal line to a rectangle, specifies the next common event. Thus, the common sequence associated with manual examine and deposit operations concludes with the branched sequence MEM→MB (examine) and SR→MB (deposit).
Note that some events specified in the rectangles of the flow chart are conditional, others unconditional. Unconditional events appear as information-transfer statements with no indication of register content. For example, LA, ST, EX, and DP key operations begin with the event 0→MAJOR STATES, which occurs in time state MFTS0, and during a load-address operation, SR transfers to the PC in MFTS2 (SR→PC).

Conditional events appear as information-transfer statements, accompanied by one or more indications of the contents of a register which are a condition to the occurrence of the event. For example, during an OPR 1 instruction, several conditional events may occur. These appear in the Fetch cycle in the leftmost sequence occurring during T3. The first event, AC→AC, occurs only if MB bit 04 contains a 0 and MB bit 06 contains a 0. When following the sequence of events in any given instruction, conditional events for which the required conditions are not met should be ignored.

To determine the method by which the processor executes an event specified in the flow diagram, refer to the appropriate engineering logic diagram and the corresponding mechanization chart. When tracing a transfer operation, first examine the control gating of the register to which the transfer is being made. Thus, to trace the operation PC→MA, examine the Major Register (Drawing BS-8I-0-8). Note that an MA LOAD pulse is required to transfer data from the register bus into the MA. An examination of the Register Input Control and Skip (Drawing BS-8I-0-9) shows the gating levels required to complete the transfer of the data through the gating network and onto the register bus. Two levels are required. These are NO SHIFT and PC ENABLE.

The method of generation of the PC ENABLE level is shown on the Register Output Gate Control (Drawing BS-8I-0-4). The PC ENABLE level activates the input gates of the major-register gating network associated with the logic 1 output of the PC. The appearance of this signal initiates a data transfer from the PC onto the major-register gating network bus. The data then passes through

the adder circuit to be transferred to the desired register (in this case the MA).

The method of generation of the NO SHIFT level is shown on the Shift and Carry Gate Control (Drawing BS-8I-0-5). The NO SHIFT levels ensure that the output of the adder circuit is transferred directly into its corresponding output gating network. That is, the output of PC00 will be transferred into MA00. The NO SHIFT level is generated during each data transfer when left or right data shifting or rotation is not desired.

4.14 TIMING

The following paragraphs discuss the internal timing of the processor.

4.14.1 Manual Function Timing Generator

When the computer is initially started, or when data is manually deposited, examined, or continued, or the memory addressed, the processor and/or memory cycles are entered by application of the Manual Function Time signals. These signals include the time-state levels MFTS0, MFTS1, MFTS2, and MFTS3, and timing pulses MFTP0, MFTP1, and MFTP2 (Drawing BS-8I-0-2). The levels and pulses are independent from, and not to be confused with, the processor time-generator signals. The generator and the manual timing signals are described in the following paragraphs. Figure 4-10 shows the timing relationship between the manual timing signals.

When any of the levels $\overline{\text{KEY LA}}$, $\overline{\text{KEY ST}}$, $\overline{\text{KEY EX + DEP}}$ or $\overline{\text{KEY CONT}}$ are activated by pressing one of the associated keys, a low-to-high level transition occurs. The transition is smoothed by an integrating filter which eliminates the noise generated by the closure of the switch (key). To perform this function, a 100 ms delay is incorporated as part of the filter. The filter output activates the ST (Schmitt Trigger) which combines with RUN(0) to generate $\overline{\text{MFTS0}}$. The RUN (0) level controls the timing generator by preventing manual time levels and pulses from occurring if a key is pressed when the computer is running. $\overline{\text{MFTS0}}$ is inverted to produce MFTS0 which sets the MFTS1 flip-flop. MFTS0 also combines with KEY EX + DEP to enable clearing of the RUN flip-flop during T3 of the processor cycle.

Timing level $\overline{\text{MFTS0}}$ generates an MFTP0 pulse. This pulse is delayed 2 μs to produce MFTP1. The MFTP1 pulse sets the MFTS2 flip-flop which clears the MFTS1 flip-flop ending that time state. MFTP1 is delayed 2 μs to generate MFTP2 which clears the MFTS2 flip-flop. When this flip-flop is cleared, the MFTS2 (0) level generates the MFTS3 timing level through a NAND gate.

The applications of the manual function timing levels and pulses are shown on the MAN FUNCTIONS flow diagram (Drawing FD-8I-0-1) and described with the key function.

4.14.2 Manual Operations

The following keys and switches are provided on the operators console: START, STOP, LOAD ADD, CONT, EXAM, DEP, SING STEP, SING INST, and the switch register (SR). All are single keys with the exception of the SR which consists of a bank of 12 switches.

These switches, used singly or in combination, permit manual intervention to start the program, stop the program, load data into a selected memory location, and run the program step-by-step, or instruction-by-instruction for troubleshooting the system, or debugging new programs.

The MAN FUNCTIONS flow chart (Drawing FD-8I-0-1; sheet 2) describes the operation of each of the five active manual control keys (Start, Load Add, Exam, Dep, Cont).

Load Add - The memory location to be addressed is toggled into the switch register, and Load Add keyed. As shown in the flow diagram (Drawing FD-8I-0-1), this clears the major state register. $\overline{\text{MANUAL PRESET}}$, which performs this operation, is generated by the MFTP0 pulse when Load Add is pressed.

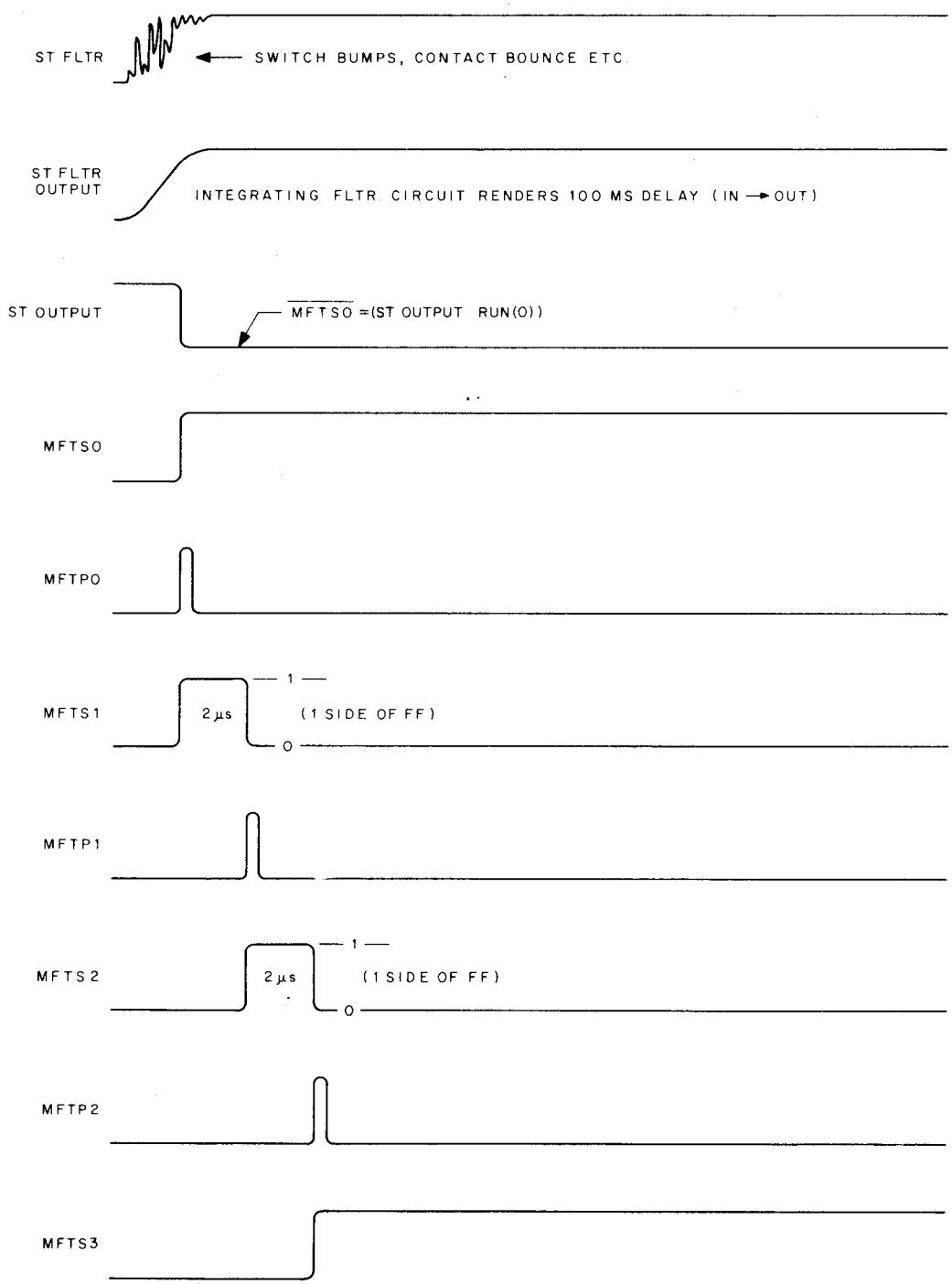


Figure 4-10 Manual Function Timing Diagram

No further operations occur until MFTS2 when the address toggled into the switches is loaded into the PC. This occurs through the generation of an SR ENABLE signal (Drawing BS-8I-0-4) and a PC LOAD signal at MFTP2 (Drawing BS-8I-0-6).

Deposit (DEP) - If data is to be manually loaded into memory, the starting location is placed in the PC, as described with Load Add, and the data is loaded word-by-word through the SR (Switch Register) by the action of Dep.

As shown in the flow diagram (Drawing FD-8I-0-1), pressing KEY DEP clears the major state registers during MFT0 by generating the $\overline{\text{MANUAL PRESET}}$ level with MFTP0. During MFT 1 the contents (starting address) of the PC are transferred to the MA. The generation of the PC ENABLE signal (Drawing BS-8I-0-4) by MFTS1 · KEY ST+EX+DEP, and the MA LOAD signal (Drawing BS-8I-0-6) by MFTP1 · KEY ST+EX+DEP perform this transfer operation.

During MFT2, the address in MA is incremented and transferred back to PC, leaving the current address in MA and placing the next consecutive memory address in PC. This transfer is accomplished by an MA ENABLE level (at MFTS2), and a PC LOAD pulse (at MFTP2). The address is incremented by the insertion of a $\overline{\text{CARRY}}$ into the adder of the least-significant bit during the transfer of the address into PC. This occurs by generating a $\overline{\text{CARRY INSERT}}$ level (Drawing BS-8I-0-5) during MFTS2.

During this transfer operation the memory cycle is started to permit loading of the SR data into the currently addressed memory location. The actuation of any manual key, except LOAD ADD, generates a MEM START pulse at MFTP2 (Drawing BS-8I-0-2), initiating the memory cycle.

During MFTP3, the actuation of Dep generates SR ENABLE (Drawing BS-8I-0-4). This gates the data, as signified by the switch positions, onto the major register bus. At TP2 of each cycle, the MB LOAD pulse (Drawing BS-8I-0-6), permits MB to accept the SR data present on the major register bus. During the write portion of the memory cycle, this data is written into the memory location specified by the contents of MA. This completes the deposit cycle.

Subsequent data is loaded in sequential memory locations by toggling the data into the SR and actuating Dep. The PC is incremented for each deposit, making further addressing unnecessary until such time as access to a non-sequential memory address is required.

Examine (Exam) - The actuation of Exam permits examination of the word in the currently addressed memory location. As shown on the flow diagram (Drawing FD-8I-0-1), this sequence is identical to the previously discussed deposit operation up to the start of the memory cycle.

During the Read portion of the memory cycle, the contents of the address specified, when the Load Add key was pressed, are transferred from core memory to the SENSE register. The memory signal $\overline{\text{STROBE}}$ allows this transfer. $\overline{\text{STROBE}}$ also ends processor time-state TS1, and initiates TS2. At the end of T2, time-pulse TP2 generates an MB LOAD pulse which completes the examine operation by transferring the contents of the SENSE register to the MB. Upon completion of this transfer cycle, the processor stops. This permits the operator to examine the contents of the location addressed by observing the MB indicator lights located on the console panel.

During the Write portion of the memory cycle, the contents of the MB containing the read word are restored to the original memory location. Thus, the contents of the examined address remain in the MB and in the memory location. The contents of the address examined may be modified through the use of the Switch Register and Key Dep. It should be noted, however, that as part of the cycle which extracted the data from memory, the PC was incremented by one to set up the address of the next instruction. The next word,

therefore, would be loaded into the core-memory address next in sequence to the address of the presently displayed word. The 12-bit SR and Key Load Add must therefore be used to set the PC back to the address of the displayed word prior to insertion of the new word.

Start - Pressing key Start initiates execution of a program previously loaded into core memory. When this key is pressed, MFTP0 is combined with the KEY ST level to generate the INITIALIZE and $\overline{\text{INITIALIZE}}$ signals (Drawing BS-8I-0-2) which clears several circuits. The MFTP0 pulse also generates $\overline{\text{MANUAL PRESET}}$.

During MFT1, the contents of the PC transfer through the major register gating network to the MA, and $\overline{\text{INITIALIZE}}$ clears the IOP flip-flops.

During MFT2, the AC, Link, and INT ENABLE (Drawing BS-8I-0-7) flip-flops are cleared, and the FETCH flip-flop is set. The memory cycle is also initiated at this time by the generation of MEM START (Drawing BS-8I-0-2).

The memory signal $\overline{\text{STROBE}}$ is activated during the memory cycle. $\overline{\text{STROBE}}$ continues the automatic sequences of the processor by clearing TS1 and loading TS2 to one. The RUN flip-flop is set by processor time-pulse TP3, and the program instructions are executed until either a halt command is encountered, or the computer is manually stopped.

Stop - Pressing key Stop can halt a program at the end of an instruction. Operation of this key generates a KEY + SI STOP level which clears the RUN flip-flop during TP3. Clearing the RUN flip-flop inhibits generation of the next MEM START pulse preventing another memory cycle.

Sing Step and Sing Inst Keys - Pressing the Sing Step (single step) or Sing Inst (single instruction) keys steps the program one cycle or one instruction respectively at a time. Operating the Sing Step key inhibits the RUN flip-flop from being set. Subsequent actuation of the Cont key generates a single MEM START pulse, but prevents the processor from automatic execution of the program. Each actuation of the Cont key therefore permits a single processor and memory cycle to be executed.

Placing the Sing Inst key in the up position permits setting of the RUN flip-flop but stops the processor, by clearing the RUN flip-flop upon generation of an F SET level, indicating that the next machine cycle is to be Fetch. Under Sing Inst an instruction consisting of more than one processor and memory cycles is completed before a halt.

4.14.3 Time States

Four time-state levels (TS1, TS2, TS3, and TS4) and associated time pulses (TP1, TP2, TP3, and TP4) are generated during each computer cycle. This train of levels and pulses is initiated at the start of each memory cycle and terminated upon its completion. The generation of the time state and time pulses and their relationships are discussed below and are shown in Figure 4-11.

TS1, the first time-state produced (Drawing BS-8I-0-2), is entered at the end of the previous processor cycle by TP4. The memory cycle is also initiated at this time by generating MEM START (Drawing BS-8I-0-2). The duration of TS1 and generation of time pulse TP1 depends on the memory signal $\overline{\text{STROBE}}$ which is produced during the READ portion of the memory cycle. Therefore, the duration of TS1 depends on the memory used, and the $\overline{\text{STROBE}}$ delay adjustment.

During processor time TS1, MEM START initiates the memory cycle by progressing through a delay chain (Drawing BS-8I-0-13). MEM START is delayed to generate MEM BEGIN which starts the read function by setting the READ flip-flop. The memory signal STROBE (Drawing BS-8I-0-13) is generated by an adjustable delay toward the end of the read portion of the memory cycle by MEM

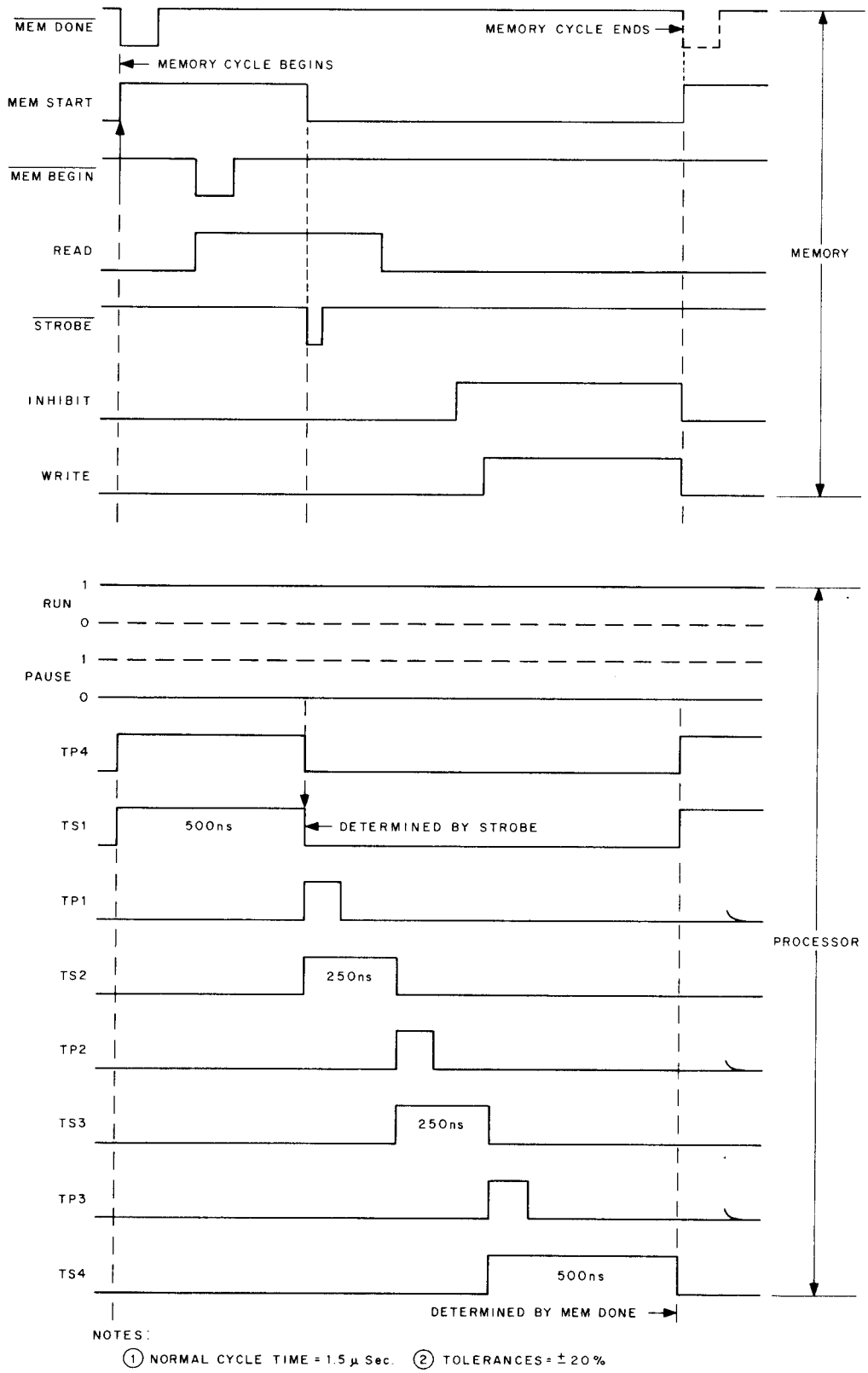


Figure 4-11 System Timing Diagram

START (delayed). STROBE allows the transition from processor time state TS1 to TS2, and generates time pulse TP1 (Drawing BS-8I-0-2). STROBE also clears the MEM IDLE flip-flop disabling MEM START and TP4.

When STROBE generates TP1, the processor-timing-progression continues. TP1 is delayed by 0.15 μs to generate TP2 which clears the TS2 flip-flop and sets the TS3 flip-flop. TP1 is also delayed 0.4 μs to generate TP3. The third time-pulse (TP3) clears TS3, and if the instruction performed is neither an input/output instruction nor an EAE function (both EAE SET and SLOW CYCLE inactive), TP3 sets the TS4 flip-flop.

During processor time-state TS3 the INHIBIT and WRITE control flip-flops (Drawing BS-8I-0-13) are set by the delayed MEM START signal. At the end of the memory cycle MEM START (delayed) generates MEM FINISH which clears the INHIBIT and WRITE flip-flops, and generates MEM DONE.
MEM DONE sets the MEM IDLE flip-flop (Drawing BS-8I-0-2). The setting of this flip-flop allows the generation of MEM START (initiating another memory cycle) and TP4, if the RUN flip-flop is still set and the PAUSE flip-flop is cleared. Time pulse TP4 clears the TS4 flip-flop and sets the TS1 flip-flop, initiating another processor cycle.

The paragraphs above describe the relationships between PDP-8/I processor, and memory timing cycles when the previous cycle initiates the next cycle by generating TP4 and MEM START. This assumes that the PDP-8/I is running. When the computer is initially started (by pressing ST, Load ADD*, DEP, EXAM, or CONT) the processor and memory cycles are entered in the following manner.

The processor timing cycle is initiated by generating MANUAL PRESET which sets the TS1 flip-flop, and simultaneously clears the TS2, TS3, and TS4 flip-flops. MANUAL PRESET is generated by pressing any of the keys mentioned above except for Cont. When this key is pressed, a TP4 pulse is forced, setting the TS1 flip-flop. The clearing characteristic of MANUAL PRESET is avoided. By pressing any of the keys mentioned above with the exception of Load Add, MEM START is generated initiating a memory cycle.
Although normal computer operation is through the use of continuously running stored program, proceeding from instruction to instruction, in sequence, this process must be started manually may be stopped manually, or may be incremented manually either instruction-at-a-time or step-at-a-time for program debugging and maintenance purposes.

4.15 MAJOR STATES

A total of six major states are provided to perform all computer operations. These states are Fetch (F), Defer (D), Execute (E), Word Count (WC), Current Address (CA), and Break (B). The major state occurs in one complete 1.5 μs computer cycle except for Fetch when there is an input/output instruction or an EAE function. The execution of a computer instruction consumes one or more major states, depending upon the operations to be performed. The following paragraphs describe the relationship between the states, their functions and their generation.

Fetch - During this state, an instruction is read into the sense register and the memory buffer at the address specified by the content of the program counter. The instruction is restored in core memory and retained in the memory buffer. The operation code of the instruction is transferred to the instruction register for decoding, and the content of the program counter is incremented by one.

The Fetch state is entered by pressing KEY ST (Drawing BS-8I-0-3). When this key is pressed the Manual Function Time Generated is activated, and MFTP2 (Drawing BS-8I-0-2) is produced. This pulse, combined with a level produced by KEY ST, sets the FETCH flip-flop.

---

*When Load Add is pressed, MEM START is inhibited, therefore, there is no memory cycle.

