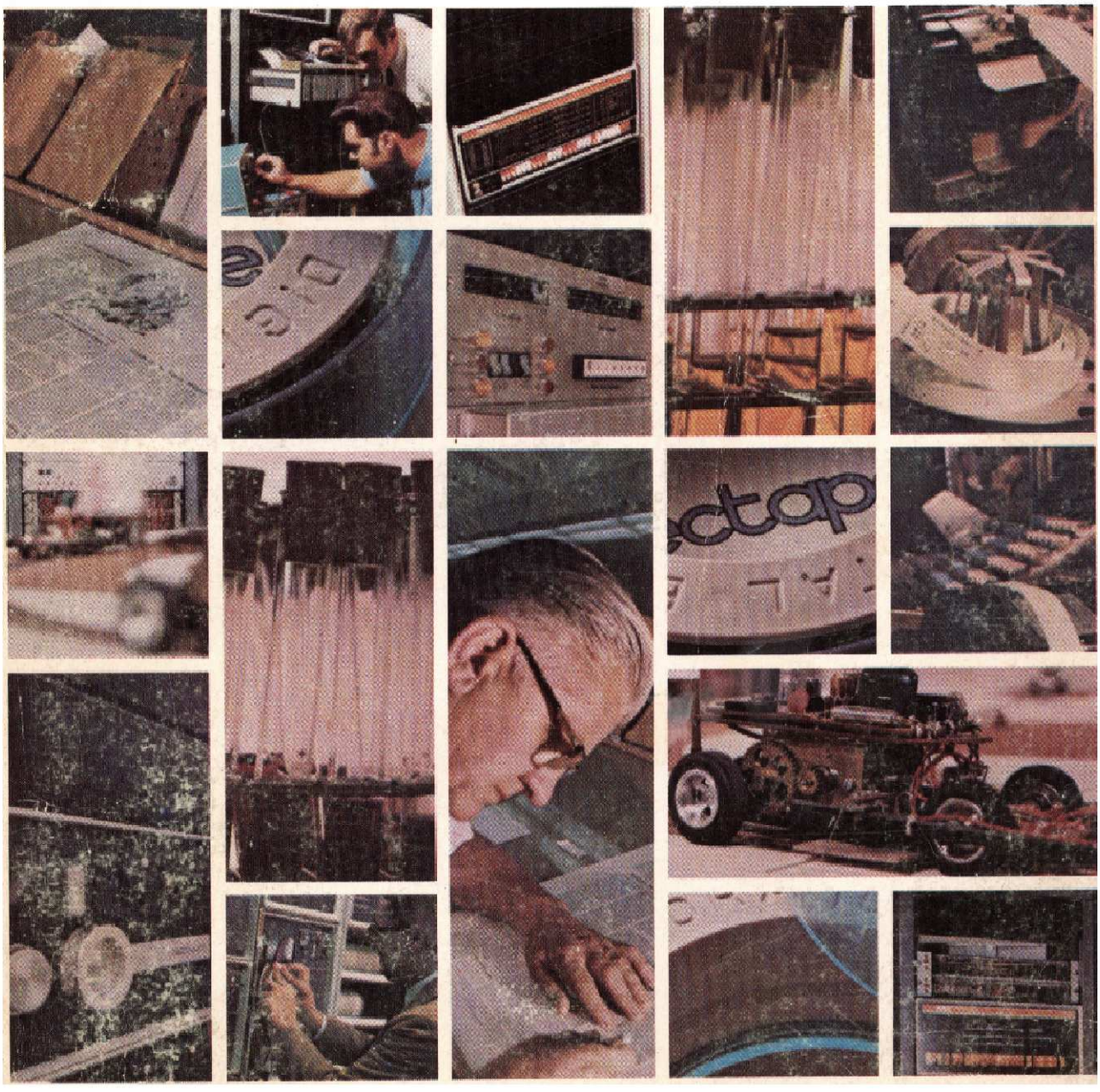
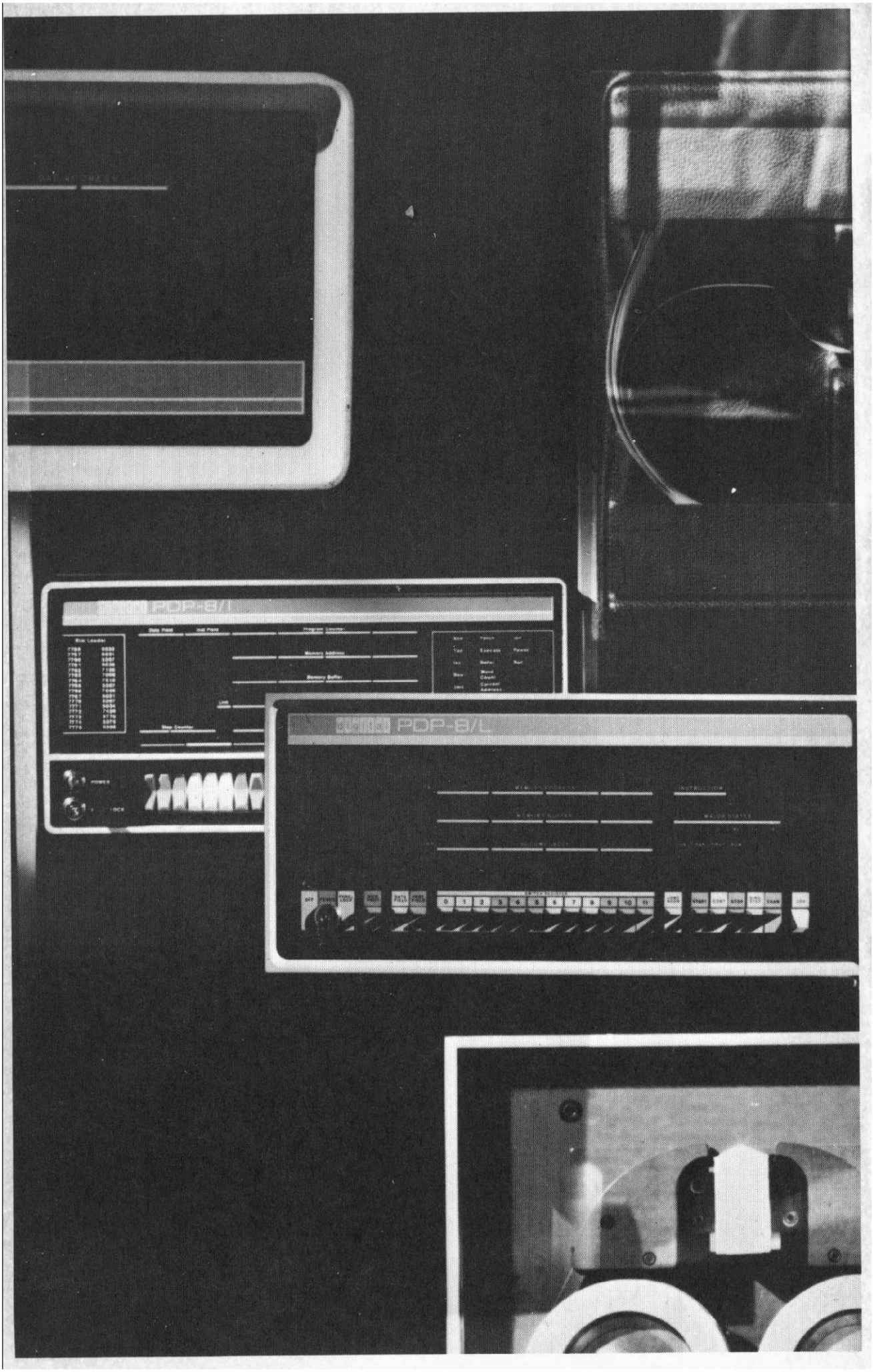
digital

small computer handbook



digital small computer handbook 1970


PDP-8/I
PDP-8/L

THE

digital

SMALL COMPUTER HANDBOOK
1970 EDITION

INTRODUCTION

This handbook is another in a series intended to familiarize the user with the newest members of the Digital Equipment Corporation (DEC) PDP-8 family of small, general-purpose computers. This handbook explains both the PDP-8/I and the PDP-8/L central processors and how they are interfaced and operated with the wide variety of peripheral equipment available.

The following topics are covered in this handbook: System Introduction; Standard System Operation; Memory and Processor Basic Programming; Memory and Processor Instructions; Data Break; Computer Internal Options; Input and Output Equipment and Instructions; Input and Output Facilities; Programmed Data Transfers; Data Break Transfers; Digital Logic Circuits; Interface Design, Technique, and Connections; and Installation and Planning. Related data is included in the appendices.

Another member of this series of handbooks is titled, *An Introduction to Programming*. This programming handbook familiarizes the user with the principles of programming the PDP-8 family of general-purpose computers; together this handbook and the programming handbook describe the complete hardware and software aspects of the PDP-8 family. Of necessity, the user will find that the material in this handbook sometimes reflects the material in the programming handbook; unfortunately, this is necessary to preserve the continuity of descriptions in this handbook. However, the purpose of this handbook is to describe the hardware and its organization into the general-purpose computer system, along with applicable programming instructions.

THE PDP-8/I PROGRAMMED DATA PROCESSOR

The PDP-8/I is a fully-parallel, 12-bit random-access, 4096-word core memory system. It performs general-purpose computations and process control operations. Transistor-Transistor Logic (TTL) integrated circuit modules make the PDP-8/I a compact, economical, and rugged system that is easy to interface and maintain. The PDP-8/I is a single-address, fixed-word-length system that uses 2's complement arithmetic. The basic PDP-8/I system includes the central processor, a 4096-word core memory, input/output transfer facilities, an operator's console, and a Teletype.

The 4096-word core memory is standard in the basic system. Central processor prewiring permits another 4096-word core memory to be plugged into the

standard cabinet. Six additional 4096-word core memory blocks can expand the PDP-8/I core memory capacity up to 32,768 words. Other low-cost storage devices such as DECdisk, DECtape, and IBM compatible magnetic tape can be easily and economically added. Peripheral and interface devices are connected to the PDP-8/I through a versatile input/output bus. A built-in instruction skipping capability provides a convenient method for checking the status of peripherals. Data is transferred between the PDP-8/I and bus-connected devices in three ways: 1) through program interrupts, 2) through programmed data transfers, and 3) through data break transfers. The user can choose the most efficient method for transferring data depending on his peripheral and interface servicing requirements.

THE PDP-8/L PROGRAMMED DATA PROCESSOR

The PDP-8/L is the lowest cost full-scale computer system available. The basic PDP-8/L system includes the same fully-parallel processor and 4096-word core memory as the PDP-8/I, along with I/O transfer facilities, an operator's console, and a Teletype.

Like the PDP-8/I, the PDP-8/L is a single-address, fixed-word-length system using 2's complement arithmetic. Operation and performance of the 8/L is essentially the same as that of the 8/I. Like the PDP-8/I, the PDP-8/L is constructed from TTL integrated-circuit modules for economy, ease of maintenance, and compactness (the PDP-8/L measures 8¾ inches high, 19 inches wide, and 20¾ inches deep). The basic difference between the PDP-8/I and the PDP-8/L is that the PDP-8/L does not have the same internal prewiring that otherwise permits peripherals to plug directly into the 8/I.

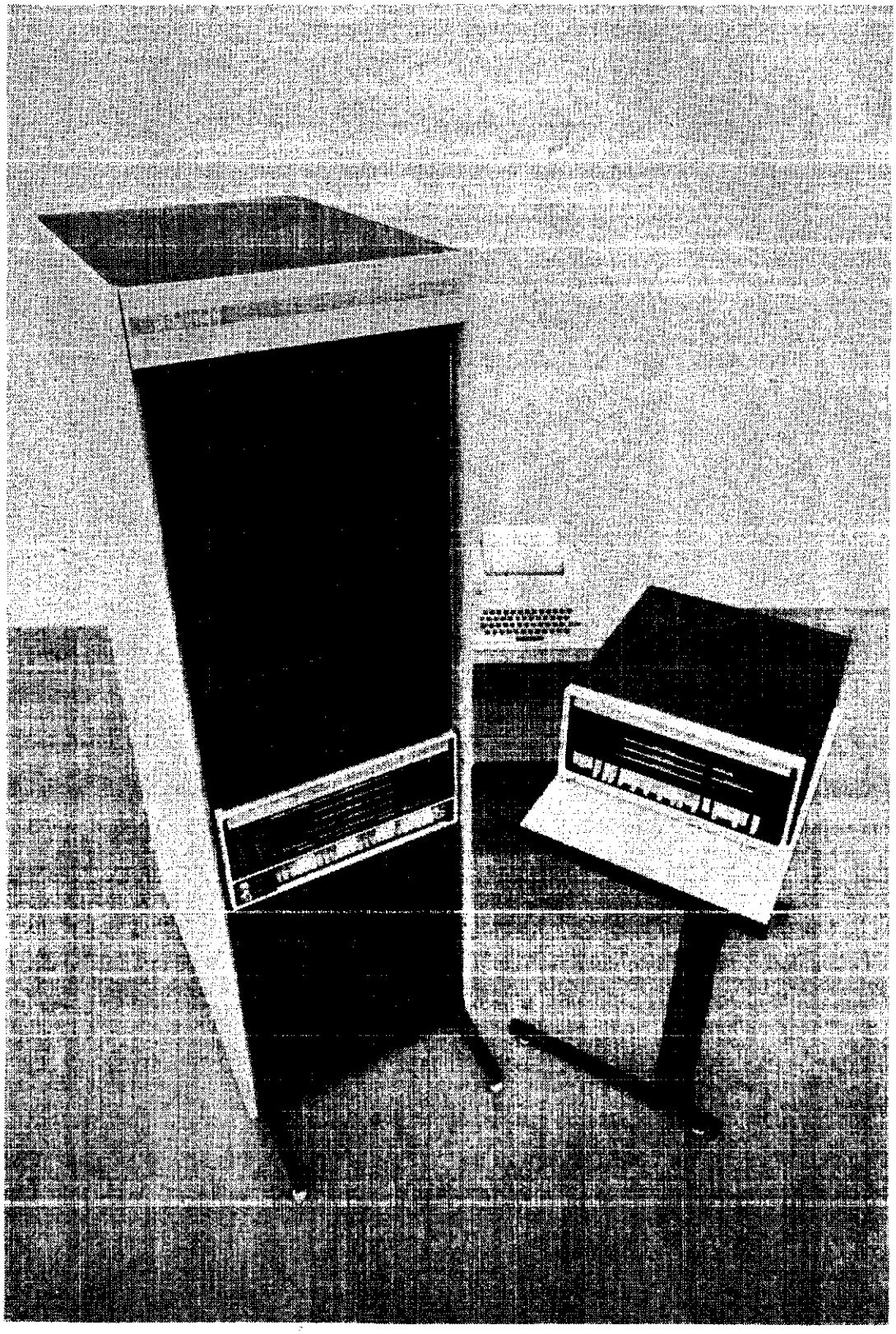
The PDP-8/L has a positive input/output bus that is functionally similar to that of the PDP-8/I. The PDP-8/L is slightly slower than the PDP-8/I: its memory-cycle time is 1.6 microseconds as compared to 1.5 microseconds for the PDP-8/I. The PDP-8/L cannot accommodate an extended arithmetic option: and core memory can only be expanded up to a capacity of 8192 words.

A memory protection feature of the PDP-8/L permits 128 12-bit core memory registers to be protected from memory modifying instructions, operations, or from break cycles. This part of memory can be used for storage of leaders, program constants, etc. If an illegal reference is attempted, the computer halts at the end of the illegal instruction without modifying memory. Memory protection is enabled by a switch on the operator's console.

CONTENTS

APPENDICES

TIMESHARED-8 is a new concept in small computers. It is a multi-language, local time-sharing system including a PDP-8 Family computer and from 8 to 32 terminals.



PDP-8/I and PDP-8/L are the latest members of the PDP-8 Family of general-purpose computers. They are the faster, smaller and more economical successors to the over 4,000 PDP-8 Family computers installed all over the world.

CHAPTER 1

SYSTEM INTRODUCTION

This handbook describes Digital Equipment Corporation's Programmed Data Processors -8/I and -8/L (PDP-8/I and PDP-8/L). All times and values discussed in this handbook are relative to both systems, unless otherwise noted. Where a difference exists, the specifications applicable to the PDP-8/L are given in brackets [ ].

The PDP-8/I and PDP-8/L are small scale, general purpose computers. TTL monolithic integrated circuit modules are used throughout, and provide efficient packaging, high reliability, and noise immunity. Both of these computers are single address, fixed word length, parallel-transfer computers using 12 bit, 2's complement arithmetic. Cycle time of the 4096-word random-address magnetic-core memory is 1.5 μs [1.6 μs]. Standard features of both computers include indirect addressing and facilities for instruction skip and program interrupt as a function of the input/output device conditon.

The 1.5 [1.6] μs cycle time of the machine provides a computation rate of 333,333 [312,500] additions per second. Each addition requires 3.0 [3.2] μs (with one number in the accumulator) and subtraction requires 6.0 [6.4] μs (with the subtrahend in the accumulator). Multiplication is performed in approximately 315 [336] μs by 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 is performed in approximately 444 [474] μs by a subroutine that produces a 12-bit quotient in the accumulator and a 12-bit remainder in core memory. Similar multiplication and division operations are performed in approximately 6.0 and 6.5 μs, respectively, utilizing the Extended Arithmetic Element which is optional only to the PDP-8/I. The flexible, high-capacity, input/output capabilities of these computers allow them to operate a large variety of peripheral equipment. Besides the standard keyboard and paper-tape punch and reader equipment, these computers are capable of operating in conjunction with a number of optional devices such as high-speed perforated-tape readers and punches, card reader equipment, line printers, analog-to-digital converters, cathode ray tube (CRT) displays, magnetic drum systems, magnetic tape equipment, a 32,000-word random-access disk file and a 225,000-word random-access disk file. Specially designed equipment is easily interfaced with either of these systems. The computers do not have to be modified when peripheral devices are added.

The PDP-8/I and PDP-8/L systems are completely self-contained, and require no special power sources or environmental conditions. A single source of 115V, 47-63 Hz, single-phase power is required for the PDP-8/L and either 115V or 220V, 50 or 60 Hz, single-phase power is required for the PDP-8/I. Internal power supplies produce the necessary operating voltages for both systems. Modules utilizing TTL monolithic integrated circuits assure reliable operation in ambient temperatures between 50 and 130 degrees Fahrenheit.

COMPUTER ORGANIZATION

The PDP-8/I and PDP-8/L systems are organized into a central processor, core memory, and input/output equipment and facilities. All arithmetic, logic, and system control operations are performed by the central processor. Permanent

(longer than one instruction time) local information storage and retrieval operations are performed by the core memory. The memory is continuously cycling, automatically performing a read and write operation during each computer cycle. Input and output address and data buffering of the core memory are performed by registers in the central processor, and the operation of the core memory is under control of timing signals produced by the central processor. Because of the close relationship of operations performed by the central processor and the core memory, both are described in this chapter.

Central processor interface circuits provide bussed connections to a variety of peripheral input/output equipment. Each input/output device is responsible for detecting its own select code and for providing all required input or output gating. Individually programmed data transfers, between the central processor and peripheral equipment, take place through the central processor accumulator. Data break transfers can be initiated by peripheral equipment, rather than under program control through the data break facilities (standard on the PDP-8/I; optional on the PDP-8/L). Standard features of both computers allow peripheral equipment to perform certain control functions, i.e., instruction skipping and a transfer of program control initiated by a program interrupt.

Standard peripheral equipment provided with each of these systems consists of a Teletype Model 33 Automatic Send-Receive (ASR) set and a teletype control logic unit. The ASR set is a standard machine operating from serial 11-unit-code characters at a rate of ten characters per second. The ASR provides a means of supplying data to the computer from keyboard-controlled perforated tape, and supplies data as an output from the computer in the form of perforated tape and/or typed copy. The teletype control serves as a serial-to-parallel converter for teletype inputs to the computer and serves as a parallel-to-serial converter for computer output signals to the Teletype unit. The Teletype and all optional input/output equipment are discussed in Chapter 7 of this handbook.

SYMBOLS

The following special symbols are used throughout this handbook to explain the function of equipment and instructions:

Symbol	Explanation
$A \rightarrow B$	The contents of register A are transferred into register B
$0 \rightarrow A$	Register A is cleared to contain all binary 0s
A_i	Any given bit in register A
A_5	The content of bit 5 of register A
$A_5(1)$	Bit 5 of register A contains a 1
$A_{6\text{-}11}$	The contents of bits 6 through 11 of register A
$A_{6\text{-}11} \rightarrow B_{0\text{-}5}$	The contents of bits 6 through 11 of register A are transferred into bits 0 through 5 of register B
Y	The contents of any core memory location
V	Inclusive OR
$\forbar$	Exclusive OR
$\wedge$	AND
$\overline{A}$	One's complement of the content of register A

MAJOR REGISTERS AND MEMORY

In order to store, retrieve, control, and modify information and to perform the required logical, arithmetic, and data processing operations, the core memory and the central processor employ the logic complement shown in Figure 1-1 and described in the following paragraphs.

Accumulator (AC)

The AC is a 12-bit register in which arithmetic and logic operations are performed. Under program control the AC can be cleared, complemented, or its contents can be rotated right or left. The contents of the memory buffer register can be added to the contents of the AC and the result stored in the AC. The contents of both of these registers can be combined by the logical AND operation with the result remaining in the AC. The inclusive OR may be performed between the AC and the switch register (on the operator's console) and the result left in the AC. The AC also serves as an input/output register; all programmed information transfers between the core memory and an external I/O device are passed through the AC.

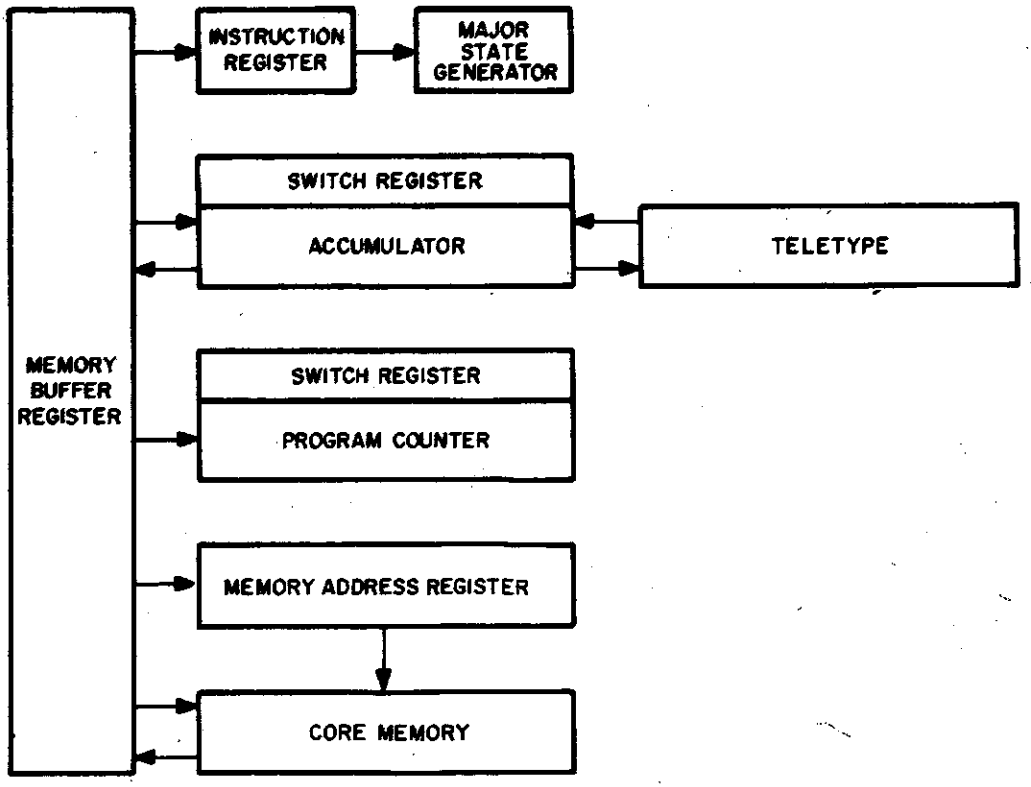


Figure 1-1. Simplified Block Diagram

Link (L)

The Link is a 1-bit register which is used to extend the arithmetic facilities of the AC. It is used as the carry register for 2's complement arithmetic. Overflow into the L from the AC can be checked by the program to greatly simplify and speed-up single and multiple-precision arithmetic routines. Under program control the L may be either cleared, complemented, or rotated as part of the AC.

Program Counter (PC)

The PC is a 12-bit register which is used to control the program sequence; that is, the order in which instructions are performed is determined by the PC. The PC contains the address of the core memory location from which the next instruction is taken. Information enters the PC from the core memory, via the memory register, and from the switch register on the operator's console. Information in the PC is transferred into the memory address register to determine the core memory address from which each instruction is taken. Incrementing the contents of the PC establishes the successive program core memory locations and provides skipping of an instruction based upon a programmed test of information or conditions.

Memory Address (MA) Register

The MA register is a 12-bit register that contains the address in core memory which is currently selected for reading or writing. Therefore, all 4096 words of core memory can be addressed directly by the MA. Data can be transferred into the MA from the memory buffer register, from the program counter, or from an I/O device through the data break facilities.

Switch Register (SR)

The SR is a 12-bit register made up of manually-operated switches that can be used to load information for transfer into the program counter as an address, by means of the load address switch, or into the memory buffer register as data to be stored in core memory, by means of the deposit switch. Both switches are on the operator's console. The contents of the SR also may be transferred to the AC under program control.

Memory Buffer (MB) Register

The MB register is a 12-bit register that is used for all information transfers between the central processor registers and the core memory. This information can be transferred and temporarily held in the MB from the AC or PC. The MB is simultaneously read and incremented by one before being read back into memory. Information can be loaded into the MB from an external device during a data break or from core memory via the sense amplifiers. Information is read from a memory location in 0.80 μs (for both systems) and rewritten in the same location in another 0.70 [0.80] μs of a single 1.5 [1.6] μs duration memory cycle.

Instruction Register (IR)

The IR is a 3-bit register that contains the operation code of the instruction currently being performed by the machine. The three most-significant bits of the current instruction are loaded into the IR from the MB during a fetch cycle. The contents of the IR are decoded to produce the eight basic instructions, and affect the cycles and states entered during each step of the program.

Core Memory

The core memory provides random-access storage for both instructions to be performed and information to be processed or distributed. This magnetic core

memory holds 4096 12-bit words in the standard PDP-8/I and PDP-8/L. Optional equipment extends the storage capacity in fields of 4096 words [the PDP-8/L is limited to one such extension] or expands the word length to 13 bits to provide parity checking. Memory location 0_8 is used to store the contents of the PC following a program interrupt, and location 1_8 is used to store the first instruction to be executed following a program interrupt. (When a program interrupt occurs, the contents of the PC are stored in location 0_8, and program control is transferred to location 1_8 automatically. Core memory locations 10_8 through 17_8 are used for auto-indexing. All other locations can be used for the storage of either instructions or data.

The core memory contains numerous circuits such as read-write switches, address decoders, inhibit drivers, and sense amplifiers. These circuits perform the electrical conversions necessary to transfer information to or from the core array; they perform no arithmetic or logic operations upon the data. Since their operation is not discernible by the programmer or operator of the PDP-8/I or PDP-8/L, these circuits are not described.

MAJOR PROCESSOR STATES

Both the PDP-8/I and the PDP-8/L utilize six processor states to either execute programmed instructions or to effect a data break. To accomplish this, a major state generator is used to establish one state for each computer timing cycle. The major processor states are: Fetch, Defer, Execute, Word Count, Current Address, and Break. Fetch, Defer and Execute states determine and execute instructions; Word Count, Current Address, and Break states are used during a data break and are based on request signals received from peripheral I/O equipment.

Fetch (F) State

Assuming the computer is in a fully automatic, running condition; that is, the computer is continuously functioning to fetch and/or execute instructions, the PC register's contents are transferred to the MA register. When this is accomplished, the MA is simultaneously incremented by one and the result is transferred to the PC register, initiating the Fetch state. At this time, TS1, the memory is also strobed. At a time TS2, the contents of the memory appear at the output of the memory (MEM) register (synonymous with the output from the sense amplifiers) and are transferred to the MB register. Simultaneously, the contents of the first three bits of the MEM register are also transferred to the IR. This completes the activity during time TS2 of the Fetch state. During times TS3 and TS4, the Fetch instruction is acted upon (executed if instruction was single-cycle, identified or deferred if instruction was two- or three-cycle). During time TS4, the contents of the PC register are transferred to the MA register (incremented if a skip condition exists), and the cycle is ready to repeat.

Defer (D) State

The Defer state is entered if a binary 1 is present in bit 03 of a memory reference instruction. This state causes the central processor to obtain the full 12-bit address of the operand from the address in either the current page or page zero, as specified by bits 4 through 11 of the instruction. This process of address deferring is called indirect addressing because access to the operand is addressed indirectly, or deferred, to another memory location.

Execute (E) State

The Execute state is entered for all memory reference instructions except jump. During an AND, two's complement add, or increment and skip if zero instruction, the contents of the core memory location specified by the address portion of the instruction are read into the MB and the operation specified by the instruction register is performed. During a deposit and clear accumulator instruction, the contents of the AC are transferred into the MB and stored in core memory at the address specified in the instruction. During a jump to a subroutine instruction, the Execute state occurs to write the contents of the PC into the core memory location as designated by the instruction and to transfer this address +1 into the PC to bring about a change in program control.

Word Count (WC) State

The WC state is entered whenever an external I/O device supplies signals requesting a data break and specifying that it should be a three-cycle data break. When this state occurs, a core memory location designated by the I/O device and simultaneously incremented by one, and read into the MB, is rewritten back into the same location. If word count overflow occurs, the central processor transmits a signal to the I/O device indicating that the desired number of data break transfers will have been enacted at the completion of the current break cycle. The Current Address state is then entered.