The Fetch state can also be entered during T4 time, by TP4 of a fetch cycle or an Execute cycle. Entry occurs from a single cycle Fetch cycle including all of the OPR and IOT commands and the JMP command when directly addressed. Entry into the Fetch state occurs from the Execute cycle if BRK REQ occurs. When these conditions are met, F SET (Drawing BS-8I-0-3) is active, enabling the FETCH flip-flop. F SET is produced when the following signals are inactive: D SET, E SET, BREAK OK, and SPECIAL CYCLE.

If a multiple-cycle instruction is fetched, the following major state will be either Defer or Execute. The multiple-cycle instructions include the And, Tad, Isz, Dca, Jms, and the indirectly-addressed Jmp instruction. When the above instructions are directly addressed, the EXECUTE flip-flop is set; when the instructions are indirectly addressed the DEFER flip-flop is set. The following major state entry is executed by TP4 in both cases.

Defer - When a 1 is present in bit 3 of a memory reference instruction, the Defer state is entered to obtain the full 12-bit address of the operand from the address in the current page or page 0, specified by bits 4 through 11 of the instruction. The process of address deferring is called indirect addressing because access to the operand is addressed indirectly, or deferred, to another memory location.

The Defer state can be entered only from the Fetch state when one of the multiple-cycle instructions And, Tad, Isz, Dca, Jms, or Jmp is indirectly addressed (MB03=1). Under these conditions, entry is made during T4 time by TP4 in the Fetch cycle. D SET (Drawing BS-8I-0-3) enables the DEFER flip-flop. It is generated by the active levels MB03=1, and B FETCH (1), and the inactive level IOT + OPR.

If the multiple-cycle instruction being performed is not a JMP command, entry into the Execute state is made from the Defer state. When the JMP instruction is performed with no BRK REQ, the instruction is completed and the Fetch state is entered.

Execute - This state is entered for all memory reference instructions except JMP. During an And, TAD, or Isz instruction the content of the core memory location specified by the address portion of the instruction is read first into the SENSE register and subsequently into the memory buffer and the operation specified by bits 0 and 2 of the instruction (instruction operation code) is performed. During a Dca instruction, the content of the accumulator is transferred into the memory buffer and is stored in core memory at the address specified by the instruction. During a Jms instruction the content of the program counter is written into the core memory address and the address specified by the instruction is written into the program counter to change program control. The Execute state can be entered at the conclusion of the Fetch, Defer, Execute, or Break state if there is a PROGRAM BRK REQ. In this event the EXECUTE flip-flop is enabled by the E SET level (Drawing BS-8I-0-3). With a PROGRAM BRK REQ, E SET is generated by INT OK (Drawing BS-8I-0-7).

Entry into the Execute state can also occur from two other methods. One of these occurs at the conclusion of the Fetch state when the instruction being performed is a directly-addressed multiple-cycle command. When the signals MB03 (0) and B FETCH (1) are active and JMP is inactive, E SET is generated enabling the Execute flip-flop.

The other Execute-state entry method is from the Defer state when any instruction except JMP is performed. In this event, E SET is produced by the DEFER (1) level and the JMP level inactive.

Regardless of the source of the Execute-state entry, the EXECUTE flip-flop is set in the previous major state during T4 by time-pulse TP4.

The Execute state is the last state that a multiple-cycle instruction enters. At the conclusion of

this state the Fetch state is entered again except when the PDP-8/I acknowledges a device requesting any type of break request at this time.

Word Count - This state is entered when an external device supplies signals requesting a data break and the break is a 3-cycle break. When this state occurs, a transfer word count in a core memory location designated by the device is read into the memory buffer, incremented by 1, and rewritten in the same location. If the word count overflows, indicating that the desired number of data break transfers will be enacted at the completion of the current break, the computer transmits a signal to the device. The Current Address state immediately follows the Word Count state.

The Word Count state is entered at the end of each major state excepting the Current Address or the Word Count states when there is a request for a 3-cycle data break. The request is acknowledged during T4 time.

The WORD COUNT flip-flop is enabled in the previous major state by the WC SET level (Drawing BS-8I-0-3). This level is generated by the active 3-CYCLE and BREAK OK levels. The WORD COUNT flip-flop is set by time-pulse TP4 in the previous major state.

At the conclusion of the Word Count state, the combination of WORD COUNT (1) and time-pulse TP4 sets the CURRENT ADDRESS flip-flop to the one state enacting Current Address entry.

Current Address - As the second cycle of a 3-cycle data break, this cycle establishes the address for the transfer that takes place in the following cycle (Break state). Normally the location following the word count is read from core memory into the memory buffer and incremented by one to establish sequential addresses for the transfers, and also transferred to the memory address register to determine the address selected for the next cycle. An inhibit signal (from the data break device) can be supplied to the computer so that the word read during the cycle is not incremented. Incrementation, if it occurs, occurs on the transfer to the memory buffer from core memory. This word is rewritten into core memory at the same location. The Break state immediately follows the Current Address state.

Since the only entry path to the Current Address state is by progressing through the Word Count state with a 3-cycle data break, the CURRENT ADDRESS flip-flop is set during T4 time by TP4 when the WORD COUNT flip-flop is set (Drawing BS-8I-0-3).

Break - This state is entered to enact a data transfer between computer core memory and an external device, either as the only state of a 1-cycle data break or as the final state of a 3-cycle data break. When a break request signal arrives and the cycle select signal specifies a 1-cycle break, the computer enters the Break state at the completion of the current instruction. Information transfers occur between the external device and a device-specified core memory location, through the memory buffer. When this transfer is complete, the program sequence resumes from the point of the break. The data break (one- or three-cycle) does not affect the contents of the accumulator, the link, or the program counter.

The Break state is entered by the active B SET level and time pulse TP4 in the final major state of the current instruction. In this event, B SET (Drawing BS-8I-0-3) is generated by WC SET and BREAK OK level. Synchronization of the asynchronous Break signal (from the device) is done with the BREAK SYNC flip-flop (which is set during TP1 of either a one- or three-cycle break).

Entry into the Break state also occurs when there is a 3-cycle data break. This is the last cycle of this type of break and Break state entry is entered directly from the Current Address state. When this occurs, B SET, generated by CURRENT ADDRESS (0) (Drawing BS-8I-0-3), enables the BREAK flip-flop. This flip-flop is set by time-pulse TP4 in the Current Address state.

At the conclusion of the Break state; the Word Count state is entered if there is still a 3-cycle break, the Execute state is entered if there is a program break, and the Fetch state is reinstated if there is no longer a break request.

4.16 INTERNAL DATA FLOW

When the content of one of the major registers (MB, MA, AC, PC) is transferred or modified, the data flow proceeds as illustrated by Figure 4-12. For ease in understanding the data flow, the sequence of events is described in three steps: (1) source, (2) route, and (3) destination.

4.16.1 Source

As Figure 4-12 illustrates, the 12-bit inter-register transfers are gated into the major register network by enable gates. The basic gating levels include: MA ENABLE, SR ENABLE, PC ENABLE, MEM ENABLE, and AC ENABLE. All enable gates are partially conditioned by processor or manual function time-state levels such as TS2(1), MFTS(1), TS3(1). The enable levels allow the data from one register to enter the major register gating network (Drawing BS-8I-0-9) in a parallel transfer.

4.16.2 Route

After the contents of a register(s) are enabled, the data enters the major register gating network including the adders, and input to the REGister BUS lines (Drawing BS-8I-0-9 all sheets). The major register gating network consists of an upper and lower gating network and adder for each of 12 bits.

The upper-level gating permits the register data to enter the adders by combining major register levels such as AC00(0), DATA00, and MEM03, and other data inputs with an enable level, or levels, depending on the operation performed.

The lower-level gating network includes the adder circuitry, and logic gates for shifting operations. The adder circuits permit propagation of carries, and provide a method of incrementing data as the ISZ, IAC, and MA+1→PC functions require. The data in the upper gating levels passes through the adders to the lower level gates (Drawing BS-8I-0-9; all sheets) whether or not the operation requires a carry or addition.

When any inter-register transfer within the PDP-8/I is performed, excepting rotate, shift operations, and the And instruction, a NO SHIFT (Drawing BS-8I-0-5) is generated. NO SHIFT allows data to pass from the adders directly through the lower level gates to the REGister BUS lines. When a shift or rotate instruction is performed, specific upper and lower gating levels direct AC data to the adder and through a particular lower-level gate to the bus. For example, when the RTR instruction (rotate every AC and Link bit right two places) is performed, AC ENABLE and DOUBLE RIGHT ROTATE levels are generated (Drawing BS-8I-0-5). AC ENABLE allows the AC data into the adders. DOUBLE RIGHT ROTATE directs each adder output two places to the right. For the RTR command, the content of the Link shifts to AC01, AC00 shifts two places to the right (AC02), AC01 shifts to AC03 etc. Identical data shifting into the REG BUS lines to the AC bits occurs with each bit.

When the AND instruction is performed, the complement of the MB is gated in on the lower level by AND ENABLE bypassing the adders. The AC ENABLE signal gates the AC onto the upper network with NO SHIFT gating the contents to the REG BUS lines. A zero appears here if either the MB or AC had a bit at zero.

4.16.3 Destination

All 12-bit inter-register data transfers enter onto the REG BUS lines 00 through 11 after passing through the major register gating network. These

lines are the data input, and the data on these lines is loaded into the specified register by a clocking pulse, i.e., if the AC is the destination, the AC LOAD pulse is generated. There are three other major register load pulses. They are: MB LOAD, MA LOAD, and PC LOAD. Each load pulse occurs at the end of a processor or manual function time period by time pulses such as TP3, TP4, and MFTP1. It should be remembered that time-state levels partially condition the enable levels, and time pulses partially condition the load pulses.

As Figure 4-12 illustrates, data can enter the major register gating network from the MB (AND instruction), MA, PC, AC, SR, the INPUT BUS, DATA BUS, DATA ADDR BUS, or from the SENSE (MEM) register. Data transfers from the REGister BUS, however, can flow only to the MB, MA, PC, or AC.

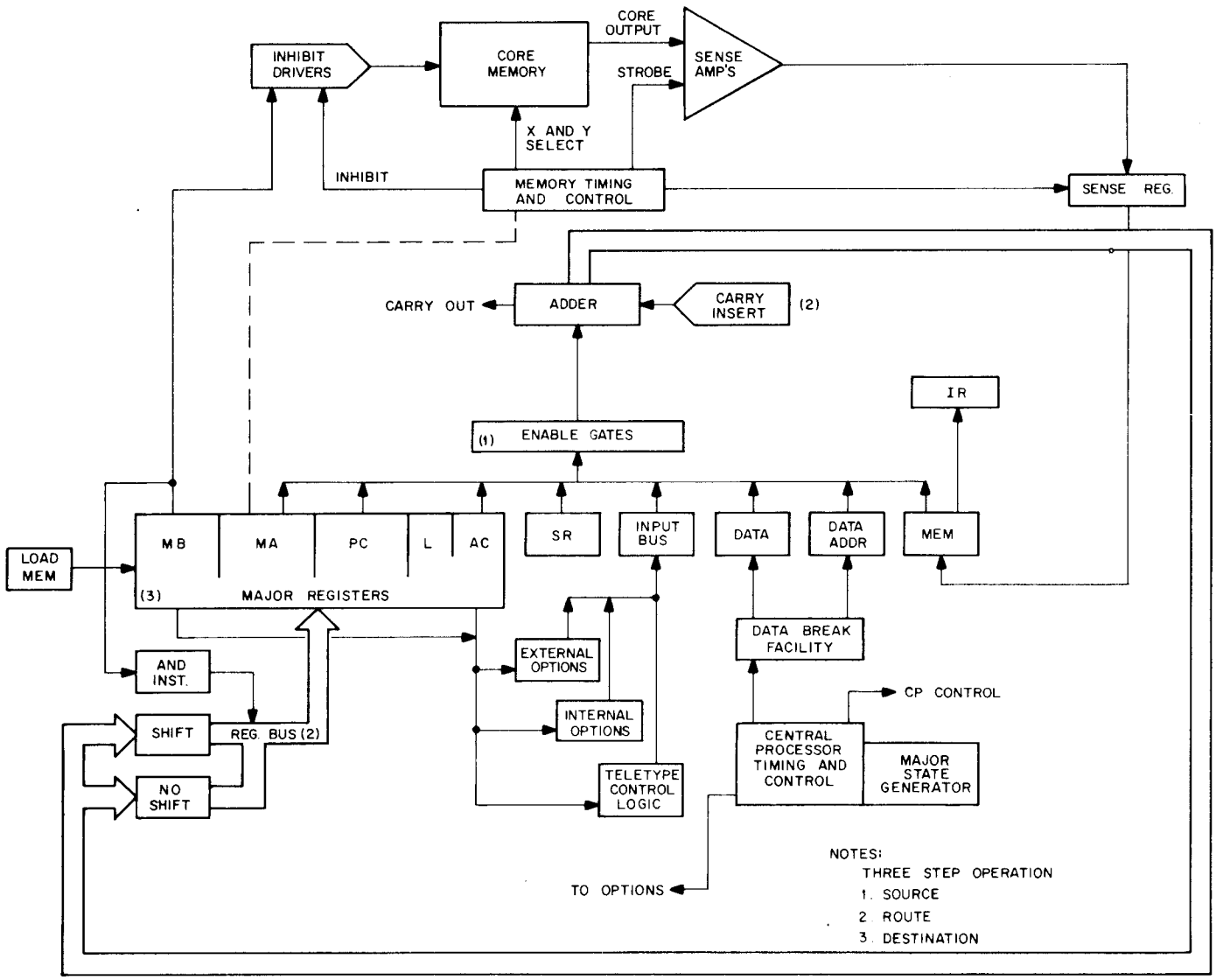


Figure 4-12 Data Flow

Data flow into core memory always occurs through the MB under MA addressing control. For example, when data in the AC is transferred to core memory (DCA instruction), the AC data transfers to the MB through the major-register gating network. The contents of the MB are sampled, inhibited if necessary, and written into memory. The contents of the MA determine the address location in which the write operation occurs. When the contents of a memory location are transferred to the MB, the data flow occurs through the sense amplifiers into the SENSE register. The data then transfers to the IR for decoding, and through the major register gating network to the MB.

The data flow from outside the major registers to one of them occurs through the INPUT BUS for devices such as the Teletype and through the DATA, and DATA ADDR lines for DATA BREAK devices.

4.17 OPERATING INSTRUCTIONS

The following paragraphs describe in detail, the dynamics of the converter operation.

The normal mode of PDP-8/I operation is execution of a prestored programmed instruction sequence. A program interrupt can modify programmed operation, or a data break can temporarily suspend programmed operation. A program interrupt transfers program control from the main program to a subroutine to effect an information transfer with an I/O device or peripheral equipment. A data break is an automatic operation suspending the main program for one or three cycles to permit a high-speed I/O device to exchange information with the core memory.

4.17.1 Instructions

The following explanations of the functions performed during the execution of each instruction assumes that the PDP-8/I is energized and is operating normally under control of the main program. Each explanation begins at the start of the fetch cycle, when the address of the instruction is in the MA and a memory read operation is initiated.

Instructions performed by the PDP-8/I are either memory reference instructions or augmented instructions. A memory reference instruction contains an operation code (in bits 0 through 2) and an address in core memory at which the operation is to occur (in bits 3 through 11). An augmented instruction is used when the operand is already in a register such as the AC; in this case, no memory address is required. Bits 0 through 2 of an augmented instruction contain the operation code which determines the general class of the instruction. Bits 3 through 11 of the instruction contain information which permits the required operations to occur during the two or three execution time states of a single (Fetch) cycle. Operations performed in this manner are said to be "microprogrammed," since several such operations may take place during a single instruction.

4.17.2 Memory Reference Instructions

The format of a memory reference instruction appears in Figure 4-2. With the exception of Jmp instructions which reference a memory address in page 0 or the current page, instructions occur in two cycles: fetch and execute. Instructions which reference any other page require three cycles: a fetch cycle in which the instruction word is brought out of memory and contains the effective address of the operand in the current page or page 0; a defer cycle (refer to Direct/Indirect Addressing in this chapter) in which the absolute address of the operand is brought out of memory and enters the MA; and the execute cycle, in which the operand is brought out of core memory and operated on.

The following explanations of memory reference instructions assume that the instruction is directly addressed; however, the JMP instruction is described with direct and indirect addressing as an example. It is also assumed that no break cycle has been initiated.

AND - The logical And operation occurs between the contents of the addressed memory cell and the contents of the AC. The result is stored in the AC. In effect, each AC bit is compared with the corresponding memory-cell bit. Only when the AC bits corresponding to the addressed memory-cell bits are both a 1 will the particular AC bit remain a 1 at the end of the operation. The logical AND is therefore a transfer of binary 0's. The original contents of the AC are lost.

The sequence of events listed below describe the order in which the And instruction is enacted. The events described in paragraphs a through i and k are common to all memory-reference instructions.

a. The Fetch state is always entered with all instructions at the completion of the last instruction performed. In all cases, the FETCH flip-flop is set during T4 by time-pulse TP4.

b. With all instructions, the functions that occur during T1 and T2 times are the same.

c. During TS1 of the Fetch cycle, the MA is incremented and its contents transferred to the PC by TP1. This will provide the address of the next instruction. The word in the currently addressed location is also read into the sense register at this time.

d. During TS2 the word in the sense register is ready to transfer into the MB. The sense bits 0 through 2 are also enabled to the IR. Time pulse TP2 loads the sense bits 0 through 2 into the IR and the complete word in the sense register into the MB. For the AND instruction, the IR will be loaded with 0's because the AND operation code is 0_8. The contents of the IR (0_8) produce the AND level. This level is used by the processor for input-gating control functions for this instruction.

e. No operations occur for the And, Tad, Isz, Dca, Jms and indirectly-addressed Jmp instructions during T3.

f. During TS4, sense register bits 5 through 11 are enabled to the corresponding MA bits, and during time pulse TP4 these bits are gated into the MA. The same enabling and gating signals affect sense register bits 0 through 4 and MA bits 0 through 4 depending on the status of MB04. This memory buffer bit determines whether the addressed cell is on the current page (MB04=1), or on page zero (MB04=0). If it is on page zero, zeros are transferred into MA00 through MA04.

g. Also, during T4, the next major state entry is determined by the status of MB03. If this bit contains a 1, indirect addressing is indicated and the DEFER flip-flop is set by TP4. If this bit contains a 0, direct addressing occurs and TP4 sets the EXECUTE flip-flop. Only one major state can be entered at any time. The state entered depends on the input gating of each major state controlling flip-flop during TS4.

h. Towards the end of the processor cycle, the memory write function occurs and the instruction is written back into core memory. When this is done, MEM DONE is generated, and TP4 is generated to start another processor cycle and clear TS4. MEM DONE also enables another memory cycle to be initiated.

i. During the strobe portion of the Execute cycle, the operand stored at the address currently held by the MA reads into the sense register. At TP2 the operand transfers to the MB.

j. During TS3 the AND ENABLE level permits the AND-combining of the AC and MB through the major register gating network. Time pulse TP3 sets the AC bits whose register inputs are active (high), and clears the other AC bits.

k. Towards the end of the Execute cycle the operand, which is unaltered in the AND process, is rewritten into memory during the Write portion of the memory cycle. The operand of other instructions such as the Dca, Jms, and Isz is altered before it is rewritten.

i. If there is no break request and SKIP=0, the contents of the PC are loaded into the MA during Execute T4 time by TP4. This time pulse also clears the IR and sets the FETCH flip-flop.

This concludes the logical And operation; the program is ready to fetch the next instruction from the location specified by the contents of the MA.

Two's Complement Add (Tad) - The contents of the addressed memory cell add to the contents of the AC in 2's complement arithmetic. The result of the addition is stored in the AC, and the operand (addend) is restored to memory. The original contents of the AC are lost.

During the Fetch cycle, the TAD instruction operates in the same manner as the AND instruction. Refer to events a through h for these operations. The operation code 1_8 is decoded by the IR to generate the TAD gating level which is used by the processor to implement the TAD operations. The actual two's complement add is performed in the Execute cycle in the following sequence:

a. During the memory strobe portion (T1) of the Execute cycle, the addend reads into the sense register from the addressed memory cell.

b. During T2 the operand in the sense register is transferrred to the MB for rewriting into memory.

c. During T3 the MB and AC outputs are enabled and applied to the major register input gating network. Carries are generated and propagated in the adders as required. Time pulse TP3 allows generation of an AC LOAD pulse during the Execute cycle of the TAD instruction. AC LOAD transfers the sum of the AC and MB into the AC.

d. In T4 of the Execute state, the memory cycle performs the write function. During this portion of the memory cycle the original operand held in the MB is restored to memory at the original address cell. If there is no break request and SKIP=0, the contents of the PC are loaded into the MA at this time by TP4. This time pulse also clears the IR and sets the FETCH flip-flop.

This concludes the TAD instruction; the program is ready to fetch the next instruction from the location specified by the contents of the MA.

Increment and Skip if Zero (Isz) - The Isz instruction reads the contents of the addressed memory cell into the sense register, and transfers the contents of this register through the major register gating network with a carry insert to the MB. If the incremented contents of the MB are not 0, the program proceeds to the next instruction. If the incremented contents of the MB are equal to 0, the contents of the PC increment by 1, and the program skips the next instruction. The events that occur in performing the Isz instruction are listed in sequence below.

a. Operations during the Fetch cycle of an ISZ instruction are similar to those during the Fetch cycle of an AND instruction. Refer to events a through h of the AND instruction. The only difference between these two instructions during the Fetch cycle is the operation code 2_8 decoded by the IR for the ISZ instruction.

b. During T1 of the Execute cycle, the word at the memory location signified by the contents of the MA is transferred into the Sense register.

c. During T2 of the Execute cycle the Sense register is transferred to the MB through the major register gating network. The incrementation occurs through the application of a carry insert level to the adder of the least significant bit during the transfer operation. If this carry insert level produces a carry out from the most significant bit, indicating an all 0 condition in the register, the SKIP flip-flop is set by TP2. TP2 also loads the MB register for rewriting into memory.

d. No event occurs during T3 of the Execute cycle, however, the SKIP flip-flop is set. This allows incrementation of the program counter register through a carry insert into the major registers during T4. The next sequential instruction will therefore be skipped.

e. During the memory write portion of the memory cycle the incremented contents of the MB are written into the address cell from which they were removed.

f. Time pulse TP4 (at the end of TS4) sets the FETCH flip-flop if there is no break request.

Deposit and Clear Accumulator (Dca) - The Dca instruction deposits the contents of the AC into the addressed memory cell and the AC clears. The original contents of the addressed cell are destroyed. The sequence of events that occur in performing the Dca instruction are listed below.

a. Operations during the Fetch cycle of a Dca instruction are similar to those occurring during the Fetch cycle of the And instruction. Refer to events a through h of the And instruction. The operation code for the two instructions differs. The operation code 3_8 decoded by the IR for the Dca command generates a Dca level which is used as a gate-enable signal for this instruction.

b. During T4 of the Fetch cycle the EXECUTE flip-flop is set to allow entry into this state.

c. During T2 of the Execute cycle, the Dca level combined with B EXECUTE (1) inhibits MEM ENABLE 0-4/5-11. This prevents the contents of the Sense register from transferring to the MB. Therefore, the contents of the addressed cell are lost.

The levels TS1(1), Dca, and B EXECUTE (1) combine to generate AC ENABLE during T2. This allows the contents of the AC to transfer through the major register gating bus to the MB. At the end of T2, time-pulse TP2 generates an MB LOAD pulse that allows the contents from the AC to be loaded into the MB.

d. During T3, time-pulse TP3 generates an AC LOAD pulse, however, no enable levels are generated. This lack of enable levels places the equivalent of all 0's on the input to the major register gating network. The AC LOAD pulse therefore, loads these 0's into the AC, effectively clearing the register.

e. During the write portion of the memory cycle the contents of the MB are written into the core location specified by the MA. During T4 if neither a break request nor a skip = 1 level is present, the contents of the PC are transferred to the MA to specify the next desired core location. Time pulse TP4 sets the FETCH flip-flop allowing entry into the Fetch cycle.

Jump to Subroutine (JMS) - The Jms instruction provides an exit from the main program into a subroutine. The contents of the PC (current program count) incremented by 1, are written into the core memory address specified by the Jms instruction. That address transfers to the PC and increments by 1; this incremented address fetches the first subroutine instruction during the next instruction cycle. When the subroutine ends, the main program is reentered by a jump indirect to the address specified by the original Jms instruction. The contents of that address are now the incremented main-program count, and transferring this count into the PC causes the main program sequence to continue.

The sequence of events in performing the Jms instruction are listed below. In addition, to further clarify the Jms operation, a sample program with this instruction is given in Table 4-1.

a. Operations during the Fetch cycle of a Jms instruction are similar to those occurring during the Fetch cycle of the And instruction. Refer to events a through h of the And instruction. The operation code for the two instructions differ. The operation code 4_8 decoded by the IR

Table 4-1
Example of Register Contents During JMS Instruction

Cycle	Time	PC Contents 0-4 (Page)	PC Contents 5-11 (Location)	MEM Contents Address	MEM Contents Contents	MB Contents 0-4	MB Contents 5-11	MA Contents 0-4	MA Contents 5-11	Command
Fetch or Execute	TS4	D	21	D/20	Unknown	Unknown		D	20	PC→MA 1→F
Fetch	TS1	D	21	D/21	JMS/0/100	Unknown		D	21	MA+1→PC Memory→MEM
	TS2	D	22	D/21	JMS/0/100	JMS/0/100		D	21	MEM→MB MEM→IR
	TS3	No operations								
	TS4	D	22	D/21	JMS/0/100	JMS/0/100		JMS/0/100		MB→Memory MEM→MA 1→E
Execute	TS1	D	22	0/100	xxx/x/xxx			0	100	Memory→MEM
	TS2	D	22	0/100	xxx/x/xxx	D	22	0	100	PC→MB
	TS3	0/101		0/100		D	22	0	100	MA+1→PC
	TS4	0/101		0/100		D	22	0	100	MB→Memory PC→MA 1→F
Fetch	TS1	0	102	0/101	1st Subroutine Instruction			0	101	MA+1→PC Memory→MEM

for the Jms command generates a Jms level which is used as a gate-enable level for this instruction.

b. During T1 of the Execute cycle no operations occur.

c. During T2 of the Execute cycle the contents of the PC are transferred to the MB if there is no skip condition (SKIP=0). In order to perform this operation, PC ENABLE and MB LOAD are generated. PC ENABLE allows the contents of the PC to enter the major register bus. The contents of the bus are loaded by MB LOAD into the MB.

d. During T3, the current address is incremented by one and transferred to the PC. To do this, the MA ENABLE, $\overline{\text{CARRY INSERT}}$, and PC LOAD signals are generated. MA ENABLE allows the contents of the MA to enter the major register network bus. $\overline{\text{CARRY INSERT}}$ adds one to the contents of the bus. Finally, PC LOAD loads the PC with the incremented contents of the bus.

e. During the write portion of the memory cycle, the contents of the MB (described in event c of this instruction) are written into memory at the location specified by the Jms instruction.

f. During T4 the FETCH flip-flop is set by TP4 if there is neither a break request nor a SKIP=1 level.

The events above describe the Jms operation. These events are easier to understand, however, if a concrete example is given. The following events describe the sample program of Table 4-1 The program sequence assumes that the main program is in page D of memory (current page), and that the 21st instruction is Jms directly, page 0, location 100_8. The following conditions are also assumed for this example: memory pages are designated 0, A, B, C, D, E; each page contains locations designated 0 through 177_8; and the subroutine is in page 0 starting at location 101_8.

(1) During T4 of instruction 20_8 in the main program, the PC contains the address of the next instruction, location 21_8 in page D (current page). This address is transferred into the MA.

(2) During TS1 of the Fetch cycle for instruction 21_8, the contents of cell D21_8 reads into the sense register. Upon completion of the read operation the sense register contains Jms/ 0/100 (4100_8). The Jms operation code (4_8) is in bits 00 through 02; page 0 is specified by bit 03 = 0 (denoting a direct address) and bit 04 = 0 (denoting page 0). Bits 05 through 11 specify location 100_8 (of page 0). Also during TS1 the contents of the MA increments as it transfers into the PC.

(3) During T2, the contents of the SENSE register (the Jms instruction) transfers into the MB. Bits 00 through 02 transfer into the IR where they are decoded to produce the Jms level.

(4) No operations occur during TS3.

(5) During T4 of the Jms Fetch cycle, the contents of the MB (the Jms instruction) is written back into its original core location (D21_8).

(6) At TP4, the contents of MB05 through 11 is transferred to the corresponding bits of the MA and, because bit MB04 = 0, bits MA00 through 04 are cleared to indicate page 0. The MA now contains ($0/100_8$), the address specified by the Jms instruction.

NOTE

MB03 or MB04 control page and whether instruction is deferred or executed.

(7) During T1 of the Jms Execute cycle, the contents of core location ($0/100_8$), as speci-

fied by the address portion of the Jms instruction (which is now in the MA), reads into the sense register and is lost.

(8) During T2 of the Jms Execute cycle, the contents of the PC, which is $D/22_8$, (address of the next sequential main-program instruction) transfers into the MB. The SKIP flip-flop is assumed as clear.

(9) During T3, the contents of the MA ($0/100_8$), which is the current subroutine address, is incremented by one as it is transferred to the PC.

(10) During T4, the contents of the MB ($D/22_8$) writes into memory location ($0/100_8$). At TP4, the contents of the PC ($0/101_8$) transfer to the MA to select the core memory location containing the first active instruction of the subroutine. At this time, the Jms Execute cycle is terminated and the Fetch cycle of the first instruction of the subroutine is entered.

(11) During T1 of the Fetch cycle, the first instruction of the subroutine reads from core location ($0/101_8$) into sense register. The program then proceeds to execute the subroutine.

(12) The last instruction of the subroutine must be a jump indirect to the location originally specified by the Jms instruction, in this case ($0/100_8$). As noted in step 10 above, location ($0/100_8$) contains the address in core memory of the next sequential main-program instruction ($D/22_8$). By this means the subroutine is terminated and the main program reentered at the point at which it was interrupted.

Jump (Jmp) - The Jmp instruction links two program instructions that are executed consecutively when the instructions are not in sequential locations. This instruction is commonly used to link a program together when the program length extends over more than one page (177_8 locations) of core memory. Jmp is also extensively used in program loops such as counting and comparing in conjunction with the Skip instructions.

The Jmp instruction contains either the absolute core-memory address of the next operand (direct addressing) or the address of a location containing the absolute core-memory address of the next operand (indirect addressing). When the next operand is located either in the current page or page zero of memory, direct addressing is used requiring only a single fetch cycle to extract the operand and prepare for its execution. If, however, the next operand is located in any other page in memory, its 12-bit absolute address must be stored in either the current page or page zero at a location specified by bits 05-11 of the Jmp instruction. This is known as indirect addressing, and requires both a fetch cycle and a defer cycle to extract the operand for processing.

The events that occur in performing the Jmp instruction are listed in sequence below.

a. Operations occurring during T1 and T2 of the Jmp Fetch cycle are identical to those events of the And instruction in the same time periods except for the operation code 5_8 decoded by the IR for the Jmp instruction. Refer to events a through d of the And instruction.

b. Operations during T3 of the Fetch cycle depends upon whether the Jmp specifies direct (MB03=0) or indirect (MB03=1) addressing. If indirect addressing is indicated no operations occur during T3. If direct addressing is indicated, the specified address (SENSE 05 through 11) is loaded into the corresponding bits of the PC. If the instruction specifies that the operand is located on page 0 (MB04=0), bits 00 through 04 of the PC are cleared. If, however, the instruction specifies that the operand is in the current page (MB04=1), bits 00 through 04 of the MA are transferred to the corresponding bits of the PC.

c. The following events occur during T4 of the Fetch cycle of a direct address Jmp (if neither a Break request nor a Skip is specified: PC transfers to the MA, the Jmp instruction is restored intact to its original core-memory location, and

the FETCH flip-flop is set. The operand is removed from core-memory during the next machine cycle (Fetch) and implemented.

d. For an indirectly addressed Jmp during T4, bits 05 through 11 of the sense register transfer to the corresponding bits of the MA and bit 04 of the MB is examined. If MB04=0 (absolute address of operand on page 0), MA00 through MA04 are cleared. If MB04=1 (absolute address of operand on current page), bits 00 through 04 of the MA (current page address) are circulated out of, and back into, the same MA bits. Also during T4, the Jmp instruction is restored intact to its original core-memory location. At the end of T4 the DEFER flip-flop is set, allowing entry into the Defer state.

The following events relate to the Jmp instruction when the Defer state is entered. This state can be entered with any of the memory reference instructions and is not restricted to the Jmp instruction exclusively.

e. During T1 of the Defer cycle, the absolute 12-bit address of the operand is read from the memory location specified by the Jmp instruction (or any of the memory reference instructions) into the sense register.

f. During T2, the contents of the sense register are transferred to the MB if an Auto Index is not required. When there is an Auto Index, the contents of sense are incremented by 1 in the major register gating network before loading into the MB. An Auto Index occurs when a memory reference instruction such as the Jmp command is indirectly addressed in one of the locations 10_8 through 17_8 on page zero of memory.

g. During T3, the contents of the sense register are transferred to the PC (intact if an Auto Index is not specified, and incremented by one if an Auto Index is specified).

h. During T4, if no break request is specified, the contents of the PC are transferred to the MA. Also during T4 the contents of the MB are written back into memory at the original location (intact if Auto Index was not performed, or incremented by 1 if Auto Index was performed). At the end of T4 the FETCH flip-flop is set allowing Fetch cycle entry for the next instruction performed. During the ensuing Fetch cycle the operand is read from memory and its operations begun.

4.17.3 Direct/Indirect Addressing

Six of the eight basic instructions in the PDP-8/I repertoire are designated as memory-reference instructions. These instructions (And, Tad, Isz, Dca, Jmp, and Jms), as part of their function either write into or read from memory.

The first three bits (0-2) of these 12-bit instructions contain the operation code designating the specific function to be performed. The remaining nine bits (3-11) are therefore available to specify the memory location involved in the required operation. A complete specification of any one of the 4096 locations in the basic PDP-8/I memory, however, requires the use of 12 address bits ($2^{12}=4096$). To minimize the number of instructions required to access memory, therefore, both direct and indirect addressing is used in the PDP-8/I.

The memory is organized into 32 pages (or blocks), each containing 128 consecutive memory locations. These pages are numbered 0 through 37_8. The specification of any of the 128 locations on a particular page requires only seven bits ($2^7=128$). Bits 5 through 11 of the memory reference instructions are used for this purpose. With the operation code carried in bits 0 through 2, bits 3 and 4 remain to specify the direct/indirect addressing mode of operation.

The status of bit 3 of the instruction specifies whether direct or indirect addressing is to be performed. When bit 3=0 direct addressing is specified, i.e.; the location specified in bits 5 through 11 contains the operand upon which the

function described in bits 0 through 2 is to be performed. If bit 3=1, indirect addressing is specified; i.e., the location specified in bits 5 through 11 contains the absolute 12-bit address of the operand. An additional computer cycle (Defer) is therefore required to extract the 12-bit address of the operand from the specified address.

The status of bit 4 determines whether the location specified by bits 5 through 11 is on the currently addressed page of memory, or on page 0 (1=current page, 0=page 0). Through the use of bit 4, therefore, a memory reference instruction can address 256 locations; 128 in the current page, and 128 in page 0.

It should be noted that the full, 12-bit absolute address of the desired location must be present in the MA to permit access to that location. The 12-bit starting address of the program is entered into the MA through the switch register and the LOAD ADD key when programmed operation commences. In normal operation, the PC is incremented during each FETCH cycle to step the address to the next sequential memory location. When a memory reference instruction is extracted from memory, only bits 5 through 11 are transferred to the MA. If bit 4 of the instruction is a 1, bits 0 through 4 of the MA are left unchanged and the next location addressed is on the current page. If, however, bit 4 is a 0, bits 0 through 4 of the MA are cleared. This addresses the bit 5 through 11 location on page 0 of the memory.

Figure 4-13 is a simplified flow chart showing the sequence of operation of both the direct and indirect addressing functions.

4.17.4 Augmented Instructions

The two classes of augmented instructions used in the PDP-8/I are: The input/output transfer (IOT) which has the operation code 6_8 and the operate instruction (OPR), which has the operation code 7_8. Augmented instructions are single-cycle (Fetch) instructions which initiate various operations as a function of bit microprogramming.

4.17.4.1 Input/Output Transfer (IOT) - The IOT class of augmented instructions generate pulses IOP pulses) that allows the PDP-8/I processor to communicate with both the internal and external devices. The IOP pulses generated by performing this instruction are used for timing, control applications, synchronizations, and data transfer functions.

The format of the IOT instructions differ from that of the memory reference instructions as illustrated in Figure 4-2. Bits 00 through 02 contain the operation code (6_8), bits 03 through 08 form a code that enables the device selector in a given I/O device, and bits 09 through 11 enable the generation of IOP pulses which control the data-transfer operation.

Two processor IOT instructions ION (6001_8) and IOF (6002_8) do not permit generation of IOP pulses when they are performed. These instructions are used to enable the program interrupt facility (ION) or, disable it (IOF).

The following events described the operation of the IOT instruction (refer to the IOT timing diagram Figure 4-14).

a. Operations during T1 and T2 of the IOT Fetch cycle are identical to the events a through d of the AND instruction, except for the operation code 6_8 decoded by the IR for the IOT instruction during T2. The IR generates the IOT level used in the IOT operations.

b. At the end of T3 of the Fetch cycle, TP3 and SLOW CYCLE (produced by the IOT level during T2 if not a processor IOT) generate IO START which sets the PAUSE flip-flop preventing the generation of another MEM START pulse and resetting of the major-state generator.

c. Approximately 150 ns after the generation of IO START, an IOP 1 pulse is generated for 700 ns(±20%) if MB11=1. After a delay of 200 ns IOP2 is generated for 700 ns (±20%) if MB10=1.

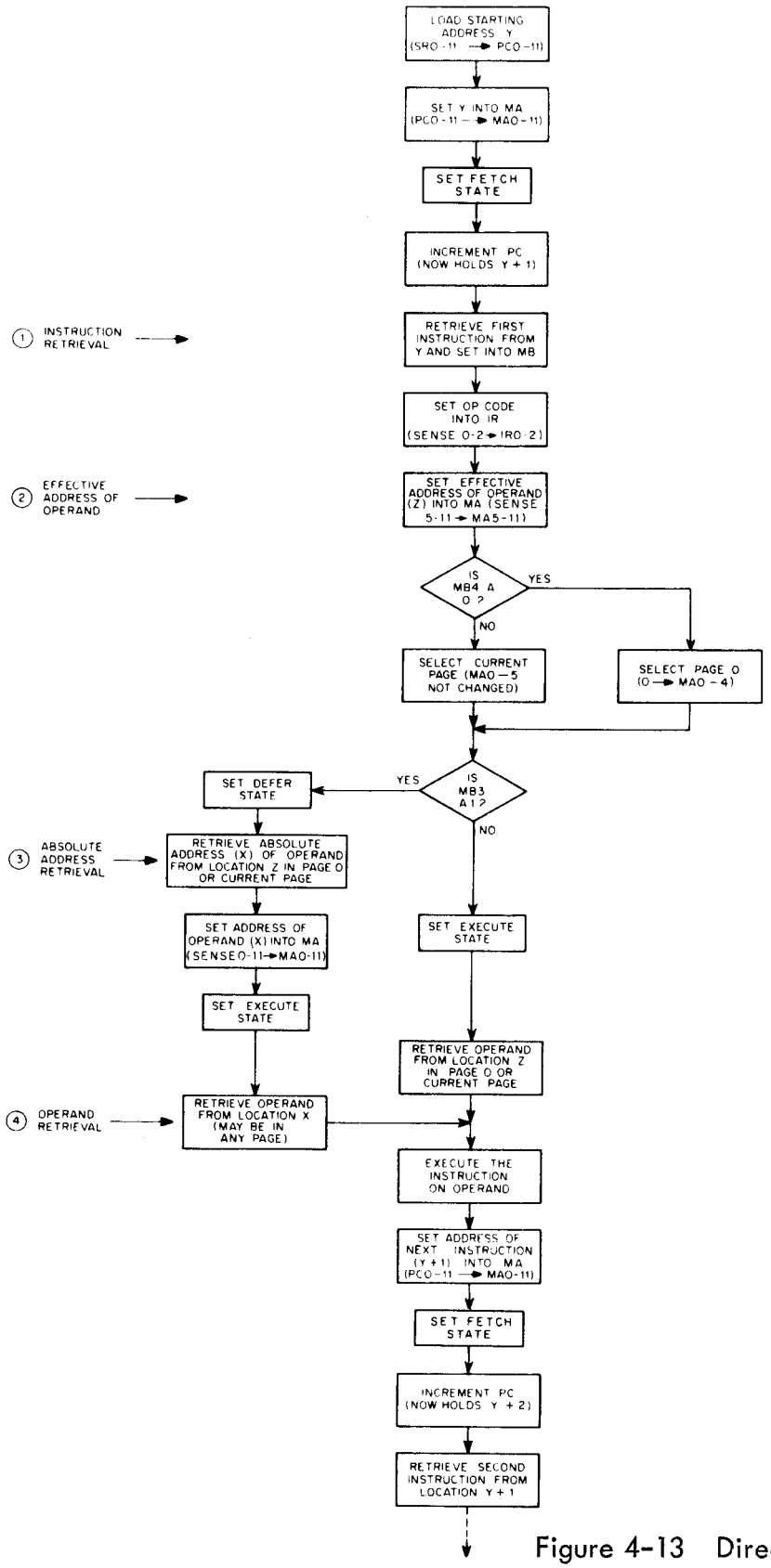


Figure 4-13 Direct and Indirect Address Selection, Simplified Flow Chart

After 200 ns IOP 4 pulse is generated for 700 ns (±20%) if MB09=1. After an additional 350 ns delay, the IO END is generated. This pulse, delayed and additional 300 ns (±20%), clears the PAUSE flip-flop to permit another MEM START level.

d. The IOP pulses are gated in the device selector of the addressed I/O device to produce IOT pulses. The IOT pulses control the operation of the device, effect a transfer of information between the device and the processor, or initiate action in the processor, such as clearing the AC, or incrementing the PC. In addition, with each IOP pulse generated, I/O STROBE is produced allowing AC LOAD (Drawing BS-8I-0-6) to clock data from extended memory or other options into the AC.

Certain IOT instructions are normally combined to clear the AC and transfer data to the accumulator in one computer cycle. This is performed by generating AC LOAD as described above, and disabling AC ENABLE (Drawing BS-8I-0-4). $\overline{\text{AC CLEAR}}$ (Drawing BS-8I-0-4), gated by I/O STROBE, disables AC ENABLE, thus 0's are loaded into each AC bit when I/O STROBE occurs (normally during IOP 2). During IOP 4 time, data is loaded into the AC by I/O STROBE.

4.17.4.2 Operate (OPR) - The OPR class of augmented instructions consist of two categories of microinstructions Group 1 and Group 2. The format of these groups appear in Figures 4-4 and 4-5, respectively. In each case, bits 00 through 02 contain the operation code 7_8. Group 1, de-

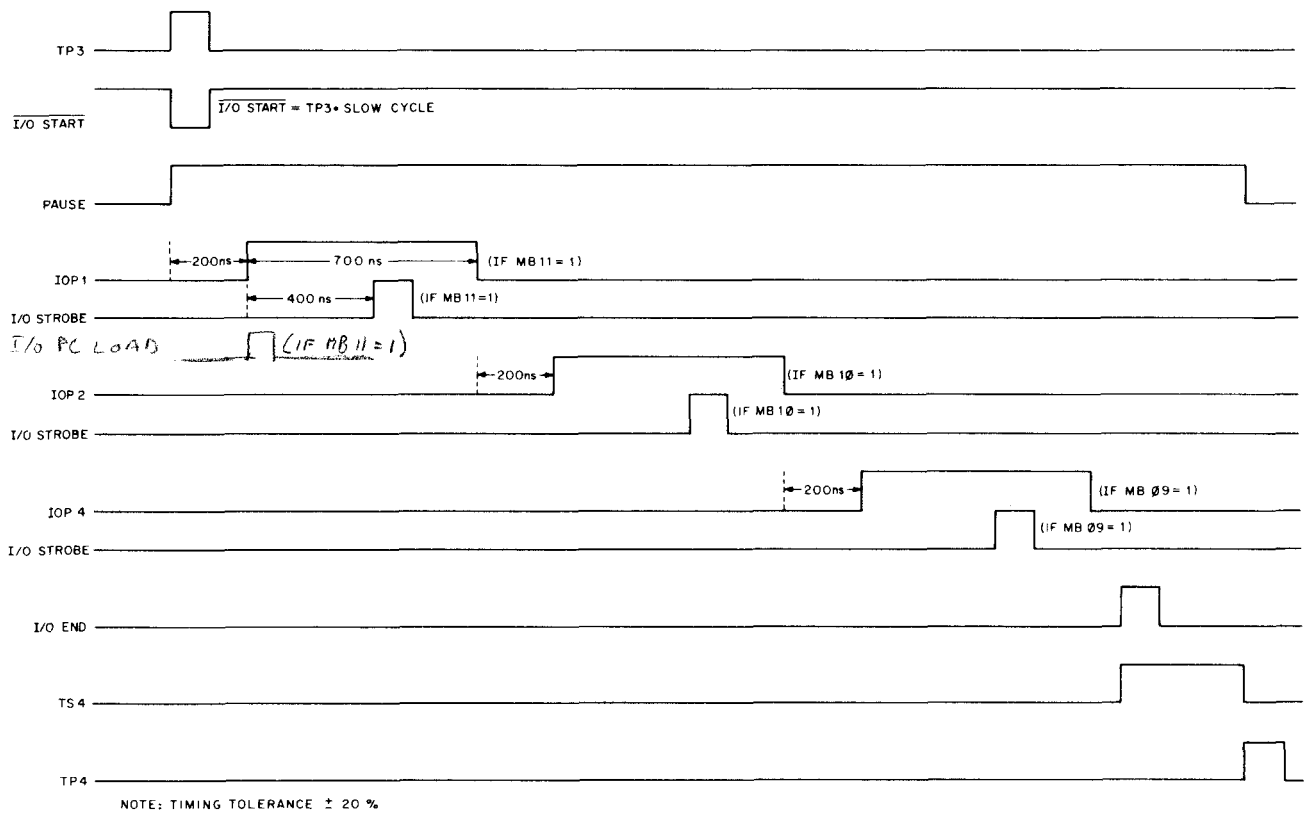


Figure 4-14 IOT Generation

signated by a 0 in bit 03, are called Opr 1 instructions. Opr 1 instructions perform AC and Link operations such as clearing, complementing, rotating, and incrementing. Group 2, designated by a 1 in bit 03 and a 0 in bit 11 are called Opr 2 instructions. These commands check the contents of the AC and Link, and on the basis of the results, determine whether the next sequential instruction is to be performed or skipped.