Current Address (CA) State

The CA state is the second state of a three-cycle data break. During this cycle the address for the data to be transferred in the next or third cycle is established. Normally the location following the word count is read from core memory and simultaneously incremented by one and transferred into the MB, thus establishing sequential addresses for the transfers; it is then transferred into the MA to determine the address selected for the next cycle (Break state). In the event the address incrementing operation is not used, the device supplies an inhibit signal to the computer so that the word read during this cycle is not incremented. Transfers occur at sequential addresses due to incrementing during the CA state. The Break state follows the CA state.

Break (B) State

The Break state is either the third cycle of a three-cycle data break or the only cycle of a single-cycle data break. This state is entered to enact a data transfer between computer core memory and an external I/O device. When a break request signal arrives and the cycle select signals specifies a single-cycle data 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 MB. When this transfer is complete, the program sequence resumes from the point of the break. The data break does not affect the contents of the AC, L, or PC.

TIMING AND CONTROL ELEMENTS

The Timing and Control Elements are determined by circuits shown in Figure 1-2.

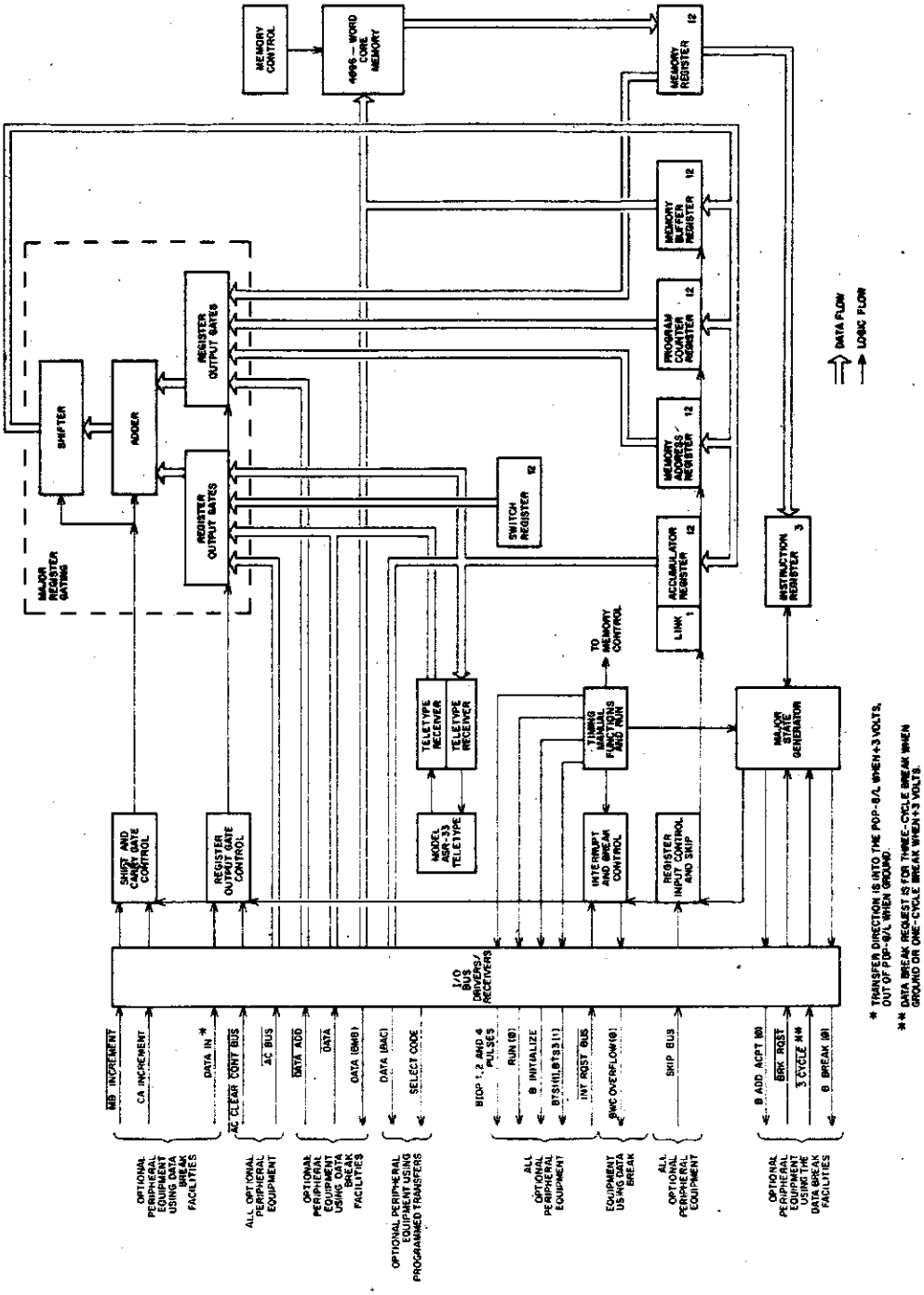


Figure 1-2. Timing and Control Element Block Diagram

This figure shows the timing and control elements described in the succeeding paragraphs and indicates their relationship to the major registers. These elements can be grouped categorically into timing generators, register controls, and program controls.

Timing Generators

The timing generators provide the timing pulses which determine the computer cycle time and which are used to initiate sequential time-synchronized gating operations. Timing pulses used during operations resulting from the use of the keys and switches on the operator's console are produced by the special pulse generator for manual control. Pulses that reset registers and control circuits during power turn-on and turn-off operations are produced by the power clear pulse generator. Several of these pulses are available for peripheral device control to be utilized with devices using programmed or data break information transfers.

Register Controls

The Register Controls are control gates which gate the contents stored within the AC, MA, MB, and/or the PC registers onto the common register bus. In both the PDP-8/I and PDP-8/L the gated input bus of each register is tied to the common register bus through the register control gates. The output of the common register bus is the major register gating circuit. The data on the common register bus originates from the various outputs of each register and can be gated directly to another register or modified by the adders within the major register gating circuit. When the contents of one register are to be transferred to another register, the contents are gated by the register output gate control onto the common register bus and are then strobed into the appropriate register. Data can thereby be transferred directly between registers, or the data can be modified during transfer to provide SHIFT, CARRY, or SKIP operations. Operations such as incrementing a register are accomplished by gating the output of the register onto the register bus, gating the carry insert into the adder, and strobing the results back into the same or some other register.

Program Control Circuits

The Program Control Circuits produce IOP pulses that initiate operations which are involved in input/output transfers, determine the advance of the computer program, and allow peripheral equipment to cause a program interrupt of the main computer program. This program interrupt transfers program control to a subroutine to perform a service for the I/O device.

INTERFACE

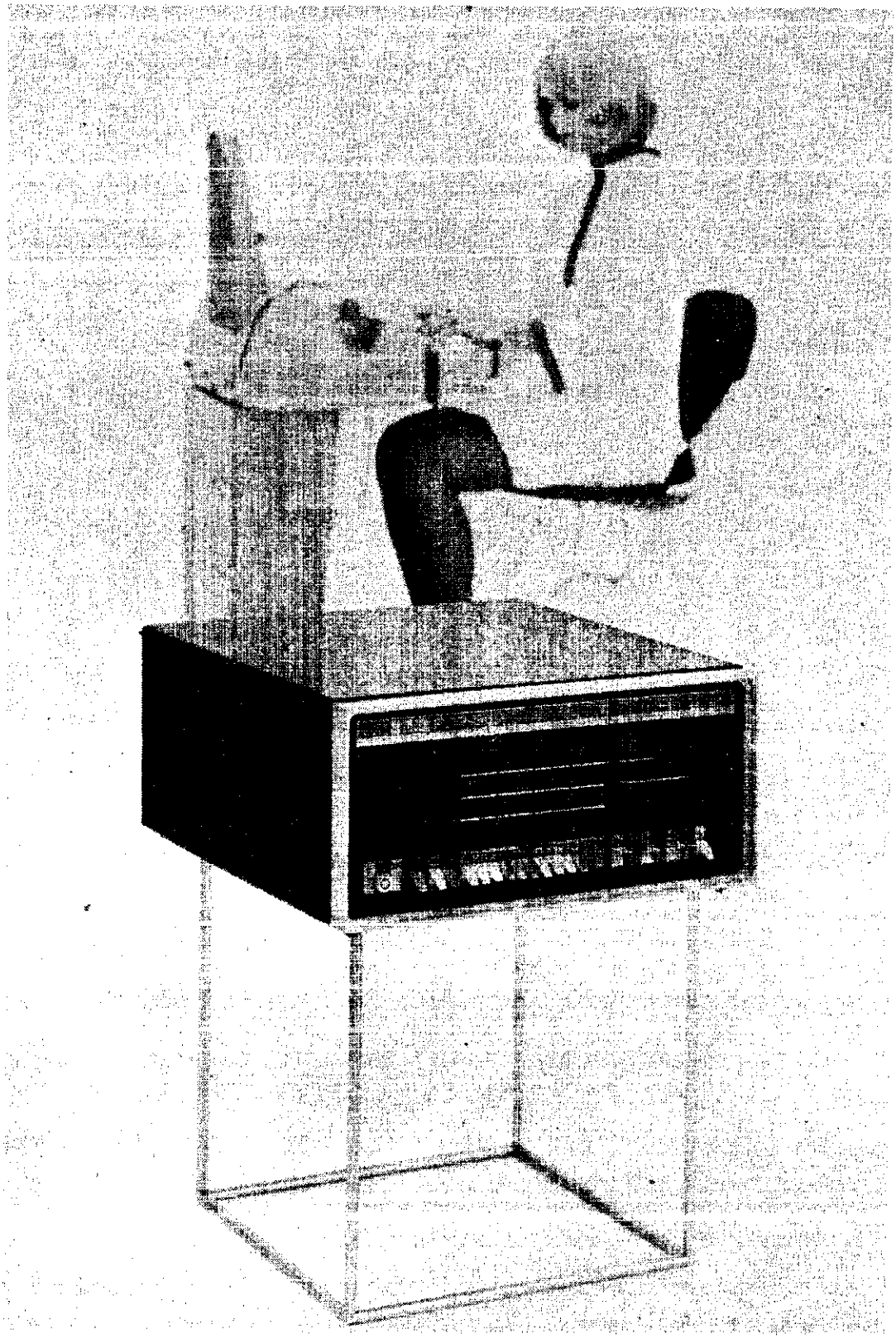
The input/output portion of both systems is extremely flexible and interfaces easily with special equipment, particularly in real-time data processing and control environments. Both PDP-8/I and PDP-8/L computers utilize positive logic within the central processor. The PDP-8/I has two bus systems which provide either a positive [KA8/I option] or a negative input/output bus system; whereas the PDP-8/L has a positive I/O bus system only. However, either system, through the use of an external option (the DW08 A or B), can have both a positive (+3V) and a negative (−3V) bus. Therefore, these systems are compatible with existing peripheral equipment offered by Digital Equipment Corporation and other manufacturers.

These computers utilize a "bus" I/O system rather than the more conventional "radial" system. The "bus" system allows a single set of data and control lines to communicate with all I/O devices by going from one device to the next. No additional connections to the computer are required. Conversely, a "radial" system requires that a different set of signals must be transmitted to each and every device; and thus the computer has to be modified for each new device added. It is not necessary to modify either the PDP-8/I or the PDP-8/L

when adding new peripheral equipments. High-speed, direct-data transfers may also be made utilizing core memory through the use of the data break facility, which is standard equipment in the PDP-8/I and optional for the PDP-8/L. It is ordinarily used with fast I/O devices such as magnetic disks or tapes. Transfers through the data break facility are interlaced with the program in progress and are initiated by a request from the peripheral device and not by programmed instruction. Therefore, the device may transfer a data word with memory whenever it is ready and does not have to wait for the program to issue an instruction. Computation may proceed on an interlaced basis with these transfers. Interface signal characteristics are discussed in Chapter 12.

FUNCTIONAL SUMMARY

The operation of these systems is accomplished on a limited scale by keys and switches on the operator's console. This manner of operation is limited to address selection and data storage by means of the switch register, core memory data examination, the normal start/stop/continue control, and the single step (both systems) or single instruction (PDP-8/I only) operation manually allowing a program to be monitored visually during maintenance or program debugging. Most manually-initiated operations are performed by executing an instruction by special pulses rather than by the normal timing pulses. During automatic operation, instructions stored in the core memory are loaded into the memory buffer register and executed during one or more computer cycles. Each instruction determines the major control states that must be entered for its execution. Each control state lasts for a single 1.5 [1.6] μs computer cycle and is divided into distinct time states which can be used to perform sequential logical operations. Performance of any function of the computer is controlled by the gating of a specific instruction during a specific major control state and a specific time state.



PDP-8/L is the lowest cost full scale digital computer available. It comes in a very neat, small package with a very neat, small price. Just right for plugging into anyone's integrated system.

CHAPTER 2

STANDARD SYSTEM OPERATION

CONTROLS AND INDICATORS

The controls and indicators on the operator's console provide manual control and indicate the program conditions of the PDP-8/I or PDP-8/L. Controls on the operator's console provide the operator with the hardware to start, stop, modify, or continue a program. The indicators on the console provide a visual indication of the machine status and current program, the contents of the major registers, and the condition of the control flip-flops. A lighted indicator denotes the presence of a binary 1 in a specific register bit position or control flip-flop. Table 2-1 lists the functions of controls and indicators for both the PDP-8/I and PDP-8/L, noting the differences in the consoles. Figures 2-1 and 2-2 illustrate the consoles. Controls and indicators of the standard Model 33 ASR Teletype unit are shown in Figure 2-3 and their functions are described in Table 2-2.

NOTE

All notations in brackets [ ] indicate data for the PDP-8/L computer only.

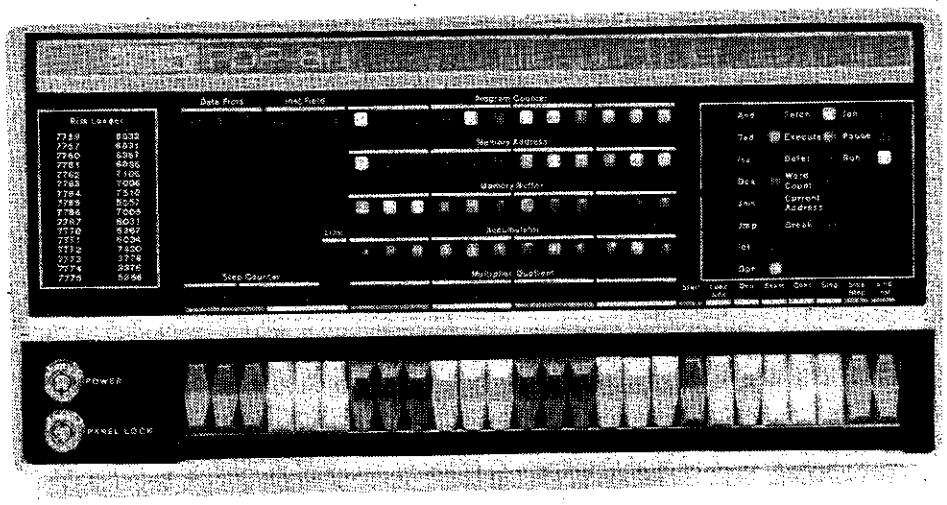


Figure 2-1. PDP-8/I Front Panel

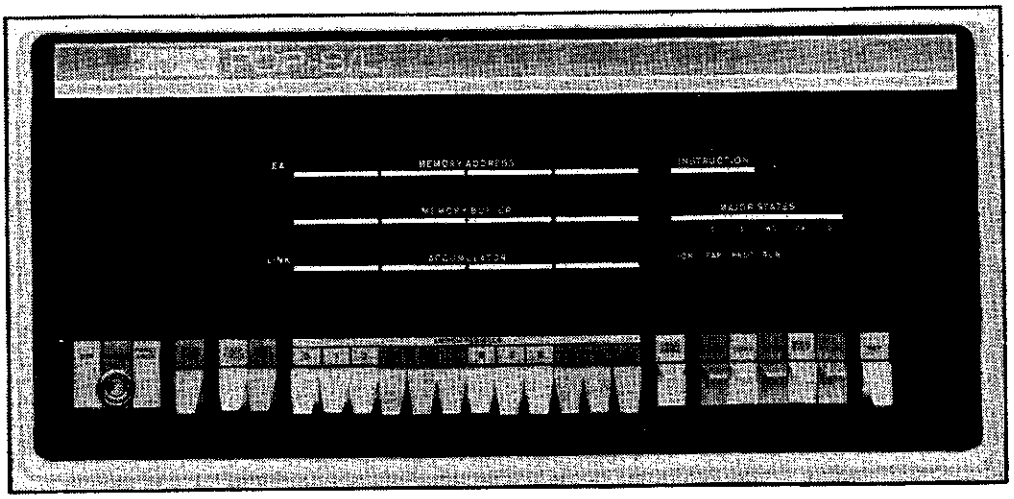


Figure 2-2. PDP-8/L Front Panel

Table 2-1. Operator's Console Control and Indicator Functions

Control or Indicator	Nomenclature PDP-8/I	Nomenclature PDP-8/L	Function
Panel lock keyswitch	Panel Lock	——	With this key-operated switch turned clockwise, all keys and switches, except the switches on the operator console, are disabled. In this condition the program cannot be disturbed by inadvertent key operation. The program can, however, monitor the contents of the SR by execution of the OSR or LAS instruction. With this switch turned counterclockwise, all operator console keys and switches function normally.
Power key switch	Power	——	In the counterclockwise position this key-operated switch removes primary power from the computer, and in the clockwise position it applies power.
Off/Power/Panel lock keyswitch	——	OFF/POWER/ PANEL LOCK	In the OFF position, this key-operated switch removes primary power from the computer. In the POWER position it applies power and enables all manual controls. In the PANEL LOCK position it maintains power, but all keys and switches, except the SWITCH REGISTER, INST FIELD, and DATA FIELD switches on the operator console, are disabled. In this condition the program cannot be disturbed by inadvertent switch operation. The program can, however, monitor the contents of the SR by execution of the OSR or LAS instruction.

Table 2-1. Operator's Console Control and Indicator Functions (Cont)

Control or Indicator	Nomenclature PDP-8/I	Nomenclature PDP-8/L	Function
Start switch	Start	START	Starts the computer program by turning off the program interrupt circuits, clearing the AC, L, MB, and IR, setting the Fetch state, transferring the contents of the PC into the MA; and setting the RUN flip-flop. Therefore, the word stored at the address currently held by the PC is taken as the first instruction.
Load Address Switch	Load Add	LOAD ADDR	Loads the contents of the SR into the PC [PC and MA], loads the contents of the instruction field switches into the IF, and loads the contents of the data field switches into the DF. [In the PDP-8/L, a memory cycle is also executed, displaying the contents of the addressed location in the MB].
Deposit Switch	Dep	DEP	Loads the contents of the SR into the MB and core memory at the address specified by the current contents of the PC. The contents of the PC are then incremented by one, to allow storing of information in sequential memory addresses by repeated operation of the deposit switch.
Examine Switch	Exam	EXAM	Sets the contents of core memory at the address specified by the contents of the PC into the MB. The contents of the PC are then incremented by one to allow examination of the contents of sequential core memory addresses by repeated operation of the examine switch.

Table 2-1. Operator's Console Control and Indicator Functions (Cont)

Control or Indicator	Nomenclature PDP-8/I	PDP-8/L	Function
Continue Switch	Cont	CONT	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. [The PDP-8/L may be started without generating a power clear by depressing LOAD ADDR, then CONT.]
Stop Switch	Stop	STOP	Causes the RUN flip-flop to be cleared at the end of the instruction in progress at the time the switch is depressed.
Single Step Switch	Sing Step	SING STEP	The switch is on in the up [down] position. In this position the switch causes the RUN flip-flop to be cleared to disable the timing circuits at the end of one cycle of operation. Thereafter, repeated depressing of the continue switch steps the program one cycle at a time so that the contents of registers can be observed in each state.
Single Instruction Switch	Sing Inst	(None)	The switch is off in the down position. In the up position the switch causes the RUN flip-flop to be cleared at the end of the next instruction executed. When the computer is started by means of the start or continue switch, this switch causes the RUN flip-flop to be cleared at the end of the last cycle of the current instruction. Therefore, repeated depressing of the continue switch steps the

Table 2-1. Operator's Console Control and Indicator Functions (Cont)

Control or Indicator	Nomenclature PDP-8/I	Nomenclature PDP-8/L	Function
			program one instruction at a time. [The single instruction function may be obtained on the PDP-8/L by physically holding down the STOP switch].
Switch register switches	(not named)	SWITCH REGISTER	Provide a means of manually setting a 12-bit word into the machine. Switches in the up position correspond to binary 1s, the down position to 0s. The contents of this register are loaded into the PC [PC and MA] by the load address switch or into the MB and core memory by the deposit switch. The contents of the SR can be loaded into the AC under program control by means of the OSR or LAS instruction.
Data field switches and indicators	Data Field	DATA FIELD (switches only)	The indicators denote the contents of the data field register (DF) (PDP-8/I only) and the switches serve as an extension of the SR to load the DF by means of the load address switch. The DF determines the core memory field of data storage and retrieval.
Instruction field switches and indicators	Inst Field	INST FIELD (switches only)	The indicators (PDP-8/I only) denote the contents of the instruction field register (IF) and the switches serve as an extension of the SR to load the IF by means of the load address switch. The IF determines the core memory field from which instructions are to be taken.

Table 2-1. Operator's Console Control and Indicator Functions (Cont)

Control or Indicator	Nomenclature PDP-8/I	Nomenclature PDP-8/L	Function
Extended address indicator light	(none)	EA	This indicates the content of the extended address field being addressed. If the current memory address is for data, it is the content of the DF flip-flop. If the current memory address is an instruction, it is the content of the IF flip-flop.
Program counter indicator lights (12)	Program Counter	(None)	Indicate the contents of the PC. When the machine is stopped the contents of the PC indicate the core memory address of the first instruction to be executed when the start or continue switch is operated. When the machine is running the contents of the PC indicate the core memory address of the next instruction.
Memory address indicator lights (12)	Memory Address	MEMORY ADDRESS	Indicate the contents of the MA, which usually denotes the core memory address of the word currently or previously read or written. After operation of either the deposit or examine switch, the contents of the MA indicate the core memory address at which information was just written or read. [Operation of the LOAD ADDR switch on the PDP-8/L causes the MA to display the contents of the SR].
Memory protector switch and associated indicator	(None)	MEM PROT (switch) PROT (indicator light)	The switch is off in the down position. In the up position the switch causes the top page, 200_8 [128_{10}] locations, of the upper memory

Table 2-1. Operator's Console Control and Indicator Functions (Cont)

Control or Indicator	Nomenclature PDP-8/I	Nomenclature PDP-8/L	Function
			field to be protected from any memory-modifying instruction. When protected, the top page may be referenced by instructions, but any change to the contents is inhibited and the machine halts with the PROT indicator on. Manual restart clears the memory protection circuits. This switch is disabled by the panel lock key switch.
Memory buffer indicator lights (12)	Memory Buffer	MEMORY BUFFER	Indicate the contents of the MB, which usually designates the word just read or written at the core memory address held in the MA.
Accumulator indicator lights (12)	Accumulator	ACCUMULATOR	Indicate the contents of the AC.
Link indicator light	Link	LINK	Indicates the contents of the L.
Multiplier quotient indicator lights* (12)	Multiplier Quotient	(None)	Indicate the contents of the multiplier quotient (MQ). The 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 a division and at the end holds the quotient.
Step counter indicator lights* (5)	Step Counter	(None)	Indicate the contents of the step counter (SC). The step counter is loaded with the complement of the contents of

*Operational only on PDP-8/I systems containing the KE8/I Extended Arithmetic Element option

Table 2-1. Operator's Console Control and Indicator Functions (Cont)

Control or Indicator	Nomenclature PDP-8/I	Nomenclature PDP-8/L	Function
			bits 7-11 of the memory location during an ASR, LSR, SCL or SHL instruction; contains the exponent after an NMI instruction; and is set to contain all zeros for the DVI or MUV instructions.
Instruction indicator lights	Instruction (indicators (8): And, Tad, Isz, Dca, Jms; Jmp, Iot, Opr)	INSTRUCTION (indicator lights (3)	On the PDP-8/I, indicator lights indicate the decoded output of the IR of the instruction currently in progress. On the PDP-8/L, indicator lights indicate the content of the 3-bit IR and, therefore, the basic instruction being executed. A binary number (0 through 7) on the PDP-8/L corresponds to the eight instructions listed on the PDP-8/I panel.
Processor states indicator lights (6)	Fetch, Execute, Defer, Word Count, Current Address, Break	MAJOR STATES F, E, D, WC, CA, B	Indicate the primary control states of the machine and that the current memory cycle is either fetch, defer, execute, or break cycle, respectively. Word count and current address indicate the first and second cycles of a three-cycle break.
Program interrupt on indicator light	Ion	ION	Indicates the 1 status of the INT ENABLE flip-flop. When lit, the program in progress can be interrupted by receipt of a program interrupt request signal from an I/O device.
Run indicator light	Run	RUN	When lit, the internal timing circuits are enabled and the machine performs instructions.

Table 2-1. Operator's Console Control and Indicator Functions (Cont)

Control or Indicator	Nomenclature PDP-8/I	Nomenclature PDP-8/L	Function
Pause indicator light	Pause	(None)	Indicates the 1 status of the PAUSE flip-flop when lit. An IOT instruction sets the PAUSE flip-flop at time TP3 to initiate operation of the IOP generator and to inhibit advance of the normal timing generator. When IOP generator operation is completed (approximately 3.8 μs later), a TP4 pulse is generated and the PAUSE flip-flop is cleared to enable advance of the timing generator.
Parity indicator light*	(None)	PAR	Indicates the 1 status of the PARITY ERROR flip-flop. When lit a parity error has been detected.

*Operational only on PDP-8/L systems containing the MP8/L Memory Parity Option

Table 2-2. 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 FREE position the reader is disengaged, permitting the paper tape to be manually moved within the reader without necessarily reloading or unloading it. In the STOP position the reader mechanism is engaged but de-energized. In the START position the reader is engaged and operated under program control. The tape may be either loaded or unloaded in either the FREE or STOP positions.

Figure 2-3. Teletype Model ASR33 Console

Keyboard	Provides a means of printing on paper when used as a typewriter and punching tape when the punch ON pushbutton is pressed, and provides a means of supplying 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 to the Teletype and control data connection to the processor. In the LINE position the Teletype is energized and connected as a computer I/O device. 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 disconnected. Both line and local use of the Teletype require that the computer be energized through the POWER switch.

OPERATING PROCEDURES

Many means are available for loading and unloading PDP-8/I and PDP-8/L information. The means used are dependent upon the form of the information, time limitations, and the peripheral equipment connected to the system. The following procedures are basic to any use of these systems and, although they may be 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.

Manual Data Storage and Modification

Programs and data can be stored or modified manually by means of the facilities on the operator's console. Chief use of manual data storage is made to load the readin mode (RIM) loader program into the computer core memory. The RIM loader is a program used to automatically load programs into the computer from perforated tape in RIM format. This program and the RIM tape format are described in Appendix E and in Digital Program Library descriptions. The RIM program listed in the Appendix can be used as an exercise in manual data storage. Use the following procedure to store data manually in the computer core memory.

a. On the PDP-8/I turn the Panel Lock switch counterclockwise and turn the Power switch clockwise. On the PDP-8/L turn the OFF/POWER/PANEL LOCK switch clockwise to the POWER position.

b. On the PDP-8/L set the MEM PROT switch down. Set the SWITCH REGISTER switches to correspond with the address bits of the first word to be stored. Press the load address key; on the PDP-8/I the address is shown in the PC, whereas on the PDP-8/L the address is shown in the MA. Observe that the address set by the switch register is held in the computer as designated by lighted Program Counter [MEMORY ADDRESS] indicators corresponding to switches in the 1 (up) position and unlighted indicators corresponding to switches in the 0 (down) position.

c. Set the switch register switches to correspond to the data or instruction word to be stored at the address just set into the PC [MA]. Operate the deposit key and check that the MB, and therefore the core memory, contain the data set in the switch register.

Also, check to see that the PC [MA after the first operation] has been incremented by one so that additional data can be stored at sequential addresses by repeated switch register setting and deposit key operation.

To check the contents of any address in core memory, set that address into the PC [PC and MA] as in step b, then operate the examine key. The contents of the address are then shown by the MB register indicators. The contents of the PC are incremented by one with the operation of the examine key, so the contents of sequential addresses may be examined by repeated operation after the original (or starting) address is loaded. The contents of any address may be modified by repeating steps b and c.

Loading Data Under Program Control

Information can be stored or modified automatically in the computer only by using programs previously stored in core memory. For example, having the RIM loader stored in core memory allows RIM format tapes to be loaded as follows:

a. On the PDP-8/I turn the Panel Lock switch counterclockwise and turn the Power switch clockwise; on the PDP-8/L turn the OFF/POWER/PANEL LOCK switch to the POWER position.

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

c. 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, moving the tape either for-

ward or backward until the punched leader section is over the read station, and setting the switch to the STOP position. Tape is loaded 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 results in three bit positions being sensed to the left of the sprocket wheel and five bit positions bing sensed to the right of the sprocket wheel.

d. Load the starting address of the RIM loader program (not the address of the program to be loaded) into the PC by means of the SR and the load address switches.

e. Press the computer start key and set the 3-position Teletype reader switch to the START position. The tape is then read automatically.

Automatic storing of the binary loader (BIN) program is performed by means of the RIM loader program as previously described. With the BIN loader stored in core memory, program tapes assembled in the 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 (usually 7777) is used in step d. When storing a program in this manner, the computer stops and the AC should contain all zeros if the program is stored properly. If the computer stops with a number other than zero in the AC, a checksum error has been detected. When the program has been stored, it can be initiated by loading the program starting address (usually designated on the leader of the tape) into the PC by means of the switch register and load address switches, then pressing the start key.

Off-Line Teletype Operation

The Teletype can be used separately from the computer for typing, punching tape, or duplicating tapes. To use the Teletype in this manner:

a. On the PDP-8/I assure that the computer Panel Lock switch is turned counterclockwise and turn the Power switch clockwise ;on the PDP-8/L the OFF/POWER/PANEL LOCK switch is positioned to POWER.