The Opr 1 operations may occur either singularly or in logical combinations. Care must be exercised, however, to ensure that contradictory or conflicting operations are not specified within the same instruction. For example, bit 08 of the Opr 1 microinstruction, when set to 1 specifies an RAR (rotate AC and L to the right one place) operation. Bit 09 similarly specifies an RAL (rotate AC and L to the left one place) operation. It is physically possible, by setting both bit 09 and 08 to the 1 state, to request an impossible, conflicting operation, i.e., rotate AC and L both right and left one place simultaneously.

Both groups of Opr microinstructions are single-cycle (Fetch) instructions. The following sequence of events describes the operations of this class of instructions. The instruction descriptions contain the primary signals necessary to perform the specified operations. Reference to the mechanization charts indicates where and how the primary signals are generated.

For ease of explanation, the signals and conditions common to most of the Opr instructions are listed below.

a. The operations code is 7_8.

b. Actual AC and L operations occur during T3 of the Fetch cycle.

c. The Opr level decoded by the IR serves as an enabling level, and also combines with MB03 to form OP1 and OP2 levels.

d. Time pulse TP3 generates the load signals such as AC LOAD, that perform the transfer of data from the major register gating network to the indicated register.

e. The major state B FETCH (1) time-state TS3 levels are used as enabling levels.

f. Except for the rotate OPR 1 instructions, the NO SHIFT signal is generated to allow all transfers from the adder of the major register gating network onto the network bus.

g. There is only one OPR 1 NOP instruction (7000_8), however, to conform with the flow chart (Drawing BS-8I-0-1) each bit is discussed in order; therefore, several events occur when this instruction is performed.

The Opr instructions and their operations are described below.

a. During T1 and T2 of the Fetch cycle, the operations that occur are identical to events a through d of the And instruction.

b. During T3, bit 03 of the Opr instruction is examined to determine whether the instruction is an Opr 1 (MB03=0) or an Opr 2 (MB03=1). Appropriate levels are generated in the control circuits to implement the requirements of each group. The descriptions of the instructions that follow should be carefully traced in the flow chart (Drawing BS-8I-0-1) to facilitate understanding of the system operation. The order of appearance of the instruction descriptions parallels the order shown in the flow diagram. It should be noted that, through microprogramming, operations may be combined during the same cycle. For instance; $\overline{AC} \rightarrow AC$ (CMA) and $\overline{L} \rightarrow L$ (CML) may occur in the same instruction to complement both the AC and the Link.

c. During T3, when the Opr 1 instructions are specified (MB03=0), the following instructions can be performed. They are:

(1) NOP (no operation). When the 7000_8 instruction is performed, MB04=0, MB06=0, and no other operand exists. The contents of the AC are circulated through the major register and adder network, and returned unaltered to the AC (AC→AC). This transfer occurs when AC ENABLE, AC LOAD, and NO SHIFT are generated. The events described in 5 and 9 of the Opr 1 instructions also occur simultaneously with the AC circulation.

(2) CMA (complement the AC). When the 7040_8 instruction is performed, MB04=0, MB06=1, and no other operand exists. The 0 side of all AC bits ($\overline{AC}$) is transferred through the major register gating network to the AC ($\overline{AC}$→AC). This transfer occurs when the $\overline{AC}$ ENABLE, NO SHIFT and AC LOAD signals are generated.

(3) CLA (clear the AC). When the 7200_8 instruction is performed, MB04=1, MB06=0, and no other operand exists, logic 0 levels are transferred to all AC bits. This transfer occurs by producing AC LOAD and NO SHIFT levels, with no enable levels existing. This operation clears (sets all bits to 0) the AC register (0→AC).

(4) CLA/CMA (clear and complement the AC). When the 7240_8 instruction is performed, MB04=1, MB06=1, and no other operand exists. The AC is cleared then complemented (AC+ $\overline{AC}$→AC, set AC=-1). This effectively sets all of the AC bits to the 1 state. This operation occurs when the AC ENABLE, $\overline{AC}$ ENABLE, NO SHIFT and AC LOAD signals are generated.

(5) NOP (no operation). When the 7000_8 instruction is performed, MB05=0, MB07=0, and no other operand exists (refer to instruction description 1). The $\overline{L}$→L circulation occurs and the contents of the Link are circulated through the major register gating network and returned unaltered to the Link. This transfer occurs when the L ENABLE, AC LOAD, and NO SHIFT signals are generated.

(6) CML (complement the Link). When the 7020_8 instruction is performed, MB05=0, MB07=1, and no other operand exists. The 0 side of the Link bit ($\overline{L}$) transfers through the major register gating network into the Link ($\overline{L}$→L). This transfer occurs when $\overline{L}$ ENABLE, AC LOAD, and NO SHIFT levels are generated.

(7) CLL (clear the Link). When the 7100_8 instruction is performed, MB05=1, and no other operand exists, and logic 0 level is transferred to the Link. The operation is performed by generating AC LOAD and NO SHIFT signals with no enable levels existing.

(8) STL (set the Link). When the 7120_8 instruction is performed, MB05=1, MB07=1, and no other operand exists, the combined operations of the CML and CLL instructions described in 6 and 7 above occur ($\overline{L}$ + L → L). The Link is set to the 1 state by clearing it, then complementing it.

(9) NOP (no operation). When the 7000_8 instruction is performed, MB03=0, MB09=0, and no other operand exists. In addition to the events described for this instruction in 1 and 5, the NO SHIFT level is generated. This level controls transfers of all data from the adder onto the major register bus. This signal is generated for all OPR 1 instructions, and is gated by MB03=0, and MB09=0. It should be noted that the NO SHIFT signal is also generated for all instructions except for the EAE shift group, but there is a different path activated.

(10) RAR (rotate the AC and Link right one place). When the 7010_8 instruction is performed, MB08=1, and no other operand exists. The contents of the AC and L are shifted one bit to the right. The Link status is transferred into AC00, while AC11 status transfers into the Link. The RAR operation occurs when the L ENABLE, AC ENABLE, RIGHT SHIFT and AC LOAD signals are generated.

(11) RAL (rotate AC and the Link left one place). When the 7004_8 instruction is performed, MB09=1, and no other operand exists. The contents of the AC and Link are shifted one bit to the left. The content of AC00 is transferred to the Link, and the content of the Link bit is transferred to AC11. The RAL operation is performed when LEFT SHIFT, L ENABLE, AC ENABLE, and AC LOAD signals are generated.

(12) RTR and RTL (double rotate, right or left). When the 7012_8 instruction (RTR) or the 7006_8 instruction (RTL) is performed, MB10=1, and the operand for either the RAR or RAL instruction exists. However, to avoid conflicting operations, only one of these operands should exist at one time. The RTR command rotates the contents of the AC and Link two places to the right. Similarly, the RTL command rotates the contents of the AC and Link two places to the left.

Both the RTR and RTL instructions occur when L ENABLE, AC ENABLE, DOUBLE RIGHT ROTATE or DOUBLE LEFT ROTATE, and AC LOAD are generated.

(13) IAC (increment the AC). When the 7001_8 instruction is performed, MB11=1 and no other operand exists. The AC is incremented by 1, by transferring the contents of the AC to the major register gating network, adding a logic 1 through the network adder, and transferring the incremented contents into the AC. The AC ENABLE, AC LOAD, and CARRY INSERT signals are generated to perform this instruction.

The IAC command can be combined with either the CLA or CMA commands to perform the functions of both during one cycle. When the CLA/IAC instruction or the CIA (IAC/CMA) instruction is performed, the operations of both separate instructions are performed simultaneously, during T3.

4.17.4.3 OPR 2 - When MB03=1, and MB11=0, the OPR 2 group of microinstructions is specified. These microinstructions may be performed singly or in useful logical combinations. The available commands include: CLA, HLT, OSR, and seven skip instructions dependent upon the status of the AC and/or Link.
The operations performed during T1 and T2 of the Fetch cycle of the Opr 2 class of instructions are identical to those previously described for those time states in the AND instruction. The following instruction descriptions therefore, describe the Opr 2 instructions during T3 of the Fetch cycle.

(1) SZA (skip on zero AC). When the 7440_8 instruction is performed MB06=1 and no other operand exists. The contents of the AC register are checked and if the AC=0, the next sequential program instruction is skipped. When the AC=0, the SKIP flip-flop is set to the one state by TP3, B FETCH (1), Opr, and a skip-enable level generated by And-combination of all AC 0-side outputs.

(2) SMA (skip on minus AC). When the 7500_8 instruction is performed MB05=1 and no other operand exists. The content of AC00 is sensed to determine its status. If AC00=1, a minus AC is specified, the SKIP flip-flop is set, and the next sequential program instruction is skipped. This flip-flop is set by a load pulse generated by the TP3, B FETCH (1) and Opr signals. The skip-enable level is produced by MB05 (1) combined with AC00 (1).

(3) SNL (skip on non-zero Link). When the 7420_8 instruction is performed, MB07=1 and no other operand exists. The Link bit is sensed to determine whether it is in the 1 state, and if it is, the next sequential program instruction is skipped. The SKIP flip-flop is enabled by LINK (1), and MB07 (1), and set by the combination of TP3, B FETCH (1), and Opr signals.

(4) SKP (skip unconditionally). When the 7410_8 instruction is performed, MB08=1, and no other operand exists. The next sequential

instruction is skipped regardless of the contents of the AC and Link. The SKIP flip-flop is enabled by MB08 (1), MB11 (0), and OP2; it is set by the combination of TP3, B FETCH (1), and Opr.

When one of the SZA, SMA, or SNL operands is combined with MB08 (1), reverse sense skipping occurs, i.e., SZA becomes SNA (skip on non-zero AC;7450_8), SMA becomes SPA (skip on plus AC;7510_8), and SNL becomes SZL (skip on zero Link;7430_8).

(5) HLT (halt operation). When the 7402_8 is performed, MB10=1, and no other operand exists. This operation clears the RUN flip-flop, inhibiting the generation of a MEM START level which prevents the start of another machine cycle. The computer stops after T4 time.

(6) OSR (inclusive OR between the SR and AC). When the 7404_8 instruction is performed MB09=1, and no other operand exists. The result of the inclusive OR remains in the AC. The OSR operation occurs when the AC ENABLE, AC LOAD, SR ENABLE, and NO SHIFT signals are generated.

(7) CLA (clear the AC). When the 7600_8 instruction is performed MB09=1 and no other operand exists. The AC is loaded to all 0's (0→AC). This instruction is identical to the Opr 1 CLA instruction with the exception of the OP 2 and MB04 levels. The CLA instruction (Opr 2) exists in order to combine with the Opr 2 microinstructions. As a result the AC can be cleared after it is sensed by one of the skip instructions, or combination between the OSR and the CLA may be made resulting in the LAS (load the AC with the contents of the SR).

(8) NOP (No Operation). When the 7400_8 instruction is performed, MB03=1, and no other operand exists. The same events described for the NOP conditions of the Opr 1 instruction descriptions 1, 5, and 9 occur.

The Opr 2 instructions listed above may be logically combined to perform more than one operation in a single Fetch cycle. Examples of the combined microinstructions are listed below; however, many other useful combinations exist.

(1) SZA CLA (7640_8). When this instruction is performed, the content of the AC is sensed. If each AC bit is a binary 0, the next instruction is skipped and the AC is cleared.

(2) SNA SZL (7470_8). When this instruction is performed the next sequential instruction is skipped if both the AC $\neq$ 0 and the Link = 0.

CHAPTER 5
MAINTENANCE

This chapter contains information pertinent to preventive maintenance, corrective maintenance, and troubleshooting techniques, of the PDP-8/I.

5.1 EQUIPMENT

Table 5-1 lists the equipment and relevant specifications needed for maintenance of the basic PDP-8/I. Also included in the list is the equivalent equipment used by Digital Equipment Corp. field service personnel.

5.2 PROGRAMS

Table 5-2 lists the Maintenance Programs supsplied by DEC for ascertaining proper operation of the PDP-8/I.

Table 5-1
Maintenance Equipment

Equipment	Specifications	Equivalent
Multimeter	10K ohms/volt-20K ohms/volt	Triplett Model 310
Oscilloscope	dc to 50 mc with calibrated deflection factors from 5 mV to 10V/div. Maximum horizontal sweep rate of 0.1 μs/div. Delaying sweep is desirable and dual trace is a necessity.	Tektronix Type 453
Probes	X10 with response characteristics matched to oscilloscope	Tektronix Type P6010
Clip-on current probe	2 mA/mV or 10 mA/mV	Tektronix Type P6019 with passive terminator
Recessed Probe Tip		Tektronix
Unwrapping tool		Gardner-Denver 505-244-475
Wire-Wrap Tool		Gardner-Denver A-20557-29
30 gauge bit for wrap tool		Gardner-Denver 504221
Sleeve for 30 gauge bit		Gardner-Denver 500350

Table 5-1
Maintenance Equipment (Cont)

Equipment	Specifications	Equivalent
Spray paint		Krylon 1501 Glossy white
Spray paint		DEC black
Module Extender (2)		DEC No. W982
Jumper Wires		Assorted lengths affixed with 30 gauge termi-point connectors

Table 5-2
Maintenance Programs*

Program Name	DEC No.	Use
Instruction Test 1	Maindec 8I-D01B	Tests And, Tad, and operate Instructions only
Instruction Test 2	Maindec 8I-D02B	Extensive test of Auto index, indirect address and the DCA instruction
Instruction Test 2B	Maindec 08-D02A	Tests 2's add and rotate logic
Random Jmp Test	Maindec 08-D04B	Extensive test of Jmp instruction
Random Jmp-Jms Test	Maindec 08-D05B	Extensive test of Jms instruction
Random ISZ Test	Maindec 08-D07B	Extensive test of ISZ instruction
Memory Checkerboard	Maindec 08-D1J0	Tests memory circuits susceptibility to noise
Memory Address Test	Maindec 08-D1B0	Tests address selection logic
Memory Power On/Off Test	Maindec 08-D1AB	Test ability to retain memory information during loss of power

*Programs are subject to change

5.3 PREVENTIVE MAINTENANCE

A systematic preventive maintenance program can be a useful deterrant against system failures. Proper application of such a program is an aid to both serviceman and user, since detection and prevention of probable failures can reduce maintenance and downtime to a minimum.

Scheduling of computer usage should always include time set aside for maintenance purposes. Careful diagnostic testing can make evident problems which may only occur intermittently during on-line operation.

We suggest weekly program checks and thorough preventive maintenance on the following criteria:

1000-hours - electrical
500-hours - mechanical

or at least every three months

5.3.1 Weekly Checks

Time should be scheduled each week to operate the MAINDEC programs. Run each program listed in Table 5-2 for at least five minutes. Take any corrective action necessary at this time and record the results in the log book.

External cleanliness of the system should also be maintained on a weekly basis.

Many hours of computer downtime can be avoided by rigid adherence to a schedule based on the condition of the air filter. A dirty filter can cause machine failure through overheating which has a number of bad effects.

The frequency of this practice depends upon system environment and usage. The condition of the air filter should be checked every week. After several weeks the frequency of cleansing for the particular environment will be determined. The procedure for filter cleansing is described under Preventive Maintenance Tasks.

5.3.2 Preventive Maintenance Tasks

The following tasks should be performed on at least a three month's schedule.

a. Clean the exterior and interior of the equipment cabinet using a vacuum cleaner and/or cloths moistened in nonflammable solvent.

b. Clean the air filter by removing the retaining bar and machine screw. Use a vacuum cleaner to remove accumulated dirt and dust. Replace filter in the PDP-8/I.

c. Lubricate hinges, slide mechanisms, and casters, with a light machine oil. Wipe off excess oil.

d. Visually inspect equipment for general condition. Repaint any scratched areas with DEC black paint or Krylon glossy white No. 1501.

e. Inspect all wiring and cables for cuts, breaks, fraying, wear, deterioration, kinks, strains, and mechanical security. Tape, solder, or replace any defective wiring or cable covering.

f. Inspect the following for mechanical security: keys, switches, control knobs, lamps, connectors, transformers, fans, capacitors, etc. Tighten or replace as required.

g. Inspect all module mounting panels to ensure that each module is securely seated in its connector. Remove and clean any module which may have collected dirt or dust due to improper air filter servicing.

h. Inspect power supply components for leaky capacitors, overheated resistors, etc. Replace any defective components.

i. Check the output voltage and ripple content of the H704 power supply as specified in Table 5-3. Use a multimeter to make these measure-

ments without disconnecting the load. Use an oscilloscope to measure p-p ripple on all dc outputs of the supply. The outputs of the supply are not adjustable; therefore, if any output voltage or ripple content is not within specifications the supply is considered defective and corrective maintenance should be performed.

Table 5-3
Type H704 Power Supply Outputs

Nominal Output dc Voltage	Output Voltage Range	Maximum Output Current	Max P-P Output Ripple
+5V	± 5%	10 amp	50 mV
+15V	± 10%	5 amp	unfiltered
-15V	± 15%	5 amp	350 mV
-30V	± 10%	6 amp	700 mV

j. Run all MAINDEC programs to verify proper equipment operation. Each program should be allowed to run for at least 10 minutes.

k. Perform all preventive maintenance operations for each peripheral device included in the system.

l. Enter preventive maintenance results in the log book.

5.4 CORRECTIVE MAINTENANCE

The PDP-8/I is constructed of highly reliable TTL M-series modules. Use of these circuits with faithful performance of the preventive maintenance tasks ensures relatively little equipment downtime due to failure. Should a malfunction occur, maintenance personnel should analyze the condition and correct it as indicated in the following procedures. Neither special test equipment nor special tools are required for corrective maintenance other than a broad-bandwidth oscilloscope, a Tektronix Type P6019 current probe, and a multimeter. The best corrective maintenance tool is a thorough understanding of the physical and electrical characteristics of the equipment. Persons responsible for maintenance should become thoroughly familiar with the system concept, the logic drawings, the operation of specific module circuits, and the location of mechanical and electrical components.

It is virtually impossible to outline any specific procedures for locating faults within complex digital systems such as the PDP-8/I. However, diagnosis and remedial action for a fault condition can be undertaken logically, and systematically in the following phases:

a. Preliminary Investigation
b. System Troubleshooting
c. Logic Troubleshooting
d. Circuit Troubleshooting
e. Repairs and Replacement
f. Validation Tests
g. Recording

5.4.1 Preliminary Investigation

Before commencing troubleshooting procedures, explore every possible source of information. Think over the problem before attempting to troubleshoot the system. Gather all available information from those users who have encountered the problem and check the system log book for any previous references to the problem.

Do not attempt to troubleshoot by use of complex system programs alone. Run the MAINDEC programs and select the shortest, simplest program available which exhibits the error conditions. MAINDEC programs are carefully written to include program loops for assistance in system and logic troubleshooting.

5.4.2 System Troubleshooting

Once the problem is understood and the proper program is selected, the logical section of the

system at fault should be determined. Obviously, the program which has been selected gives a reasonable idea of what section of the system is failing. However, faults in equipment which transmit or receive information, or improper connection of the system, frequently give indications similar to those caused by computer malfunctions.

Disconnect any peripheral devices which are not necessary to operate the failing program.

At this time, reduce the program to its simplest scope loop and duplicate this loop in a dissimilar portion of memory to verify, for instance, that an operation failure is not dependent upon memory location. This process can aid in distinguishing memory failures from processor failures. Use of the techniques described above often pinpoints the problem to a few modules.

5.4.3 Logic Troubleshooting

Before attempting to troubleshoot the logic, make sure that proper and calibrated test equipment is available. Always calibrate the vertical preamp and probes of an oscilloscope before using. Make sure the oscilloscope has a good ac ground and keep the dc ground from the probe as short as possible.

Use the oscilloscope to trace signal flow through the suspected logic element. Oscilloscope sweep can be synchronized by control pulses or by level transitions which are available on individual module terminals at the wiring side of the logic. Care should be exercised when probing the logic, to prevent shorting between pins. Shorting of signal pins to power supply pins can result in damaged components.

Within modules, unused gate inputs are held at +3V. This voltage is introduced from pin U1 or V1 of modules M113, M117, or M617. The number in parenthesis beside each +3V input represents the wiring run number for that +3V line. Each line can handle a maximum of 15 loads. There are several terminals which must be tied to either ground or +3V when prewired options are not installed. Refer to engineering specifications (Drawing A-SP-8I-0-23) for the locations of these temporary jumpers.

5.4.4 Circuit Troubleshooting

Engineering schematic diagrams of each module are supplied with each PDP-8/I system and should be referred to for detailed circuit information. Copies of engineering schematic diagrams are contained in Volume II.

Visually inspect the module on both the component side and the printed wiring side to check for overheated or broken components or etch. If this inspection fails to reveal the cause of trouble or to confirm a fault condition observed, use the multimeter to measure resistances.

CAUTION

Do not use the lowest or highest resistance ranges of the multimeter when checking semiconductor devices. The X10 range is suggested. Failure to heed this warning may result in damage to components.

Measure the forward and reverse resistances of diodes. Diodes should measure approximately 20 Ω forward and more than 1000 Ω reverse. If readings in each direction are the same and no parallel paths exist, replace the diode.

Measure the emitter-collector, collector-base, and emitter-base resistances of transistors in both directions. Short circuits between collector and emitter or an open circuit in the base-emitter path cause most failures. A good transistor indicates an open circuit in both directions between collector and emitter. Normally 50 to 100 Ω exist between the emitter and the base, or between the collector and the base in the forward direction, and an open circuit condition exists in the reverse direction. To determine forward

and reverse directions, consider a transistor as two diodes connected back to back. In this analogy, PNP transistors would have both cathodes connected together to form the base, and both the emitter and collector would assume the function of an anode. In NPN transistors the base would be a common-anode connection; and both the emitter and collector, the cathode.

Multimeter polarity must be checked before measuring resistance, since many meters apply a positive voltage to the common lead when in the resistance mode.

Since IC's contain complex integrated circuits with only the input, output, and power terminals available, static multimeter testing is limited to continuity checks for shorts between terminals. IC checking is best done under dynamic conditions using a module extender to make terminals readily accessible. Using PDP-8/I logic diagrams and M-series module schematics, you may locate an IC on a circuit board as follows.

a. Hold the module with the handle in your left hand; component side facing you.

b. IC's are numbered starting at the contact side of the board; upper right hand corner.

c. The numbers increase toward the handle.

d. When a row is complete, the next IC is located in the next row at the contact end of the board. (See Figure 5-1)

e. The pins on each IC are located as Figure 5-2 illustrates.

5.4.5 Repairs and Replacement

When soldering semiconductor devices (transistor, diodes, rectifiers or integrated circuits) which may be damaged by heat, physical shock, or excessive electrical current, take the following special precautions.

a. Use a heat sink, such as a pair of pliers, to grip the lead between the joint and device being soldered.

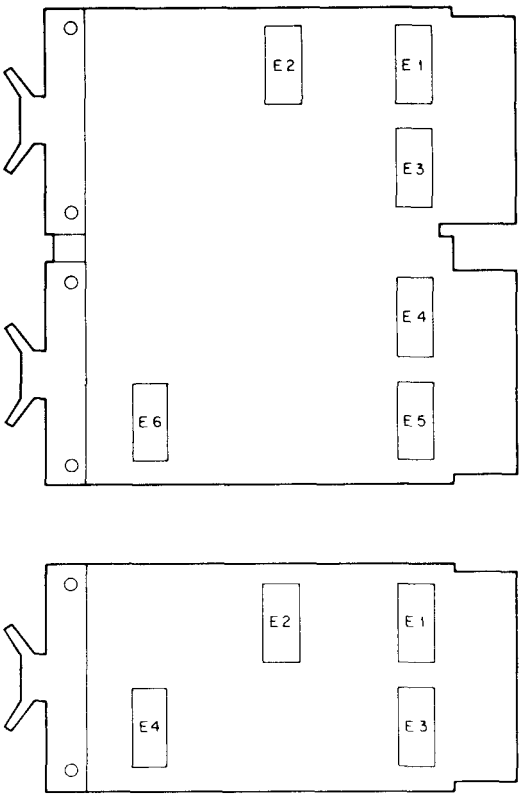


Figure 5-1 IC Location

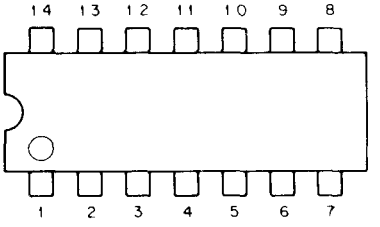


Figure 5-2 IC Pin Location

b. Use a 6V iron with an isolation transformer. Use the smallest iron adequate for the work. Use of an iron without an isolation transformer may result in excessive voltages presented at the iron tip.

c. Perform the soldering operation in the shortest possible time to prevent damage to the component and delamination of the module etched wiring.

d. IC's may be easily removed by using a solder puller to remove all excessive solder from contacts and then by straightening the leads, lift the IC from its terminal points. If it is not desirable to save the defective IC for test purposes, then the terminals may be cut at the IC body and each terminal removed from the board individually.

CAUTION

Never attempt to remove solder from terminal points by heating and rapping module against another surface. This practice can result in module or component damage. Always remove solder by the use of a solder-sucking tool.

When removing any part of the equipment for repair and replacement, make sure that all leads or wires which are unsoldered, or otherwise disconnected, are legibly tagged or marked for identification with their respective terminals. Replace defective component only with parts of equal or better quality and equal tolerance.

In all soldering and unsoldering operations in the repair and replacement of parts, avoid placing excessive solder or flux on adjacent parts or service lines. When repair has been completed, remove all excess flux by washing junctions with a solvent such as trichlorethylene. Be very careful not to expose painted or plastic surfaces to this solvent.

5.4.6 Validation Tests

Always return repaired modules to the location from which they were taken. If a defective module is replaced by a new one while repairs are being made; tag the defective module noting the location from which it was taken, and the nature of the failure. When repairs are completed, return the repaired module to its original location and ascertain that the repairs have resolved the problem.

To confirm that repairs have been completed, run all tests which originally exhibited the problem. If modules have been moved during the troubleshooting period, return all modules to their original positions before running the validation tests.

5.4.7 Recording

A log book is supplied with each PDP-8/I system. Corrective maintenance is not complete until all activities are recorded in the log book. Record all data indicating the symptoms given by the fault, the method of fault detection, the component at fault, and any comments which would be helpful in maintaining the equipment in the future.

The log should be maintained on a daily basis, recording all operator usage and preventive maintenance results.

NOTE

The measurements and adjustments made in the following paragraphs are analog in nature, and are not of the ON-OFF, HI-LO nature. These represent tuned sections of the machine that must be properly aligned by qualified personnel.

5.5 ADJUSTMENTS

Adjustments of the PDP-8/I should never be undertaken until it has been confirmed that a failure is due to circuit aging or misalignment rather than a component failure. Replacement of certain components or removal of an excessive environment may eliminate the need for adjustment.

5.5.1 Power-Up Threshold Adjustment

This adjustment is preset at the factory and should only be attempted in the field by personnel trained in this skill.

The G826 Negative Regulator Control module contains a difference amplifier that compares the +5V supply voltage with an adjustable reference threshold voltage. The threshold reference is properly set when the voltage at test point A2U2 changes from a low voltage (-6V) to a higher voltage (-2V) as the +5V supply voltage passes through 4.75V after turn-on. The threshold voltage is adjusted by the G826 variable resistor R2, 1000 Ω and performs several functions.

When the +5V supply voltage is less than this threshold, the regulated memory supply voltage is held off. This threshold also controls various logic signals to the central processor by initiating POWER OK and a power failure level SHUT DOWN; a STOP OK level from the processor allows a shut down of the memory supply voltage when the +5V supply voltage fails. As the threshold is reached during turn-on, POWER CLEAR produces a pulse to ground and then returns to a high level. Also, with these conditions, POWER OK is a low logic level, and SHUT DOWN is a high logic level. The inverse of these levels should exist before the threshold is reached. The STOP OK level originates from the central processor and its undriven input should remain high, having no effect during alignment.

5.5.2 Memory Alignment Procedure

To adjust or check the memory currents, the PDP-8/I should be allowed to warm up for approximately one hour before measurements are made. In addition, the measurements should be performed at an ambient temperature of 25°C.

The G826 negative regulator control module at location AB2 and the G805 negative regulator module at location AP1 control the memory voltage regulation, and therefore control the memory currents.

The voltage difference between MEMORY SUPPLY + and MEMORY SUPPLY - provides the read/write, and inhibit current through the memory stack.

The negative regulator control (G826) serves to vary the G805 regulator for temperature variations by a thermistor and difference amplifier tracking circuit which compares the regulated memory voltage to an adjustable reference. A trimpot (R28, 2000 Ω) located on the G826 module varies this reference voltage, and therefore, varies the memory currents.

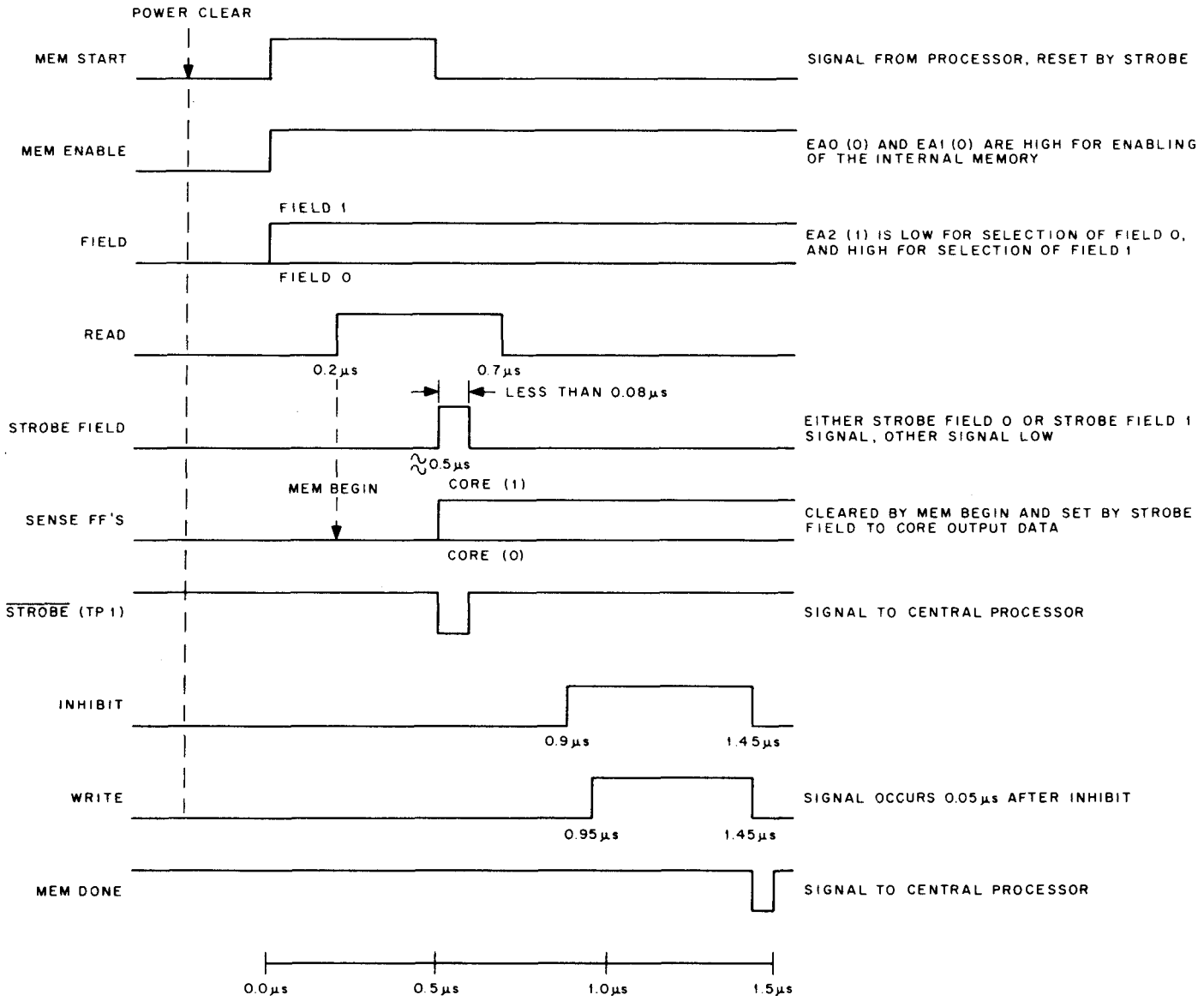
Adjustment of the trimpot varies the MEMORY SUPPLY + voltage between approximately -1V and -12V; the subsequent variation of the regulated memory supply voltage should be between -18V and -29V. Normally, the regulated memory voltage should be set to approximately -22.5V, measured with a multimeter across MEMORY SUPPLY + and MEMORY SUPPLY-.

If the voltage can be adjusted but not to the above value, the thermistor across the regulator control (pins B2R2 and B2S2) should be investigated. The thermistor is located within the memory stack and outputs on the inhibit connector card (pins B35S2 and B35T2); its resistance should be 330 Ω ± 10% at 25°C. Shunting resistance in the regulator control would reduce this value if a measurement is made with the modules connected.

If no voltage adjustment is possible, either the Negative Regulator (G805) or the Regulator Control (G826) can be at fault.

Removal of the Negative Regulator allows a separate test of the Regulator Control and, therefore, a determination of which module has failed. With the Negative Regulator (G805 at 81) removed and a stack with its associated thermistor present, the regulator control output (pin B2R2) can be varied between ground and -13V by adjusting R28. The absence of such a variation indicates a failing regulator control and the need for replacement. Variation suggests a failed Negative Regulator (G805).

In alignment of the memory, the actual outputs of the various memory control flip-flops should be checked against those of Figure 5-3. An approximate initial STROBE adjustment can be such that its leading edge occurs 500 ns after the leading edge of MEM START. The width of the strobe signal should be less than or equal to 80 ns



NOTE. TRANSITION TIMES ARE MEASURED FROM THE POSITIVE TRANSITION OF MEM START AND ARE APPROXIMATE.

Figure 5-3 Memory Control Waveforms

and have an approximate adjustment range for its leading edge from MEM START from 350 ns to 650 ns. A clockwise rotation of the adjustment on the variable delay line (M360 at B23) increases this delay. The final adjustment of strobe must be made in relation to the analog data signal from core memory.

The R/W selector switches are examined by inspecting the current waveforms on the current loops at the SOURCE and RETURN signals. The current waveforms are similar to those of Figure 5-4 which represents the equal amplitude read/write currents measured on the memory stack input. When running a multiple-selection program, such as MEMORY CHECKERBOARD, the waveforms differ in the amplitudes of the read/write currents due to the contribution of the base currents to the emitter currents of the address decoding and selection switches.

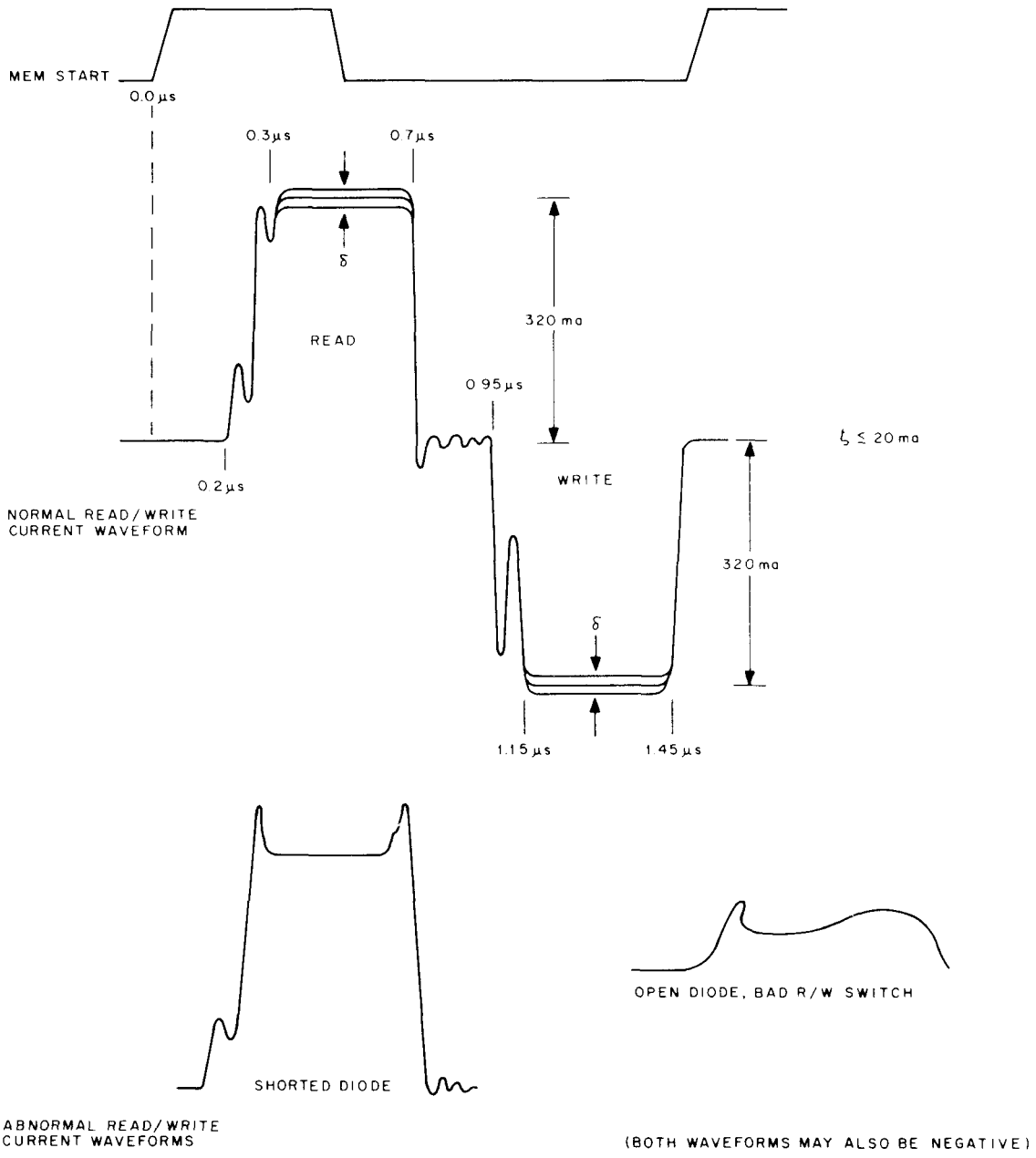


Figure 5-4 Representative Read/Write Current Waveforms

The additional current (approximately 30 mA) appears on the return current through the R/W selection switches to MEMORY SUPPLY-. Depending upon the loop examined, it is sometimes additional read current and sometimes additional write current. This difference in current should not be confused with amplitude variations within the read/write current due to bad address decoding selection switches; these differences are also apparent at the memory stack inputs and vary with different addresses. The current loops, provided for use with a Tektronix type P6019 current probe, are as follows.

X Source	C38T2 to C39K1
Y Source	C33T2 to C39S1
X Return	D33T2 to D39K1
Y Return	D38T2 to D39S1

Examination of the current at these loops has the advantage that the currents for all memory addresses pass through these common points. The waveshape should be inspected on each of the four loops to check that no leakage to ground exists between the voltages, MEMORY SUPPLY + and MEMORY SUPPLY -

The regulated memory voltage was previously adjusted in a static condition for -22.5V; there should be no change although the memory is now cycling. The resultant stack current should be approximately 320 mA (this value varies with stack vendors, and is best determined by adequate core output with minimum noise). The lower read/write current amplitudes on the current amplitudes on the current loops equal the current amplitudes in the stack windings; the current measurements can conveniently be made here. Correlation between the voltage and current should exist. Variations in current amplitude at successive addresses should be less than 20 mA. Failure of the R/W selection switches is indicated by the lack of proper read/write currents of any address. If all of the R/W selection switches appear to have failed, inspection should be made at the input logic signals, READ (1) and WRITE (1), and the input supply voltages. To analyze the read/write currents for individual addresses, the following program may be used:

```
0000    Beg,  LAS          7604
0001          DCA Temp     3004
0002          TAD I Temp   1404
0003          JMP Beg      5000
0004    Temp, 0            0000
```

The read/write currents for individual addresses may be examined by setting the desired address into the switch register. Of course, since the currents for all memory addresses pass through the common test point, you will also be examining the currents for those locations which the test program occupies. You must first ascertain that the currents are proper for those addresses within the program; if not, relocate the program to some other area of memory. By selecting individual addresses via the switch register, waveform deformities may be traced to defective associated R/W switches.

If the improper waveform remains fixed to a specific address and replacement of the associated R/W switch does not correct the problem; then, the logic signal inputs from the memory address (MA) register should be inspected. If those signals are correct and vary according to the selected address, attention should be turned to the memory stack with its attendant Diode Selection Matrices.

The Diode Selection Matrices (G611 and G610) sandwich the memory planes of ferrite cores between them. Also connected to this unit are two W025 cable connectors, one for the inhibit inputs, the other for the SENSE outputs.

The X-axis Diode Selection Matrix (D-BS-8I-0-15) has half its diodes on the G610 board and half on the G611 board; the same is true of the Y-axis Diode Selection Matrix (D-BS-8I-0-16). The inductor symbol connecting the centers of the two sets of diodes represents the stack winding traversing the 12 core planes. The windings are identified on the Diode Matrix Selection boards as X, 0-64 and $\underline{X}$. 0-64 for the X-axis windings and Y,

0-64 and $\overline{Y}$. 0-64 for the Y-axis windings. Suspected opens and shorts in the windings, detected during dynamic tests, should be verified by measurements across these points with an ohmmeter. The resistance of the read/write winding is 3.5 Ω ±10%. The forward and reverse resistance of the diodes in the matrix should also be checked when address selection failures are attributed to stack failures. Drawing D-CS-3005256-D-3 correlates the memory diode location and function.

CAUTION

The memory stack is expensive and fragile; it is easily damaged and must be handled with care.

Twelve inhibit drivers are associated with each memory stack. To inspect the inhibit currents, you will need two module extender boards (W998) at AB35. These extender boards provide current loops for each of the 12 inhibit lines. Inhibit current should be inspected and compared with the waveform in Figure 5-5. Inhibit current amplitude is approximately 290 mA as noted; more important, however, is the ratio of inhibit to read/write currents. This ratio is 0.85 of the read/write current and should exist regardless of the read/write current amplitude. Failure of a specific inhibit driver can be determined by movement of the suspected driver to another location. Movement of the failure indicated that the module should be repaired or replaced. No movement of the failure indicates that either the input signals or output load is causing the failure.

The logic inputs, B INHIBIT and B FIELD (0), from the memory control; the connection through the current limiting resistor; and the memory buffer signals should be checked. If the stack is suspected, the specific winding should be measured for a resistance of 14 Ω ±10%.

Twelve sense amplifiers (G020 or G021) are associated with each memory stack; each amplifier transforms the analog pulse output of ferrite core to a usable logic level. The G020 module is used in machines which have a basic 4K memory; when an additional 4K memory is added, this module is replaced with a G021 which provides an additional amplifier input (Drawing BS-8I-0-14). The sense windings for each bit enter a differential amplifier with a threshold voltage established as a function of the fixed SLICE voltage. The test points (pins E1 and K2) after the amplifiers allow observation of the waveforms for comparison with those of Figure 5-6. The preliminary setting of STROBE should now be modified as a function of the data output from core memory.

The strobe leading edge is set approximately at the center or just past the midpoint of the amplifier "one" output. This adjustment should be late enough to sense all "one" data with normal variations in delay, and yet centered in the "zero" data output. Each sense amplifier test point should be examined and the final adjustment of the strobe signal should be made using the most sensitive sense amplifier as a criteria.

The lack of proper waveforms at all addresses indicates that the sense amplifier or the core winding is in error. The sense amplifier may be checked by exchanging a known good amplifier with the suspected one. The absence of an input signal indicates the stack sense winding should be checked or the sense connector (W025 at 34AB) for a resistance of 21 Ω ±10%.

Testing and adjustment of the memory section of the PDP-8/I is now completed by running the memory address test and memory checkerboard test (worst pattern). Final adjustment of the read/write current may be necessary to increase or decrease the amplitude of the core input and corresponding noise.

5.6 ASR33 TELEPRINTER AND CONTROL MAINTENANCE

This section contains information pertinent to the maintenance of the ASR33 and its associated control logic.