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

c. If the punch is to be used, load it by raising the cover, manually feeding the tape from the top of the roll into the guide at the back of the punch, advancing the tape through the punch by manually turning the friction wheel, and then closing the cover. Energize the punch by pressing the ON pushbutton, and produce about two feet of leader. The leader-trailer can be code 200 or 377. To produce the code 200 leader, simultaneously press and hold the CTRL and SHIFT keys with the left hand; press and hold the REPT key, press the @ key. When the required amount of leader has been punched release the @ key, and then all keys. To produce the 377 code, 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 noted after typing and punching any number (n) of 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.

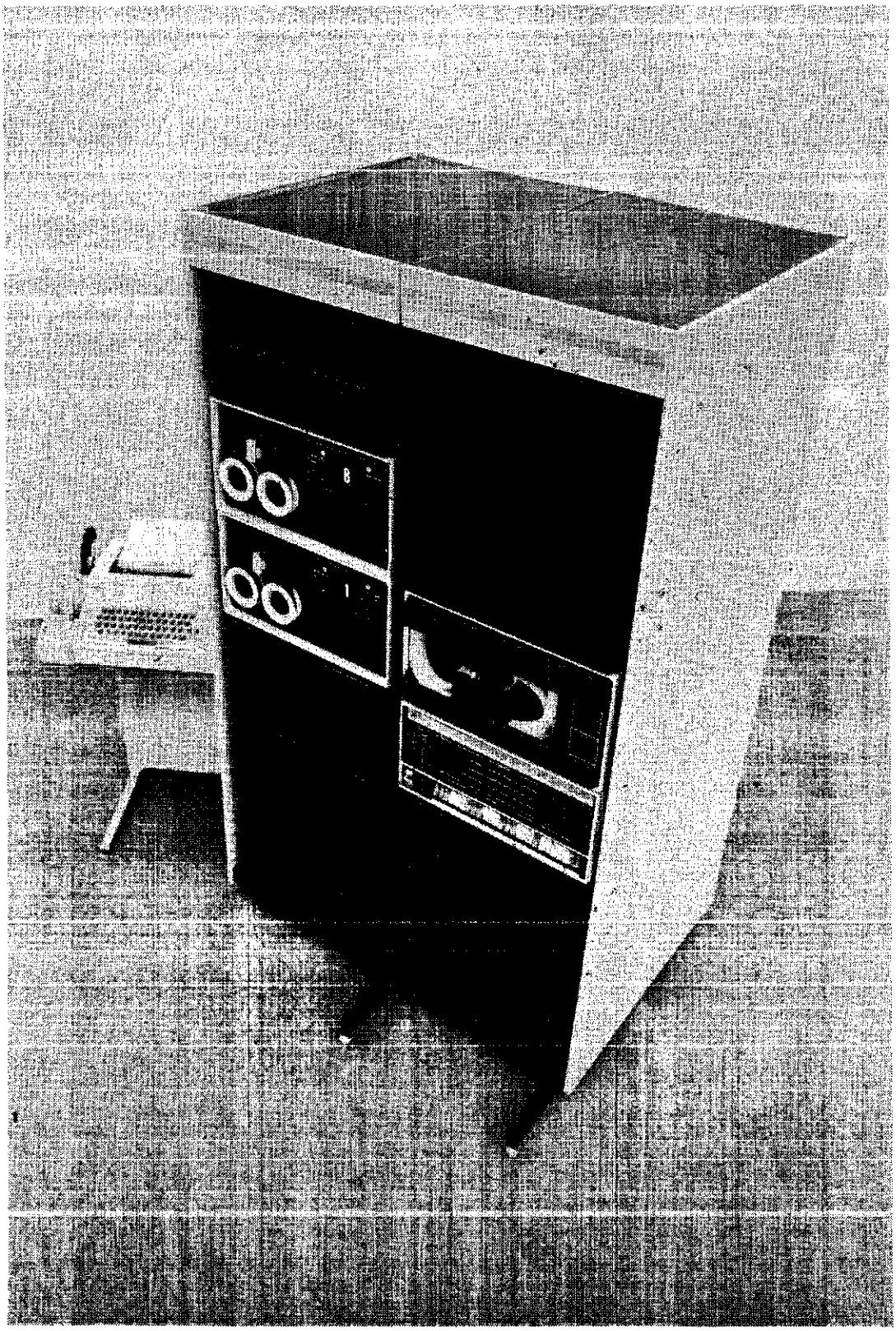
To duplicate and obtain a listing of an existing tape: Perform the procedure under steps a through c above. Then load the tape to be duplicated as de-

scribed in step b of the procedure listed under Loading Data Under Program Control. Initiate tape duplication by setting the reader START/STOP/FREE switch to the START position. The punch and teleprinter stops when the tape being duplicated is completely read. Corrections to insert or delete information on a perforated tape can be made by duplicating the correct portion of the tape and manually punching additional information or inhibiting punching of information to be deleted. This is accomplished by duplicating the tape and carefully observing the information being typed as the tape is read. In this manner the reader START/STOP/FREE switch can be set to the STOP position just before the point of the correction is typed. Information to be insertd can then be punched manually by means of the keyboard. Information can be deleted by pressing the punch OFF pushbutton and operating the reader until the portion of the tape to be deleted has been typed. It may be necessary to backspace and rub out one or two characters on the new tape if the reader is not stopped precisely on time. The number of characters to be rubbed out can be determined exactly by the typed copy. Be sure to count spaces when counting typed characters. Continue duplicating the tape in the normal manner after making the corrections.

New, duplicated, or corrected perforated tapes should be verified by reading them off-line and carefully proofreading the typed copy.

Program Control

When the program is stopped at the end of an instruction by raising the single step key, then the load address, examine, and deposit keys may be used without changing the AC. The program may then be resumed by resetting the PC to one less than the address wanted, operating the examine key and then operating the continue key. [On the PDP-8/L, the program may be resumed by resetting the PC to the desired address and then operating the CONT key.]



PDP-8/I is the most powerful and most versatile small computer available anywhere. Because of its flexible input/output facilities, PDP-8/I is the ideal starting point in an application that requires expansion.

CHAPTER 3

MEMORY AND PROCESSOR BASIC PROGRAMMING*

All notations in brackets [ ] indicate data for the PDP-8/L computer only.
The following terms are used in memory address programming:

Term	Definition
Page	A block of 128 core memory locations (200_8 addresses).
Current Page	The page containing the instruction being executed; as determined by bits 0 through 4 of the program counter.
Page Address	An 8-bit number contained in bits 4 through 11 of an instruction which designates one of 256 core memory locations. Bit 4 of a page address indicates that the location is in the current page when a 1, or indicates it is in page 0 when a 0. Bits 5 through 11 designate one of the 128 locations in the page determined by bit 4.
Absolute Address	A 12-bit number used to address any location in core memory.
Effective Address	An effective address is an absolute address obtained from core memory by indirect addressing.

Organization of the standard core memory or any 4096-word field of extended memory is summarized as follows:

Total locations (decimal)	4096
Total addresses (octal)	7777
Number of pages (decimal)	32
Page designations (octal)	0-37
Number of locations per page (decimal)	128
Addresses within a page (octal)	0-177

Four methods of obtaining the effective address are used as specified by combinations of bits 3 and 4.

Bit 3	Bit 4	Effective Address
0	0	The operand is in page 0 at the address specified by bits 5 through 11.
0	1	The operand is in the current page at the address specified by bits 5 through 11.
1	0	The absolute address of the operand is taken from the contents of the location in page 0 designated by bits 5 through 11.
1	1	The absolute address of the operand is taken from the contents of the location in the current page designated by bits 5 through 11.

*See Appendix A for Program Abstracts

The following example indicates the use of bits 3 and 4 to address any location in core memory.

Suppose it is desired to add the contents of locations A, B, C, and D to the contents of the accumulator by means of a routine stored in page 2. The instructions in this example indicate the operation code, the content of bit 4, the content of bit 3, and a 7-bit address. This routine would take the following form:

Page 0		Page 1		Page 2		
Location	Content	Location	Content	Location	Content	Remarks
A	xxxx	C	xxxx	B	xxxx	
M	C	D	xxxx	N	D	
				R	TAD 00 A	/DIRECT TO DATA IN PAGE 0
				S	TAD 01 B	/DIRECT TO DATA IN CURRENT PAGE
				T	TAD 10 M	/INDIRECT TO ADDRESS SPECIFIED IN PAGE 0
				U	TAD 11 N	/INDIRECT TO ADDRESS SPECIFIED IN CURRENT PAGE

Routines using 128 instructions or less can be written in one page using direct addresses for looping and indirect addresses for data stored in other pages. When planning the location of instructions and data in core memory, the following locations are reserved for special purposes:

Address	Purpose
0_8	Stores the contents of the program counter following a program interrupt.
1_8	Stores the first instruction to be executed following a program interrupt.
10_8 through 17_8	Auto-indexing.

INDIRECT ADDRESSING

When indirect addressing is specified, the address part (bits 5-11) of a memory reference instruction is interpreted as the address of a location containing the address of the operand. Consider the instruction TAD A. Normally, A is interpreted as the address of the location containing the quantity to be added to the content of the AC. Thus, if location 100 contains the number 5432, the instruction TAD 100 causes the quantity 5432 to be added to the contents of the AC. Now suppose that location 5432 contains the number 6543. The instruction TAD I 100 (where I signifies indirect addressing) causes the computer to take the number 5432, which is in location 100, as the effective address of the instruction and the number in location 5432 as the operand. Hence, this instruction results in the quantity 6543 being added to the contents of the AC.

AUTO-INDEXING

When a location between 10_8 and 17_8 in page 0 of any core memory field is addressed indirectly (by an instruction in which bit 3 is a 1) the contents of that location are read, incremented by one, rewritten in the same location, and then taken as the effective address of the instruction. This feature is called auto-indexing. If location 12_8 contains the number 5432 and the instruction DCA I 12 is given, the number 5433 is rewritten in location 12_8 and the contents of the accumulator are deposited in core memory location 5433.

STORING AND LOADING

Data is stored in any core memory location by use of the DCA instruction. This instruction clears the AC to simplify loading of the next data. If the data deposited is required in the AC for the next program operation, the DCA must be followed by a TAD for the same address. All loading of core memory information into the AC is accomplished by means of the TAD instruction, preceded by an instruction that clears the AC such as CLA or DCA.

Storing and loading of information in sequential core memory locations can make excellent use of an auto-index register to specify the core memory address.

PROGRAM CONTROL

Transfer of program control to any core memory location uses the JMP or JMS instructions. The JMP I (indirect address, 1 in bit 3) is used to transfer program control to any location in core memory which is not in the current page or page 0.

The JMS Y is used to enter a subroutine which starts at location Y + 1 in the current page or page 0. The contents of the PC + 1 are stored in the specified address Y, and address Y + 1 is transferred into the PC. To exit a subroutine the last instruction is a JMP I Y, which returns program control to the location stored in Y.

INDEXING OPERATIONS

External events can be counted by the program and the number can be stored in core memory. The core memory location used to store the event count can be initialized (cleared) by a CLA command followed by a DCA instruction. Each time the event occurs, the event count can be advanced by a sequence of commands such as CLA, TAD, IAC, and DCA.

The ISZ instruction is used to count repetitive program operations or external events without disturbing the contents of the accumulator. Counting a specified number of operations is performed by storing a two's complement negative number equal to the number of operations to be counted. Each time the operation is performed, the ISZ instruction is used to increment the contents of this stored number and to check the result. When the stored number becomes zero, the specified number of operations have occurred and the program skips out of the loop and back to the main sequence.

This instruction is also used for other routines in which the contents of a memory location are incremented without disturbing the contents of the accumulator, such as storing information from an I/O device in sequential memory locations or using core memory locations to count I/O device events.

LOGIC OPERATIONS

The PDP-8/I and PDP-8/L instruction lists include the logic instruction AND. Short routines can be written from this instruction to perform the inclusive OR and exclusive OR operations.

Logical AND

The logical AND operation between the contents of the accumulator and the contents of a core memory location Y is performed directly by means of the AND Y instruction. The logical AND performs an AND operation AC00 • MB00, AC01 • MB01, etc. The result remains in the AC, the original contents of the AC are lost, and the contents of location Y are unaffected.

Inclusive OR

Assuming value A is in the AC and value B is stored in a known core memory address, the following sequence performs the inclusive OR. The sequence is stated as a utility subroutine called IOR.

```
/CALLING SEQUENCE                          JMS  IOR
/                                          (ADDRESS  OF  B)
/ENTER WITH ARGUMENT IN AC: EXIT WITH LOGICAL RESULT IN AC
          /ADDRESS LABEL  INSTRUCTION REMARKS
               IOR,       0
                          DCA TEM1       /SAVE "A" ARGUMENT
                          TAD I IOR      /GET ADDRESS OF "B"
                          DCA TEM2
                          TAD TEM1
                          CMA
                          AND I TEM2     /Ā • B + A = A + B
                          TAD TEM1
                          ISZ IOR
                          JMP I IOR
               TEM1,      0
               TEM2,      0
```

Exclusive OR

The exclusive OR operation for two numbers, A and B, can be performed by a subroutine called by the mnemonic code XOR. In the following general purpose XOR subroutine, the value A is assumed to be in the AC, and the address of the value B is assumed to be stored in a known core memory location.

```
/CALLING SEQUENCE                          JMS XOR
/                                          (ADDRESS OF B)
/                                          (RETURN)
/ENTER WITH ARGUMENT IN AC: EXIT WITH LOGICAL RESULT IN AC

               XOR,       0
                          DCA TEM1
                          TAD I XOR
                          DCA TEM2
                          TAD TEM1
                          AND I TEM2
                          CMA IAC
                          CLL RAL
                          TAD TEM1
                          TAD I TEM2
                          ISZ XOR
                          JMP I XOR
               TEM1,      0
               TEM2,      0
```

An XOR subroutine can be written using fewer core memory locations by making use of the IOR subroutine; however, such a subroutine takes more time to execute. A faster XOR subroutine can be written by storing the value B instead of the address of B, in the second instruction of the calling sequence; however, the resulting subroutine is not as useful as the subroutine given here.

ARITHMETIC OPERATIONS

One arithmetic instruction is included in the order code, the two's complement add: TAD. Using this instruction, routines can easily be written to perform addition, subtraction, multiplication, and division in two's complement arithmetic.

Two's Complement Arithmetic

In two's complement arithmetic, addition, subtraction, multiplication, and division of binary numbers are performed in accordance with the common rules of binary arithmetic. In the PDP-8/I and the PDP-8/L, as in other machines utilizing complementation techniques, negative numbers are represented as the complements of positive numbers, and subtraction is achieved by complement addition. Representation of negative values in one's complement arithmetic is slightly different from that in two's complement arithmetic.

The one's complement of a number is the complement of the absolute positive value; that is, all 1s are replaced by 0s and all 0s are replaced by 1s. The two's complement of a number is equal to the one's complement of the positive value plus one.

In one's complement arithmetic a carry from the sign bit (most significant bit) is added to the least significant bit in an end-around carry. In two's complement arithmetic a carry from the sign bit complements the link (a carry would set the link to 1 if it were properly cleared before the operation), and there is no end-around carry.

PROGRAMMING SYSTEM

The programming system for the computers includes: symbolic assemblers of varying complexity and versatility, compilers, monitor systems, a symbolic tape editor, a floating point package and a mathematical function routines package, and utility and maintenance programs. All operate with the basic computer. The programming system was designed to simplify and accelerate the process of learning to program. At the same time, experienced programmers will find that it incorporates many advanced features. The system is intended to make available to the user the full, general-purpose data processing capability of the computer and to serve as the operating nucleus for a growing library of programs and routines to be made available to all installations. New techniques, routines, and programs are constantly being developed, field-tested, and documented in the Digital Program Library for incorporation into the user's systems.

Assemblers

The use of an assembly program has become standard practice in programming digital computers. The programmer codes his program in a symbolic language, using alphanumeric mnemonics for instructions and labels, rather than the binary numbers actually used in the computer. The assembler then translates the symbolic program into the equivalent machine code. The advantages are significant: the symbolic language is more meaningful to the programmer; instructions and data can be referred to by symbolic labels without regard to their actual locations in the computer; data characters can be expressed directly; programs can be altered without great changes and then reassembled; and debugging is considerably simplified. Three assemblers are available: PAL III, MACRO-8, and SABR.

PAL III Assembler — PAL III (Programming Assembly Language-III) is a basic assembler allowing symbolic references, symbolic origins, and expressions. The output of the PAL III Assembler is in a form suitable for input to the binary loader. The PAL III Assembler can be loaded into the computer by either high- or low-speed paper tape.

MACRO-8 Assembler — MACRO-8 is an advanced assembler which has the same basic features as PAL-III and, in addition, also has MACRO capability, literals, off-page references, and high/low speed paper tape input and output.

SABR Assembler — SABR (Symbolic Assembler for Binary Relocatable programs) is a single-pass assembler producing binary relocatable code, independent of core paging or memory fields. Psuedo-ops permit the use of subroutines and external references, and the binary tapes produced are loaded using the 8K System Linking Loader.

Compilers

Compilers for higher-level languages provide computer translation from programs written in a mixture of English phrases and symbolic notation. The more expressive statements or high level languages require less programming effort to produce the same sequence of machine instructions, yet the languages are so close to patterns of speech that the programmer can learn the language easily and with no need for knowledge of the internal organization of the computer. Compilers used with the PDP-8/I and PDP-8/L computers consist of FOCAL, FORTRAN II, ALGOL, BASIC, and INDAC-8.

FOCAL Language — FOCAL (FOrmula CALculator) is a conversational language useful in mathematical and statistical problems. A programmer can write, debug, and execute his program at a teletype console using this language. The simplicity of the language structure and command set makes it possible to begin programming immediately, without any prior knowledge of the organization or operation of the computer. FOCAL includes instructions for text editing, logical, control, input/output, and mathematical operations; mathematical functions; and error diagnostics for debugging in both compilation and execution.

FORTRAN II Language — The FORTRAN (FORmula TRANslation) II Language is provided in both 4K and 8K memory systems. The 4K system includes a one-pass compiler and an operating system to translate mathematical equations and logical decision making processes into binary code and execute the programs, producing diagnostic messages both in compilation and in execution. FORTRAN II includes instructions for mathematical, logical, control, and input/output operations, as well as mathematical functions. The FORTRAN II 8K system differs from the 4K system primarily by using a two-pass compiler (which uses a version of the SABR assembler as a second pass). A linking loader, in addition to an operating system, can be used on computers with a minimum of 8K of core memory. Additions to the 8K compiler (over the 4K compiler), include:

a. U.S.A. Standard FORTRAN Syntax
b. Subroutines
c. Two levels of subscripting
d. Function subprograms
e. I/O supervisor
f. Relocatable link loadable outputs
g. Common storage
h. I, E, F, H, A, X format specification
e. Arithmetic and trigonometric library

The use of the SABR assembler makes this compiler independent of core paging and field boundaries.

ALGOL Language — The ALGOrithmic Language compiler and operating system provide a facility similar to the FORTRAN II programming system, but based on slightly different structural features. ALGOL is provided to ensure compatibility with other user's systems. One of the most widely used international programming languages, ALGOL emphasizes formal, well-defined procedures for solving problems with computers.

BASIC Language — A conventional language similar to FOCAL and with similar features, the BASIC language is designed to run under the control of the Time-Sharing System (TSS/8), which is optional only to the PDP-8/I computer. The BASIC language is made available to provide compatibility with other user's systems and to allow many users to have equal access to the computer, while it maintains records of the time used by each user.

INDAC-8 Language — INDAC-8 (INdustrial Data Acquisition and Control), is a combined hardware and software system designed for use with a PDP-8 family computer. INDAC provides real-time data acquisition of digital or analog inputs which can be processed in a time-shared mode to provide tabulated output and direct process control. A background system makes use of unused processor time for general program execution.

Monitoring Systems

TSS/8 Monitoring System — The Time-Sharing System designed only for the PDP-8/I is a general-purpose, stand-alone monitor to provide simultaneous access of the computer to up to 16 users (a maximum of two pages per user), each of whom interacts with the system as though he were the only user due to the great difference in response times between the computer and the user. TSS/8 provides text-editing and file-maintenance functions for use with a library of systems including loaders, assemblers, editors, debugging systems, FORTRAN, and the conversational languages BASIC and FOCAL.

Disk/DECtape Monitoring System — This system permits the user to control the flow of programs through his computer and takes full advantage of the extended memory capabilities of disk or DECtape. In addition to the Monitor, the system also contains a library of system programs. Together they provide the user with the capabilities of compiling, assembling, editing, loading, saving, calling, and debugging his own programs.

Symbolic On-Line Debugging Program — On-line debugging with DDT-8 provides the user with dynamic printed program status information, close control over program execution, and prevents errors ("bugs") from destroying other portions of his program. The user can monitor the execution of single instructions or subsections, change instructions or data in any format, and output a corrected program at the end of the debugging session. Using the standard Teletype keyboard/reader and teleprinter/punch, the user can communicate conveniently with the PDP-8/I or the PDP-8/L in the symbols of his source language. He can control the execution of any portion of his object program by inserting breaks, or traps, in it. When the computer reaches a break, it transfers control of the object program to DDT. The user can then examine and modify the contents of individual core memory registers to correct and improve his object program.

Symbolic Tape Editor

The Symbolic Tape Editor program is used to edit, correct, and update symbolic program tapes using the PDP-8/I or PDP-8/L, the Teletype unit and/or the high-speed reader. With the editor in core memory, the user reads in portions

of his symbolic tape, removes, changes, or adds instructions or operands, moves single or multiple lines of text from place to place, and receives in return a complete new symbolic tape with errors removed. The user can work through the program instruction-by-instruction, spot check it, or concentrate on new sections. A character string search is also available.

Floating Point Package

The Floating Point Package permits the PDP-8/I or the PDP-8/L to perform arithmetic operations that many other computers can perform only after the addition of costly optional hardware. Floating point operations automatically align the binary points of operands, retaining the maximum precision available by discarding leading zeros. In addition to increasing accuracy, floating point operations relieve the programmer of scaling problems common in fixed point operations. This is of particular advantage to the inexperienced programmer.

Mathematical Function Routines

The programming system also includes a set of mathematical function routines to perform the following operations in both single and double precision: addition, subtraction, multiplication, division, square root, sine, cosine, arc tangent, natural logarithm, and exponential.

Utility and Maintenance Programs

PDP-8/I and PDP-8/L utility programs provide printouts or punchouts of core memory content in octal, decimal, or binary form, as specified by the user. Subroutines are provided for octal or decimal data transfer and binary-to-decimal, decimal-to-binary, and Teletype tape conversion. A complete set of standard diagnostic programs is provided to simplify and expedite system maintenance. Program descriptions and manuals permit the user to effectively test the operation of the computer for proper core memory functioning and proper execution of instructions. In addition, diagnostic programs to check the performance of standard and optional peripheral devices are provided with the devices.

CHAPTER 4
MEMORY AND PROCESSOR INSTRUCTIONS

All notations in brackets [ ] indicate data for the PDP-8/L computer only.
An instruction is a coded program step that tells the computer what to do for a single operation in a program. It is a set of characters, together with one or more addresses (or no address), that defines an operation and which, as a unit, causes the computer to operate accordingly on the indicated quantities. In the PDP-8 family of computers, there are two types of instruction words: 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 (op) 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 [1.6] μs memory cycle. Indirect addressing increases the execution time of a memory reference instruction by 1.5 [1.6] μs. The augmented instructions (input/output transfer and operate) are performed in 4.25 and 1.5 [1.6] μs, respectively. (All computer times are ±20%.)

MEMORY REFERENCE INSTRUCTIONS

Since the PDP-8/I and PDP-8/L systems contain 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 either the current page or page 0 containing the full 12-bit absolute address of the operand.

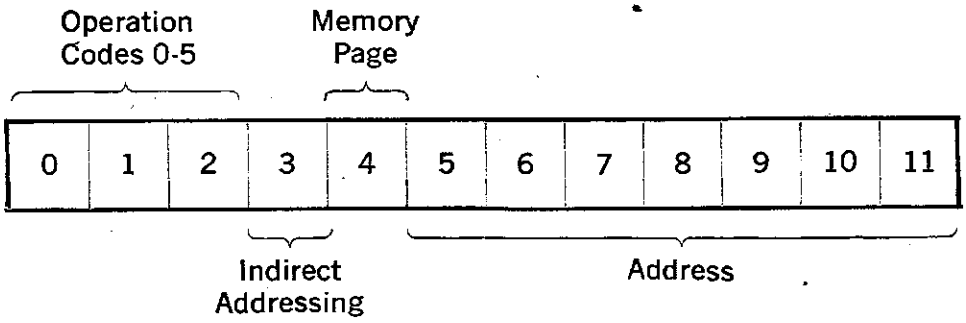


Fig. 4-1. Memory Reference Instruction Bit Assignments

Word format of memory reference instructions is shown in Figure 4-1 and the instructions perform as follows:

Logical AND (AND Y)

Octal Code: 0

Indicators: And, Fetch, Execute [IR = 0, F, E]

Execution Time: 3.0 [3.2] μs with direct addressing, 4.5 [4.8] μs with indirect addressing.

Operation: The AND operation is performed between the contents of memory location Y and the contents of the AC. The result is left in the AC, the original contents of the AC are lost, and the contents of Y are 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 [IR = 1, F, E]

Execution Time: 3.0 [3.2] μs with direct addressing, 4.5 [4.8] 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: $AC_{0-11} + Y_{0-11} \rightarrow AC_{0-11}$

Increment and Skip if Zero (ISZ Y)

Octal Code: 2

Indicators: Isz, Fetch, Execute [IR = 2, F, E]

Execution Time: 3.0 [3.2] μs with direct addressing, 4.5 [4.8] μs with indirect addressing.

Operation: The contents of memory location Y are incremented by one in two's complement arithmetic. If the resultant contents of Y equal zero, the contents of the PC are incremented by one and the next instruction is skipped. If the resultant contents of Y do not equal zero; the program proceeds to the next instruction. The incremented contents of Y are restored to memory. The contents of the AC are not affected by this instruction.

Symbol: Y + 1 → Y

If resultant $Y_{0-11} = 0$, then PC + 1 → PC

Deposit and Clear AC (DCA Y)

Octal Code: 3

Indicators: Dca, Fetch, Execute [IR = 3, F, E]

Execution Time: 3.0 [3.2] μs with direct addressing, 4.5 [4.8] μs with indirect addressing.

Operation: The contents of the AC are deposited in core memory at address Y and the AC is cleared. The previous contents of memory location Y are lost.

Symbol: AC → Y
then 0 → AC

Jump to Subroutine (JMS Y)

Octal Code: 4

Indicators: Jms, Fetch, Execute [IR = 4, F, E]

Execution Time: 3.0 [3.2] μs with direct addressing, 4.5 [4.8] μs with indirect addressing.

Operation: The contents of the PC are deposited in core memory location Y and the next instruction is taken from core memory location Y + 1. The contents of the AC are not affected by this instruction.

Symbol: PC → Y
Y + 1 → PC

Jump to Y (JMP Y)

Octal Code: 5

Indicators: Jmp, Fetch [IR = 5, F]

Execution Time: 1.5 [1.6] μs with direct addressing, 3.0 [3.2] 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 contents of the PC are lost. The contents of the AC are not affected by this instruction.

Symbol: Y → PC

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.

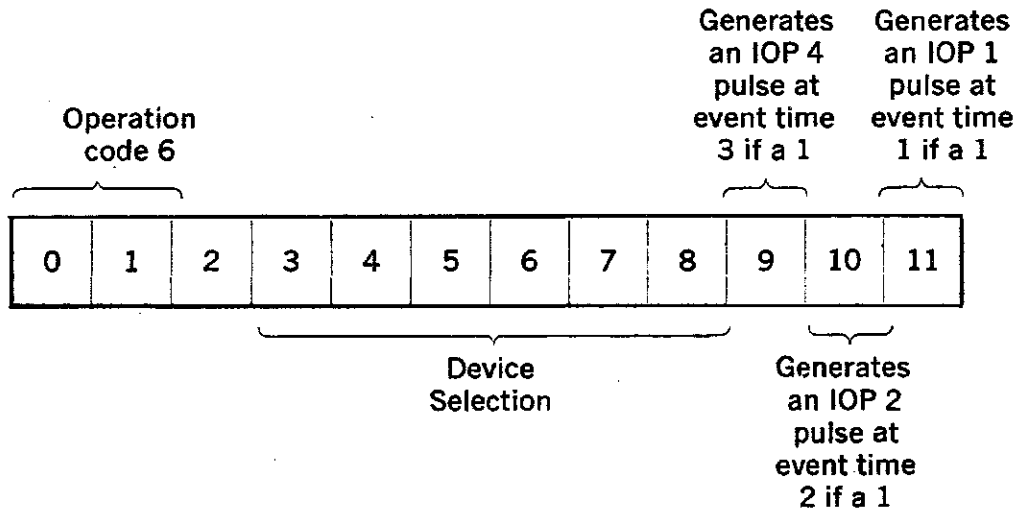
Input/Output Transfer Instruction

Microinstructions of the input/output transfer (IOT) initiate the 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 the 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 4-2. Operations performed by IOT microinstructions are explained in Chapter 7.