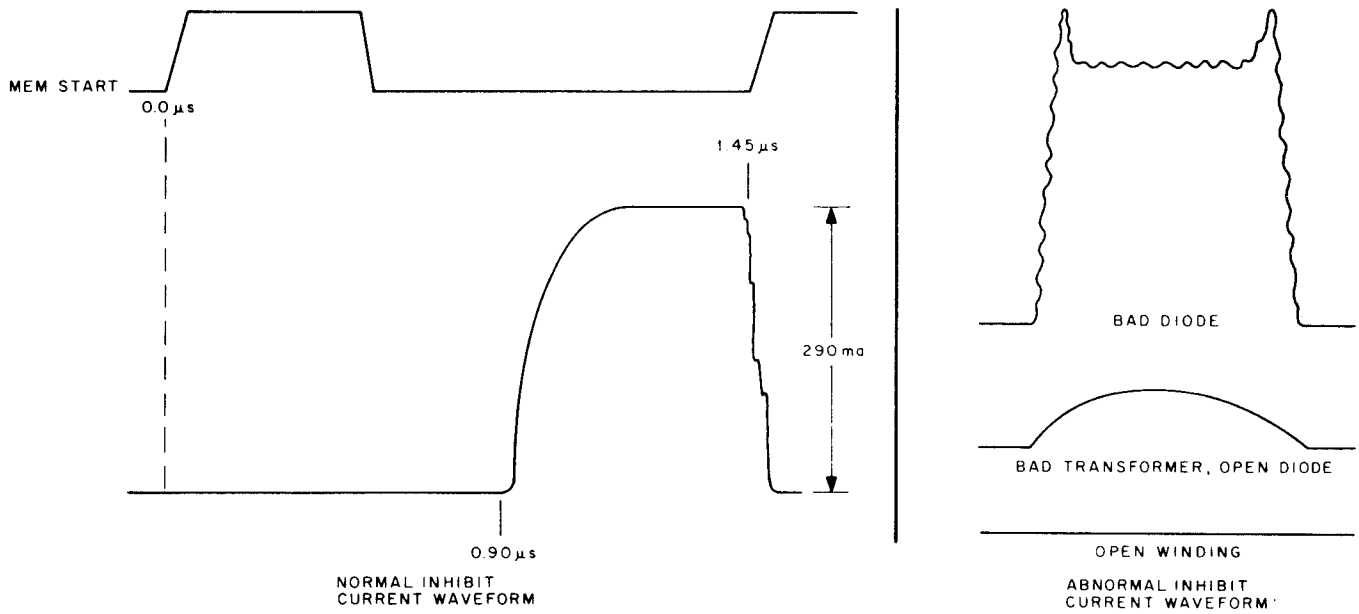
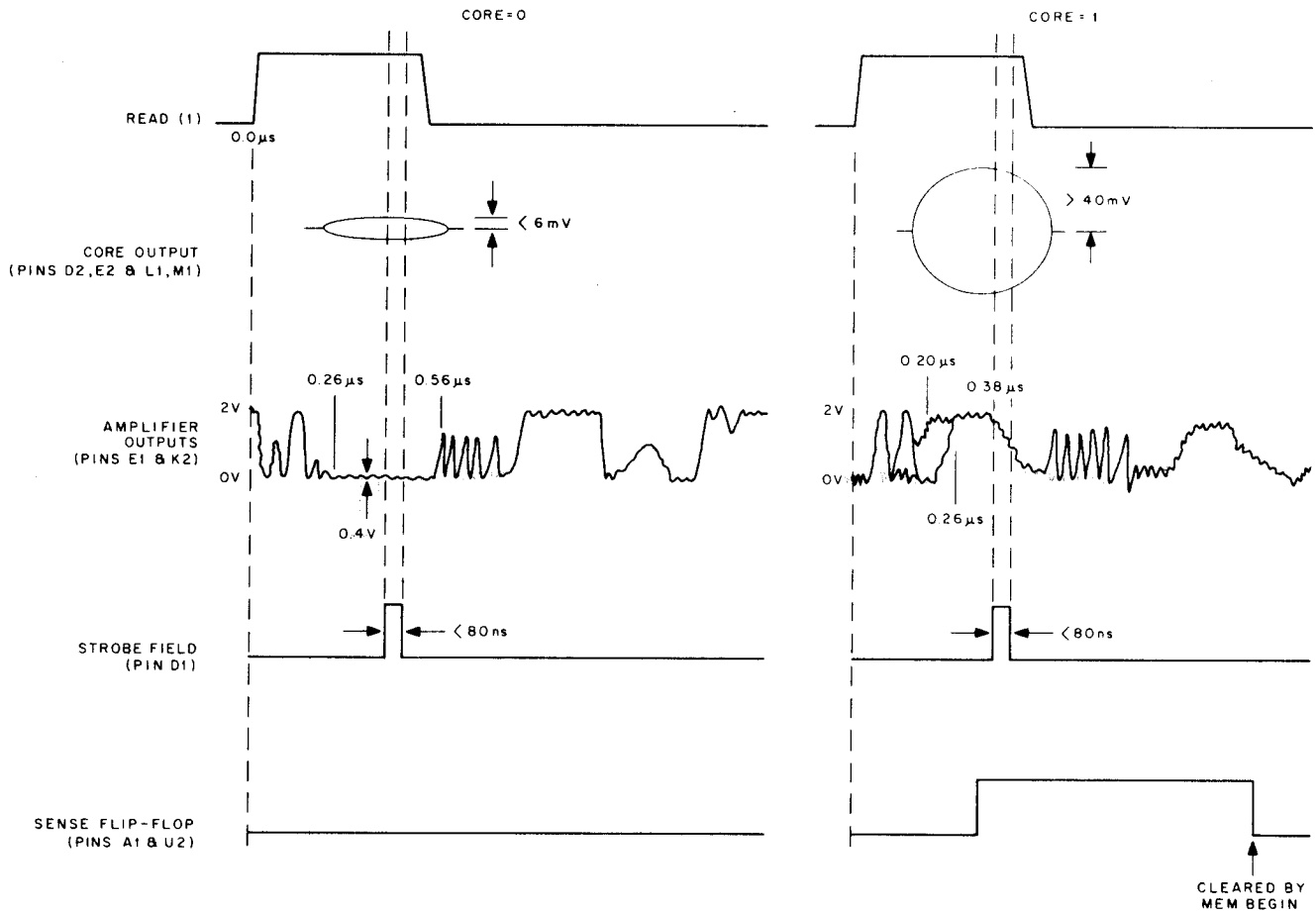


Figure 5-5 Representative Inhibit Current Waveforms



NOTE: CORE OUTPUTS ARE OBSERVED DIFFERENTIALLY AT THE PINS NOTED.

Figure 5-6 Representative Sense Amplifier Waveforms

5.6.1 Equipment

Table 5-4 lists the special tools needed for maintenance of the ASR33 teleprinter. All of these items can be obtained from Digital Equipment Corporation or from the Teletype Corporation.

Table 5-4
Teletype Maintenance Tools

Item	Part No.
8 oz. scale	110443
32 oz. scale	110444
64 oz. scale	82711
Set of gauges	117781
Offset screwdriver	94644
Offset screwdriver	94645
Handwheel	161430
Handwheel adaptor	181465
Contact adjustment tool	172060
Gauge	180587
Gauge	180588
Gauge	183103
Bending Tool	180993
Extractor	182697
Tweezer	151392
Spring hook (push)	142555
Spring hook (pull)	142554
Screw holder	151384

5.6.2 Programs

Table 5-5 lists the maintenance programs supplied by DEC for aid in maintaining the ASR33 and associated control logic.

5.6.3 Preventive Maintenance

Teletype preventive maintenance is scheduled on the same frequency as discussed in Section 5.4.

CAUTION

Do not use alcohol, mineral spirits, or other solvents to clean plastic parts with protective decorative finishes. Normally, a soft, dry cloth should be used to remove dust, oil, grease, or otherwise clean parts or subassemblies.

To clean plastic surfaces, we recommend using any of several household cleaner-waxer liquids such as "Jubilee" or "Jato". To clean the printer platen, we recommend using a lacquer thinner.

During a overhaul, subassemblies and metal parts can be cleaned in a bath of trichlorethylene. Proper lubrication should be performed often.

Table 5-5
Teleprinter Maintenance Programs

Program Name	DEC No.	Use
Reader Test	Maindec-8I-D2PB	Function test and exerciser for ASR33/35 Teletype paper tape reader
Punch Test	Maindec-8I-D2QB	Function test and exerciser for ASR33/35 Teletype paper tape punch
Keyboard Test	Maindec-8I-D2RB	Function test and exerciser for ASR33/35 Teletype keyboard
Combination Test	Maindec-8I-D2TB	Exerciser program used to test ASR33/35 printer and punch simultaneously

Weekly Tasks

a. Inspect platen and paper guides. Wipe clean, using a soft, dry cloth.

b. Clean external areas of paper tape punch and reader, using a soft brush or cloth.

c. Remove and empty paper tape punch chad box.

d. Run teleprinter combination test (Maindec-8I-D2TB) for approximately 15 min.

Preventive Maintenance Tasks

a. Inspect platen and paper guides. Clean platen, using a lacquer thinner to remove shiny surface.

b. Clean ribbon guides and replace ribbon, if necessary.

c. Remove cover and check for vibration effects; loose nuts, screws, retaining clips, etc.

d. Remove distributor rotor and clean disk surface, using cotton swab moistened in "freon" or "trichlorethylene."

e. Check selector magnet coil for signs of overheating.

f. Clean between selector magnet pole piece and armature with bond paper to remove any lubricant or dirt.

g. Clean and lubricate teletype as per Teletype Bulletin 273B. Follow instructions literally so as not to over lubricate.

h. The following adjustments should be checked Pages indicated are in Bulletin 273B, Volume 2.

Trip Shaft	574-122-700 Page 13
Trip Lever	574-122-700 Page 14
Brush Holder (Distributor)	574-122-700 Page 15
Clutches	574-122-700 Pages 16-24
Code Bar Reset	574-122-700 Pages 30-34
Print Suppression	574-122-700 Page 35
Blocking Levers	574-122-700 Page 37
Print Suppression	574-122-700 Page 43
Carriage Drive Bail	574-122-700 Page 44
Print Trip Lever	574-122-700 Pages 61-62
Dashpot	574-122-700 Page 78
Final Printing Alignment	574-122-700 Page 85
Line Feed	574-122-700 Page 89-95
Keyboard Trip Lever	574-122-700 Page 141
Reader Trip Lever	574-124-700 Pages 6-9
Detent Lever	574-124-700 Page 10
Sensing Pin	574-124-700 Page 15
Tape Lid Latch Handle	574-124-700 Page 18
Feed Pawl	574-125-700 Page 11
Registration	574-125-700 Page 12

i. Run each of the teletype Maindec Programs for at least two passes each.

j. Check that tape holes are being punched cleanly.

5.6.4 Corrective Maintenance

Figure 5-7 is a simplified drawing of the control circuits for the ASR33 Teleprinter. Details of the cable connector are included to show how a teleprinter is modified to operate with the PDP-8/I. During off-line operation, the keyboard distributor effectively drives the printer selector magnet. This means that any character received from the keyboard or paper tape reader is automatically reproduced on the printer and paper tape punch. During on-line operation, this continuity is broken and a teletype receiver (M706) is used to accept the input from the reader or keyboard while a teletype transmitter (M707) is used to drive the printer and paper tape punch.

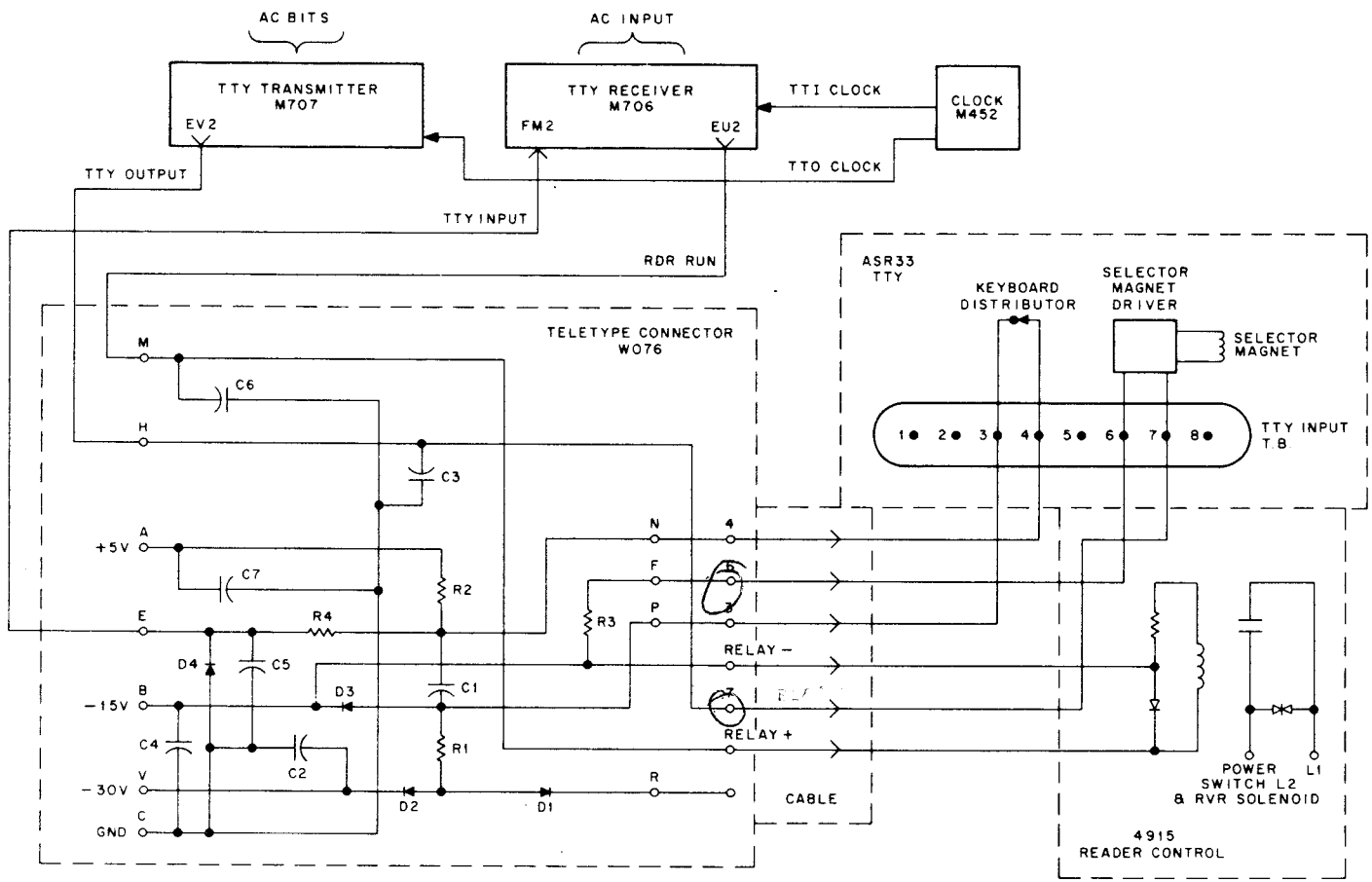


Figure 5-7 Teletype Connections

The clock (M452) develops a TTI clock (880 HZ) and a TTO clock (220 HZ).

These clocks are used to shift the bits through the transmitter and receiver buffers. Adjustment is made by viewing the TTO clock output with the oscilloscope probe on F3K2 and adjusting the trimpot for a 4.5 ms to 4.6 ms repetition rate. Most teletype problems can be traced to one of four areas:

a. ASR33 keyboard or reader
b. ASR33 printer or punch
c. M706 receiver
d. M707 transmitter

Printer/punch problems can sometimes be isolated by comparing the printed character with the output of the paper tape punch. If the printed character agrees with the punch output, and both are incorrect then, the problem lies in the selector mechanism or in the TTY transmitter module (M707). If the printed character and the paper tape punch output disagree, and the paper tape punch output is correct then, the problem lies within the printer assembly.

Figure 5-8 shows the teletype signal produced at pin EV2 while transmitting from the computer and also at FM2 when receiving information from the teletype.

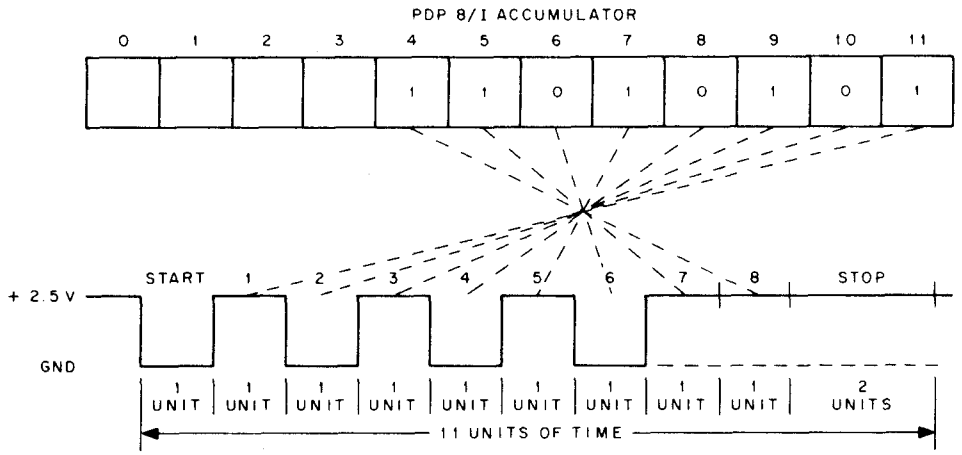


Figure 5-8 Teletype Signal Waveform and Bit Relationship for the Character "U"

Isolation of bit related problems is fairly simple. Off-line duplication can usually determine whether the problem is in the teleprinter or the control logic. Steps may also be taken to isolate the problem to subassemblies within the teleprinter. Picking up bits during a read operation can be caused by a defect in any of three sets of contacts which are tied in parallel. Reader, keyboard, and answer-back contacts provide parallel inputs to the distributor contact disk. Bit pick-up problems can be isolated to one of these three areas by disengaging the related contact from the suspected contact set.

5.7 SPARE PARTS

Each user of the PDP-8/I system should establish a spare parts stock in accordance with the extent of the available repair facilities. The following considerations are helpful in determining what spare parts should be stocked.

Teletype - Users who do not have maintenance personnel trained in the maintenance and repair of Teletype units should keep a complete Model 33 Automatic Send Receive Teletype near the computer. If the on-line unit becomes defective,

substitute the spare to avoid computer down-time. However, many users have facilities for the maintenance of Teletype units, in which case it is suggested that spare parts be stocked as listed in Table 5-6, and that one of each Teletype maintenance tool listed in Table 5-4 be stocked. All of these items can be obtained from the Digital Equipment Corporation, or from the Teletype Corporation.

Table 5-6
Spare Parts for Keyboard-Model ASR33 Teletype

Quantity	Item	Part No.
1	Circuit Board	181821
2	Tape Feed Sprocket	183071
2	Lever, universal	182240
2	Fuse (3.2 amp)	120167
2	Distributor Brush	180979
1	Belt Driven Gear	181420
1	Drive Gear	181411
2	Belt	181409
1	Shaft	181007
2	Bearing	181002

Modules And Components - All of the module types in the basic PDP-8/I are listed in Table 5-7. It is suggested that one module of each type be stocked as a spare part, except for the Type M113, and Type M310 modules for which the suggested quantity is two each. If modules are to be repaired at the installation site, reduce this list of spare modules, and stock the components listed in Tables 5-8 through 5-11.

Table 5-7
Spare Module List

Name	Type
Sense Amplifier	G020
Memory Selector	G221
Inhibit Driver	G228

Table 5-7 (Cont)
Spare Module List

Name	Type
Resistor Board	G624
Negative Regulator	G805
Regulator Control	G826
Ten, 2-Input Nand	M113
Eight, 3-Input Nand	M115
Six, 4-Input Nand	M117
Three, And/Nor Expanders	M160
Six Flip-Flops	M216
Major Registers	M220
Delay Line	M310
Variable Delay	M360
Variable Clock	M452
Negative Input Converter	M506
Six, 4-Input Power Nand	M617
Negative Output Converter	M650
Manual Timing Generator	M700
Teletype Receiver	M706
Teletype Transmitter	M707

Table 5-8
Recommended Spare Diodes

Code Number	Description	Quantity
11-00113	D662	10
11-00114	D664	10
11-02451	1N753	2
11-03183	MR2064	2
11-03309	D671	2
11-05275	D672	10

Table 5-9
Recommended Spare Transistors

Code Number	Description	Quantity
15-02155	DEC 1008-S	5
15-02937	2N3568	2
15-03100	DEC 3009B-S	10
15-03399	DEC 3790-S	2
15-03409	6534 D	10

Table 5-10
Recommended Spare Integrated Circuits

Code Number	Description	Quantity
19-05521	MC 1540G	2
19-05547	SN7474	5
19-05575	SN7400N	5
19-05576	SN7410N	5
19-05577	SN7420N	5
19-05578	SN7430N	2
19-05579	SN7440N	5
19-05580	SN7450N	2
19-05581	SN7460N	5
19-05582	SN7453N	5
19-05584	SN7482N	2

5.8 MECHANIZATION CHARTS

This paragraph contains function mechanization charts intended to serve both as a quick reference table for the experienced technician and as a troubleshooting analysis guide for less experienced maintenance personnel.

The charts are designed to serve as a bridge between the functions specified on the flow diagrams and those logic elements in the block schematics by which the functions are implemented.

The charts contain, reading left to right, the desired function, the time period (per flow-diagram notation) during which it occurs, the time state (or pulse) during which it is implemented, the primary signal(s) which perform the function, the signals which generate the primary signals, and the engineering drawings upon which all signals previously described are generated.

The portion of the chart shown Table 5-12 represents the T1 portion of the FETCH cycle of a typical memory reference instruction.

In the Function column, the notation MA+1 → PC indicates that during T1 the content of MA is to be incremented and transferred to the PC. This is to be initiated during TS1 and completed at TP1. The signals required during TS1 are MA ENABLE and $\overline{\text{CARRY INSERT}}$. MA ENABLE is generated on Drawing BS-8I-0-4, and $\overline{\text{CARRY INSERT}}$ on Drawing BS-8I-0-5.

MA ENABLE is generated by the output of the logic 1 side of the TS1 flip-flop [TS1(1)] combined with the PC INCREMENT signal. These two signals are generated on Drawings BS-8I-0-2 and -3 respectively.

$\overline{\text{CARRY INSERT}}$ is generated by the combination of TS1(1) and FETCH=0. These two signals are

Table 5-11
Spare Miscellaneous Components and Parts

Component	Code Number	Description	Quantity
Delay Line	16-05530	L501 D501A	1
Pulse Transformer	52-00672	2037	5
Pulse Transformer	52-02123	2052	5
Rocker Switch	12-5375 (RS-9-3-FB)		4
Rocker Switch	12-5941 (RS-50-FB-PC)		4
Rocker Handles	12-5317*	4	
Indicator Bulbs	12-5550 (2313)	6	
Power Lock Switch	34-4235	1	

*Must specify color ... gray or orange.

generated on Drawings BS-8I-0-2 and -3, respectively.

The implementation of this function is completed at TP1 by the signal PC LOAD, generated on Drawing BS-8I-0-6. PC LOAD is, in turn, generated by the combination of TP1 and PC INCREMENT. These two signals are generated on Drawings BS-8I-0-2 and -5, respectively.

The charts are so structured that the events of a particular time state are fully explained before the description of the events governed by the associated time pulse is begun. The user, therefore, after examining all the time state events of a particular function, should examine the time pulse activities of that same function.