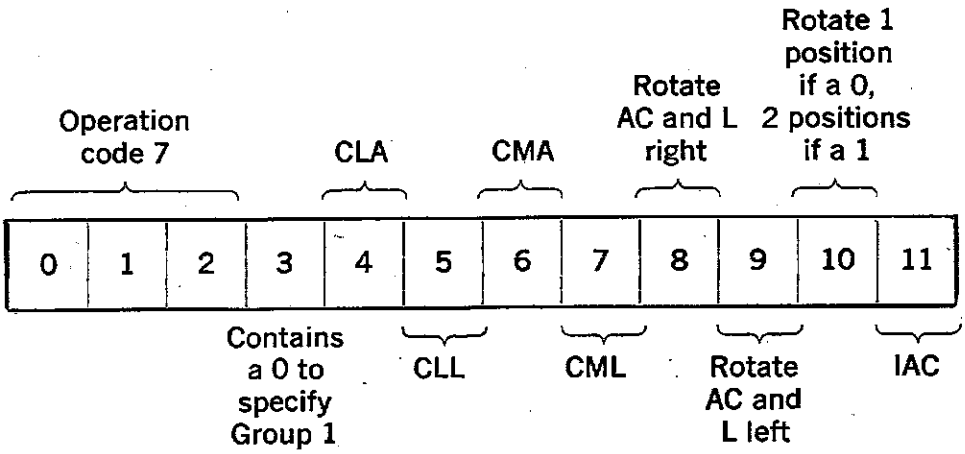


Operate Instruction

By using operate instructions, the programmer can consider the logical sequences occurring during one computer Fetch cycle to 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 contents 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.

GROUP 1

The Group 1 operate microinstruction format is shown in Figure 4-3, 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 1s to bits 5, 6, and 11; although it is not logical to assign 1s to bits 8 and 9 simultaneously since they specify conflicting operations. (The most frequently used combinations are listed in Appendix B).



LOGICAL SEQUENCE:
1 — CLA, CLL
2 — CMA, CML
3 — IAC
4 — RAR, RAL, RTR, RTL

Figure 4-3. Group 1 Operate Instruction Bit Assignments

No Operation (NOP)
Octal Code: 7000
Sequence: None
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs
Operation: This command causes a 1-cycle delay in the program before 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.

Symbol: None

Increment Accumulator (IAC)
Octal Code: 7001
Sequence: 3
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs
Operation: The contents of the AC are incremented by one in two's complement arithmetic.

Symbol: AC + 1 → AC

Rotate Accumulator Left (RAL)
Octal Code: 7004
Sequence: 4
Indicators: Opr, Fetch [IR = 7, F]

Execution Time: 1.5 [1.6] μs

Operation: The contents of the AC are rotated one binary position to the left with the content of the link. The contents of bits AC1 — 11 are shifted to the next greater significant bit, the content of AC0 is shifed into the L, and the content of the L is shifted into AC11.

Symbol: ACj → ACj — 1
AC_0 → L
L → AC_{11}

Rotate Two Left (RTL)
Octal Code: 7006
Sequence: 4
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The contents of the AC are rotated two binary positions to the left with the content of the link. This instruction is logically equal to two successive RAL operations.

Symbol: $ACj \rightarrow ACj - 2$
$AC_1 \rightarrow L$
$AC_0 \rightarrow AC_{11}$
$L \rightarrow AC_{10}$

Rotate Accumulator Right (RAR)
Octal Code: 7010
Sequence: 4
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The contents of the AC are rotated one binary position to the right with the content of the link. The contents 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: $ACj \rightarrow ACj + 1$
$AC_{11} \rightarrow L$
$L \rightarrow AC_0$

Rotate Two Right (RTR)
Octal Code: 7012
Sequence: 4
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs
Operation: The contents of the AC are rotated two binary positions to the right with the content of the link. This instruction is logically equal to two successive RAR operations.

Symbol: $ACj \rightarrow ACj + 2$
$AC_{10} \rightarrow L$
$AC_{11} \rightarrow AC_0$
$L \rightarrow AC_1$

Complement Link (CML)
Octal Code: 7020
Sequence: 2
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The content of the L is complemented.

Symbol: $\overline{L} \rightarrow L$

Complement Accumulator (CMA)
Octal Code: 7040
Sequence: 2
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The contents of the AC are set to the one's complement of the current contents of the AC. The content of each bit of the AC is complemented individually.

Symbol: $\overline{ACj} \rightarrow ACj$

Complement and Increment Accumulator (CIA)
Octal Code: 7041
Sequence: 2, 3
Indicators: Opr, Fetch [IR = 7, R]
Execution Time: 1.5 [1.6] μs

Operation: The contents of the AC are converted from a binary value to their equivalent two's complement number. This conversion is accomplished by combining the CMA and IAC commands, thus the contents of the AC are complemented during sequence 2 and are incremented by one during sequence 3.

Symbol: $\overline{ACj} \rightarrow ACj$,
then $AC + 1 \rightarrow AC$

Clear Link (CLL)
Octal Code: 7100
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The content of the L is cleared to contain a 0.

Symbol: $0 \rightarrow L$

Set Link (STL)
Octal Code: 7120
Sequence: 1, 2
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The L is set to contain a binary 1. This instruction is logically equal to combining the CLL and CML commands.

Symbol: $1 \rightarrow L$

Clear Accumulator (CLA)
Octal Code: 7200
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The content of each bit of the AC is cleared to contain a binary 0.

Symbol: $0 \rightarrow AC$

Set Accumulator (STA)
Octal Code: 7240
Sequence: 1, 2
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

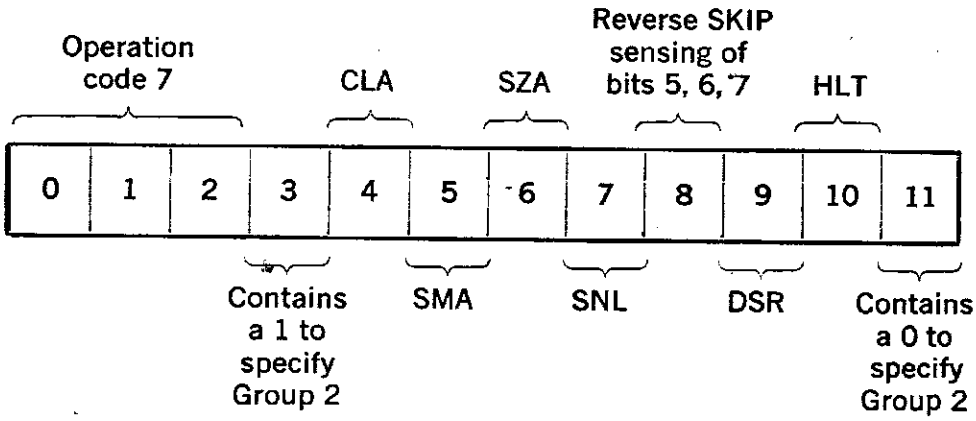
Operation: Each bit of the AC is set to contain a binary 1. This operation is logically equal to combining the CLA and CMA commands.

Symbol: 1 → ACj

GROUP 2

The Group 2 operate microinstruction format is shown in Figure 4-4 and the primary microinstructions are explained in the following paragraphs. Any logical combination of bits within this group can be composed into one microinstruction. (The instructions constructed by most logical command combinations are listed in Appendix B.)

If skips are combined in a single instruction the inclusive OR of the conditions 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 1s are designed 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 1s 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.



LOGICAL SEQUENCE:
1 (Bit 8 is a 0) — Either SMA or SZA or SNL
(Bit 8 is a 1) — Both SPA and SNA and SZL
2 — CLA
3 — OSR, HLT

Figure 4-4. Group 2 Operate Instruction Bit Assignments

Halt (HLT)

Octal Code: 7402
Sequence: 3
Indicators: Opr, not Run [IR = 7 not RUN]
Execution Time: 1.5 [1.6] 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 or 2, and so are performed before the program stops.

Symbol: 0 → RUN

OR with Switch Register (OSR)

Octal Code: 7404

Sequence: 3
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The inclusive OR operation is performed between the contents of the AC and the contents of the SR. The result is left in the AC, the original contents of the AC are lost, and the contents of the SR are unaffected by this command. When combined with the CLA command, the OSR performs a transfer of the contents of the SR into the AC.

Symbol: ACj V SRj → ACj

Skip, Unconditional (SKP)
Octal Code: 7410
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The contents of the PC are incremented by one so that the next sequential instruction is skipped.

Symbol: PC + 1 → PC

Skip on Non-Zero Link (SNL)
Octal Code: 7420
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The content of the L is sampled, and if it contains a 1, the contents of the PC are 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 → PC

Skip on Zero Link (SZL)
Octal Code: 7430
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The content of the L is sampled, and if it contains a 0 the contents of the PC are 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 → PC

Skip on Zero Accumulator (SZA)
Octal Code: 7440
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The content of each bit of the AC is sampled, and if all bits contain a 0 the contents of the PC are incremented by one so that the next sequential instruction is skipped. If any bit of the AC contains a 1, no operation occurs and the next sequential instruction is initiated.

Symbol: If $AC_0 - _{11} = 0$, then PC + 1 → PC

Skip on Non-Zero Accumulator (SNA)
Octal Code: 7450
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The content of each bit of the AC is sampled, and if any bit contains a 1 the contents of the PC are 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 $AC_0 - _{11} \neq 0$, then PC + 1 → PC

Skip on Minus Accumulator (SMA)
Octal Code: 7500
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: The content of the most significant bit of the AC is sampled, and if it contains a 1, indicating that the AC contains a negative two's complement number, the contents of the PC are 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: $AC_0 = 1$, then PC + 1 → PC

Skip on Positive Accumulator (SPA)
Octal Code: 7510
Sequence: 1
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] 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 contents of the PC are 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 $AC_0 = 0$, then PC + 1 → PC

Clear Accumulator (CLA)
Octal Code: 7600
Sequence: 2
Indicators: Opr, Fetch [IR = 7, F]
Execution Time: 1.5 [1.6] μs

Operation: Each bit of the AC is cleared to contain a binary 0.

Symbol: 0 → AC

PROGRAM INTERRUPT

The program interrupt features allow certain external conditions to interrupt the computer program. Program interrupts are used to either speed the information processing of input/output devices or allow certain alarms to halt the program in progress and initiate another routine. When a program interrupt request is made, the computer completes execution of the instruction in

progress before acknowledging the request and entering the interrupt mode. A program interrupt is similar to a JMS to location 0; that is, the contents of the program counter are stored in location 0, and the program resumes operation in location 1 with the interrupt disabled. The interrupt program commencing in location 1 is responsible for identifying the signal causing the interruption, for removing the interrupt condition; and for returning to the original program with the interrupt re-enabled. Exit from the interrupt program, back to the original program, can be accomplished by a JMP I 0 instruction.

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

Interrupt Turn On (ION)
Octal Code: 6001
Event Time: Not applicable
Indicators: Iot, Fetch, Ion [IR = 6, F, ION]
Execution Time: 1.5 [1.6] 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 [IR = 6, F]
Execution Time: 1.5 [1.6] μs

Operation: This command disables the program interrupt synchronization element to prevent interruption of the current program.

Symbol: 0 → INT ENABLE, INT DELAY

Programming
When an interrupt request is acknowledged, the interrupt is automatically disabled by the program interrupt synchronization circuits (not by instructions). The next instruction is taken from core memory location 1. Usually, the instruction stored in location 1 is a JMP, which transfers program control to a subroutine which services the interrupt. At some time during this subroutine, an ION instruction must be given. The ION can be given at the end of the subroutine to allow other interrupts to be serviced after program control is transferred back to the original program. In this application, the ION instruction immediately precedes the last instruction in the routine. A delay of one instruction (regardless of the execution time of the following instruction), is inherent in the ION instruction to allow transfer of program control back to the original program before enabling the interrupt. Exit from the subroutine usually is accomplished by a JMP I 0 instruction.

The ION command can be given during the subroutine as soon as it has determined the I/O device causing the interrupt. This latter method allows the subroutine which is handling a low priority interrupt to be interrupted, possibly by a high priority device. Programming of an interrupt subroutine, which checks for priority and allows itself to be interrupted, must make provisions to relocate the contents of the program counter stored in location 0; so that the return address to the original program is not lost if another interrupt occurs.

CHAPTER 5
DATA BREAK

All notations in brackets [ ] indicate data for the PDP-8/L computer only.

Programmed transfers of data, including program interrupt transfers, pass through the accumulator. The accumulator must therefore be cleared while the transfer is performed. This type of transfer is often too slow for use with extremely fast peripheral devices. Devices which operate at very high speed, or which require very rapid response from the computer, use the data break facility (standard on the PDP-8/I; optional on the PDP-8/L). When using the data break facility, an external device is permitted to insert or extract words from the computer memory, bypassing all program control. Because the computer program has no cognizance of the transfers made in this manner, the program must check for the presence of this data prior to its use. The data break is particularly well suited for devices that transfer large amounts of data in block form, for example, random-access disk files, high-speed magnetic tape systems, or high-speed drum memories.

Peripheral equipment connected to the data break facility can cause a temporary suspension of the in-progress program, causing a halt in information being transferred with the computer core memory, via the MB. From two to nine I/O devices can be connected directly to the data break facility through the use of a Type DM04 Data Multiplexer (one DM04 for each three break devices), which has a positive I/O bus. (A data multiplexer is not necessary when only one I/O device is used.) The Type DM01 Data Multiplexer performs a similar function on computers equipped with a negative I/O bus, and can handle up to seven break devices. This cycle stealing mode of operation provides a high-speed transfer of individual words or blocks of information to core memory.

Since program execution is not involved in data break transfers, the program counter, accumulator, and instruction register are not disturbed or involved in these transfers. The program is suspended at the conclusion of an executed instruction and the data break is initiated to perform the transfer; then the Fetch cycle is initiated to continue the main program.

There are two types of data breaks: single-cycle and three-cycle. During a single-cycle data break, registers in either the I/O device or the device's interface specify the core memory address for each transfer and count the number of transfers, thus determining the end of the data blocks. During a three-cycle data break, two core memory locations within the computer are used to perform these functions, thereby simplifying the I/O interface by omitting two hardware registers. The computer receives the following signals from the I/O device during a data break:

Signal	−3 [+3] V	0 V
Break Request	No break request	Break request
Cycle Select	One-cycle break	Three-cycle break
Transfer Direction	Data into computer	Data out of computer
Increment CA Inhibit	CA incremented	CA not incremented
Increment MB (pulse)	MB not incremented	MB incremented
Address (12 bits)	Binary 0	Binary 1
Data (12 bits)	Binary 0	Binary 1

The computer sends the following signals to the device during a data break:

Signal	Characteristics
Data (12 bits)	−3 [+3] V = binary 0, 0 V = binary 1
Address Accepted	0.35-0.45 μs negative going pulse (0 to −3 V) [+3 V to 0 V] beginning at time TP4 of a break cycle.
WC Overflow	−3 [+3 V to 0 V] V level change beginning at time TP2 of the WC state and lasting to the end of the current machine cycle.
Buffered Break	−3 [0 V] V when in Break state.

NOTE: Signal voltages for the positive bus PDP-8/I (with the KA8/IB option installed), are like the PDP-8/L, +3 V and 0 V.

The I/O device must supply four signals simultaneously to initiate a data break: break request, that sets the BRK SYNC flip-flop in the major state generator thus controlling entry into the data break states (either the Word Count for a three-cycle break or the Break for a single-cycle break); transfer direction (supplied to the MB allowing data to be strobed into the MB from the I/O device and inhibiting reading from core memory); cycle select (controlling gating in the major state generator, thus determining if the single-cycle or three-cycle data break is selected); and a core memory address of the transfer for the data supplied. When the data break request is made, the data break cycle replaces the fetch cycle of the next instruction. Therefore, the data break is entered at the end of either the execute cycle of a memory reference instruction, the fetch cycle of augmented instructions, or a previous break cycle. Once a data break is established, the computer continues to execute breaks until the break request signal by the I/O device is removed.

More exactly, the break request signal sets the BRK SYNC flip-flop.

At time TP4 of each computer cycle, the major state generator is set to establish the state for the next cycle. At this time, the status of the BRK SYNC flip-flop is sampled and if it is set, the Word Count or Break state is set into the major state generator and a data break cycle commences. Therefore, to initiate a data break, the break request input line must be at ground potential at time TP1 of the cycle preceding the data break cycle. The break request, address, data, transfer direction, and cycle select signals all should be simultaneously supplied to the computer.

When a data break occurs, the address designated by the I/O device is loaded into the MA during time TP4 of the last cycle of the current instruction (instruction in progress at the time of the break request signal), and the major state generator is set according to the cycle select signal. If the cycle select signal is at ground then the major state generator is set to the Word Count state, but if the cycle select signal is at −3 [+3] V, then the major state generator is set to the Break state. The program is then delayed for the duration of the data break which commences in the following cycle. A break request is granted only after completion of the current instruction under the following conditions:

a. At the end of the fetch cycle of an OPR or IOT instruction, or a directly addressed JMP instruction.

b. At the end of the defer cycle of an indirectly addressed JMP instruction.

c. At the end of the execute cycle of a JMS, DCA, ISZ, TAD, or AND instruction.

At either the beginning of the word count cycle of a three-cycle data break or the break cycle of a single-cycle data break, the address supplied to the MA is strobed into the MA and the computer supplies an address accepted signal to the I/O device. Entry into the Break state is indicated to the I/O device by a buffered break signal. These signals may be used to enable gates in the I/O device and its interface to perform tasks that are associated with the data transfer.

The address accepted signal is the most convenient control available to the I/O equipment to disable the break request signal, since this signal must be removed to prevent a second data break from occurring in the next cycle. If the transfer direction signal establishes that the direction of the data flow is from the computer, the contents of the core memory register at the address specified are transferred into the MB and are immediately available for strobing by the I/O equipment. If the transfer direction signal specifies that the direction of data flow is into the computer, the reading from core memory is inhibited and data is transferred into the MB from the I/O device. The status of the BRK SYNC flip-flop is sensed at time TP1 of a break cycle to determine if an additional break cycle is required. If the break request signal is present, the Break state is maintained in the major state generator; if a break request signal is not received at this time, the Fetch state is set into the major state generator to continue the program. The break request signal should be removed before the end of the address accepted signal if additional break cycles are not required.

SINGLE-CYCLE DATA BREAK

Single-cycle data breaks either transfer a data word into the computer core memory from the I/O device, transfer a data word into an I/O device from the core memory, or increment the contents of a device-specified core memory location. In each of these types of data break, one computer cycle is stolen from the program during each transfer; break cycles occur singly (interleaved with the program steps) or continuously (as in a block transfer), depending upon the timing of the break request signal, at rates of up to 660 kHz [625 kHz].

During the first portion of the break cycle, the contents of the addressed cell are read into the MB; if the transfer direction is from the computer to the I/O device. The write operation then restores the original contents of the addressed cell to memory. If the transfer direction is into the computer, information is transferred from the output data register of the I/O device into the MB at time TP2, and is written into core memory during the later portions of the break cycle. If there is a further break request, another break cycle is initiated. If there is no break request, the contents of the PC are transferred into the MA, and the major state generator is set to Fetch. The program then executes the next instruction.

The Increment MB facility is useful for counting iterations or events by means of a data break, so that the PC and AC are not disturbed. Within one break cycle of 1.5 [1.6] μs, a word is fetched from a device-specified core memory location, is incremented by one, and is restored to the same memory location. The increment MB signal input must be supplied to the computer only during a break cycle and the direction of transfer must be from the computer. These restrictions can be met by a simple AND gate in the device; an increment MB signal is generated only when an event occurs, the buffered break signal from the computer is present, and the transfer direction signal supplied to the computer is at ground potential.

THREE-CYCLE DATA BREAK

The three-cycle data break provides an economical method of controlling the transfer of data between the computer core memory and fast peripheral devices. Transfer rates in excess of 220 kHz [208 kHz] are possible using this feature.

The three-cycle data break differs from the one-cycle break in that a ground-level cycle select signal is supplied at the same time as the break request is made. When the request is honored, the Word Count state is set to increment the fixed memory location containing the word count. The I/O device requesting the break supplies this address as in the single-cycle break, except that it is a fixed address supplied by wired ground and −3 [+3] V signals rather than from a register.

Following the Word Count state a Current Address state occurs in which the location following the word count address is read, incremented by one, restored to memory, and loaded into the MA to be used as the transfer address. Then the Break state is initiated to effect the transfer between the I/O device and the computer memory cell specified by the MA. The following paragraphs detail each of these states.

Word Count State

The Word Count state is initiated and the contents of the core memory address specified by the external I/O device plus 1 are loaded into the MB at time TP2. The word count, established previously by instructions, is the two's complement negative number equal to the required number of transfers. When the word becomes 0 following an increment, the computer generates a WC overflow signal and supplies it to the I/O device. The incremented word count is rewritten in memory, the contents of the MA are incremented by 1 to establish the next location as the address for the following memory cycle, and the major state generator is set to the Current Address state.

Current Address State

The Current Address state is entered as the second cycle of a three-cycle data break. Normally the current address location is incremented before use, although the incrementing can be suppressed by grounding INCREMENT CA INHIBIT [+1 → CA Inhibit]. In either case, the memory word from the address following the word count address is loaded (directly or incremented) into the MB at time TP2. The contents of the MB are rewritten into core memory, the address word in the MB is transferred into the MA to designate the address to be used in the succeeding memory cycle, and the major state generator is set to the Break state.

Break State

The Break state is the third cycle of a three-cycle data break. The actual transfer of data between the I/O device and core memory, through the MB, occurs during this cycle in the same manner as during a single-cycle data break, except that the address has been determined from a word in memory rather than directly from the I/O device.

CHAPTER 6
COMPUTER INTERNAL OPTIONS

All notations in brackets [ ] indicate data for the PDP-8/L computer only. Chapter 6 is divided into six sections; Section 6-1 describes the computer options to the PDP-8/I and PDP-8/L systems for memory equipment and instructions, including the MC8/IA and B [MC8/LA and B] Memory Extension Control, MM8/IA-F Memory Modules (for PDP-8/I only), and MP8/I [MP8/L] Memory Parity Control; Section 6-2 describes the Power Failure Detection and Restart unit KP8/I [KP8/L]; Section 6-3 describes the Extended Arithmetic Element (EAE) KE8/I (for PDP-8/I only); Section 6-4 describes the various Real Time Clock options KW8/IA-F [KW8/LA-F]; Section 6-5 describes the Time Sharing Hardware Modification Unit KT8/I (for PDP-8/I only); and Section 6-6 describes the Positive Input/Output Bus KA8/I (for PDP-8/I only). The KD8/L Data Break facility is also considered a computer internal option, but is discussed in Chapter 5.

SECTION 6-1 MEMORY EQUIPMENT AND INSTRUCTIONS

MEMORY EXTENSION CONTROL MC8/I [MC8/L] AND MEMORY MODULE MM8/I

(The logic for the Memory Extension Control, Type MC8/I, is located in the PDP-8/I central processor, whereas the Memory Extension Control, Type MC8/L, is located in the BA08 Peripheral Expander.) There are two types of memory extension controls and memory modules available for use with these systems. The MC8/IA and MC8/LA are designed for operation in the systems not utilizing the memory parity options. When the MP8/I or MP8/L Memory Parity options are installed in these systems then either the MC8/IB or the MC8/LB should be used in lieu of the MCA/8IA and MCA/8LA, respectively.

The memory extension control expands the storage capacity of the standard 4096-word core memory by adding fields of 4096-word core memories. Field select control and extended-address control for up to 32,768 [8192] words are provided through maximum use of these options. In the PDP-8/I the MC8/I adds a 4096-word core memory internal to the computer for a total of 8K of memory. Each Memory Module (MM8/I) adds either 4096-words of core memory or 8192 words of core memory external to the computer. Up to six fields of 4096-word core memory can be added externally to the PDP-8/I, providing a maximum storage of 32,768 words (internal and external). The MM8/I cannot be attached to the MC8/L. Direct addressing of 32,768 words requires 15 bits ($2^{15} = 32{,}768$). However, since programs and data do not need to be directly addressed for execution of each instruction, a field can be program-selected, and thus all 12-bit addresses are assumed to be within the current memory field. Program interrupt of a program in any field automatically specifies field 0, address 0 for storage of the program count. The memory extension control consists of several 3- [1-] bit flip-flop registers extending addresses to 15 [13] bits to establish or select the proper field.

Addition of a memory extension control to a standard PDP-8/I requires a simple modification of the operator's console to activate indicators and switches associated with the IF register and the DF register of the memory extension control. These switches function, in the same manner as the switch register, to load information into associated registers when the Load Add [LOAD ADDR] switch is depressed.

The functional circuit elements which comprise the memory extension control perform as follows:

Instruction Field Register (IF) — The IF is a 3- [1-] bit register that serves as an extension of the PC. The contents of the IF determine the field from which all instructions are taken and the field from which operands are taken in directly-addressed AND, TAD, ISZ, or DCA instructions. Depressing the Load Add [LOAD ADDR] switch transfers the contents of the instruction field switch register on the operator's console into the IF register. During a JMP or JMS instruction, the IF is set by a transfer of information contained in the instruction buffer register. When a program interrupt occurs, the contents of the IF are automatically stored in bits 0 through 2 [bit 0 only] of the save field register for restoration to the IF from the instruction buffer register at the conclusion of the program interrupt subroutine.

Data Field Register (DF) — This 3- [1-] bit register determines the memory field from which operands are taken in indirectly-addressed AND, TAD, ISZ, or DCA instructions. Depressing the Load Add [LOAD ADDR] switch transfers the contents of the DATA FIELD switch register on the operator's console into the DF register. The DF is set by a transfer of information from bits 6 through 8 [bit 8 only] of the MB during a CDF microinstruction to establish a microprogrammed data field. When a program interrupt occurs, the contents of the DF are automatically stored in the save field register. The DF is set by a transfer of information from bits 3 through 5 [bit 1 only] of the save field register by the RMF microinstruction to restore the data field at the conclusion of the program interrupt subroutine.

Instruction Buffer Register (IB) — The IB serves as a 3- [1-] bit input buffer for the instruction field register. All field number transfers into the instruction field register are made through the instruction buffer, except transfers from the operator's console switches. The IB is set by depressing of the Load Add [LOAD ADDR] switch in the same manner as the instruction field register. A CIF microinstruction loads the IB with the programmed field number contained in MB5-8 [MB8]. An RMF microinstruction transfers the contents of bits 0 through 2 [bit 0 only] of the save field register into the IB to restore the instruction field to the conditions that existed prior to a program interrupt.

Save Field Register (SF) — When a program interrupt occurs, this 6- [-2] bit register is loaded from the instruction field and data field registers. The SF is loaded during the cycle in which the program count is stored at address 0000 of the JMS instruction forced by a program interrupt request, then the instruction field and data field are cleared. An RMF microinstruction can be given immediately prior to the exit from the program interrupt subroutine to restore the instruction field and data field by transferring the contents of the SF into the instruction buffer and the data field register.

Break Field Register (BF) — This 3- [1-] bit register receives three [one] ADDRESS EXTEND signals from any I/O device using the data break facility. When the B set signal arrives from the processor, this register is loaded with the bit combination of the three inputs. During a three-cycle data break, the word count and current address cycles always occur in field 0.

Extended Address Signal Generator — When the PDP-8/I or PDP-8/L core memory capacity is extended, the standard memory is designated as field 0. The extended address signal generator circuit produces the extend address field 0 signal when data field 0 is selected, or instruction field 0 is selected. This circuit will produce the other seven [one] possible extend address field signals determined by the bit or bit combination applied to its input.

Accumulator Transfer Gating — This gating allows the contents of the save field register, instruction field register, or the data field register to be strobed into the accumulator. Transfer of information in this manner is accomplished by circuits which sample the contents of registers and supply positive pulses to the AC upon receipt of IOT command pulses. During an RIB microinstruction, bits 6 through 11 [bits 8 and 11] of the AC are set by the contents of the save field register. During an RIF microinstruction, bits 6 through 8 [bit 8] of the AC are set by the contents of the instruction field register. During an RDF microinstruction, bits 6 through 8 [bit 8] of the AC are set by the contents of the data field register.

Device Selector — Bits 3 through 5 of the IOT instruction are decoded to produce the IOT command pulses for the memory extension control. Bits 6 through 8 of the instruction are not used for device selection since they specify a field number in some commands. Therefore, the select code for this device selector is designated as 2X.

The Memory Module Option, Type MM8/I, adds memory and read/write switches for each 8K of additional memory. One or two 4K memory modules may be specified. Each 4K memory consists of a core array, address selection circuits and inhibit selection circuits which are identical with those housed in the PDP-8/I.

Instructions

The instructions for the Type MC8/I option do not use the IOP generator. The instructions for the MC8/L, however, do use the IOP generator. Use of these instructions extends the IOT instruction list to include the following:

Change to Data Field N (CDF)

Octal Code: 62N1
Event Time: Not applicable
Indicators: Iot, Fetch [IR = 6, F]
Execution Time: 1.5 [4.25] μs
Operation: The data field register is loaded with the program-selected field number (N = 0 to 7 [0 to 1].) All subsequent memory requests for operands are automatically switched to that data field until the data field number is changed by a new CDF command, or during a program interrupt.
Symbol: $MB_{6-8} \rightarrow DF$ [$MB_8 \rightarrow DF$]

Change Instruction Field (CIF)

Octal Code: 62N2
Event Time: Not applicable
Indicators: Iot, Fetch [IR = 6, F]
Execution Time: 1.5 [4.25] μs
Operation: the instruction buffer register is loaded with the program-selected field number (N = 0 to 7 [0 or 1]). The next JMP or JMS instruction causes the new field to be entered at the conclusion of that instruction.
Symbol: $MB_{6-8} \rightarrow IB$ [$MB_8 \rightarrow IB$]

Read Data Field (RDF)

Octal Code: 6214
Event Time: Not applicable
Indicators: Iot, Fetch [IR =, F]
Execution Time: 1.5 [4.25] μs
Operation: The contents of the data field register are ORed into bits 6, 7, 8 [bit 8] of the AC. All other bits of the AC are unaffected.
Symbol: $DF \vee AC_{6-8} \rightarrow AC_{6-8}$ [$DF \vee AC_8 \rightarrow AC_8$]

Read Instruction Field (RIF)

Octal Code: 6224
Event Time: Not applicable
Indicators: Iot, Fetch [IR = 6, F]
Execution Time: 1.5 [4.25] μs
Operation: The contents of the instruction field register are ORed into bits 6, 7, 8 [bit 8] of the AC. All other bits of the AC are unaffected.
Symbol: IF V $AC_{6-8} \rightarrow AC_{6-8}$ [IF V $AC_8 \rightarrow AC_8$]

Read Interrupt Buffer (RIB)

Octal Code: 6234
Event Time: Not applicable
Indicators: Iot, Fetch [IR = 6, F]
Execution Time: 1.5 [4.25] μs
Operation: The contents of the save field register (which is loaded from the instruction and data fields during a PI) are ORed into bits 6 through 8 [bit 8 only] and 9 through 11 [bit 11 only] of the AC, respectively. Thus AC_{6-8} [AC_8] contains the instruction field and AC_{9-11} [AC_{11}] contains the data field which were in use before the last PI.
Symbol: SF_{0-2} V $AC_{6-8} \rightarrow AC_{6-8}$ [SF_0 V $AC_8 \rightarrow AC_8$]
SF_{3-5} V $AC_{9-11} \rightarrow AC_{9-11}$ [SF_1 V $AC_{11} \rightarrow AC_{11}$]

Restore Memory Field (RMF)

Octal Code: 6244
Event Time: Not applicable
Indicators: Iot, Fetch [IR = 6, F]
Execution Time: 1.5 [4.25] μs
Operation: This command is used upon exit from the program interrupt subroutine in another field. The data and instruction fields that were in use at the time of the interrupt, are restored by transferring the contents of the save field register into the instruction buffer and data field registers.
Symbol: $SF_{0-2} \rightarrow IB$ [$SF_0 \rightarrow IB$]
$SF_{3-5} \rightarrow DF$ [$SF_1 \rightarrow DF$]

Programming

Instructions and data are accessed from the currently assigned instruction and data fields, where instructions and data may be stored in the same or different memory fields. When indirect memory references are executed, the operand address refers first to the instruction field to obtain an effective address which, in turn, refers to a location in the currently assigned data field. All instructions and operands are obtained from the field designated by the contents of the instruction field register, except for indirectly addressed operands which are specified by the contents of the data field register. In other words, the DF is effective only in the execute cycle that is directly preceded by the defer cycle of a memory reference instruction, as follows:

Indirect (Bit 3)	Page or Z Bit (Bit 0)	Field In IF	Field In DF	Effective Address
0	0	m	n	The operand is in page 0 of field m at the page address specified by bits 5 through 11.
0	1	m	n	The operand is in the current page of field m at the page address specified by bits 5 through 11.
1	0	m	n	The absolute address of the operand in field n is taken from the contents of field m located in page 0 designated by bits 5 through 11.
1	1	m	n	The absolute address of the operand in field n is taken from the contents of field m located in the current page, designated by bits 5 through 11.

Each field of extended memory contains eight auto-index registers in addresses 10 through 17. For example, assume that a program in field 2 is running (IF = 2) and using operands in field 1 (DF = 1) when the instruction TAD I 10 is fetched. The defer cycle is entered (bit 3 = 1) and the contents of location 10 in field 2 are read, incremented, and rewritten. If address 10 in field 2 originally contained 4321, it now contains 4322. In the execute cycle the operand is fetched from location 4322 of field 1. Program control is transferred between memory fields by the CIF instruction. The instruction does not change the instruction field directly, since this would make it impossible to execute the next sequential instruction; instead it loads the new instruction field into the IB for automatic transfer into the IF when either a JMP or JMS instruction is executed. The DF is unaffected by the JMP and JMS instructions.

The 12-bit program counter is set in the normal manner and, since the IF is an extension on the most significant end of the PC, the program sequence resumes in the new memory field following a JMP or JMS. Entry into a program interrupt is inhibited after the CIF instruction until a JMP or JMS is executed.

To call a subroutine that is out of the current field, the data field register is set to indicate the field of the calling JMS, which establishes the location of the operands as well as the identity of the return field. The instruction field is set to the field of the starting address of the subroutine. The following sequence returns program control to the main program from a subroutine that is out of the current field.

```
/PROGRAM OPERATIONS IN MEMORY FIELD 2
/INSTRUCTION FIELD = 2; DATA FIELD = 2
/CALL A SUBROUTINE IN MEMORY FIELD 1
/INDICATE CALLING FIELD LOCATION BY THE CONTENTS OF THE DATA FIELD

            CIF         10          /CHANGE TO INSTRUCTION
                                    /FIELD 1 = 6212
            JMS         1 SUBRP     /SUBRP = ENTRY ADDRESS
            CDF         20          /RESTORE DATA FIELD
              .
              .
              .
SUBRP,      SUBR                    /POINTER
/CALLED SUBROUTINE, LOCATED IN FIELD 1
            0                       /RETURN ADDRESS STORED HERE
SUBRP,      CLA
            RDF                     /READ DATA FIELD INTO AC
            TAD RETURN              /CONTENTS OF THE AC = 6202 + DATA
                                    /FIELD BITS
            DCA EXIT                /STORE INSTRUCTION SUBROUTINE
              .                     /NOW CHANGE DATA FIELD IF
                                    DESIRED
              .
              .
              .
EXIT,       0                       /A CIF INSTRUCTION
            JMP 1 SUBR              /RETURN
RETURN,     CIF                     /USED TO CALCULATE EXIT
                                    INSTRUCTION
```

When a program interrupt occurs, the current instruction and data field numbers are automatically stored in the 6- [2-] bit save field register, then the IF and DF are cleared. The 12-bit program count is stored in location 0000 of field 0 and program control advances to location 0001 of field 0. At the end of the program interrupt subroutine, the RMF instruction restores the IF and DF from the contents of the SF. The following instruction sequence at the end of the program interrupt subroutine continues the interrupted program after the interrupt has been processed:

```
              .             /RESTORE MQ IF REQUIRED [PDP-8/I only]
              .
              .
              .             /RESTORE L IF REQUIRED
              .
              .
              .
              .
            CLA
            TAD AC          /RESTORE AC
            RMF             /LOAD IB FROM SF
            ION             /TURN ON INTERRUPT SYSTEM
            JMP  I 0        /RESTORE PC WITH CONTENTS OF
                            /LOCATION 0 AND LOAD IF FROM IB
```

An I/O device, using the computer data break facility, supplies a 12-bit address to the MA and a 3-bit [1-bit] address extension to the memory extension

control. The address extension is received by a break field decoder which selects the memory field used for the data break. Word count and current address are located in field 0 in a three-cycle data break.

MEMORY PARITY MP8/I [MP8/L]

(The logic for this option is housed within the central processor.) Data transmission checking of each word written into and read from core memory is provided by this option. The option replaces the 12-bit core memory with a 13-bit system (driving, inhibiting, and sensing circuits as well as a core array constructed of 13 planes) and includes a parity generator and a parity checking circuit. The parity generator produces the 13th bit for each 12-bit data word written in core memory so that the entire word contains an odd number of binary 1s (odd parity). The parity checking circuit monitors each word read from core memory to assure that the odd parity is maintained. If a word read contains an even number of 1s a transmission error is indicated by initiating a parity error flag. This flag is connected to the program interrupt system to initiate a program interrupt subroutine. This routine sequentially checks all equipment error flags to determine the option causing the interrupt and initiates an appropriate service and returns to the main program, or provides a suitable error printout and halts programmed operations. Upon determining that a memory parity error has occurred, the program interrupt subroutine can repeat the main program step that caused the error to check the reliability of the error condition, can perform a simple write/read/check routine at the error address, or can determine the status of the machine when the error was detected and re-establish or print out these conditions and halt operations.

Instructions

Two instructions are associated with the memory parity options. They are:

Skip on No Memory Parity Error (SMP)

Octal Code: 6101
Event Time: 1
Indicator: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The memory parity error flag is sensed and, if it contains a 0 (signifying no error has been detected), the PC is incremented so that the next successive instruction is skipped.
Symbol: If Memory Parity Error Flag = 0, then PC + 1 → PC

Clear Memory Parity Error Flag (CMP)

Octal Code: 6104
Event Time: 3
Indicator: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The memory parity error flag is cleared.
Symbol: 0 → Memory Parity Error Flag

Programming

Both instructions for this option are used in the program interrupt subroutine and in diagnostic maintenance programs. The SMP command is used as a programmed check for memory parity error. In the program interrupt subroutine this command can be followed by a jump to a portion of the routine that services the memory parity option as described previously. The CMP command is used to initialize the memory parity option in preparation for normal programmed operation of the computer.

SECTION 6-2 POWER FAILURE DETECTION AND RESTART KP8/I [KP8/L]

(The logic for this option is housed within the central processor.) This prewired option protects an operating program in the event of failure of the source of computer primary power. If a power failure occurs, this option causes a program interrupt and enables continued operation for 1 millisecond, allowing the interrupt routine to detect the low power condition as initiator of the interrupt, and to store both the contents of active registers (AC, L, MQ, etc.), and the program count in known core memory locations. When power is restored, the power low flag clears and a routine beginning in address 0000 starts automatically. This routine restores the contents of the active registers and program counter to the conditions that existed when the interrupt occurred, then continues the interrupted program.

The power failure option consists of three logic circuits:

A power interrupt circuit monitors the status of the computer power supply, and sets a power low flag when power is interrupted (due to a power failure or to the operation of the POWER lock on the operator's console). This flag causes a program interrupt when an interruption in computer power is detected.

A shut-down sequence circuit assures that when a power interrupt occurs, the computer logic circuits continue operation for 1 ms to allow a program subroutine to store the contents of the active registers. The computer operation is halted at the end of the 1-ms interval if it is still running. A restart circuit clears the power low flag and restarts the program when power conditions are suitable for computer operation. A manual RESTART switch located on the power failure module enables or disables the automatic restart operation. With this switch in the ON (down) position, the option clears the MA immediately and produces a signal to simulate operation of the START key on the operator's console 200 ms after power conditions are satisfactory. The MA is cleared so that operation restarts by executing the instruction in address 0000. This instruction must be a JMP to the starting address of the subroutine which restores the contents of the active registers and the program counter to the conditions that existed prior to the power low interrupt. The 200-ms delay assures that slow mechanical devices, such as Teletype equipment, have completed any previous operation before the program is resumed. Simulation of the manual START function causes the processor to generate a power clear pulse to clear internal controls and I/O device registers. With the RESTART switch in the OFF (up) position, the power low flag is cleared but the program must be started manually, possibly after resetting peripheral equipment or by starting the interrupted program from the beginning. The shut-down circuitry is unaffected by the switch.

A skip circuit provides programmed sensing of the condition of the power low flag by adding the following instruction to the computer repertoire:

Skip on Power Low (SPL)

Octal Code: 6102

Event Time: 2

Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25 μs

Operation: The content of the power low flag is sampled, and if it contains a 1 (indicating a power failure has been detected) the contents of the PC are incremented by one so the next sequential instruction is skipped.

Symbol: If Power Low flag = 1, then PC + 1 → PC

Since the time that computer operation can be extended after a power failure is limited to 1 ms, the condition of the power low flag should be the first status check made by the program interrupt subroutine. The interrupt subroutine, starting with the SPL microinstruction and including the power fail program sequence can be executed in 30 μs on a basic PDP-8/I with an EAE [27 μs on a PDP-8/L]. The power fail program sequence stores the contents of the active registers and program count in designated core memory locations, then relocates the calling instruction of the power restore subroutine to address 0000, as follows:

Address	Instruction	Remarks
0000	—	/STORAGE FOR PC AFTER PROGRAM INTERRUPT
0001	JMP FLAGS	/INSTRUCTION EXECUTED AFTER PROGRAM /INTERRUPT
FLAGS,	SPL	/SKIP IF POWER LOW FLAG = 1
	JMP OTHER	/INTERRUPT NOT CAUSED BY POWER LOW, /CHECK OTHER FLAGS
	DCA AC	/INTERRUPT WAS CAUSED BY POWER LOW, /SAVE AC
	RAR	/GET LINK
	DCA LINK	/SAVE LINK
	MQA	/GET MQ [PDP-8/I only]
	DCA MQ	/SAVE MQ [PDP-8/I only]
	TAD 0000	/GET PC
	DCA PC	/SAVE PC
	TAD RESTRT	/GET RESTART LOCATION
	DCA 0000	/DEPOSIT RESTART LOCATION IN 0000
	HLT	
RESTRT,	JMP ABCD	/ABCD IS LOCATION OF RESTART ROUTINE

Automatic program restart begins by executing the instruction stored in address 0000 by the power fail routine. This instruction must be a JMP (either direct or indirect). The power restore subroutine restores the contents of the active registers, enables the program interrupt facility, and continues the interrupted program from the point at which it was interrupted, as follows:

Address	Instruction	Remarks
0000	JMP ABCD	
ABCD,	TAD MQ	/GET MQ [PDP-8/I only]
	MQL	/RESTORE MQ [PDP-8/I only]
	TAD LINK	/GET LINK
	CLL RAL	/RESTORE LINK
	TAD AC	/RESTORE AC
	ION	/TURN ON INTERRUPT
	JMP I PC	/RETURN TO INTERRUPTED PROGRAM

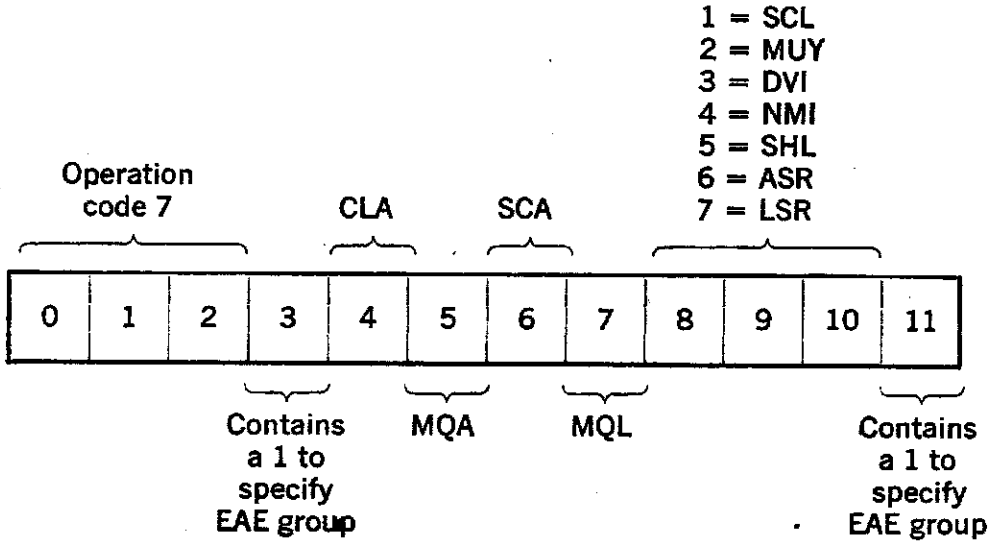
SECTION 6-3 EXTENDED ARITHMETIC ELEMENT (EAE) KE8/I

(The logic for this option is located within the PDP-8/I central processor.)
This option consists of circuits that perform parallel arithmetic operations on positive binary numbers. A 12-bit multiplier quotient register (MQ), a 5-bit

step counter (SC), and various shifting and control logic elements constitute the option. The AC and MB are used in conjunction with these logic elements to perform arithmetic operations. With the addition of this option to a PDP-8/I system, indicators on the operator's console for the content of each bit of the MQ are activated and a class of instructions is added to the Group 2 Operate instruction list.

INSTRUCTIONS

The Extended Arithmetic Element (EAE) microinstructions are specified by an operate instruction (operation code 7) in which bits 3 and 11 contain binary 1s. Being augmented instructions, the EAE commands are microprogrammed and can be combined with each other to perform non-conflicting logical operations. Format and bit assignments of the EAE commands are indicated in Figure 6-1.



Logical Sequence:
1 — CLA
2 — MQA, MQL, SCA
3 (Bits 8 through 10 = 1) — SCL
3 (Bits 8 through 10 = 2) — MUY
3 (Bits 8 through 10 = 3) — DVI
3 (Bits 8 through 10 = 4) — NMI (cannot be combined with MQL)
3 (Bits 8 through 10 = 5) — SHL
3 (Bits 8 through 10 = 6) — ASR
3 (Bits 8 through 10 = 7) — LSR

Figure 6-1. EAE Microinstruction Bit Assignments

Multiply (MUY)

Octal Code: 7405
Sequence: 3
Indicators: Opr, Fetch, Pause
Execution Time: 4.8 to 7.2 μs
Operation: The number held in the MQ is multiplied by the number held in core memory location PC + 1 (or the next successive core memory location after the MUY instruction), and the PC is made equal to PC + 2. At the conclusion of this instruction, the link contains a 0, the most significant 12

bits of the product are contained in the AC, and the least significant 12 bits of the product are contained in the MQ.

Symbol: $Y \times MQ \rightarrow AC, MQ$
$0 \rightarrow L$
$PC + 2 \rightarrow PC$

Divide (DVI)

Octal Code: 7407
Sequence: 3
Indicators: Opr, Fetch, Pause
Execution Time: 5.2 to 7.8 μs
Operation: The 24-bit dividend held in the AC (most significant 12 bits) and the MQ (least significant 12 bits) is divided by the divisor held in core memory location PC + 1 (or the next successive core memory location following the DVI instruction). At the conclusion of this instruction, the quotient is held in the MQ, the remainder is in the AC, and the L contains a 0. If the L contains a 1, divide overflow occurred and the operation was concluded after the first cycle of the division.

Symbol: $AC, MQ \div Y \rightarrow MQ$
$PC + 2 \rightarrow PC$

Normalize (NMI)

Octal Code: 7411
Sequence: 3
Indicators: Opr, Fetch, Pause
Execution Time: 1.5 μs + 0.25 μs for each shift
Operation: This instruction is used as part of the conversion of a binary number to a fraction and an exponent for use in floating-point arithmetic. Regardless of sign, the combined contents of the AC and the MQ are shifted left by this one instruction until the content of AC0 is not equal to the content of AC1, or until 6000 0000 is contained in the combined AC and MQ, to form the fraction. Zeros are shifted into vacated MQ11 positions for each shift. At the conclusion of this instruction, the step counter contains a number equal to the number of shifts performed, which can be loaded into the AC by an SCA instruction to form the exponent. The content of L is lost.

Symbol: $ACj \rightarrow MCj - 1$
$AC_0 \rightarrow L$
$MQ_0 \rightarrow AC_{11}$
$MQj \rightarrow MQj - 1$
$0 \rightarrow MQ_{11}$ until $AC_0 \pm AC_1$ or until AC MQ = 6000 000

Shift Left (SHL)

Octal Code: 7413
Sequence: 3
Indicators: Opr, Fetch, Pause
Execution Time: 3.0 μs + 0.25 μs for each shift
Operation: This instruction is used for scaling by shifting the combined contents of the AC and MQ to the left one position more than the number of positions indicated by the contents of core memory at address PC + 1 (or the next successive core memory location following the SHL instruction). During the shifting, zeros are shifted into vacated MQ11 positions. The L, AC, and MQ are treated as one long register during this operation. Bits shifted out of AC0 enter the L, and bits shifted out of the L are lost.

Symbol: Shift Y + 1 positions as follows:

$AC_j \rightarrow AC_j - 1$
$AC_0 \rightarrow L$
$MQ_0 \rightarrow AC_{11}$
$MQ_j \rightarrow MQ_j - 1$
$0 \rightarrow MQ_{11}$
$PC + 2 \rightarrow PC$

Arithmetic Shift Right (ASR)

Octal Code: 7415
Sequence: 3
Indicators: Opr, Fetch, Pause
Execution Time: 3.0 μs + 0.25 μs for each shift
Operation: This instruction is used for scaling and treats the AC and MQ as one long register. The combined contents of the AC and the MQ are shifted right one position more than the number contained in memory location PC + 1 (or the next successive core memory location following the ASR instruction). The sign bit, contained in AC0, enters vacated positions, the sign bit is preserved in the link, information shifted out of MQ11 is lost, and the L is set to correspond to the sign bit during this operation.

Symbol: Shift Y + 1 positions as follows:

$AC_0 \rightarrow L$
$AC_0 \rightarrow AC_0$
$AC_j \rightarrow AC_j + 1$
$AC_{11} \rightarrow MQ_0$
$MQ_j \rightarrow MQ_j + 1$
$PC + 2 \rightarrow PC$

Logical Shift Right (LSR)

Octal Code: 7417
Sequence: 3
Indicators: Opr, Fetch, Pause
Execution Time: 3.0 μs + 0.25 μs for each shift
Operation: This instruction is used for scaling and treats the AC and MQ as one long register. The combined contents of the AC and MQ are shifted right one position more than the number contained in memory location PC + 1 (or the next successive core memory location following the LSR instruction). This instruction is similar to the ASR instruction except that zeros instead of the sign bit enter vacated positions. Information shifted out of MQ11 is lost and the L is cleared during this operation.

Symbol: Shift Y + 1 positions as follows:

$0 \rightarrow L$
$0 \rightarrow AC_0$
$AC_j \rightarrow AC_j + 1$
$AC_{11} \rightarrow MQ_0$
$MQ_j \rightarrow MQ_j + 1$
$PC + 2 \rightarrow PC$

Load Multiplier Quotient (MQL)

Octal Code: 7421
Sequence: 2
Indicators: Opr, Fetch
Execution Time: 1.5 μs
Operation: This instruction clears the MQ, loads the contents of the AC into the MQ, then clears the AC. This operation is essential to initializing any mul-

tiply or divide routine and can be combined with a MUY or DVI, but not with an NMI instruction.
Symbol: AC → MQ
0 → AC

Step Counter Load from Memory (SCL)

Octal Code: 7403
Sequence: 3
Indicators: Opr, Fetch, Execute
Execution Time: 3.0 μs
Operation: Loads the complement of bits 7 through 11 of the word into memory following the instruction into the step counter.
Symbol: $\overline{MB_{7-11}}$ → SC
PC + 2 → PC

Step Counter Load into Accumulator (SCA)

Octal Code: 7441
Sequence: 2
Indicators: Opr, Fetch
Execution Time: 1.5 μs
Operation: The contents of the step counter are transferred into the AC. This instruction is used following an NMI instruction to establish the exponent of a normalized number to be used in floating point arithmetic. The AC should be cleared prior to issuing this instruction; or the CLA instruction can be combined with the SCA to clear the AC and then effect the transfer.
Symbol: SC V AC → AC

Multiplier Quotient Load into Accumulator (MQA)

Octal Code: 7501
Sequence: 2
Indicators: Opr, Fetch
Execution Time: 1.5 μs
Operation: The contents of the MQ are transferred into the AC. This instruction is given to load the 12 least significant bits of the product into the AC following a multiplication or to load the quotient into the AC following a division. The AC should be cleared prior to issuing this instruction, or the CLA instruction can be combined with the MQA to clear the AC and then effect the transfer.
Symbol: MQ V AC → AC

Clear Accumulator (CLA)

Octal Code: 7601
Sequence: 1
Execution Time: 1.5 μs
Indicators: Opr, Fetch
Operation: The AC is cleared, allowing this instruction to be combined with the other EAE instructions that load the AC (such as SCA and MQA).
Symbol: 0 → AC

PROGRAMMING

Multiplication

Multiplication is performed as follows:

a. Load the AC with the multiplier using the TAD instruction.
b. Transfer the contents of the AC into the MQ using the MQL command.
c. Give the MUY command.

(Note: Steps b and c can be combined into one instruction.)

The contents of the MQ are then multiplied by the contents of the next successive core memory address (PC + 1). At the conclusion of the multiplication the most significant 12 bits of the product are held in the AC and the least significant bits are held in the MQ. This operation takes a maximum of 7.2 μs; at the end of this time the next instruction is executed.

The following multiplication program examples indicate the operation of the KE8/I option in closed subroutines (routines which are incorporated into larger routines and are not written in a form which allows them to be called as a normal mathematical subroutine).

Multiplication of 12-Bit Unsigned Numbers — Enter with a 12-bit multiplicand in AC and a 12-bit multiplier in core memory. Exit with high order half of product in a core memory location labeled HIGH, and with low order half of product in the AC. Program time is from 9.3 to 11.7 μs.

```
        MQL MUY         /LOAD MQ WITH MULTIPLICAND, INITIATE
                        /MULTIPLICATION
        MLTPLR          /MULTIPLIER
        DCA HIGH        /STORE HIGH ORDER PRODUCT
        MQA             /LOAD AC WITH LOW ORDER PRODUCT
```

Multiplication of 12-bit Signed Number, 24-bit Signed Product — Enter with a 12-bit multiplicand in AC and a 12-bit multiplier in core memory. Exit with signed 24-bit product in core memory locations designated HIGH and LOW. Program time is from 36.3 to 54.2 μs.

```
            CLL
            SPA                     /MULTIPLICAND POSITIVE?
            CMA CML     IAC         /NO. FORM TWO'S COMPLEMENT
            MQL                     /LOAD MULTIPLICAND INTO MQ
            TAD         MLTPLR
            SPA                     /MULTIPLIER POSITIVE?
            CMA CML     IAC         /NO. FORM TWO'S COMPLEMENT
            DCA         MLTPLR
            RAL
            DCA         SIGN        /SAVE LINK AS SIGN INDICATOR
            MUY                     /MULTIPLY
MLTPLR,0                            /MULTIPLIER
            DCA         HIGH
            TAD         SIGN
            RAR                     /LOAD LINK WITH SIGN INDICATOR
            MQA
            SNL                     /IS PRODUCT NEGATIVE?
            JMP         LAST        /NO. MULT DONE, EXIT
            CLL CMA     IAC         /YES
            DCA         LOW         /COMPLEMENT RESULT
            TAD         HIGH
            CMA
            SZL                     /LINK USED TO COUPLE CARRY
                                    /FROM BIT 12 TO BIT 11
            IAC                     /OF DOUBLE-LENGTH PRODUCT
            DCA         HIGH
            SKP
LAST,       DCA         LOW
```

Division

Division is performed as follows:

a. Load the 12 least significant bits of the dividend into the AC using the TAD instruction, then transfer the contents of the AC into the MQ using the MQL command.

b. Load the 12 most significant bits of the dividend into the AC.

c. Give the DVI command.

The 24-bit dividend contained in the AC and MQ is then divided by the 12-bit divisor contained in the next successive core memory address (PC + 1). This operation takes a maximum of 7.8 μs and is concluded with the 12-bit quotient held in the MQ, the 12-bit remainder in the AC, and the link holding a 0 if divide overflow did not occur. To prevent divide overflow, the divisor in the core memory must be greater than the 12 bits of the dividend held in the AC.

When divide overflow occurs, the link is set and the division is concluded after only one cycle. Therefore, the instruction following the divisor in core memory should be an SZL microinstruction to test for overflow. The instruction following the SZL can be a jump to a subroutine that services the overflow. This subroutine can either cause the program to type out an error indication, rescale the divisor or the dividend, or perform other mathematical corrections and repeat the divide routine.

The following division program examples indicate the operation of the Type KE8/I option in closed subroutines.

Division of 12-Bit Unsigned Numbers — Enter with a 12-bit unsigned dividend in the AC and a 12-bit unsigned divisor in core memory. Exit with remainder in core memory location labeled REMAIN and with the quotient in the AC. Program time is a maximum of 15.3 μs.

```
CCL
MQL   DVI          /LOAD MQ, INITIATE DIVISION
DIVSOR             /DIVISOR
SZL                /OVERFLOW?
JMP                /YES, EXT
DCA REMAIN
MQA                /LOAD AC WITH QUOTIENT
```

Division of 12-Bit Signed Numbers — Enter with a 12-bit signed dividend in the AC and a 12-bit signed divisor in core memory. Exit with unsigned remainder in core memory location REMAIN and a 12-bit signed quotient in the AC. Program time is a maximum of 39.3 μs.

```
CLL
SPA                /DIVIDEND POSITIVE?
CMA CML IAC        /NO
MQL
TAD DIVSOR
SPA                /DIVISOR POSITIVE?
CMA CML IAC        /NO
DCA DIVSOR
SNL                /QUOTIENT NEGATIVE?
CMA                /NO
CLL
DCA SIGN           /SET SIGN INDICATOR
DVI
DIVISOR            /DIVISOR
SZL                /OVERFLOW
JMP                /EXIT ON OVERFLOW
DCA REMAIN
MQA
ISZ SIGN
CMA IAC
```

SECTION 6-4 REAL TIME CLOCK OPTIONS KW8/IA-F [KW8/LA-F]

The KW8/I [KW8/L] is a prewired option for the PDP-8/I [PDP-8/L] and provides a method of accurately measuring time intervals. There are six KW8 configurations. The first three basic clocks are referred to as fixed interval clocks and consist of a frequency source and a clock control. The next three clocks are referred to as programmable-interval clocks and consist of a frequency source, clock control, and a 12-bit clock counter .The clock counter provides preset and readout capability with associated controls. Addition of this module allows hardware interval counting, reducing program instruction time and, therefore, more efficient use is made of computer time.

For the fixed interval clocks (Types KW8/IA, KW8/IB, and KW8/IC [KW8/LA, KW8/LB, and KW8/LC]), the clock control module decodes various IOT instructions to allow a SKIP and/or INTERRUPT after a frequency source has set the FLAG. Counting is performed by incrementing a memory location which has been loaded with 2s complement of the desired count. Due to finite instruction time, the fixed interval clock has a program upper frequency limit of approximately 66.6 kHz for a 12-bit counter and approximately 50 kHz for a 24-bit counter.

For the programmable interval clocks (Types KW8/ID, KW8/IE, and KW8/IF [KW8/LD, KW8/LE, and KW8/LF]), the counter module produces an OVERFLOW signal to the clock control module that sets the FLAG for a SKIP and/or INTERRUPT. The number of clock pulses before OVERFLOW is determined by a number in the accumulator containing the desired number of counts. This number is complemented and incremented by the clock counter logic as it is loaded into the count register so that when the desired number of counts minus one have occurred, OVERFLOW is generated. This signal enables the FLAG within the clock control and permits the next clock pulse to set the FLAG. During counting, the counter contents are transferred to a buffer register on the rising edge of the clock pulse and the count is incremented by one on the falling edge. The output buffer can be read into the accumulator with the appropriate IOT instruction without disturbing the contents of the count register. The count read is the last counter value, if the instruction occurs during the clock pulse. Due to the propagation delay of the counter, the programmable interval clock has an upper frequency limit of 1 MHz.