Table 5-12
Sample Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MA+1→PC	T1	TS1	MA ENABLE		BS-8I-0-4
				TS1(1)	BS-8I-0-2
				PC INCREMENT	BS-8I-0-5
			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				TS1(1)	BS-8I-0-2
				FETCH=0	BS-8I-0-3
MA+1→PC		TP1	PC LOAD		BS-8I-0-6
				TP1	BS-8I-0-2
				PC INCREMENT	BS-8I-0-5

Table 5-13
Fetch Cycle, AND, TAD, ISZ, DCA, and JMS Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MA+1 →PC	T1	TS1	MA ENABLE		BS-8I-0-4
				TS1(1)	BS-8I-0-2
				PC INCREMENT	BS-8I-0-5
			CARRY INSERT		BS-8I-0-5
				TS1(1)	BS-8I-0-2
				FETCH=0	BS-8I-0-3
MA+1 →PC		TP1			BS-8I-0-6
				TP1	BS-8I-0-2
				PC INCREMENT	BS-8I-0-5

Table 5-13 (Cont)
Fetch Cycle, AND, TAD, ISZ, DCA, and JMS Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM → MB	T2	TS2	MEM ENABLE 0-4/5-11		BS-8I-0-4
				TS2(1)	BS-8I-0-2
				All of the following NOT true: KEY DP, DCA, JMS, BREAK, EXECUTE	BS-8I-0-2,3
MEM → MB					
MEM → IR		TP2	MB LOAD		BS-8I-0-6
				TP2	BS-8I-0-2
			Load Instruction Register		BS-8I-0-3
				TP2	BS-8I-0-2
				B FETCH=1	BS-8I-0-3
	T3	TS3/ TP3	No Operations		
MEM5-11 → MA5-11	T4	TS4	MEM ENABLE 5-11		BS-8I-0-4
				TS4(1)	BS-8I-0-2
				$\overline{\text{E SET}}$	BS-8I-0-3
				$\overline{\text{PC ENABLE}}$	BS-8I-0-4
				$\overline{\text{INT OK}}$	BS-8I-0-7
Address Current Page MA0-4 → MA0-4			If MB04=1 then: MA ENABLE 0-4		BS-8I-0-4
				MEM ENABLE 0-4	BS-8I-0-4
				MEM ENABLE 5-11	BS-8I-0-4
Address Page Zero 0 → MA0-4			If MB04=0 then: MA ENABLE 0-4 not generated		
Defer State entry	T4 (cont)	TS4 (cont)	If MB03=1 then: 1 → DEFER (enable)		BS-8I-0-4
				B FETCH (1)	BS-8I-0-3
				IR0=0 or IR1=0	BS-8I-0-3
				MB03=1	BS-8I-0-8
Execute State entry			If MB03=0 then: 1 → EXECUTE (enable)		BS-8I-0-3
				JMP	BS-8I-0-3
				B FETCH=1	BS-8I-0-3
				IR0=0 or IR1=1	BS-8I-0-3
				MB0=0	BS-8I-0-8

Table 5-13 (Cont)
Fetch Cycle, AND, TAD, ISZ, DCA, and JMS Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM5-11 → MA5-11		TP4	MA LOAD	TP4	BS-8I-0-6 BS-8I-0-2
			1 → DEFER (Set) or		BS-8I-0-2
			1 → EXECUTE (Set)	TP4	BS-8I-0-2 BS-8I-0-2

Table 5-14
Fetch Cycle, OPR Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MA+1 → PC	T1	TS1/ TP1	Same as AND instruction		
MEM → MB MEM → IR	T2	TS2/ TP2	Same as AND instruction		
No operation AC → AC	T3	TS3	$\overline{MB03-0\ (OP1)}$ NOP (MB04 0·MB06=0) AC ENABLE	 OP1 MB04=MB06=0	 BS-8I-0-4 BS-8I-0-5 BS-8I-0-8
Complement AC $\overline{AC}$ → AC			CMA (MB04=0·MB06=1) $\overline{AC}$ ENABLE	 OP1 MB06=1/MB04=0	 BS-8I-0-4 BS-8I-0-5 BS-8I-0-8
0 → AC			CLA (MB04=1·MB06=0) NO OPERATION		

Table 5-14 (Cont)
Fetch Cycle, OPR Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
Set AC=-1 AC+$\overline{AC}$ → AC			CLA/CMA (MB04=1·MB06=1) AC ENABLE		
					BS-8I-0-4
				OP1	BS-8I-0-5
				MB04·MB06=1	BS-8I-0-3
AC+$\overline{AC}$ → AC			$\overline{AC}$ ENABLE		BS-8I-0-4
				OP1	BS-8I-0-4
				MB06=1	BS-8I-0-8
No operation L → L			NOP (MB05=0·MB07=0) L ENABLE		
					BS-8I-0-4
				MB05·MB07	BS-8I-0-8
Complement the Link $\overline{L}$ → L			CML (MB05=0·MB07=1) $\overline{L}$ ENABLE		
					BS-8I-0-4
				OP1	BS-8I-0-5
				MB07=1	BS-8I-0-8
Clear the Link 0 → L	T3 (cont)	TS3 (cont)	CLL (MB05=1·MB07=0) No operation		
Set Link to 1 +1 → L			CLL/CML (MB05=1·MB07=1) L ENABLE		
					BS-8I-0-4
				OP1	BS-8I-0-5
				MB05·MB07=1	BS-8I-0-8
CLL and STL			$\overline{L}$ ENABLE		BS-8I-0-4
				OP1	BS-8I-0-5
				MB07=1	BS-8I-0-8
No Operation AC → AC			(MB08=0·MB09=0) NO SHIFT		
					BS-8I-0-5
				MB08=0	BS-8I-0-8
				MB09=0	BS-8I-0-8
				OPR	BS-8I-0-3
				TS3 (1)	BS-8I-0-2
				MB03=0	BS-8I-0-8
				B FETCH (1)	BS-8I-0-3

Table 5-14 (Cont)
Fetch Cycle, OPR Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
RAR Rotate AC and Link one right			MB10=0		
			(MB08=1)		
			RIGHT SHIFT		BS-8I-0-5
				MB08=1	BS-8I-0-8
				MB10=0	BS-8I-0-8
				OPR	BS-8I-0-3
				TS3 (1)	BS-8I-0-2
				MB03=0	BS-8I-0-8
				B FETCH (1)	BS-8I-0-3
RAL Rotate AC and Link one left			(MB09=1)		
			LEFT SHIFT		BS-8I-0-5
				MB09=1	BS-8I-0-8
				MB10=0	BS-8I-0-8
				OPR	BS-8I-0-3
				TS3 (1)	BS-8I-0-2
				MB03=0	BS-8I-0-8
				B FETCH (1)	BS-8I-0-3
Rotate AC and Link right two	T3 (cont)	TS3 (cont)	MB10=1		
			(MB08=1)		
			DOUBLE RIGHT ROTATE		BS-8I-0-5
				MB08=1	BS-8I-0-8
				MB10=1	BS-8I-0-8
				OPR1	BS-8I-0-5
Rotate AC and Link left two			(MB09=1)		
			DOUBLE LEFT ROTATE		BS-8I-0-5
				MB09=1	BS-8I-0-8
				MB10=1	BS-8I-0-8
				OPR	BS-8I-0-3
				TS3 (1)	BS-8I-0-2
				MB08=0	BS-8I-0-8
				B FETCH (1)	BS-8I-0-3
+1 → AC			(MB11=1)		
			AC ENABLE		BS-8I-0-4
				MB04=MB06	BS-8I-0-8
+1 → AC			CARRY INSERT		BS-8I-0-5
				OP1	BS-8I-0-5
				MB11=1	BS-8I-0-8

Table 5-14 (Cont)
Fetch Cycle, OPR Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
Skip on a zero AC			MB03=1 (OP2) SZA (MB06=1·AC=0) 1 → SKIP (Enable)		BS-8I-0-6
				MB08=1	BS-8I-0-8
				MB11=0	BS-8I-0-8
				OP2	BS-8I-0-4
				AC00=1	BS-8I-0-8
				MB05=1	BS-8I-0-8
Skip on a non-zero Link			SNL (MB07=1·L=1) 1 → SKIP (Enable)		BS-8I-0-6
				MB08=0	BS-8I-0-8
				MB11=0	BS-8I-0-8
				OP2	BS-8I-0-4
				AC00=1	BS-8I-0-8
				MB05=1	BS-8I-0-8
Skip unconditionally 1 → SKIP	T3 (cont)	TS3 (cont)	Reverse SKIP Sens (MB08=1) 1 → SKIP (Enable)		BS-8I-0-6
If combined with SMA, SZA and SNL instructions, the inverse occurs i.e., SPA, SNA, and SZL				MB11=1	BS-8I-0-8
				MB08=1	BS-8I-0-8
				OP2	BS-8I-0-4
				AC/L Sense Out	BS-8I-0-6
Halt operation 0 → RUN			HLT (MB10=1) 0 → RUN (Enable)		BS-8I-0-2
				MB10=1	BS-8I-0-8
				MB11=0	BS-8I-0-8
				OP2	BS-8I-0-4
No operation AC → AC			NOP (MB04=0·MB09=0) AC ENABLE		BS-8I-0-4
				MB04=0	BS-8I-0-8
				AC → MQ ENABLE	BS-KE8I-0-2
				MB03=1	BS-8I-0-8
				OPR	BS-8I-0-3
				TS3 (1)	BS-8I-0-2
				B FETCH (1)	BS-8I-0-3

Table 5-14 (Cont)
Fetch Cycle, OPR Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
AC+SR → AC			OSR (MB04=0·MB09=1)		
			AC ENABLE		BS-8I-0-4
				(Same as NOP above)	
AC+SR → AC			SR ENABLE		BS-8I-0-4
				OP2	BS-8I-0-4
				MB09=1	BS-8I-0-8
				MB11=0	BS-8I-0-8
0 → AC			CLA (MB04=1·MB09=0)		
			No Operation		
SR → AC			CLA/OSR (MB04=1·MB09=1)		
			SR ENABLE		BS-8I-0-4
				OP2	BS-8I-0-4
				MB09=1	BS-8I-0-8
				MB11=0	BS-8I-0-8
NOP (MB04=0·MB06=0)	T3	TP3	MB03=0 (OP1)		
CMA (MB04=0·MB06=1)			AC LOAD		BS-8I-0-6
CLA (MB04=1·MB06=0)					
CLA/CMA (MB04=1·MB06=1)					
				TP3	BS-8I-0-2
				B FETCH (1)	BS-8I-0-3
				OPR	BS-8I-0-3
				AC LOAD	BS-8I-0-6
			NOTE: No operations occur at TP3 of an OP1 FETCH cycle for any combination of MB08, MB09, or MB10. These bits generate levels which extend throughout T3.		
+1 → AC			IAC (MB11=1)		
			AC LOAD		BS-8I-0-6
				TP3	BS-8I-0-2
				B FETCH (1)	BS-8I-0-3
				OPR	BS-8I-0-3

Table 5-14 (Cont)
Fetch Cycle, OPR Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
SZA(MB06=1•AC=0) SNA(MB05=1•AC00=1) SNL(MB07=1•L=1)			Reverse SKIP sens (MB08=0) 1 → SKIP (strobe)		
					BS-8I-0-6
				TP3	BS-8I-0-2
				B FETCH (1)	BS-8I-0-3
				OPR	BS-8I-0-3
Halt Operation 0 → RUN			HLT (MB10=1) 0 → RUN (strobe)		
					BS-8I-0-2
				TP3	BS-8I-0-2
NOP(MB04=0•MB09=0) OSR(MB04=0•MB09=1) CLA(MB04=1•MB09=0) CLA/OSR(MB04=1•MB09=1)			AC LOAD		BS-8I-0-6
				TP3	BS-8I-0-2
				B FETCH (1)	BS-8I-0-3
				OPR	BS-8I-0-3
PC → MA	T4	TS4	If $\overline{\text{BRK REQ}}$ and (SKIP=0) then: PC ENABLE		
					BS-8I-0-4
				$\overline{\text{TS4 (1)}}$	BS-8I-0-2
				$\overline{\text{F SET}}$	BS-8I-0-3
PC+1 → MA			(SKIP=1) PC ENABLE		
					BS-8I-0-4
				$\overline{\text{TS4 (1)}}$	BS-8I-0-2
				$\overline{\text{F SET}}$	BS-8I-0-3
PC +1 → MA			$\overline{\text{CARRY INSERT}}$		
					BS-8I-0-5
				SKIP=1	BS-8I-0-6
				$\overline{\text{PC ENABLE}}$	BS-8I-0-4
Fetch State entry			1 → F F SET (Enable)		
					BS-8I-0-3
				$\overline{\text{D SET}}$	BS-8I-0-3
				$\overline{\text{E SET}}$	BS-8I-0-3
				$\overline{\text{BREAK OK}}$	BS-8I-0-3
				$\overline{\text{SPECIAL CYCLE}}$	BS-8I-0-3

Table 5-14 (Cont)
Fetch Cycle, OPR Instruction Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
PC → MA PC+1 → MA	TP4		$\overline{\text{BRK REQ}}$ and SKIP = 0 or SKIP = 1 then: MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2
DATA ADD → MA	T4	TS4	If there is a DATA BRK REQ then: DATA ENABLE		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				BREAK OK	BS-8I-0-3
Break State entry		TP4	If 1 cycle DATA BREAK then: 1 → B		BS-8I-0-3
				TP4	BS-8I-0-2
				B SET	BS-8I-0-3
Word Count State entry	T4 (cont)	TP4 (cont)	If 3 cycle DATA BREAK then: 1 → WC		BS-8I-0-3
				TP4	BS-8I-0-2
				WC SET	BS-8I-0-2
DATA ADD → MA			MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2
0 → MA			If the BRK REQ is a PROGRAM BREAK REQUEST then: MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2
JMS → IR			JMS (forced) → IR		BS-8I-0-3
				TP4	BS-8I-0-2
				INT OK	BS-8I-0-7
Execute State entry			1 → EXECUTE		BS-8I-0-3
				TP4	BS-8I-0-2
				E SET	BS-8I-0-3

Table 5-15
Fetch Cycle, JMP Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MA+1 → PC	T1	TS1/ TP1	Same as AND instruction		
MEM → MB MEM → IR	T2	TS2/ TP2	Same as AND instruction		
MEM5-11 → PC5-11	T3	TS3	If MB03=0 then: MEM ENABLE 5-11		BS-8I-0-4
				TS3 (1)	BS-8I-0-2
				JMP	BS-8I-0-3
				B FETCH (1)	BS-8I-0-3
				MB03 (0)	BS-8I-0-8
0 → PC0-4			If MB04=0 then: PC ENABLE not generated		
MA0-4 → PC0-4			If MB04=1 then: MA ENABLE 0-4		BS-8I-0-4
				MEM ENABLE 5-11	BS-8I-0-4
				MB04 (1)	BS-8I-0-8
				(MEM ENABLE 0-4 not generated)	BS-8I-0-4
MA0-4 → PC0-4 MEM5-11 → PC5-11		TP3	PC LOAD		BS-8I-0-6
				TP3	BS-8I-0-2
				JMP	BS-8I-0-3
				MB03 (0)	BS-8I-0-8
				B FETCH (1)	BS-8I-0-3
	T4	TS4/ TP4	Same as OPR instruction		

Table 5-16
Fetch Cycle, IOT Instructions, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MA+1 →PC	T1	TS1/ TP1	Same as AND instruction		
MEM →MB MEM →IR	T2	TS2/ TP2	Same as AND instruction		
1 →PAUSE	T3	TS3 TP3	1 →PAUSE	$\overline{\text{IO START}}$	BS-8I-0-2 BS-8I-0-2
Generate IOP1			IOP1 (1)	$\overline{\text{IO START}}$ MB11(1)	BS-8I-0-2 BS-8I-0-2 BS-8I-0-8
Generate IOP2			IOP2 (1)	$\overline{\text{IO START}}$ MB10 (1)	BS-8I-0-2 BS-8I-0-2 BS-8I-0-8
Generate IOP4			IOP4 (1)	$\overline{\text{IO START}}$ MB09 (1)	BS-8I-0-2 BS-8I-0-2 BS-8I-0-8

Table 5-17
Execute Cycle, AND Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM →SENSE	T1	TS1/ TP1	NONE Operand to Sense	$\overline{\text{STROBE}}$	BS-8I-0-14 BS-8I-0-13
MEM →MB	T2	TS2	MEM ENABLE0-4/5-11	TS2 (1) All of the following NOT true: KEY DEP, DCA, JMS and EXECUTE BREAK	BS-8I-0-4 BS-8I-0-2

Table 5-17 (Cont)
Execute Cycle, AND Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM → MB		TP2	MB LOAD		BS-8I-0-6
				TP2	BS-8I-0-2
AC·MB → AC	T3	TS3	AC ENABLE		BS-8I-0-4
				$\overline{\text{AND ENABLE}}$	BS-8I-0-5
			$\overline{\text{AND ENABLE}}$		BS-8I-0-5
				AND	BS-8I-0-3
				TS3 (1)	BS-8I-0-2
				B EXECUTE (1)	BS-8I-0-3
			AND ENABLE		BS-8I-0-5
				$\overline{\text{AND ENABLE}}$	BS-8I-0-5
AC·MB → AC		TP3	AC LOAD		BS-8I-0-6
				$\overline{\text{TP3}}$	BS-8I-0-2
				$\overline{\text{AND}}$	BS-8I-0-3
				B EXECUTE (1)	BS-8I-0-3
PC → MA	T4	TS4	$\overline{\text{BRK REQ}}$ and SKIP=0 then: PC ENABLE		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				$\overline{\text{F SET}}$	BS-8I-0-3
PC+1 → MA			(SKIP=1) PC ENABLE		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				$\overline{\text{F SET}}$	BS-8I-0-3
PC+1 → MA			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				SKIP=1	BS-8I-0-6
				$\overline{\text{PC ENABLE}}$	BS-8I-0-4
Fetch State entry	T4 (cont)	TS4 (cont)	1 → F F SET (Enable)		BS-8I-0-3
				$\overline{\text{D SET}}$	BS-8I-0-3
				$\overline{\text{E SET}}$	BS-8I-0-3
				$\overline{\text{BREAK OK}}$	BS-8I-0-3
				$\overline{\text{SPECIAL CYCLE}}$	BS-8I-0-3

Table 5-17 (Cont)
Execute Cycle, AND Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
PC → MA PC+1 → MA		TP4	$\overline{\text{BRK REQ}}$ and (SKIP=0) then: (SKIP=1) MA LOAD	 TP4	 BS-8I-0-6 BS-8I-0-2
DATA ADD → MA	T4	TS4	If there is a DATA BRK REQ then: DATA ENABLE	 TS4 (1) BREAK OK	 BS-8I-0-4 BS-8I-0-2 BS-8I-0-3
Break State entry		TP4	If 1 cycle DATA BREAK then: 1 → B	 TP4 B SET	 BS-8I-0-3 BS-8I-0-2 BS-8I-0-3
Word Count State entry	T4	TP4	If 3 cycle DATA BREAK then: 1 → WC	 TP4 WC SET	 BS-8I-0-3 BS-8I-0-2 BS-8I-0-2
DATA ADD → MA			MA LOAD	 TP4	BS-8I-0-6 BS-8I-0-2
0 → MA			If the BRK REQ is a PROGRAM BREAK REQUEST then: MA LOAD	 TP4	 BS-8I-0-6 BS-8I-0-2
JMS → IR			JMS (forced) → IR	 TP4 INT OK	BS-8I-0-3 BS-8I-0-2 BS-8I-0-7
Execute State entry			1 → EXECUTE	 TP4 E SET	BS-8I-0-3 BS-8I-0-2 BS-8I-0-3

Table 5-18
Execute Cycle, TAD Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM → SENSE	T1	TS1	NONE		
		TP1	Operand to SENSE		BS-8I-0-14
				$\overline{\text{STROBE}}$	BS-8I-0-13
MEM → MB	T2	TS2	MEM ENABLE0-4/5-11		BS-8I-0-4
				TS2 (1)	BS-8I-0-2
				All of the following NOT true: KEY DP, (DCA V JMS), EXECUTE, BREAK	
MEM → MB MEM → IR		TP2	MB LOAD		BS-8I-0-6
				TP2	BS-8I-0-2
AC+MEM → AC	T3	TS3	AC ENABLE		BS-8I-0-4
				TAD	BS-8I-0-3
				B EXECUTE (1)	BS-8I-0-3
				TS3 (1)	BS-8I-0-2
			MEM ENABLE0-4/5-11		BS-8I-0-4
				$\overline{\text{ADD}}$	BS-8I-0-4
AC+MEM → AC		TP3	AC LOAD		BS-8I-0-6
				B EXECUTE (1)	BS-8I-0-3
				$\overline{\text{TAD}}$	BS-8I-0-3
				TP3	BS-8I-0-2
	T4	TS4/ TP4	Same as AND instruction during T4 of EXECUTE Cycle.		

Table 5-19
Execute Cycle, ISZ Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM → SENSE	T1	TS1/ TP1	NONE Operand to SENSE	 STROBE	 BS-8I-0-14 BS-8I-0-13
MEM+1 → MB	T2	TS2	MEM ENABLE0-4/5-11	 TS2 (1)	BS-8I-0-4 BS-8I-0-2
MEM+1 → MB			$\overline{\text{CARRY INSERT}}$	 TS2 (1) ISZ B EXECUTE (1)	BS-8I-0-5 BS-8I-0-2 BS-8I-0-3 BS-8I-0-3
		TP2	If MB=0 after MEM+1 → MB then: 1 → SKIP	 TP2 B EXECUTE (1) TS2 (1) B EXECUTE (1) CARRY OUT 0 ISZ	BS-8I-0-6 BS-8I-0-2 BS-8I-0-3 BS-8I-0-2 BS-8I-0-3 BS-8I-0-8 BS-8I-0-2
MEM+1 → MB			MB LOAD	 TP2	BS-8I-0-6 BS-8I-0-2
	T3	TS3/ TP3	NONE		
	T4	TS4/ TP4	Same as AND instruction during T4 of EXECUTE Cycle.		

Table 5-20
Execute Cycle, DCA Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM → SENSE	T1	TS1	No Operations		
		TP1	Operand from memory to SENSE		
				$\overline{\text{STROBE}}$	BS-8I-0-14 BS-8I-0-13
AC → MB	T2	TS2	AC ENABLE		BS-8I-0-4
				TS2 (1)	BS-8I-0-2
				B EXECUTE (1)	BS-8I-0-3
				DCA	BS-8I-0-3
AC → MB		TP2	MB LOAD		BS-8I-0-6
				TP2	BS-8I-0-2
0 → AC	T3	TS3	No Operations		
		TP3	AC LOAD		BS-8I-0-6
				TP3	BS-8I-0-2
				$\overline{\text{DCA}}$	BS-8I-0-3
				B EXECUTE (1)	BS-8I-0-3
	T4	TS4/ TP4	Same as AND instruction during T4 of EXECUTE Cycle.		