The basic accuracy of the real time clock is that of the frequency source. The overall accuracy also involves the frequency source and its relationship to the instruction time of the service loop, and the number of clock pulses counted for a time interval. The period of the frequency source should be greater than the time to service the loop including the SKIP and IOT instructions, to prevent occurrence of more than one pulse count during the service routine.

The crystal and line frequency sources are asynchronous to the program execution. This allows the variation of the first clock pulse occurrence to be as long as one full period. The variable frequency source has less accuracy in its first period because the initial pulse output from this clock is "masked" by the clock control, therefore, the first pulse allowed is approximately that of a full period. Because of this, the overall accuracy of the clocks using the variable frequency source may exceed that of the clocks using the crystal and line frequency sources. Table 6-1 generally describes each of the six KW8 options and how they differ.

Table 6-1 Real Time Clock Options

Type	Description
KW8/IA [KW8/LA]	Real Time Clock, Fixed Interval, Line Frequency Timing Source
KW8/IB [KW8/LB]	Real Time Clock, Fixed Interval, Variable Frequency Timing Source
KW8/IC [KW8/LC]	Real Time Clock, Fixed Interval, Crystal Frequency Timing Source
KW8/ID [KW8/LD]	Real Time Clock, Programmable Interval, Line Frequency Timing Source
KW8/IE [KW8/LE]	Real Time Clock, Programmable Interval, Variable Frequency Timing Source
KW8/IF [KW8/LF]	Real Time Clock, Programmable Interval, Crystal Frequency Timing Source

INSTRUCTION DESCRIPTIONS

Tables 6-2 and 6-3 list the instructions for the basic clock and the preset and read-out clock, respectively, and contains the descriptions of their operation.

Table 6-2 Basic Clock Instruction Descriptions for clocks KW8/IA, KW8/IB, KW8/IC, [KW8/LA, KW8/LB, KW8/LC]

Mnemonic	Octal Code	Description
CCFF	6132	The flag, flag buffer, clock enable, and interrupt enable flip-flops are cleared. This disables the real-time clock.
CCEC	6136	All clock control flip-flops are first cleared, then the clock enable flip-flop is set. For the variable frequency clock, the frequency source is enabled synchronously with program operation. With all clocks the data input to the flag is enabled after IOP2 time. This represents an 800-ns mask, after the clock is enabled.
CECI	6137	All clock control flip-flops are cleared, then the clock enable, and interrupt enable flip-flops are set. The clock enable flip-flop is described with the CCEC instruction. The interrupt enable flip-flop allows an IO BUS IN INT signal when the flag is set.
CSCF	6133	When the flag flip-flop has been set by a clock pulse, the flag buffer flip-flop is set to a one. Upon execution of this instruction an IO BUS IN SKIP is generated if the flag is set. The content of the PC is incremented by one so the next sequential instruction is skipped. The flag flip-flop is then cleared. If the flag flip-flop has not been set, no skip is generated nor is the flag flip-flop cleared.

Table 6-3 Preset and Read-out Instruction Descriptions for clocks KW8/ID, KW8/IE, KW8/IF [KW8/LD, KW8/LE, KW8/LF]

Mnemonic	Octal Code	Description
CCFF	6132	The flag, flag buffer, clock enable, and interrupt enable flip-flops are cleared. This disables the real-time clock. In addition, the OVERFLOW gating is disabled.
CECL	6136	The operations are the same as that of the CCEC instruction for the basic clock, except that the data input to the flag is not enabled until **both** CLOCK ENABLE and OVERFLOW are set. All counter bits are set at IOP2, and then cleared according to the accumulator prior to IOP4. At IOP4 the contents of the counter (the one's complement of the accumulator) are transferred to the output buffer. At the end of IOP4, the counter is incremented by one to provide the two's complement of the accumulator.
CEIL	6137	Operations are the same as those described in the CECI instruction for the basic clock, except that the counter is loaded according to the CECL instruction.
CRCA	6134	The output buffer is gated to the I/O BUS during IOP4, and a CLK AC CLR signal generated. This register contains the last count in the count register. The transfer from the count register is synchronized with this instruction so that a transfer that would occur during this instruction is not made.
CSCF	6133	When the flag flip-flop has been set by a clock pulse, the flag buffer flip-flop is set to a one. Upon execution of this instruction an IO BUS IN SKIP is generated if the flag is set. The content of the PC is incremented by one so the next sequential instruction is skipped. The flag flip-flop is then cleared. If the flag flip-flop has not been set, no skip is generated nor is the flag flip-flop cleared.

PROGRAMMING EXAMPLES

The following examples further clarify the instruction set.

Counting program for the basic KW8/IA, KW8/IB, and KW8/IC [KW8/LA, KW8/LB, and KW8/LC] clocks:

```
        CCEC          /ENABLE CLOCK
        CSCF          /TEST CLOCK FLAG
        JMP .-1
        ISZ B         /COUNT CLOCK PULSES UNTIL B OVERFLOWS
        JMP .-3
          .
          .
          .
B,      XXXX          /TWO'S COMPLEMENT OF DESIRED COUNT
```

Counting with Preset and Read-out Clock KW8/ID, KW8/IE, and KW8/IF [KW8/LD, KW8/LE, and KW8/LF], using interrupt:

```
      CLA
      TAD B
      CEIL          /LOAD CLOCK COUNTER, ENABLE CLOCK,
                     CONNECT CLOCK
                    /TO INTERRUPT
        .
        .
        .
B,    YYYY          /DESIRED COUNT (NOT COMPLEMENT)
```

The preset count register with the interrupt facility enabled allows a more efficient use of machine time. While counting out an interval of time, the machine can process other programs instead of looping.

Single count with KW8/I [KW8/L]:

By using the Variable Clock (KW8/IB [KW8/LB]), one clock cycle can be counted accurately. This improvement of first cycle accuracy can be used for multiple counts either in a fixed interval clock or with the preset and read-out counter.

```
      CCEC          /OR CECL WITH AC = 1 IF "E" CLOCK USED
      CSCF          /TEST CLOCK FLAG
      JMP .-1
        .
        .
        .
```

Measurement of an Interval:

```
      CLA           /LOAD COUNTER WITH ZERO, ENABLE CLOCK
      CCEC          /BEGIN INTERVAL
        .
        .
        .
        .
        .
        .           /END OF INTERVAL
CRCA                /READ CLOCK COUNTER
```

The initialization of the preset and read-out clock with zero at the beginning of the interval allows a direct read-out of the number of clock pulses occurring during the interval. If the interval is greater than 7777_8 pulses, a service of OVERFLOW would have to be effected.

SECTION 6-5 TIME SHARING HARDWARE MODIFICATION KT8/I

The KT8/I Time-Sharing Option is unique to the PDP-8/I computer and provides the additional logic circuits required for use by the computer when it is used in a time sharing system. Type MC8/I Memory Extension Control is a prerequisite for use of the KT8/I. The KT8/I uses three instructions to permit the time sharing system's monitor to handle user interrupts and to control the user interrupt logic circuits. These instructions are listed and described in Table 6-4.

Table 6-4 Instruction Descriptions

Mnemonic Code	Octal Code	Description
CINT	6204	Clear user interrupt. Resets the user interrupt (UINT) flip-flop to the 0 state.
SINT	6254	Skip on user interrupt. When the user interrupt (UINT) flip-flop is in the 1 state, sets the user skip flag (USF) flip-flop to the 1 state and causes the program to skip the next instruction.
CUF	6264	Clear user flag. Clears the user buffer (UB) flip-flop.
SUF	6274	Set user flag. Sets user buffer (UB) and inhibits processor interrupts until the next JMP or JMS instruction. Generation of IB → IF during the next JMP or JMS instruction transfers the state of UB to the user field (UF) flip-flop.

The KT8/I operates in two modes as denoted by the user flag (UF) flip-flop. When the UF flip-flop is in the logic 1 state, operation is in the user mode and a user program is running in the central processor. When the UF flip-flop is in the logic 0 state, operation is in the executive mode and the time sharing system's monitor is in control of the central processor. The three added instructions are used by the time sharing system's monitor only in the executive mode and are never used by a user program. If a user program attempted to use one of these instructions, execution of the instruction would be blocked and a user interrupt would result because they are IOT (octal code 6XXX) instructions. The KT8/I option adds the necessary hardware to the PDP-8/I to implement these three instructions.

In executive mode, the computer operates normally, as described in Chapter 5. When the computer is operated in user mode, operation is normal except for IOT, HLT, LAS, and OSR instructions. When one of these instructions is encountered, the KT8/I hardware inhibits the normal instruction sequence (other than rewriting the instruction in memory), and generates an interrupt at the end of the current memory cycle by setting the UINT flip-flop. The time sharing system's monitor program then analyzes the source of interrupt and takes appropriate action.

SECTION 6-6 POSITIVE INPUT/OUTPUT BUS KA8/IB

(PDP-8/I only; the PDP-8/L is available only with a positive bus.) The normal output of the PDP-8/IB computer is available on a negative bus. Since many I/O devices used require positive inputs, a KA8/IB option is available to replace the modules which provide the negative bus. The KA8/IB consists of M660 and M661 Positive Level Driver modules and M516 Positive Bus Receiver modules. When these modules are inserted, the PDP-8/I is converted to a positive-bus computer.

The positive levels produced are +0.2V (maximum) for low logic states and +3.3V (minimum) for high logic states, which allows direct TTL logic interfacing with the appropriate diode clamp protection.

CHAPTER 7
INPUT/OUTPUT EQUIPMENT INSTRUCTIONS

All notations in brackets [ ] indicate data for the PDP-8/L computer only.

Chapter 7 discusses the input/output equipment which is available for use with the PDP-8/I or PDP-8/L. Section 7-1 describes the Teletype Model ASR 33, and high-speed paper reader and punch. Section 7-2 describes the interface units which may be required for use with the input/output devices. Sections 7-3 through 7-11 describe, respectively, the card reader option, the incremental plotter options, the CRT display options, the magnetic tape options, the disk options, analog input subsystem options, digital-to-analog conversion subsystem options, communications subsystems options, and the A/D and D/A subsystem option.

SECTION 7-1 TELETYPE AND CONTROL

All notations in brackets [ ] indicate data for the PDP-8/L Computer only.

MODEL ASR 33 TELETYPE

(The control circuitry for this device is located in the central processor).
The standard Teletype Model ASR 33 (automatic send-receive) is used to type in or print out information at a rate of up to ten characters per second, or to read in or punch out perforated tape at ten characters per second. Signals transferred between the Model ASR 33 and the control logic are standard, serial, 11-unit code, Teletype signals. The signals consist of marks and spaces which correspond to idle and bias current in the Teletype, and to 0's and 1's in the Teletype control and computer. The start mark and subsequent eight character bits are one-unit-of-time duration and are followed by the stop bit, which occupies two units.

The 8-bit code used by the Model ASR 33 Teletype unit is the American Standard code for Information Interchange (ASCII) modified. To convert the ASCII code to Teletype code add 200 octal (ASCII + 200_8 = Teletype). This code is read in the normal octal form used in the computer. Bits are numbered from right to left, from 1 through 8, with bits 1 through 3 containing the least significant octal number. Figure 7-1 illustrates the context and description of the ASCII Teletype code and its associated bit content in the AC.

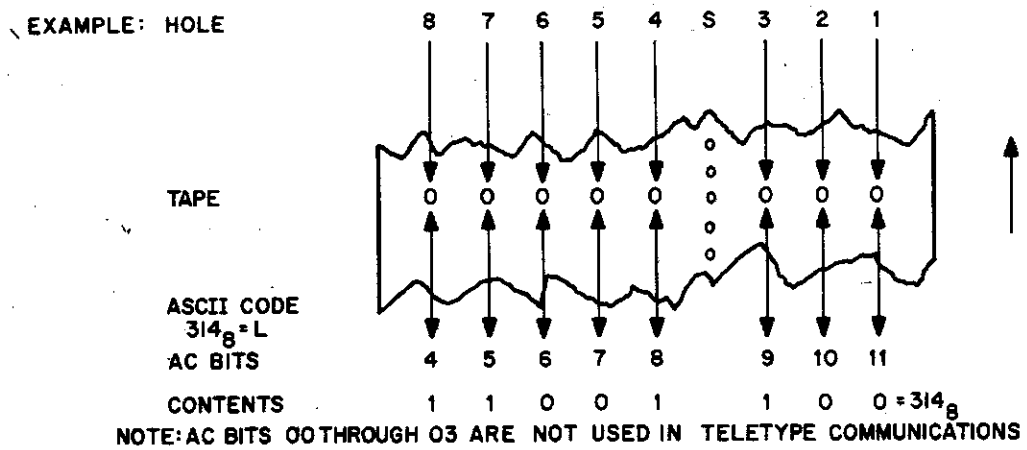


Figure 7-1 Relationship between Teletype Tape AC contents and binary and octal number being transferred.

The ASR 33 generates all assigned codes except 340 through 374 and 376 which are not assigned. Generally, codes 207, 212, 215, 240 through 337, and 377 are sufficient for Teletype operation. The ASR 33 detects all characters, but does not interpret all of the codes that it generates as commands.

The standard number of characters printed per line by the ASR 33 is 72. The sequence for proceeding to the next line is a carriage return followed by a line feed (as opposed to a line feed followed by a carriage return). Appendix C lists the Teletype character code. Punched tape format (for 264_8) is as follows:

Binary Code (Punch = 1)	Tape Channel								
	8	7	6	5	4	S	3	2	1
Octal Code	1	0	1	1	0	(Sprocket)	1	0	0

Teletype Control

Serial information read or written by the Teletype unit is assembled or disassembled by the Teletype control for parallel transfer to the AC. The control also provides the program that causes either a program interrupt or an instruction skip, depending on the availability of the Teletype and the processor.

In all programmed operation, the Teletype unit and control are considered as a Teletype in (TTI) for input intelligence from the keyboard or the perforated-tape reader, and as a Teletype out (TTO) for computer output information to be printed and/or punched on tape. Therefore, two device select codes are used; select code 03 initiates operations associated with the Keyboard/reader (TTI), and select code 04 performs operations associated with the teleprinter/punch (TTO). Parallel input and output functions are performed by corresponding IOT pulses produced by the two device selectors. Pulses produced by the IOP1 pulse trigger skip gates; pulses produced by the IOP2 pulse clear the control flags and/or the accumulator; and pulses produced by the IOP4 pulse initiate data transfers to and from the control.

Keyboard/Reader

The keyboard and tape reader control contains an 8-bit shift register (TTI) which assembles and holds the code for the last character struck on the keyboard or read from the tape. Teletype characters from the keyboard/reader are received serially by register TTI. The teletype character code is loaded into the TTI so that spaces (the absence of holes) correspond with binary 0's and holes (marks) correspond to binary 1's. Upon program command the contents of the TTI are transferred in parallel to the AC.

When a Teletype character starts to enter the SR, the control de-energizes a relay in the Teletype unit to release the tape feed latch. When released, the latch mechanism stops tape motion only when a complete character has been sensed, and before sensing of the next character is started.

When an 8-bit character has been assembled in the TTI, the keyboard flag is set to cause a program interrupt. The program senses the flag with a KSF instruction and, if the flag is set, issues a KRB microinstruction which clears the AC, clears the keyboard flag, transfers the contents of the TTI into the AC, and enables advance of the tape feed mechanism.

Instructions for use in supplying data to the computer from the Teletype are:

Ship on Keyboard Flag (KSF)

Octal Code: 6031
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F,]
Execution Time: 4.25 μs
Operation: If the keyboard flag is set, the contents of the PC are incremented by one so that the next sequential instruction is skipped.
Symbol: If keyboard Flag = 1, then PC + 1 → PC

Clear Keyboard Flag (KCC)

Octal Code: 6032
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The AC and the keyboard flag are cleared. This allows the hardware to begin assembling the next input character in the TTI. If there is tape in the reader and the reader is activated, the character over the read head is loaded into the TTI and the tape is advanced one frame. The keyboard can be used to load characters into the TTI provided the reader is deactivated (STOP or FREE). If the reader and keyboard are inadvertently used at the same time, a garbled message results. In either case, when the character is completely assembled in the TTI the hardware sets the keyboard flag. KCC can be microprogrammed with KRS.

Symbol: 0 → AC
0 → Keyboard flag allowing the hardware to cause:
Keyboard/Tape Character → TTI
1 → Keyboard flag when completed

Read Keyboard Buffer Static (KRS)

Octal Code: 6034
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The contents of the TTI are transferred into AC bits 4 through 11. This is a static command in that neither the AC nor the keyboard flag is cleared. KRS can be microprogrammed with KCC.

Symbol: TTI V AC4-11 → AC4-11

Read Keyboard Buffer Dynamic (KRB)

Octal Code: 6036
Event Time: 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: This microinstruction combines the functions of the KCC and KRS. The AC and keyboard flag are both cleared and the contents of the TTI are transferred into AC bits 4-11. Clearing the keyboard flag allows the hardware to begin assembling the next input character into the TTI (as described for KCC). When the character is completely assembled in the TTI, the hardware causes the keyboard flag to be set, indicating TTI again has a character ready for transfer.

Symbol: 0 → AC TTI V AC4-11 → AC4-11
0 → Keyboard Flag allowing the hardware to cause:
Keyboard/Tape Character → TTI
1 → Keyboard flag when completed

A typical TTI instruction sequence is:

```
LOOK,   KSF          /SKIP IF KEYBOARD FLAG = 1
        JMP  LOOK    /JMP BACK & TEST FLAG AGAIN
        KRB          /TRANSFER TTI CONTENTS INTO AC
```

This sequence waits for the TTI to set its flag, indicating that it has a character ready to be transferred. It then skips to the KRB command which causes the character to be transferred from the TTI to the AC.

If this sequence of instructions is made a subroutine of the main program, it can be accessed each time an input character is desired. Consequently,

```
        KCC          /CLEAR TTI FLAG
        .
        .
        .
READ,   0            /STOP PC HERE FOR RETURN ADDRESS
        KSF          /SKIP IF FLAG = 1
        JMP .-1      /TEST FLAG AGAIN
        KRB          /READ CHAR INTO AC
        JMP I READ   /EXIT TO MAIN PROGRAM
        .
        .
        .
```

Teleprinter/Punch

On program command a character is transferred from the AC to the output shift register (TTO) for transmission to the teleprinter/punch unit. The Teletype control generates the start space, shifts the eight character bits serially into the printer selector magnets of the Teletype unit, and then generates the stop marks. Bit transfer rate from the TTO to the teleprinter/punch unit is at the normal teletype rate. A character transfer requires 100 ms for completion. The teleprinter flag is set when the last bit of the character code is sent to the teleprinter/punch, indicating that the TTO is ready to receive a new character from the AC. The flag activates either the program interrupt synchronization element or the instruction skip element. When using instruction skip, the program checks the flag by means of the TSF microinstruction. If the flag is set, the program issues the TLS microinstruction which clears the flag and sends a new character from the AC to the TTO. AC-to-TTO transfer-time is short compared to the print/punch time, so the program must wait for the flag to set before issuing another TLS.

Instructions for use in outputting teletype data are as follows:

Skip on Teleprinter Flag (TSF)

Octal Code: 6041
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: If the teleprinter flag is set, the contents of the PC are incremented by one so that the next sequential instruction is skipped.
Symbol: If Teleprinter Flag = 1, then PC + 1 $\rightarrow$ PC

Clear Teleprinter Flag (TCF)

Octal Code: 6042
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The teleprinter flag is cleared. This instruction can be microprogrammed with TPC.

Symbol: 0 → Teleprinter Flag

Load Teleprinter and Print (TPC)

Octal Code: 6044
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: AC bits 4-11 are transferred to the TTO at which time the hardware starts shifting the character out to the printer/punch unit. This instruction does not clear the teleprinter flag. This instruction can be microprogrammed with TCF.

Symbol: AC4-11 → TTO causing:
TTO → printed and (if punch is on) punched

Load Teleprinter Sequence (TLS)

Octal Code: 6046
Event Time: 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: This is a microinstruction that combines TCF and TPC. The teleprinter flag is cleared and the contents of AC bits 4-11 are transferred to the TTO, where the hardware shifts the character out to the printer/punch unit. When the printer/punch has finished outputting the character and is ready for another character, the hardware sets the teleprinter flag. The whole operation, from the time at which the TLS has cleared the flag and TPC starts character transfer, until the time at which the hardware finishes with the character and again sets the flag, requires 100 ms.

Symbol: 0 → Teleprinter flag
AC 4-11 → TTO, causing:
TTO → Printed and (if punch is on) punched
1 → Teleprinter flag when completed

A typical TTO instruction sequence is:

```
          .
          .
          .
          CLA
          TAD X          /PUT CHARACTER CODE INTO AC FROM
                         /LOCATION
          TLS            /LOAD TTO FROM AC & PRINT/PUNCH
FREE,     TSF            /TEST FLAG TO SEE IF DONE PRINTING,
                         /SKIP IF = 1
          JMP FREE       /TEST FLAG AGAIN
          CLA            /CLEAR CHARACTER CODE FROM AC
          .
          .
          .
```

This sequence sends one character to the TTO and waits for printing/punching before sending another character. It does not require that the flag be set to output the character. By making the instruction sequence a subroutine of a larger program, it can be accessed by a JMS each time a character is to be output. Assume that the subroutine is entered with the character code in the AC:

```
TYPE,    0
         TLS          /LOAD TTO FROM AC AND PRINT/PUNCH
         TSF          /TEST FLAG, SKIP IF = 1
         JMP .-1      /JMP BACK & TEST FLAG AGAIN
         CLA          /CLEAR CHARACTER FROM AC
         JMP I TYPE   /EXIT TO MAIN PROGRAM
         .
         .
         .
```

By rearranging this subroutine the 100 ms spent waiting for the character to be output and the flag to be set is used to continue the main program, making more efficient use of program time.

```
TYPE,    0            /TEST FLAG TO SEE IF TELEPRINTER FREE;
         TSF          /SKIP IF YES OR . . .
         JMP .-1      /WAIT TILL IT IS BY TESTING AGAIN AND
                      /AGAIN
         TSL          /OUTPUT CHARACTER
         CLA          /CLEAR CHARACTER FROM AC
         JMP I TYPE   /EXIT TO CONTINUE PROGRAM
```

This subroutine tests the flag first and waits only if a previous character is still being output. If clears the AC and exits immediately after sending the character to the TTO and continues to run the user's program instead of waiting while the Teletype (a much slower I/O device) is typing/punching the preceding character. The computer clears all flags which are on the clear flag bus (including teleprinter flags), when the START pushbutton is depressed. This means that the user program must account for setting the teleprinter flag initially and after each TCF (if any), or the program hangs up in the wait loop of the print routine. The only way to set the flag is by initializing it. This instruction should appear among the first few executed, and must appear before any attempt to output a character. The following example initializes the flag with the TLS as the first instruction of the program and makes optimum use of the punch/print time.

```
BEGIN,   TLS
         .
         .
         .
TYPE,    0
         TSF          /SKIP IF FLAG = 1 or . . .
         JMP .-1      /WAIT UNTIL IT IS LOAD TTO &
         TLS          /TYPE CHARACTER
         CLA          /CLEAR CHARACTER FROM AC
         JMP I TYPE   /EXIT & CONTINUE PROGRAM WHILE
         .            /TELEPRINTER IS FINISHING CHARACTER
         .
         .
```

TYPE PR8/I [PR8/L] HIGH SPEED PERFORATED TAPE READER AND CONTROL

(The control circuitry for this device is located in the PDP-8/I [PDP-8/L] central processor.)

The PR8/I [PR8/L] senses 8-hole paper or Mylar tape perforations photoelectrically at 300 characters per second. The PR8/I [PR8/L] control requests reader movement, transfers data from the reader into the reader buffer (RB) and signals the computer when incoming data is present. Reader tape movement is started by the reader control, simultaneously releasing the brake and engaging the clutch. When the RB contains a complete 8-bit character, the reader flag is set. The character is then transferred from RB to AC4-11 by instruction RRB. The reader flag is sensed by instruction RSF and cleared by instruction RRB. Tape format is as described for the Teletype Model ASR 33. Instructions for inputting data from the reader are:

Skip on Reader Flag (RSF)

Octal Code: 6011
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The reader flag is sensed and, if it contains a binary 1, the PC is incremented by one so that the next sequential instruction is skipped.
Symbol: If Reader Flag = 1, then PC + 1 → PC

Read Reader Buffer (RRB)

Octal Code: 6012
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the reader buffer is ORed into AC4-11 and the reader flag is cleared. This command does not clear the AC.
Symbol: RB V AC4-11 → AC4-11
0 → Reader Flag

Reader Fetch Character (RFC)

Octal Code: 6014
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The reader flag and the RB are both cleared, one character is loaded into RB from the tape, and the reader flag is set when the RB is full.
Symbol: 0 → Reader Flag, 0 → RB
Tape Data → RB
1 → Reader Flag when RB full

A program sequence loop to read a character from perforated tape can be written as follows:

```
        RFC         /FETCH CHARACTER FROM TAPE
LOOK    RSF         /SKIP IF READER FLAG = 1
        JMP LOOK    /JUMP BACK & TEST FLAG AGAIN
        CLA         /CLEAR CHARACTER FROM AC
        RRB         /LOAN AC FROM RB, CLEAR READER FLAG
```

TYPE PP8/I [PP8/L] HIGH-SPEED PERFORATED TAPE PUNCH AND CONTROL

(The control circuitry from this device is located in the PDP-8/I [PDP-8/L] central processor.) The PP8/I consists of a Royal McBee paper tape punch that perforates 8-hole paper or Mylar tape at a punch rate of 50 characters per second. A character to be punched on a line of tape is loaded in an 8-bit punch buffer (PB) from AC bits 4 through 11. The punch flag is set when the character has been punched and the PB is empty. The punch flag indicates that new information may be transferred into the punch buffer. The punch flag is as described for the Teletype Model ASR 33. Instructions for outputting data to the punch are:

Skip on Punch Flag (PSF)

Octal Code: 6021
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The punch flag is sensed and, if it contains a binary 1, the PC is incremented by one so that the next sequential instruction is skipped.
Symbol: Skip if Punch Flag = 1, then PC + 1 → PC

Clear Punch Flag (PCF)

Octal Code: 6022
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Both the punch flag and the PB are cleared in preparation for receiving a new character from the computer.
Symbol: 0 → Punch Flag, 0 → PB

Load Punch Buffer and Punch Character (PPC)

Octal Code: 6024
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The 8-bit character in AC4-11 is transferred into the PB and then this character is punched. The instruction does not clear the punch flag or the PB.
Symbol: AC4-11 → PB

Load Punch Buffer Sequence (PLS)

Octal Code: 6026
Event Time: 2, 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: The punch flag and punch buffer are both cleared, the content of AC bits 4 through 11 is transferred into the punch buffer, the character in the PB is punched in tape, and the punch flag is set when the operation is completed.
Symbol: 0 → Punch Flag, 0 → PB
AC4-11 → PB
PB → Punch Tape
1 → Punch Flag when PB empty

A program sequence loop to punch a character when the punch buffer is "free" can be written as follows:

```
FREE,   PSF          /SKIP IF PUNCH FLAG = 1
        JMP FREE     /JUMP BACK & TEST FLAG AGAIN
        PLS          /CLEAR PUNCH FLAG & PB, LOAD PB
                      FROM AC, PUNCH CHARACTER, SET
                      PUNCH FLAG WHEN DONE
```

SECTION 7-2 INTERFACE ADAPTER UNITS

Four types of interface units may be required for use with the PDP-8/I and the PDP-8/L computers — the Type DW08A/DW08-B Input/Output Conversion Panel, and the DM01/DM04 Data Channel Multiplexer. A fifth interface unit, the Type BA08 Peripheral Expander Unit, is used with the PDP-8/L only.

TYPE DW08-A/DW08-B INPUT/OUTPUT CONVERSION PANEL

The basic DW08 panel converts the input and output bus of the computer (either the PDP-8/I or the PDP-8/L) from one voltage level to another, to achieve compatibility between the computer and various I/O devices.

The DW08-A I/O Conversion Panel is used on the positive bus of the PDP-8/L (or the PDP-8/I with the KA8/I Positive I/O Bus option installed), to convert the positive bus to a negative bus to achieve compatibility with negative bus peripherals. Conversely, the DW08-B panel converts the negative bus of a PDP-8/I to a positive bus. The DW08-B only can be used on a negative bus PDP-8/I because the PDP/L bus has only a standard positive bus.

Figure 7-2 illustrates the differences between the two configurations. The basic DW08 panel has I/O cables connected from the computer to its bus. These cables contain buffered signals that are amplified and inverted for the optional I/O device. The I/O device communicates with the computer via the I/O cables through the DW08 panel.

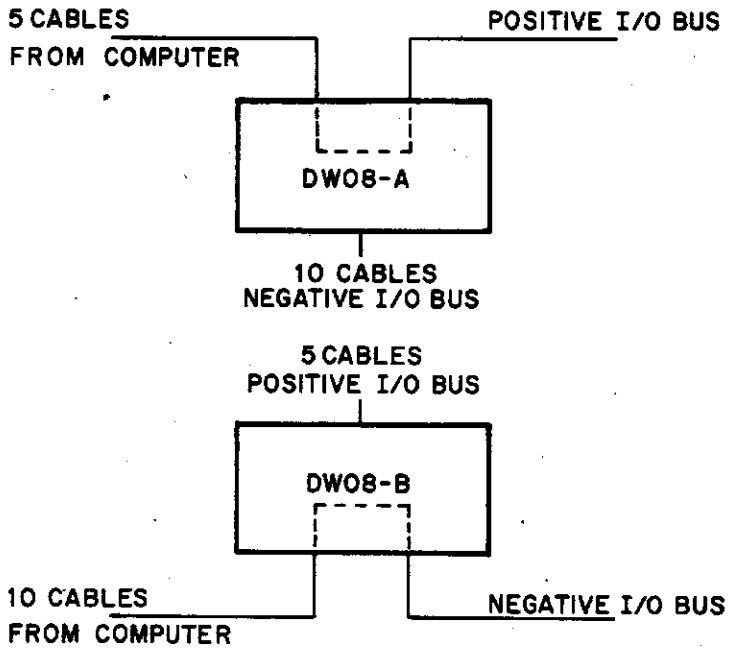


Figure 7-2 I/O Conversion Panel Configurations

The original bus of the computer is parallel-wired on the DW08 to an additional set of input/output slots, enabling the continuation of the original bus to additional devices compatible with it.

TYPE BA08 PERIPHERAL EXPANDER UNIT

The BA08 option is used exclusively with the PDP-8/L system to facilitate the installation of one or more PDP-8/L options that cannot be installed directly in the PDP-8/L chassis. The BA08 option is the same physical size as the PDP-8/L and is similar in appearance except for a blank front panel.

Options requiring the use of the BA08 are listed in Table 7-1. The listed options can be installed in the BA08 at the same time, except where variations of a major option are mutually exclusive. For example, only one KW8/L Real-Time Clock option can be used at a time. Similarly, the VC8/L and KV8/L options are mutually exclusive. A maximum of one MC8/L Memory Extension Control and 4k of core memory can be accommodated in the BA08.

The BA08 provides interface connections to the PDP-8/L and power for the modules used in the applicable I/O options. A power supply, identical to the power supply of the PDP-8/L, is contained in the BA08. BA08 circuitry is limited to bus driver circuits, inverters, loads, and diode clamps. Signal amplification for interfacing signals is provided by the bus driver circuits.

Table 7-1 Options Used with the BA08

TYPE	DESCRIPTION
CR8/L Card Reader and Control	Allows the PDP-8/L to read 12 rows, 80-column punched cards at a nominal rate of 200 cards/minute.
DC02 Multiple Teletype Control	Allows the PDP-8/L user to add from one to four teletypewriters or other serial data stations. This option consists of a DC02-A Multiple Control and from one to four DC02-D Data Stations.
KV8/L Storage Tube Display Control Logic only (table space is required for VT01 Storage Tube Display)	A display controller which is basically an analog stroke vector generator that provides programmed selection and control to the VT01 Storage Tube Display. The KV8/L and the VC8/L cannot operate in the same BA08 unit.

Table 7-1 Options Used with the BA08 (continued)

TYPE	DESCRIPTION
KW8/L Real-Time Clock (consists of six configuration):	Allows the PDP-8/L to establish a reference time to be used to associate data with real time.
KW8/LA Fixed-Interval Line Frequency KW8/LB Fixed-Interval Variable Frequency KW8/LC Fixed-Interval Crystal Frequency	Program interrupt occurs each time clock pulse is received; counts at a specific rate.
KW8/LD Programmable-Interval Line Frequency KW8/LE Programmable-Interval Variable Frequency KW8/LF Programmable-Interval Crystal Frequency	Program interrupt occurs only when overflow occurs. Counts at fixed frequency rate and will not cause a interrupt until the number of counts desired is reached. Automatically takes 2's complements of desired number in the AC.
MC8/LA Memory Extension Control and 4096 words of memory without memory parity	Provides the PDP-8/L with an additional 4096-word (4K) memory with parity.
MC8/LB Memory Extension Control and 4096 words of memory with parity	Provides the PDP-8/L with an additional 4096-word (4K) memory with parity.
VC8/L Oscilloscope Display Control (without display)	Provides the controls to display and plot data on an optional display (VR01A, VR01B, VR02, etc.
NOTE: The VC8/L and KV8/L options cannot be used in the same BA08 Peripheral Expander Unit.	
VP8/L Incremental Plotter Control	Allows the PDP-8/L to plot and display data on an Incremental Plotter.

TYPE DM01/DM04 DATA CHANNEL MULTIPLEXER

Type DM01 multiplexer is a negative-bus data break device which provides the PDP-8/I and PDP-8/L systems with the capability of operating up to seven high-speed devices, via the data break facility. Only one DM01 can be used with either computer system. The Type DM04 differs from the DM01 in that the DM04 is a positive-bus data break device and provides the capability of operating up to three high-speed devices; as many as three DM04's can be used simultaneously with either computer. Thus, use of the DM04 option provides a maximum use of up to nine optional high-speed devices.

The multiplexer is essentially a switching device for use between the computer and the high-speed I/O devices. The high-speed peripheral devices include high-speed magnetic tape systems, high-speed drum memories, and disk systems, all of which use the data break facility.

SECTION 7-3 TYPE CR8/I [CR8/L] CARD READER

The Type CR8/I Card Reader reads standard 12 row, 80-column punched cards at a nominal rate of 200 cards per minute. Cards are read by column, beginning with column 1. One select instruction starts the card moving past the read station. Once a card is in motion, all 80 columns are read. Data in a card column is photoelectrically sensed. Column information is read in one of two program-selected modes: alphanumeric or binary. In the alphanumeric mode the 12 information bits in one column are automatically decoded and transferred into the least significant half of the accumulator as a 6-bit Hollerith code. Appendix C lists the Hollerith card codes. In the binary mode the 12 bits of a column are transferred directly into the accumulator so that the top row (12) is transferred into AC and the bottom row (9) is transferred into AC11. A punched hole is interpreted as a binary 1 and the absence of a hole is interpreted as a binary 0. Three program flags indicate card reader conditions to the computer. The data ready flag appears and requests a program interrupt when a column of information is ready to be transferred into the AC. A read-alphanumeric or read-binary command must be issued within 1.4ms after the data ready flag appears to prevent data loss. The card done flag appears and requests a program interrupt when the card leaves the read station. A new select command must be issued immediately after the card done flag appears, to keep the reader operating at maximum speed. Sensing of this flag can eliminate the need for counting columns or, combined with column counting, can provide a check for data loss. The reader-not-ready flag can be sensed by a skip command to provide indication of card reader power off, pick failure, a dark check indication, a stacker failure, hopper empty, stacker full, Sync failure, or light check indication. When this flag appears the reader cannot be selected and select commands are ignored. The reader-not-ready flag is not connected to the program interrupt facility and cannot be cleared under program control. Manual intervention is required to clear the reader-not-read flag. Instructions for the CR8/I [CR8/L] are:

Skip On Data Ready (RCSF)

Octal Code: 6631
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the data ready flag is sensed and, if it contains a 1 (indicating that information for one card column is ready to be read), the content of the PC is incremented by one so the next sequential instruction is skipped.
Symbol: If Data Ready Flag = 1, then PC + 1 → PC

Read Alphanumeric (RCRA)

Octal Code: 6632
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The 6-bit Hollerith code for the 12 bits of a card column are transferred into bits 6 through 11 of the AC, and the data ready flag is cleared.
Symbol: AC6-11 V Hollerith Code → AC6-11
0 → Data Ready Flag

Read Binary (RCRB)

Octal Code: 6634
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The 12-bit binary code for a card column is transferred directly into the AC, and the data ready flag is cleared. Information from the card column is transferred into the AC so that card row 12 enters AC0, row 11 enters AC1, row 0 enters AC2, . . . and row 9 enters AC11.
Symbol: AC V Binary Code → AC
0 → Data Ready Flag

Skip on Card Done Flag (RCSD)

Octal Code: 6671
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the card done flag is sensed and, if it contains a 1 (indicating that the card has passed the read station), the content of the PC is incremented to skip the next sequential instruction.
Symbol: If Card Done Flag = 1, then PC + 1 → PC

Select Card Reader and Skip if Ready (RCSE)

Octal Code: 6672
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the reader-not-ready flag is sensed and, if it contains a 1 (indicating that the card reader is ready for programmed operation), the PC is incremented to skip the next sequential instruction; a card is started towards the read station from the feed hopper; and the card done flag is cleared. If the reader-not-ready flag contains a 0 (indicating power is off or no card is in the read station), card selection (motion) and skip do not occur.
Symbol: If Reader-Not-Ready Flag = 1, then PC + 1 → PC
0 → Card Done Flag

Clear Card Done Flag (RCRD)

Octal Code: 6674
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The card done flag is cleared. This command allows a program to stop reading at any point in a deck.
Symbol: 0 → Card Done Flag

A logical instruction sequence to read cards is:

```
START,   RCSE            /START CARD MOTION AND SKIP IF READY
         JMP NOT RDY     /JUMP TO SUBROUTINE THAT TYPES OUT
                         /"CARD READER MANUAL INTERVENTION
                         /REQUIRED" OR HALTS
NEXT?    RCSF            /DATA READY?
         JMP. DONE       /NO, CHECK FOR END OF CARD
         RCRA or RCRB    /YES, READ ONE CHARACTER OR ONE
                         /COLUMN AND CLEAR DATA READY
         DCA I STR       /STORE DATA
DONE,    RCSD            /END OF CARD?
         JMP NEXT        /NO, READ NEXT COLUMN
         JMP OUT         /YES, JUMP TO SUBROUTINE THAT CHECKS
                         /CARD COUNT OR REPEATS AT START FOR
                         /NEXT CARD
```

No validity checking is performed by the CR8/I [CR8/L]. A programmed validity check can be made by reading each card column in both the alphanumeric and the binary mode (within the 1.4 ms time limitation) and then performing a comparison check.

Before commencing a card reading program, energize the reader, load the input hopper with cards, and press the MOTOR START and READ START pushbuttons. The function of the manual controls and indicators are shown in Table 7-2 (as they appear from right to left on the card reader):

Table 7-2 CR8 Controls and Indicators

Control or Indicator	Function
POWER switch	On-Off toggle switch. Applies power to all circuits except drive motor.
MOTOR START	Momentary action pushbutton, with separate indicator. Applies power to main drive motor. Motor start is also used as a reset to clear error indicators and therefore will not operate if there is an unremedied condition such as: 1. Input hopper is empty. 2. Output hopper is full. 3. All photo cells are not lit. 4. Internal power supply is not operational.
READ START	Momentary action pushbutton, with separate indicator. Causes ready line to go high, which enables card reading under control of the external read command. If read command is open or high, card reading begins immediately at full rated speed.

Table 7-2 CR8 Controls and Indicators (Cont.)

Control or Indicator	Function
READ STOP	Momentary action pushbutton, with separate indicator. Inhibits further card reading until READ START switch is pressed again. Ready line goes low and READ STOP condition is indicated. Does not stop drive motor. However, a READ STOP condition is indicated whenever the drive motor is stopped.
INDICATORS	Several detection circuits are incorporated in the card reader. Whenever any red indicator lights, the drive motor is stopped after allowing the completion of the current card cycle.
PICK FAIL Indicator	Lights when a card fails to enter the read station after two successive pick attempts.
DARK CHECK Indicator	After the card enters the read station, a check is made at the hypothetical 0th and 81st hole positions to be sure all photocells are dark. If not, the DARK CHECK indicator lights and data outputs are inhibited immediately.
STACKER FAIL Indicator	When three cards have passed the read station and none have been stacked, a STACK FAIL is indicated. Prevents more than three cards from being in the track at once.
HOPPER EMPTY Indicator	Indicates input hopper is empty.
STACKER FULL Indicator	Switch closure detects when approximately 400 cards are in the stacker hopper.
SYNC FAIL Indicator	Internal timing signals are derived from an oscillator which is synchronized to the track speed. If the sync signal is lost, a SYNC FAIL is indicated.
LIGHT CHECK Indicator	All photo cells must always be lit except during the time a card is being read. The detector is inhibited each time a card enters the read station until position (count of) 84. If a card fails to leave the read station by this time, a LIGHT CHECK is indicated.

SECTION 7-4 PLOTTER OPTIONS

TYPE VP8/I [VP8/L] INCREMENTAL PLOTTER CONTROL

(The control circuitry for this device is located in the PDP-8/I central processor [BA08 Peripheral Expander Box when used with the PDP-8/L computer].)
Four models of California Computer Product's Digital Incremental Recorder can be operated from a Type VP8/I [VP8/L] Incremental Plotter Control. Characteristics of the four recorders are:

CCP Model	Step Size (inches)	Speed (steps/minute)	Paper Width (inches)
563	*0.01 or 0.005	12,000	30
565	0.01 or 0.005	18,000	12

(NOTE: Each of the two models is available in one of two sizes; either step size 0.01 or 0.005.)

The principles of operation are the same for each of the models. Bidirectional rotary stepping motors are employed for both the X- and Y-axes. Plotting is produced by movement of a pen, relative to the surface of the graph paper, with each pen direction instruction causing an incremental step in one of four directions. X-axis deflection is produced by motion of the drum; Y-axis deflection, by motion of the pen carriage. Instructions are used to raise and lower the pen from the surface of the paper. Each incremental step can be in any one of eight directions through appropriate combinations of the X- and Y-axis instructions. All recording (discrete points, continuous curves, or symbols) is accomplished by the incremental stepping action of the paper drum and pen carriage. Front panel controls permit single-step or continuous-step manual operation of the drum and carriage, and manual control of the pen solenoid. The recorder and control are connected to the computer program interrupt and instruction skip facility. Instructions for the recorder and control are:

Skip on Plotter Flag (PLSF)
Octal Code: 6501
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter flag is sensed and, if it contains a 1, the content of the PC is incremented by one so the next sequential instruction is skipped.
Symbol: If Plotter Flag = 1, then PC + 1 → PC

Clear Plotter Flag (PLCF)
Octal Code: 6502
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter flag is cleared in preparation for issuing a plotter operation command.
Symbol: 0 → Plotter Flag

Pen Up (PLPU)
Octal Code: 6504
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter pen is raised from the surface of the paper.
Symbol: None

Pen Right (PLPR)
Octal Code: 6511
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter pen is moved to the right in either the raised or lowered position.
Symbol: None

Drum Up (PLDU)
Octal Code: 6512
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter paper drum is moved upward. This command can be combined with the PLPR and PLDD commands.
Symbol: None

Instructions for the plotter and control are:

Drum Down (PLDD)
Octal Code: 6514
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter paper drum is moved downward.
Symbol: None

Pen Left (PLPL)
Octal Code: 6521
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter pen is moved to the left in either the raised or lowered position.
Symbol: None

Drum Up (PLUD)
Octal Code: 6522
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter paper drum is moved upward. This command is similar to command 6512, except that it can be combined with the PLPL or PLPD commands.
Symbol: None

Pen Down (PLPD)

Octal Code: 6524
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The plotter pen is lowered to the surface of the paper.
Symbol: None
Program sequence must assume that the pen location is known at the start of a routine since there is no means of specifying an absolute pen location in an incremental plotter. Pen location can be preset by the manual controls on the recorder. During a subroutine, the PDP-8/I [PDP-8/L] can track the location of the pen on the paper by counting the instructions that increment the position of the pen and the drum.

VP8/IA AND VP8/IB [VP8/LA AND VP8/LB] INCREMENTAL PLOTTERS AND CONTROLS

The A and B versions of the basic VP8 indicate that the user can purchase the controls and/or plotters separately and for varying step sizes and paper widths. Table 7-3 lists the options available to the user.

Table 7-3 VP8 Options

Control Only	Control and Plotter (30 inch)	Control and Plotter (12 inch)
VP8/I	VP8/IA 0.01 in. or 0.005 in.	VP8/IB 0.01 in. or 0.005 in.
VP8/L	VP8/LA 0.01 in. or 0.005 in.	VP8/LB 0.01 in. or 0.005 in.

SECTION 7-5 CRT DISPLAYS

The CRT display equipment for use with either computer consist of a Type VC8/I [VC8/L] Oscilloscope Display Control Unit and a KV8/I [KV8/L] Storage Tube Display Control. However, to use either the VC8/L or the KV8/L with the PDP-8/L computer, the BA08 Peripheral Expander unit must also be used. Only one type of display can be accommodated at any time.

TYPE VC8/I [VC8/L] OSCILLOSCOPE DISPLAY CONTROL UNIT

(The VC8/I control circuitry is located in the PDP-8/I central processor; the VC8/L control circuitry is located in the BA08 unit.)

The Type VC8/I [VC8/L] is a two-axis, digital-to-analog converter with an intensifying circuit, which provides the deflection and intensifying pulses needed to plot data on Models RM503 and RM564 Tektronix oscilloscope. (See VR01-A and VR01-B for more information.) Coordinate data in 10-bit characters are loaded into an X buffer (XB) and a Y buffer (YB) from AC2-11. The binary data in the XB and YB buffers is converted to 0V to +2.7V analog deflection signals. The 20V (max) intensifying signal, which connects to the grid of the CRT, is controlled by the output of a 2-bit brightness register (BR). The BR controls the amplitude of the intensifying pulse, depending on the binary content of BR, which is loaded from a number contained in IOT instruction DSB. The BR is initialized to a load of maximum intensity by application of power to the computer or by depressing the Start [START] pushbutton switch. Points can be displayed at a rate of approximately 30 kHz. The instructions for outputting data to the display are:

Clear X-Coordinate Buffer (DCX)
Octal Code: 6051
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The X-coordinate buffer (XB) is cleared.
Symbol: 0 → XB

Load X-Coordinate Buffer (DXL)
Octal Code: 6053
Event Time: 1, 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The X-coordinate buffer (XB) is loaded with new X-axis data from AC2-11.
Symbol: AC2-11 → XB

Clear Y-Coordinate Buffer (DCY)
Octal Code: 6061
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The Y coordinate buffer (YB) is cleared.
Symbol: 0 → YB

Clear and Load Y-Coordinate Buffer (DYL)
Octal Code: 6063
Event Time: 1, 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The Y-coordinate buffer (YB) is loaded with the new Y-axis data from AC2-11.
Symbol: AC2-11 → YB

Intensify (DIX)
Octal Code: 6054
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Intensify the point defined by the content of XB and YB. This command can be combined with the DXL command.
Symbol: None

Intensify (DIY)
Octal Code: 6064
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Intensify the point defined by the content of XB and YB. This command is identical to the DIX command except that it can be combined with the DYL command.
Symbol: None

X-Coordinate Sequence (DXS)

Octal Code: 6057
Event Time: 1, 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: DXS is a microinstruction which combines instructions DXL and DIX. XB is loaded from AC2-11, the point defined by the content of the X and Y buffers is intensified.
Symbol: AC2-11 → XB then intensify

Y-Coordinate Sequence (DYS)

Octal Code: 6067
Event Time: 1, 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: DYS is a microinstruction which combines instructions DYL and DIY. YB is loaded from AC2-11, then the point defined by the content of the X and Y coordinate buffers is intensified.
Symbol: AC2-11 → YB then intensify

Set Brightness Control (DSB)

Octal Code: 607X
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The brightness register (BR) is loaded from the content of bits 10 and 11 of the instruction. Instruction code 6074 sets zero brightness; instruction code 6075 sets minimum brightness; instruction code 6076 sets medium brightness; instruction code 6077 sets maximum brightness.
Symbol: MB10-11 → BR

The following program sequence, to display a point, assumes that the coordinate data is stored in known addresses X and Y.

```
X,
Y,
BEG,    CLA         /CLEAR CHARACTER FROM AC
        TAD X       /LOAD AC WITH X
        DXL         /LOAD XB
        CLA         /CLEAR X FROM AC
        TAD Y       /LOAD AC WITH Y
        DYS         /LOAD YB, /DISPLAY POINT
        CLA         /CLEAR Y FROM AC
```

TYPE VR01-A OSCILLOSCOPE DISPLAY AND MOUNTING HARDWARE

The VR01-A is a Tektronix RM503 Oscilloscope Display unit and its associated mounting hardware which is used to plot points on a self-contained 5-inch CRT. The VR01-A is controlled by the VC8/I [VC8/L].

TYPE VR01-B OSCILLOSCOPE DISPLAY MOUNTING HARDWARE AND TYPE 370 LIGHT PEN

The Type VR01-B consists of the VR01-A and a Type 370 Light Pen, photosensitive device, which detects the presence of information displayed on a CRT. If the pen is held against the face of the CRT at a bright point, the light pen display flag is set. The flag activates the computer instruction skip facility, the instructions for which are:

Skip on Display Flag (DSF)

Octal Code: 6071
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The display flag is sensed and, if it is set, the content of the PC is incremented by one so that the next sequential instruction is skipped.
Symbol: If Display Flag = 1, then PC + 1 → PC

Clear the Display Flag (DCF)

Octal Code: 6072
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The display flag is cleared prior to sensing another bright point on the CRT.
Symbol: 0 → Display Flag

TYPE KV8/I [KV8/L] GRAPHIC SYSTEM

The KV8/I [KV8/L] Graphic System is available for the PDP-8/I [PDP-8/L] computer system. This display system can be utilized in interactive graphic, graphical output, and alphanumeric terminal applications. The KV8/I controller plugs into the internal option bus of the PDP-8/I [the BA08 Peripheral Expander unit on the PDP-8/L].

The KV8/I [KV8/L] Graphic System utilizes the VT01 storage tube display with a 6½ inch by 8¼ inch screen size. Inputs to the system are entered via the TTY keyboard, the H306 Joystick graphic input device, or other compatible devices. The system includes the EDGRIN (EDitor with GRaphic INterpreter) software-support package.

The controller portion of the graphic system is basically an analog stroke vector generator that provides programmed selection and control of the VT01 storage tube display. Hardware capabilities include: single command point plotting, linear vectors, circular vectors, and arc vectors. Alphanumeric character generation is accomplished by the use of an efficient, programmable subroutine supplied with the basic system.

The controller accepts analog input signals produced by the H306 or other compatible source on the X or Y cursor channels. The system provides a means to display these signals as a visible but non-stored cursor on the VT01, and to perform the required analog-to-digital conversion to enter these signals into the computer. The controller can operate up to eight independent VT01/VT02 terminals by adding the M713 multiplex option. These terminals consist of the VT01 storage tube display and the VT02 terminal, which includes an electronic keyboard (128-character code), interface electronics, power supply, Joystick comparator and hooded terminal case. Each terminal is serviced by the multiplexer and is operated independently of other terminals on the system, thus allowing a single system to service multiple applications.

Specifications

These specifications are based on using the KV8/I with the PDP-8/I computer. (Operating rates must be adjusted for the slower speeds of the PDP-8/L.)

Hardware

a. No refresh is necessary. Drift and flicker are eliminated by the direct-view storage tube.
b. High information density. Over 4000 characters (76 characters/line, 54 lines), and 30,000 discrete resolvable points or vectors.
c. Linear stroke vector generation at 225 full-screen vectors/second, or 1750 short vectors/second.
d. Circle generation accomplished by single-command circular stroke generator.
e. Point plotting capable of addressing a page-size of 32 x 32 cm with 1024 x 1024 points.
f. Joystick-controlled pointer incorporating a write-through cursor.
g. Variable font symbol generator (average writing speed of 350 Hz with 64 basic alphanumeric characters, program expandable).
h. Origin shifting and scalling (zooming) to view area 21 x 16.3 cm.

Software

a. General-purpose Text Editor with more than 20 commands for rapid editing of text strings.
b. General-purpose Graphics Software package ready for immediate use and easily expandable.
c. On-line editing for picture manipulation and changing a display copy.
d. Intermixing of characters with vectors permissible.
e. Generation of Macro programs that can be referenced as nested subroutines.
f. Graph plotting with incrementing on either the X or Y axis.
g. Automatic generation of preselected dash/dot sequences.
h. Function buttons for program interaction.
i. Automatic generation of alphanumeric linkage listing describing the topology of the generated graphical structure.

INSTRUCTIONS

Skip if Interrupt is Low (SNC)
Octal Code: 6051
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Senses the condition of the cursor interrupt flag (CIF). The flag produces an interrupt request when set by operation of the interrupt push-button on the Joystick. The flag is initially cleared when the computer is started. As with all flag-sent instructions, SNC can be used under interrupt conditions to detect the source of the interrupt, or it can be used under interrupt on (ION) when the interrupt request has been caused by the operation of the cursor interrupt button. In a program running with the interrupt off, SNC can be used to ignore the cursor successive approximation subroutine in the program if a request for service has not been made from the Joystick controller.
Symbol: If CIF = 0, then PC + 1 → PC

Clear Interrupt Flag (CCF)

Octal Code: 6052
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: This instruction is used to clear the CIF after a request for service has been acknowledged by the program.
Symbol: 0 → CIF

Select Analog Comparator (SAC)

Octal Code: 6062
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The analog comparator (ANC) is set to compare the analog content of any one of six analog sources with the content of the digital-to-analog converter. The analog sources are chosen according to a 3-bit binary code. This code establishes the parameter for choosing the wanted register according to the content of AC2, AC3, and AC6.
Symbol: AC2, AC3, AC6 → ANC

Accumulator Bits			Selected Source
2	3	6	
0	0	0	X INT
0	0	1	Y INT
1	0	0	X SH
1	0	1	Y SH
1	1	0	X CUR
1	1	1	Y CUR

Load and Select Format (LDF)

Octal Code: 6063
Event Time: 1, 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: This instruction is used to establish the mode in which a wanted graphic is to be produced according to a 2-bit binary code. This code determines whether the desired vector will be linear absolute relative, whether the point plot mode will be used, or whether the cursor will be displayed. This code establishes the parameters for these formats by loading the content of AC2 and AC3 into the format register (FR). The LDF instruction must precede the LDX and LDY instructions.
Symbol: AC2-3 → FR

Load AC into X Sample and Hold (LDX)

Octal Code: 6064
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The X-axis sample and hold register (XSH) is loaded with the binary equivalent of the X-axis coordinate according to the contents of AC2-11. This data appears at the output of the digital-to-analog converter (DAC) as the

analog equivalent of the X-axis value of the binary word stored in the AC. The LDX instruction clears an existing ready flag (RF) and sets the RF after 100 ±20 μs.

Symbol: 0 → RF
AC → DAC
DAC → XSH (Analog Voltage)
1 → RF

Load AC into Y Sample and Hold (LDY)

Octal Code: 6065
Event Time: 1, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The Y-axis sample and hold register (YSH) is loaded with the binary equivalent of the Y-axis coordinates according to the contents of AC2-11. This data appears at the output of the digital-to-analog converter as the analog equivalent of the binary word in the AC. The LDY instruction clears an existing RF and sets the RF after 100 ± 20 μs.

Symbol: 0 → RF
AC → DAC
DAC → YSH (Analog Voltage)
1 → RF

Execute Microprogrammed Instruction (EXC)

Octal Code: 6066
Event Time: 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Used to execute the desired vector in the function register (FUR) and AC6-11 (bit 4 inhibits bit 6). The parameter word establishes long or short formats, circular vectors, display erasure, reset of the integrators, and intensification of the vector. The EXC instruction clears an existing RF and sets the RF as follows:

a) after 20 ± 5 μs for a point or vector continue
b) after 250 μs for short vectors
c) after 4.05 ms for long vectors
d) after 500 ms for an erase

Symbol: 0 → RF
AC2-4, AC6-11 → FUR
Execute Operation
1 → RF

Execute Long Vector Microprogrammed Instruction (EXL)

Octal Code: 6067
Event Time: 1, 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Execute long vector with bit positions the same as in EXC (6066g). Long vector is inhibited by bit 4. (Operation requires 4.05 ms.)

Symbol: 0 → RF
AC2-4, AC6-11 → FUR
Execute Long Vector Operation
1 → RF

Skip if Ready Flag is High (SRF)

Octal Code: 6071
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Used to determine when the controller is ready to perform the next execute instruction. The RF produces an interrupt condition when set. The RF can be set by pressing the erase pushbutton on the VT01 unit. Normally, however, the state of this flag is determined by the controller. This flag is initially cleared when the computer is started and prior to an LDX, LDY, or EXC instruction.
Symbol: If RF = 1, PC + 1 → PC

Clear Ready Flag (CRF)

Octal Code: 6072
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: This instruction clears the RF after a skip instruction has been acknowledged.
Symbol: 0 → RF

Skip if D/A is Greater than Analog Comparison Signal (SDA)

Octal Code: 6073
Event Time: 1, 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Used in the successive approximation subroutine to determine the digital equivalent of the selected analog holding register. This instruction is used with the SAC (6062_8) instruction. When the voltage in the analog comparator is greater than the selected value (the value in the register), ANF = 1. When ANF = 1, the next instruction is skipped.
Symbol: If ANF = 1, PC + 1 → PC

Load D/A (LDA)

Octal Code: 6074
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: This instruction is used to load the content of AC2-11 into the analog comparator (ANC) and is used with SDA (6073_8) in the successive approximation subroutine to determine the digital value of the content of the selected analog holding register. This instruction does not change flag states.
Symbol: AC2-11 → DAC
DAC → ANC

SAMPLE PROGRAM

SNC SKIP IF INTERRUPT FLAG IS LOW
CCF CLEAR INTERRUPT FLAG
SAC SELECT ANALOG COMPARATOR:

AC BIT 2	3	6	FUNCTION
0	0	0	READ X INTEGRATOR
0	0	1	READ Y INTEGRATOR
1	0	0	READ X SAMPLE/HOLD
1	0	1	READ Y SAMPLE/HOLD
1	1	0	DISPLAY CURSOR AND READ X CURSOR SIGNAL
1	1	1	DISPLAY CURSOR AND READ Y CURSOR SIGNAL

LDF /LOAD AND SELECT FORMAT, CLEAR VECTOR CONTINUE:

PATHS CLOSED	AC BIT 2	3	FORMAT
VM	0	0	VECTOR MODE, RELATIVE INPUT
VM, AB	0	1	VECTOR MODE, ABSOLUTE INPUT
PM	1	0	POINT MODE
CUR, AB	1	1	DISPLAY CURSOR

NOTE: SAC and LDF do not affect flag status

LDX LOAD THE CONTENTS OF THE ACCUMULATOR INTO THE X SAMPLE AND HOLD, CLEAR READY FLAG AND RAISE WHEN X IS LOADED (APPROX 120 μs)

LDY SAME AS 6064 EXCEPT THE Y SAMPLE AND HOLD IS LOADED

EXC EXECUTE MICROPROGRAMMED INSTRUCTIONS:

BIT	FUNCTION WHEN SET	NOT SET
2	See under 6063	
3	See under 6063	
4	Set execution to short vector speed.	Normal stored mode with speed set by bit 6
5	Spare	
6	Execute long vector unless inhibited by bit 4.	Execute short vector.
7	Set vector continue when flag is raised.	Vector continue disabled. Can also be disabled using 6062 or 6063.
8	Execute circle vector when enabled by bit 2.	Execute linear vector when enabled by bit 2.