Table 5-21
Execute Cycle, JMS Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM → SENSE	T1	TS1/ TP1	NONE Operand to SENSE	 STROBE	 BS-8I-0-14 BS-8I-0-13
PC → MB	T2	TS2	When SKIP=0 PC ENABLE	 INT SKIP ENABLE JMS	 BS-8I-0-4 BS-8I-0-4 BS-8I-0-3
			When SKIP=1 PC ENABLE	 (as above)	 BS-8I-0-4
			When SKIP=1 CARRY INSERT	 SKIP (1) INT SKIP ENABLE	 BS-8I-0-5 BS-8I-0-6 BS-8I-0-5
PC → MB		TP2	MB LOAD	 TP2	BS-8I-0-6 BS-8I-0-2
MA+1 → PC	T3	TS3	MA ENABLE0-4/5-11	 TS3 (1) JMS B EXECUTE (1)	BS-8I-0-4 BS-8I-0-2 BS-8I-0-3 BS-8I-0-3
			CARRY INSERT	 TS3 (1) B EXECUTE (1) JMS	BS-8I-0-5 BS-8I-0-2 BS-8I-0-3 BS-8I-0-3
MA+1 → PC		TP3	PC LOAD	 TP3 JMS B EXECUTE (1)	 BS-8I-0-2 BS-8I-0-3 BS-8I-0-3
Fetch State entry	T4	TS4/ TP4	Same as AND instruction during T4 EXECUTE Cycle.		

Table 5-22
Defer Cycle, JMP Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM → SENSE	T1	TS1	None		
		TP1	Address from memory to SENSE		BS-8I-0-14
				$\overline{\text{STROBE}}$	BS-8I-0-13
MEM+1 → MB [Auto Index]	T2	TS2	MEM ENABLE0-4/5-11		BS-8I-0-4
				TS2 (1)	BS-8I-0-2
MEM+1 → MB [Auto Index]			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				$\overline{\text{AUTO INDEX}}$	BS-8I-0-5
			MEM ENABLE0-4/5-11		BS-8I-0-4
				TS2 (1)	BS-8I-0-2
$\overline{\text{MEM → MB}}$ [Auto Index]		TP2	MB LOAD		BS-8I-0-2
				TP2	BS-8I-0-2
MEM+1 → MB [Auto Index]					
MEM → IR					
	T3	TS3/ TP3	No Operations		
MEM+1 → MA [Auto Index]	T4	TS4	(Auto Index) MEM ENABLE0-4/5-11		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				DEFER (1)	BS-8I-0-3
				$\overline{\text{JMP}}$	BS-8I-0-3
MEM+1 → MA [AUTO INDEX]			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				$\overline{\text{AUTO INDEX}}$	
MEM → MA [Auto Index]			$\overline{\text{(AUTO INDEX)}}$ MEM ENABLE0-4/5-11		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				DEFER (1)	BS-8I-0-3
				$\overline{\text{JMP}}$	BS-8I-0-3

Table 5-22 (Cont)
Defer Cycle, JMP Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
Execute State entry	T4	TS4	1 → EXECUTE (Enable)		BS-8I-0-3
				JMP	BS-8I-0-3
				DEFER (1)	BS-8I-0-3
MEM+1 → MA		TP4	(AUTO INDEX/AUTO INDEX)		
			MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2
Execute State entry			1 → EXECUTE (Set)		BS-8I-0-3
				TP4	BS-8I-0-2

Table 5-23
Defer Cycle, JMP Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM → SENSE	T1	TS1/ TP1	Address from memory to SENSE		
				$\overline{\text{STROBE}}$	BS-8I-0-14 BS-8I-0-13
MEM+1 → MB [AUTO INDEX]	T2	TS2	Same as $\overline{\text{JMP}}$ instruction		
$\overline{\text{MEM → MB [AUTO INDEX]}}$		TP2			
MEM+1 → PC	T3	TS3	[Auto Index]		
			MEM ENABLE0-4/5-11		BS-8I-0-4
				TS3 (1)	BS-8I-0-2
				DEFER (1)	BS-8I-0-3
				JMP	BS-8I-0-3

Table 5-23 (Cont)
Defer Cycle, JMP Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
MEM+1 →PC			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				$\overline{\text{AUTO INDEX}}$	BS-8I-0-5
MEM →PC			$\overline{\text{[AUTO INDEX]}}$		
			MEM ENABLE0-4/5-11		BS-8I-0-4
				TS3 (1)	BS-8I-0-2
				DEFER (1)	BS-8I-0-3
				JMP	BS-8I-0-3
MEM+1 →PC MEM →PC		TP3	PC LOAD		BS-8I-0-6
				JMP	BS-8I-0-3
				TP3	BS-8I-0-2
				DEFER (1)	BS-8I-0-3
PC →MA	T4	TS4	If BRK REQ and SKIP=0 then:		
			PC ENABLE		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				$\overline{\text{F SET}}$	BS-8I-0-3
PC+1 → MA			(SKIP 1)		
			PC ENABLE		BS-8I-0-4
				TS (1)	BS-8I-0-2
				$\overline{\text{F SET}}$	BS-8I-0-3
PC+1 →MA			CARRY INSERT		BS-8I-0-5
				SKIP=1	BS-8I-0-6
				$\overline{\text{PC ENABLE}}$	BS-8I-0-4
Fetch State entry	T4 (cont)	TS4 (cont)	1 →F		
			F SET (Enable)		BS-8I-0-3
				$\overline{\text{D SET}}$	BS-8I-0-3
				$\overline{\text{E SET}}$	BS-8I-0-3
				$\overline{\text{BREAK OK}}$	BS-8I-0-3
				$\overline{\text{SPECIAL CYCLE}}$	BS-8I-0-3
PC →MA PC+1 →MA		TP4	$\overline{\text{BRK REQ}}$ and (SKIP=0) then (SKIP=1)		
			MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2

Table 5-23 (Cont)
Defer Cycle, JMP Instruction, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
DATA ADD → MA	T4	TS4	If there is a DATA BRK REQ then:		
			DATA ENABLE		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				BREAK OK	BS-8I-0-3
Break State entry		TP4	If 1 cycle DATA BREAK then:		
			1 → B		BS-8I-0-3
				TP4	BS-8I-0-2
				B SET	BS-8I-0-3
Word Count State Entry			If 3 cycle DATA BREAK then:		
			1 → WC		BS-8I-0-3
				TP4	BS-8I-0-2
				WC SET	BS-8I-0-2
DATA ADD → MA			MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2
0 → MA			If the BRK REQ is a PROGRAM BREAK REQUEST then:		
			MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2
JMS → IR			JMS (forced) → IR		BS-8I-0-3
				TP4	BS-8I-0-2
				INT OK	BS-8I-0-7
Execute State entry			1 → EXECUTE		BS-8I-0-3
				TP4	BS-8I-0-2
				E SET	BS-8I-0-3

Table 5-24
Word Count Cycle, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
	T1	TS1/ TP1	NONE		
MEM → MB		TS2	If $\overline{\text{CA INCREMENT}}$ then:		
			MEM ENABLE0-4/5-11		BS-8I-0-4
				TS2 (1)	BS-8I-0-2
MEM+1 → MB		TS2	If CA INCREMENT then:		
			MEM ENABLE0-4/5-11		BS-8I-0-4
				As above	
MEM+1 → MB			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				CURRENT ADDRESS (1)	BS-8I-0-3
				CA INCREMENT	BS-8I-0-5
MEM → MB		TP2	MB LOAD		
MEM+1 → MB					BS-8I-0-6
				TP2	BS-8I-0-2
	T3	TS3/ TP3	NONE		
MEM+1 → MA	T4	TS4	If CA INCREMENT then:		
			MEM ENABLE0-4/5-11		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				CURRENT ADDRESS (1)	BS-8I-0-3
MEM+1 → MA			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				CURRENT ADDRESS (1)	BS-8I-0-3
				CA INCREMENT	BS-8I-0-5
MEM → MA			If $\overline{\text{CA INCREMENT}}$ then:		
			MEM ENABLE0-4/5-11		BS-8I-0-4
				As above	
MEM+1 → MA		TP4	MA LOAD		
MEM → MA					BS-8I-0-6
				TP4	BS-8I-0-2
Break State entry		TP4	1 → B		BS-8I-0-3
				TP4	BS-8I-0-2
				B SET	BS-8I-0-3

NOTE ALL instructions suspended. (The first cycle of the 3-cycle Break)

Table 5-25
Current Address Cycle, Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
	T1	TS1/ TP1	NONE		
MEM → MB		TS2	If $\overline{\text{CA INCREMENT}}$ then: MEM ENABLE0-4/5-11		BS-8I-0-4
				TS2 (1)	BS-8I-0-2
MEM+1 → MB		TS2	If CA INCREMENT then: MEM ENABLE0-4/5-11		BS-8I-0-4
				As above	
MEM+1 → MB			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				CURRENT ADDRESS (1)	BS-8I-0-3
				CA INCREMENT	BS-8I-0-5
MEM → MB		TP2	MB LOAD		
MEM+1 → MB					BS-8I-0-6
				TP2	BS-8I-0-2
	T3	TS3/ TP3	NONE		
MEM+1 → MA	T4	TS4	If CA INCREMENT then: MEM ENABLE0-4/5-11		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				CURRENT ADDRESS (1)	BS-8I-0-3
MEM+1 → MA			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				CURRENT ADDRESS (1)	BS-8I-0-3
				CA INCREMENT	BS-8I-0-5
MEM → MA			If $\overline{\text{CA INCREMENT}}$ then: MEM ENABLE0-4/5-11		BS-8I-0-4
				As above	
MEM+1 → MA		TP4	MA LOAD		
MEM → MA					BS-8I-0-6
				TP4	BS-8I-0-2
Break State entry		TP4	1 → B		BS-8I-0-3
				TP4	BS-8I-0-2
				B SET	BS-8I-0-3

NOTE ALL Instructions (2nd cycle of 3-cycle Break)

Table 5-26
Break Cycle Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
	T1	TS1/ TP1	NONE		
DATA → MB	T2	TS2	If DATA IN then:		
			DATA ENABLE		BS-8I-0-4
				TS2(1)	BS-8I-0-2
				DATA IN	BS-8I-0-4
				BREAK (1)	BS-8I-0-3
MEM → MB			If DATA OUT then:		
			MEM ENABLE0-4/5-11		BS-8I-0-4
				TS2 (1)	BS-8I-0-2
MEM+1 → MB			If MEM INCREMENT then:		
			MEM ENABLE0-4/5-11		BS-8I-0-4
				As above	
MEM+1 → MB			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				TS2 (1)	BS-8I-0-2
				MEMORY INCREMENT	BS-8I-0-10
				BREAK (1)	BS-8I-0-3
DATA → MB MEM → MB MEM+1 → MB		TP2	MB LOAD		BS-8I-0-6
				TP2	BS-8I-0-2
PC → MA	T4	TS4	If $\overline{\text{BRK REQ}}$ and SKIP=0 then:		
			PC ENABLE		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				F SET	BS-8I-0-3
PC+1 → MA			(SKIP=1)		
			PC ENABLE		BS-8I-0-4
				TS4 (1)	BS-8I-0-2
				F SET	BS-8I-0-3

NOTE ALL Instructions suspended (Third cycle of 3-cycle Break)

Table 5-26 (Cont)
Break Cycle Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
PC+1 →MA			$\overline{\text{CARRY INSERT}}$		BS-8I-0-5
				SKIP=1	BS-8I-0-6
				$\overline{\text{PC ENABLE}}$	BS-8I-0-4
FETCH STATE entry	T4	TS4	1 →F		
			F SET (Enable)		BS-8I-0-3
				$\overline{\text{D SET}}$	BS-8I-0-3
				$\overline{\text{E SET}}$	BS-8I-0-3
				$\overline{\text{BREAK OK}}$	BS-8I-0-3
				$\overline{\text{SPECIAL CYCLE}}$	BS-8I-0-3
PC →MA		TP4	If $\overline{\text{BRK REQ}}$ and		
PC+1 →MA			(SKIP=0) then:		
			(SKIP=1)		
			MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2
DATA ADD →MA	T4	TS4	If there is a DATA BRK REQ then:		
			DATA ENABLE		BS-8I-0-4
				TS4(1)	BS-8I-0-2
				BREAK OK	BS-8I-0-3
BREAK STATE entry		TP4	If 1 cycle DATA BREAK then:		
			1 →B		BS-8I-0-3
				TP4	BS-8I-0-2
				B SET	BS-8I-0-3
WORD COUNT STATE entry			If 3 cycle DATA BREAK then:		
			1 →WC		BS-8I-0-3
				TP4	BS-8I-0-2
				WC SET	BS-8I-0-2
DATA ADD →MA			MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2

NOTE ALL Instructions suspended (Third cycle of 3-cycle Break)

Table 5-26 (Cont)
Break Cycle Mechanization Chart

Function	Time	State or Pulse	Signal Required	Generated By	Drawing Reference
0 → MA			If the BRK REQ is a PROGRAM BREAK REQUEST then:		
			MA LOAD		BS-8I-0-6
				TP4	BS-8I-0-2
JMS → IR			JMS (forced) → IR		BS-8I-0-3
				TP4	BS-8I-0-2
				INT OK	BS-8I-0-7
EXECUTE STATE entry			1 → EXECUTE		BS-8I-0-3
				TP4	BS-8I-0-2
				E SET	BS-8I-0-3

NOTE ALL Instructions suspended (Third cycle of 3-cycle Break)

5.9 Table 5-27 provides maintenance personnel with a reference to the origin of selected signals generated within the PDP-8/I. The signal name appears in the left column, exactly as it is shown on the engineering drawings. The specific drawing upon which that signal is generated appears in the right column.

Table 5-27
Signal Origins

Signal	Drawing No.
$\overline{\text{AC CLEAR}}$	D-BS-8I-0-4
AC ENABLE	D-BS-8I-0-4
$\overline{\text{AC}}$ ENABLE	D-BS-8I-0-4
AC LOAD	D-BS-8I-0-6
AC → MQ ENABLE	D-BS-KE8I-0-2
$\overline{\text{AC}}$ → MQ ENABLE	D-BS-KE8I-0-2
ADD	D-BS-8I-0-4
ADD ACCEPTED	D-BS-8I-0-7
ADDER L	D-BS-8I-0-8
$\overline{\text{ADDER L}}$	D-BS-KE8I-0-3
AND	D-BS-8I-0-3
$\overline{\text{AND}}$	D-BS-8I-0-3
AND ENABLE	D-BS-8I-0-5
ASR ENABLE	D-BS-KE8I-0-2
$\overline{\text{ASR L SET}}$	D-BS-KE8I-0-2
AUTO INDEX	D-BS-8I-0-5
B EAE ON	D-BS-KE8I-0-2
B EXECUTE (1)	D-BS-8I-0-3
B FETCH (1)	D-BS-8I-0-3
B FIELD (0)	D-BS-8I-0-13
B FIELD (1)	D-BS-8I-0-13
B INHIBIT	D-BS-8I-0-13
B MEM ENABLE	D-BS-8I-0-13
B SET	D-BS-8I-0-3
BREAK	D-BS-8I-0-3
BREAK OK	D-BS-8I-0-3
$\overline{\text{BREAK OK}}$	D-BS-8I-0-3
CARRY INSERT	D-BS-8I-0-5
CURRENT ADDRESS	D-BS-8I-0-3
D SET	D-BS-8I-0-3
$\overline{\text{D SET}}$	D-BS-8I-0-3
DATA ADD ENABLE	D-BS-8I-0-4
DATA ENABLE	D-BS-8I-0-4
DCA	D-BS-8I-0-3

Signal	Drawing No.	Signal	Drawing No.
DCA	D-BS-8I-0-3	I/O STROBE	D-BS-8I-0-2
DEFER	D-BS-8I-0-3	INHIBIT	D-BS-8I-0-13
DIV LAST	D-BS-KE8I-0-2	INITIALIZE	D-BS-8I-0-2
DIV LAST	D-BS-KE8I-0-2	INITIALIZE	D-BS-8I-0-2
DOUBLE LEFT ROTATE	D-BS-8I-0-5	INT DELAY	D-BS-8I-0-7
DOUBLE RIGHT ROTATE	D-BS-8I-0-5	INIT ENABLE	D-BS-8I-0-7
DVI	D-BS-8I-0-2	INT OK	D-BS-8I-0-7
DVI	D-BS-8I-0-2	INT OK	D-BS-8I-0-7
EAE AC ENABLE	D-BS-KE8I-0-2	INT SKIP ENABLE	D-BS-8I-0-4
EAE AC ENABLE	D-BS-KE8I-0-2	INT STROBE	D-BS-8I-0-2
EAE BEGIN	D-BS-KE8I-0-2	INT SYNC	D-BS-8I-0-7
EAE COMPLETE	D-BS-KE8I-0-2	IOP 1	D-BS-8I-0-2
EAE E SET	D-BS-KE8I-0-2	IOP 2	D-BS-8I-0-2
EAE END	D-BS-KE8I-0-2	IOP 4	D-BS-8I-0-2
EAE EXECUTE	D-BS-KE8I-0-2	IOT	D-BS-8I-0-3
EAE INST	D-BS-KE8I-0-2	IOT	D-BS-8I-0-3
EAE IR0	D-BS-KE8I-0-2	ISZ	D-BS-8I-0-3
EAE IR1	D-BS-KE8I-0-2	ISZ	D-BS-8I-0-3
EAE IR2	D-BS-KE8I-0-2	JMP	D-BS-8I-0-3
EAE L DISABLE	D-BS-KE8I-0-2	JMP	D-BS-8I-0-3
EAE LEFT SHIFT ENABLE	D-BS-KE8I-0-2	JMS	D-BS-8I-0-3
EAE MEM ENABLE	D-BS-KE8I-0-2	JMS	D-BS-8I-0-3
EAE MQ 0 ENABLE	D-BS-KE8I-0-2	KEY CONT	D-BS-8I-0-2
EAE MQ 0 ENABLE	D-BS-KE8I-0-2	KEY DP	D-BS-8I-0-2
EAE NO SHIFT ENABLE	D-BS-KE8I-0-2	KEY EX + DP	D-BS-8I-0-2
EAE ON (0)	D-BS-KE8I-0-2	KEY LA	D-BS-8I-0-2
EAE ON (1)	D-BS-KE8I-0-2	KEY SI + STOP	D-BS-8I-0-2
EAE RIGHT SHIFT ENABLE	D-BS-KE8I-0-2	KEY ST	D-BS-8I-0-2
EAE RUN (1)	D-BS-KE8I-0-2	KEY SS	D-BS-8I-0-2
EAE SET	D-BS-KE8I-0-2	L ENABLE	D-BS-8I-0-4
EAE START	D-BS-KE8I-0-2	L ENABLE	D-BS-8I-0-4
EAE TG (1)	D-BS-KE8I-0-2	LEFT SHIFT	D-BS-8I-0-5
EAE TP	D-BS-KE8I-0-2	LEFT SHIFT	D-BS-KE8I-0-3
EAE TP	D-BS-KE8I-0-2	MA ENABLE 0-4	D-BS-8I-0-4
E SET	D-BS-8I-0-3	MA ENABLE 5-11	D-BS-8I-0-4
EXECUTE	D-BS-8I-0-3	MA LOAD	D-BS-8I-0-6
F SET	D-BS-8I-0-3	MANUAL PRESET	D-BS-8I-0-2
F SET	D-BS-8I-0-3	MB LOAD	D-BS-8I-0-6
FETCH	D-BS-8I-0-3	MB → SC ENABLE	D-BS-KE8I-0-2
FIELD (0)	D-BS-8I-0-13	MEM BEGIN	D-BS-8I-0-13
FIELD (1)	D-BS-8I-0-13	MEM DONE	D-BS-8I-0-13
I/O ENABLE	D-BS-8I-0-4	MEM ENABLE 0-4	D-BS-8I-0-4
I/O END	D-BS-8I-0-2	MEM ENABLE 5-11	D-BS-8I-0-4
I/O END	D-BS-8I-0-2	MEM ENABLE	D-BS-8I-0-13
I/O START	D-BS-8I-0-2	MEM FINISH	D-BS-8I-0-13

Signal	Drawing No.	Signal	Drawing No
MEM START	D-BS-8I-0-2	RUN	D-BS-8I-0-2
MFTP0	D-BS-8I-0-2	SC 0-3 = 0	D-BS-KE8I-0-3
MFTP1	D-BS-8I-0-2	$\overline{\text{SC 0-3 = 0}}$	D-BS-KE8I-0-3
MFTP2	D-BS-8I-0-2	SC0 - SC4	D-BS-KE8I-0-3
MFTS0	D-BS-8I-0-2	$\overline{\text{SC = 0}}$	D-BS-KE8I-0-2
$\overline{\text{MFTS0}}$	D-BS-8I-0-2	SC ENABLE	D-BS-KE8I-0-2
MFTS1	D-BS-8I-0-2	SC FULL	D-BS-KE8I-0-3
MFTS3	D-BS-8I-0-2	$\overline{\text{SCL}}$	D-BS-KE8I-0-2
MQ00→MQ11	D-BS-KE8I-0-3	SC LOAD	D-BS-KE8I-0-2
MQ ENABLE	D-BS-KE8I-0-2	SHUTDOWN	D-BS-8I-0-13
MQ LOAD	D-BS-KE8I-0-2	SKIP	D-BS-8I-0-6
MQ and LOW AC = 0	D-BS-KE8I-0-2	SLOW CYCLE	D-BS-8I-0-2
MUY	D-BS-KE8I-0-2	$\overline{\text{SLOW CYCLE}}$	D-BS-8I-0-2
$\overline{\text{MUY}}$	D-BS-KE8I-0-2	SPECIAL CYCLE	D-BS-8I-0-3
MUY + DVI	D-BS-KE8I-0-2	SR ENABLE	D-BS-8I-0-4
NMI	D-BS-KE8I-0-2	STOP OK	D-BS-8I-0-13
$\overline{\text{NMI}}$	D-BS-KE8I-0-2	$\overline{\text{STROBE}}$	D-BS-8I-0-13
NORM	D-BS-KE8I-0-2	STROBE FIELD 0	D-BS-8I-0-13
$\overline{\text{NORM}}$	D-BS-KE8I-0-2	STROBE FIELD 1	D-BS-8I-0-13
NO SHIFT	D-BS-8I-0-5	TAD	D-BS-8I-0-3
OP1	D-BS-8I-0-5	$\overline{\text{TAD}}$	D-BS-8I-0-3
OP2	D-BS-8I-0-4	TP1	D-BS-8I-0-2
OPR	D-BS-8I-0-3	TP2	D-BS-8I-0-2
$\overline{\text{OPR}}$	D-BS-8I-0-3	TP3	D-BS-8I-0-2
PAUSE	D-BS-8I-0-2	TP4	D-BS-8I-0-2
PC ENABLE	D-BS-8I-0-4	TS1	D-BS-8I-0-2
$\overline{\text{PC ENABLE}}$	D-BS-8I-0-4	TS2	D-BS-8I-0-2
PC LOAD	D-BS-8I-0-6	TS3	D-BS-8I-0-2
$\overline{\text{POWER CLEAR}}$	D-BS-8I-0-13	TS4	D-BS-8I-0-2
$\overline{\text{POWER OK}}$	D-BS-8I-0-13	WC OVERFLOW	D-BS-8I-0-7
$\overline{\text{PROCESSOR IOT}}$	D-BS-8I-0-7	$\overline{\text{WC SET}}$	D-BS-8I-0-3
PC INCREMENT	D-BS-8I-0-5	WORD COUNT	D-BS-8I-0-3
READ	D-BS-8I-0-13	WRITE	D-BS-8I-0-13
RIGHT SHIFT	D-BS-8I-0-5		

Digital Equipment Corporation
Maynard, Massachusetts

digital

printed in U.S.A.