BIT	FUNCTION WHEN SET	— NOT SET
9	Erase the scope.	No action
10	Reset integrators and sample/holds.	No effect
11	Intensify vector	Blank vector

The ready flag is cleared and will go high at the end of 30 μs for point plot, 250 μs for short vector, and 4 ms for long vector. The flag will also go high after 30 μs in vector continue, signaling the start of the vector.

Any logical combination of bits may be grouped together to form an execute command.

EXL — SAME AS 6066 EXCEPT THAT A LONG VECTOR WILL BE EXECUTED (UNLESS INHIBITED BY BIT 4) REGARDLESS OF THE STATE OF BIT 6.

SRF — SKIP IF READY FLAG IS HIGH:

CRF — CLEAR READY FLAG

SDA — SKIP IF D/A IS GREATER THAN (SELECTED) ANALOG COMPARISON SIGNAL

LDA — LOAD D/A

SECTION 7-6 MAGNETIC TAPE OPTIONS

DECTAPE SYSTEM

The DECtape system is a standard option for the PDP-8/I [PDP-8/L] which serves as an auxiliary magnetic tape data storage facility. The DECtape system stores information at fixed positions on magnetic tape, as in magnetic disk or drum storage devices, rather than at unknown or variable positions, as in conventional magnetic tape systems. This feature allows replacement of blocks of data on tape in a random fashion without disturbing other previously recorded information. In particular, during the writing of information on tape, the system reads format (mark) and timing information from the tape and uses this information to determine the exact position at which to record the information to be written. Similarly, in reading, the same mark and timing information is used to locate data to be played back from the tape.

The system has a number of features to improve its reliability and make it exceptionally useful for program updating and program editing applications. These features are: phase or polarity sensed recording on redundant tracks, bidirectional reading and writing, and a simple mechanical mechanism utilizing hydrodynamically lubricated tape guiding (the tape floats on air and does not touch any metal surfaces).

DECtape Format

DECtape utilizes a 10-track read/write head. Tracks are arranged in five non adjacent redundant channels: a timing channel, and three information channels. Redundant recording of each character bit on non adjacent tracks materially reduces bit drop outs and minimizes the effect of skew. The series-connection of corresponding track heads within a channel and the use of Manchester phase recording techniques, rather than amplitude sensing techniques, virtually eliminate drop outs.

The timing and mark channels control the timing of operations within the control unit and establish the format of data contained on the information channels. The timing and mark channels are recorded prior to all normal data reading and writing on the information channels. The timing of operations performed by the tape drive and some control functions is determined by the information on the timing channel. Therefore, wide variations in the speed of tape motion do not affect system performance. Information read from the mark channel is used during reading and writing data to indicate the beginning and end of data blocks and to determine the functions performed by the system in each control mode. During normal data reading, the control assembles 12-bit computer-length words from four successive lines read from the information channels of the tape. During normal data writing, the control disassembles 12-bit words and distributes the bits so they are recorded on four successive lines on the information channels. A mark-channel error-check circuit assures that one of the permissible marks is read in every six lines on the tape. This 6-line mark-channel sensing requires that data be recorded in 12-line segments (12 being the lowest common multiple of 6-line marks and 4-line data words) which correspond to three 12-bit words.

A tape contains a series of data blocks that can be of any length which is a multiple of three 12-bit words. Block length is determined by information on the mark channel. Usually a uniform block length is established over the entire length of a reel of tape by a program which writes mark and timing information at specific locations. The ability to write variable-length blocks is useful for certain data formats. For example, small blocks containing index or tag information can be alternated with large blocks of data. (Software supplied with DECtape allows writing for fixed block lengths only.)

Between the blocks of data are areas called interblock zones. The interblock zones consist of 30 lines on tape before and after a block of data. Each of these 30 lines is divided into five 6-line control words. These 6-line control words allow compatibility between DECtape written on any of DEC's 12, 18-, or 36-bit computers. As used on the PDP-8/I, [PDP-8/L] only the last four lines of each control word are used.

Block numbers normally occur in sequence from 1 to n. There is one block numbered 0 and one block n + 1. Programs are entered with a statement of the first block number to be used and the total number of blocks to be read or written. The total length of the tape is equivalent to 849,036 lines which can be divided into any number of blocks up to 4096 by prerecording of the mark track. The maximum number of blocks is determined by the following equation in which n_b equals number of blocks and n_w equals number of words per block (n_w must be divisible by 3).

$$n_b = \frac{212112}{n_w + 15} - 2$$

DECtape format is illustrated in Figures 7-3 through 7-6.

DECTAPE TRANSPORT (TYPE TU55) AND DECTAPE CONTROL (TYPE TC01)

A DECtape system configuration contains up to eight TU55 transports operated from one TC01 control. All data transfers occur between the computer and the control and are effected by the three-cycle data break facility. A 12-bit data buffer in the control synchronizes transfers between the TC01 and the PDP-8/I data break facility. [When the TC01 control is used with the PDP-8/L computer, the following interfacing units are required: the KD8/L data break facility, and the DW08-A I/O conversion panel.] Data read from four consecutive lines on tape by the transport are assembled into 12-bit words by a read/write buffer in the tape control. Data read from the computer is disassembled by the read/write buffer and is supplied to the transport for writing on four lines of tape. Transfer of command and control signals between the computer and the control is effected by normal IOT instructions. Small registers and control flip-flops in the TC01 are joined to serve as two status registers for the transfer of command and control information with the PDP-8/I [PDP-8/L] accumulator. Bit assignments of these registers are indicated in Figure 7-7 and Figure 7-8.

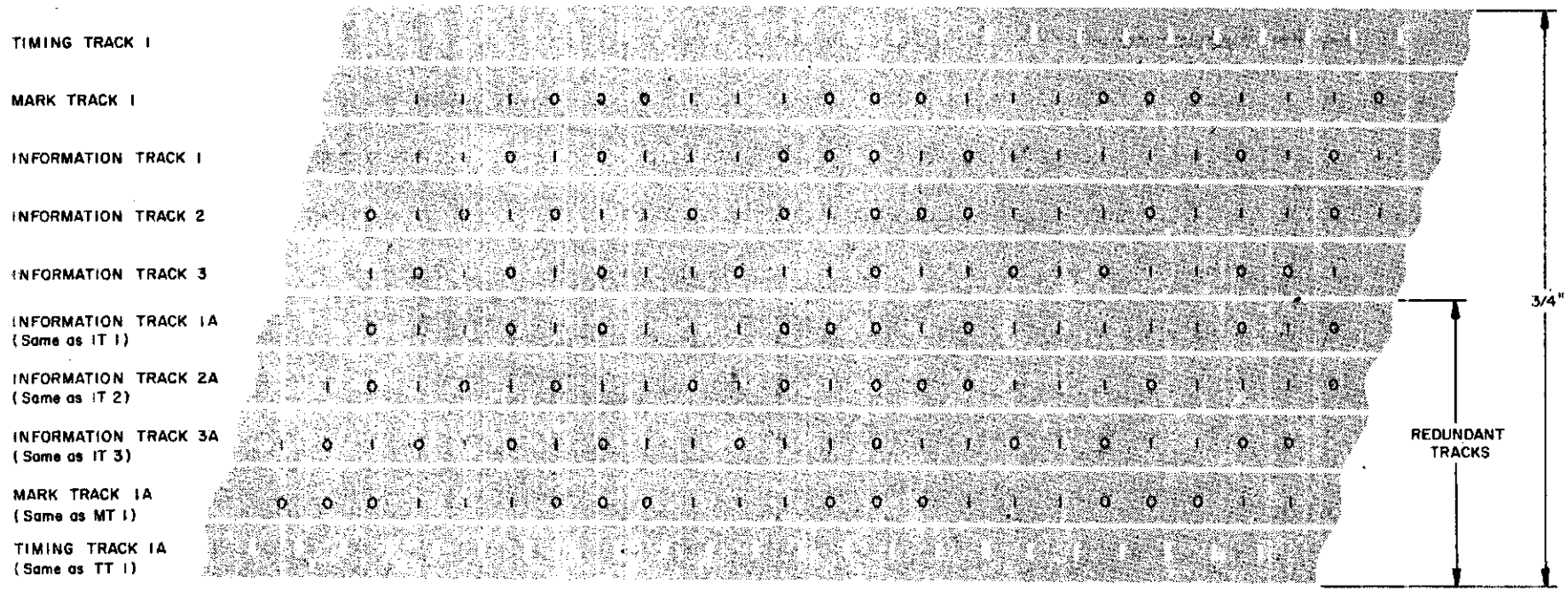


Figure 7-3. DECtape Track Allocations

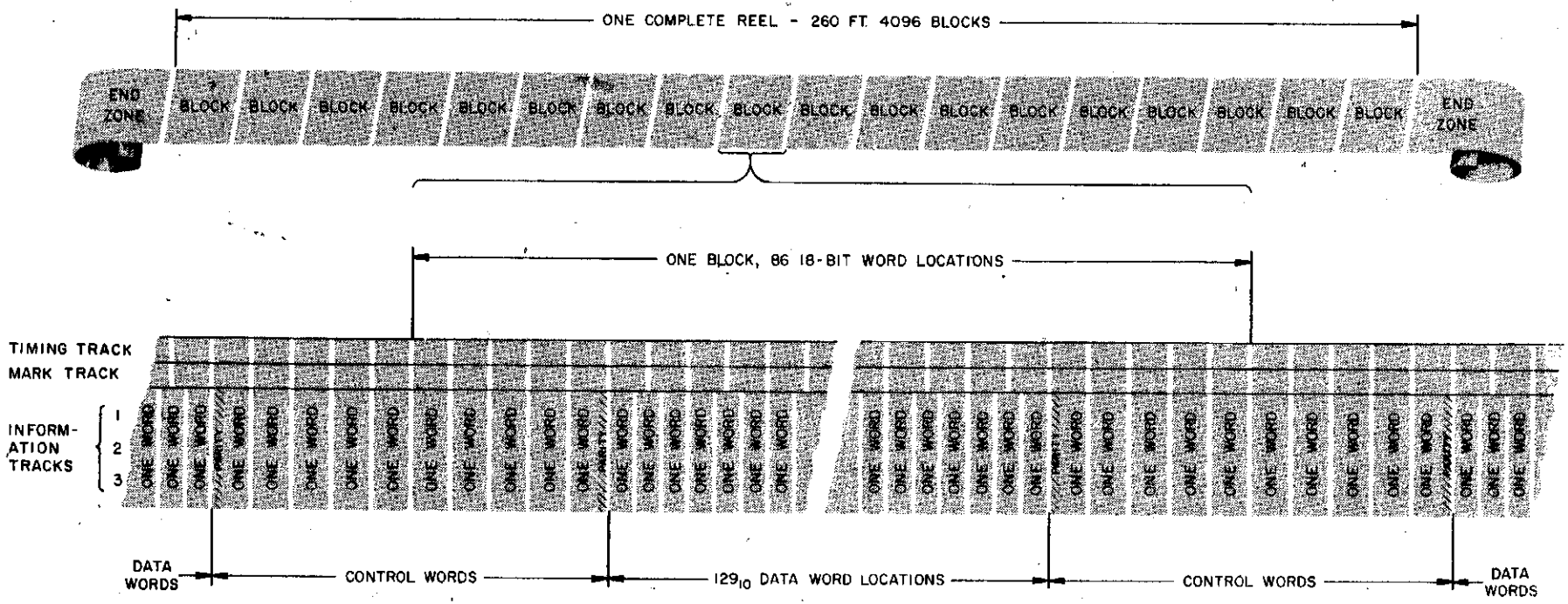


Figure 7-4. DECtape Mark Channel Format

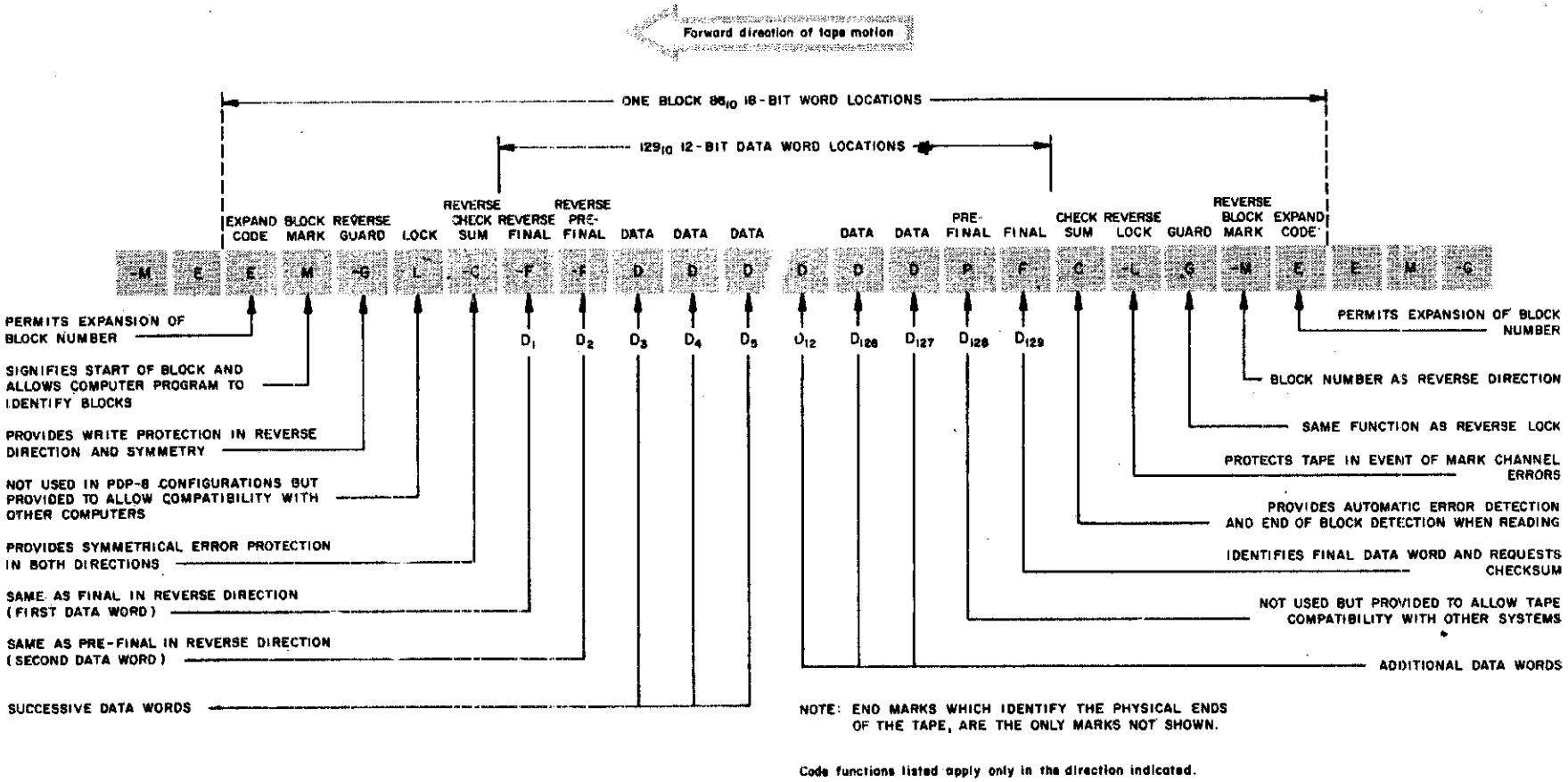


Figure 7-5. DECtape Control Word and Data Word Assignments

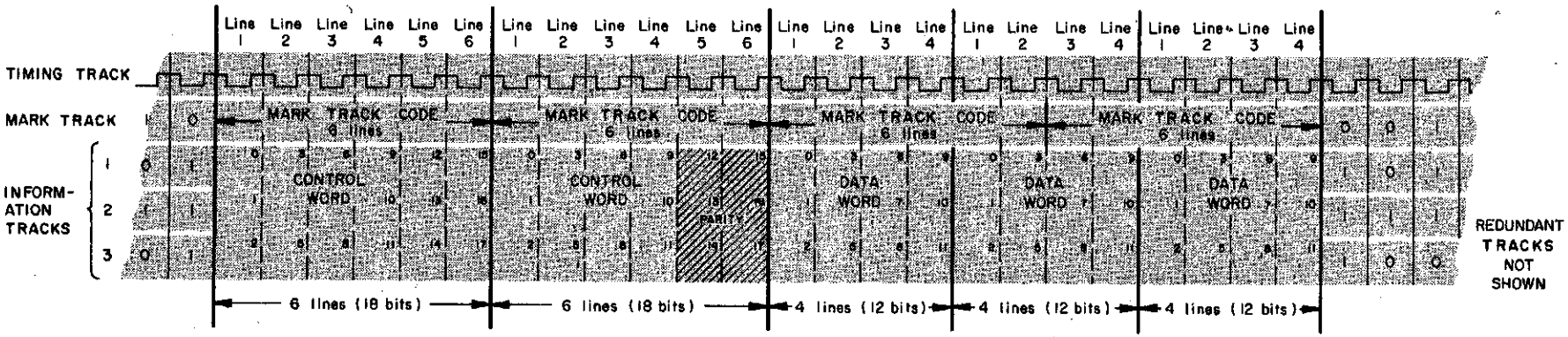


Figure 7-6. DECtape Format Details

DECtape Transport (Type TU55)

The TU55 is a bidirectional magnetic-tape transport consisting of a read/write head for recording and playback of information on five tape channels.

Connections from the read/write head are made directly to the external control, which contains the read and write amplifiers.

The logic circuits of the TU55 control tape movement in either direction over the read/write head. Tape drive motor control is exercised completely through the use of solid state switching circuits to provide fast reliable operation. These switching circuits contain silicon controlled rectifiers which are controlled by normal DEC logic circuits. These circuits control the torque of the two motors which transport the tape across the head according to the established function of the device; i.e., go, stop, forward, or reverse. In normal tape movement, full torque is applied to the forward or leading motor and a reduced torque is applied to the reverse or trailing motor to keep proper tension on the tape. Since tape motion is bidirectional, each motor serves as either the leading or trailing drive for the tape, depending upon the forward or reverse control status of the TU55. A positive stop is achieved by an electromagnetic brake mounted on each motor shaft. When a stop command is given, the trailing motor brake latches to stop tape motion. Enough torque is then applied to the leading motor to take up slack in the tape.

Tape movement can be controlled by commands originating in the computer or by manual operation of rocker switches on the front panel of the transport. Manual control is used to mount new reels of tape on the transport, or as a quick maintenance check for proper operation of the control logic in moving the tape.

Since DECtape is a fixed address system, the programmer need not know accurately where the tape has stopped. To locate a specific point on tape he must only start the tape motion in the search mode. The address of the block currently passing over the head will be automatically transferred to core where it can be compared with the desired block address and tape motion continued or reversed accordingly. TU55 typical time characteristics are provided below, but are not accurately controlled.

Start Time	200 ms*
Stop Time	200 ms*
Turn Around Time	275 ms*

*Also, see control specifications. These times are frequently lengthened by the particular control.

DECTAPE CONTROLS

Type TC01 Control

The TC01 DECtape Control operates up to eight TU55 DECtape Transports. Binary information is transferred between the tape and the computer in 12-bit computer words approximately every 133-1/3 μs. In writing, the control disassembles 12-bit computer words so that they are written at four successive lines on tape. Data transfers between the computer and the control always occur in parallel for a 12-bit word and require the use of the three-cycle data break facility. As the start and end of each block of data are detected by the mark track detection circuits, the control raises a DECtape control flag (DTCF) which requests a computer program interrupt. The program interrupt is used by the computer program to determine the block number. When it determines

that the forthcoming block is the one selected for a data transfer it establishes the appropriate read or write function. Each time a word is assembled or the DECtape system is ready to receive a word from the computer, the control raises a data flag (DF). This flag is connected to the computer data break facility to request a data break. Therefore, when each 12-bit computer word is assembled, the data flag causes a transfer via the three-cycle data break facility. By using the mark channel decoding circuits and the data break in this manner, computation in the main computer program can continue during DECtape operations.

Four program flags in the control serve as condition indicators and request originators.

a. DECtape Flag (DT): This flag indicates the active/done status of the current function.

b. Data Flag (DF): This flag requests a data break to transfer a block number into the computer during a search function, or when a data word transfer is required during a read or write function.

c. DECtape Control Flag (DTCF): This flag, when enabled by a binary 1 in bit 9 of status register A, requests a program interrupt if either the DECtape flag or the error flag is set and is connected to the instruction skip facility.

d. Error Flag (EF): Detection of any nonoperative condition by the control sets this flag in status register B and stops (except for parity errors) the selected transport. The error conditions indicated by this flag are:

(1) Mark Track Error: This error occurs any time the information read from the mark channel is erroneously decoded.

(2) End of Tape: The end zone on either end of the tape is over the read head.

(3) Select Error: This error occurs five μs after loading status register A to indicate any one of the following conditions:

(a) Specifying a unit select code which does not correspond to any transport select number, or which is set to multiple transports.

(b) Specifying a write function with the WRITE ENABLED/WRITE LOCK switch in the WRITE LOCK position on the selected transport.

(c) Specifying an unused function code (i.e., AC6-8 = 111).

(d) Specifying any function except read all with the NORMAL/WRTM/RDMK switch in the RDMK position. (The read all function is used to read tape in an unusal format.)

(e) Specifying any function except write timing and mark track with the NORMAL/WRTM/RDMK switch in the WRTM position.

(f) Specifying the write timing and mark track function with the NORMAL/WRTM/RDMK switch in a position other than WRTM.

(4) Parity Error: This error occurs during a read data function if the longitudinal parity over the entire data word, the reverse check character, and the checks character is not equal to 1.

(5) Timing Error: This error indicates a program fault caused by one of the following conditions:

(a) A data break did not occur within 66 μs ($\pm$30%) of the data break request.

(b) The DT flag was not cleared by the program before the control attempt to set it.

(c) The read data or write data function was specified while a data block was passing the read head.

Type TC08 Control

The TC08 control unit is very much like the TC01 control unit. This unit buffers and controls information transfers between from one to eight TU55 DECtape Transports and the PDP-8/I [PDP-8/L] computer. Variations of the TC08 control unit allow it to be connected directly onto either a PDP-8/I or PDP-8/L (negative and positive bus, respectively) computer.

During both input and output operations, the TC08 receives data and control information from the processor and generates the appropriate signals to the selected transport to execute the programmed commands. Binary information is transferred between the tape transport and the computer as one 12-bit word every 133-1/3 μs. When writing, the TC08 disassembles the 12-bit word into four successive 3-bit words to be written on tape. During read operations, the TC08 assembles the four successive 3-bit words into a single 12-bit word for transfer to the computer. Transfers between the computer and the control always occur in parallel for a 12-bit word. Data transfers use the 3-cycle data break (high-speed channel) facility of the computer.

Instructions for TC01 and TC08 Controls — Instructions for a TC01/TU55-TC08/TU55 system are microprogrammed commands of the PDP-8/I [PDP-8/L] IOT instruction and are defined as follows:

Read Status Register A (DTRA)

Octal Code: 6761
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of status register A is transferred into the accumulator by an OR transfer.

The AC is not cleared before the transfer. The AC bit assignments are:

AC0-2 = Transport unit select number
AC3(0) = Forward
AC3(1) = Reverse
AC4(0) = Stop
AC4(1) = Go
AC5(0) = Normal mode
AC5(1) = Continuous mode
AC6-8 = 0 = Move function
AC6-8 = 1 = Search function
AC6-8 = 2 = Read data function
AC6-8 = 3 = Read all function
AC6-8 = 4 = Write data function
AC6-8 = 5 = Write all function
AC6-8 = 6 = Write timing and mark tracks function
AC6-8 = 7 = Unused (causes a select error if issued)
AC9(0) = DECtape Control Flag (DTCF) and error flag disabled from causing a program interrupt
AC9(1) = DECtape Control Flag (DTCF) and error flag enabled to cause a program interrupt

Symbol: AC0-9 V Status Register A → AC0-9

Clear Status Register A (DCTA)

Octal Code: 6762
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Status register is cleared. The DECtape flag and error flags are undisturbed.
Symbol: 0 → Status Register A

Load Status Register A (DTXA)

Octal Code: 6764
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The exclusive OR of the content of bits 0 through 9 of the accumulator is loaded into status register A, and bits 10 and 11 of the accumulator are sampled to control clearing of the error and DECtape flags, respectively. Loading status register A from AC0-9 establishes the transport unit select code, motion control, function, and enables or disables the DECtape control flag to request a program interrupt as described in the DTRA instruction. The sampling of AC10 and AC11 is as follows:

AC10(0) = Clear all error flags
AC10(1) = All error flags undisturbed
AC11(0) = Clear DECtape flag
AC11(1) = DECtape flag undisturbed

The accumulator is cleared at the end of this instruction.
Symbol: AC0-9 V Status Register A → Status Register A
If AC10 = 0, then 0 → Error Flag
If AC11 = 0, then 0 → DT Flag
0 → AC

Skip on Flags (DTSF)

Octal Code: 6771
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of both the error flag and the DECtape flag are sampled, and if any flag contains a binary 1, the content of the program counter is incremented by one to skip the next sequential instruction.
Symbol: If EF Flag = 1 V DT Flag = 1, then PC + 1 → PC

Read Status Register B (DTRB)

Octal Code: 6772
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25 μs
Operation: The content of status register B is transferred into the accumulator by an OR transfer. The AC is not cleared before the transfer. The AC bit assignments are:

AC0 = Error Flag (flag — mark track error ∀ end of tape ∀ select error ∀ parity error ∀ timing error)
AC1 = Mark track error
AC2 = End of tape
AC3 = Select error
AC4 = Parity error
AC5 = Timing error
AC6-8 = Memory field
AC9-10 = Unused
AC11 = DECtape flag

Symbol: A∈ V Status Register B → AC

Load Status Register B (DTLB)
Octal Code: 6774
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The memory field register portion of status register B is loaded from the content of bits 6 through 8 of the accumulator. The accumulator is then cleared.
Symbol: AC6-8 → Memory Field, then 0 → AC

Three cycle data break locations: The TC01 uses location 7754 of field 0 for word count and 7755 of field 0 for current address.

Control Modes — The DECtape system operates in either the normal or continuous mode, as determined by bit 5 of status register A during a DTXA command. Operation in each mode is as follows:

a. Normal (NM): Data transfers and flag settings are controlled by the format of information on the tape.

b. Continuous (CM): Data transfers and flag settings are controlled by a word count read from core memory during the first cycle of each three-cycle data break, and by tape format.

Functions — The DECtape system performs one of seven functions, as determined by the octal digit loaded into status register A during a DTXA command. These functions are:

a. Move: Initiates movement of the selected transport tape in either direction. Mark channel decoding is inhibited in this mode except for end of tape.

b. Search: As the tape is moved in either direction, sensing of a block mark causes a data transfer of the block number. If the word count overflows in either NM or CM, the DT flag is set and causes a program interrupt. After finding the first block number, the CM can be used to avoid all intermediate interrupts between the current and the desired block number. This makes a virtually automatic search possible.

c. Read Data: This function is used to transfer blocks of data into core memory with the transfer controlled by the tape format. In NM, the DT flag is set at the end of a block and causes a program interrupt. In CM, transfers stop when the word count overflows, the remainder of the block is read for parity checking, and then the DT flag is set.

d. Read All: Read all is used to read tape in an unusual format, since it causes all lines to be read. In NM the DT flag is set at each data transfer. In CM the DT flag is set when WCO occurs. In either case the DT flag causes a program interrupt.

e. Write Data: This function is used to write blocks of data with the transfer controlled by the standard tape format. After word count overflow occurs, zeros are written in all lines of the tape to the end of the current block. Then the parity checksum for the block is written. The DT flag rises as in the read function.

f. Write All: The write all function is used to write an unusual tape format (e.g., block numbers). The DT flag raisings are similar to the read all function.

g. Write Timing and Mark Track: This function is used to write on the timing and mark tracks permitting blocks to be established or block lengths to be changed. The DT flag raisings are also similar to the read all function. This function is illegal unless a manual switch in the control is positioned on.

Programmed Operation — Prerecording of a reel of DECtape, prior to its use for data storage, is accomplished in two passes. During the first pass, the timing and mark channels are placed on the tape. During the second pass, forward and reverse block mark numbers, the standard data pattern, and the automatic parity checks are written. These functions are performed by the DECTOG program. Prerecording utilizes the write timing and mark channel function and a manual switch on the control, which permits writing on the timing and mark channels, activates a clock which produces the timing channel recording pattern, and enables flags for program control. Unless both this control function and switch are used simultaneously, it is physically impossible to write on the mark or timing channels. A red indicator lamp on the control panel lights when the manual NORMAL/WRTM/RDMK switch is in the WRTM position. Under these conditions only, the write register and write amplifier, used to write on information channel 1 (bits 0, 3, 6, and 9), are used to write on the mark channel. This prerecording operation need only be performed once for each reel of DECtape.

There are two registers in the TC01 DECtape Control that govern tape operation and provide status information to the operating program. Status register A (see Figure 7-7) contains three unit selection bits, two motion bits, the continuous mode/normal mode bit, three function bits, and three bits that control the flags. Status register B (see Figure 7-8) contains the three memory field bits and the error status bits. PDP-8/I [PDP-8/L] IOT microinstructions are used to clear, read, and load these registers. In addition, there is an IOT skip instruction to test control status.

Since all data transfers between DECtape and the computer memory are controlled by the data break facility, the program must set the WC and CA registers (locations 7754 and 7755, respectively) before a data break. After initiating a DECtape operation, the program should always check for error conditions (a program interrupt would be initiated if the error flag is enabled and if the program interrupt system is enabled). The DECtape system should be started in the search function to locate the block number selected for transfer and then, when the correct block is found, the transfer is accomplished by programmed setting of the WC, CA, and status register A.

When searching, the DECtape control reads block numbers only. These are used by the operating program to locate the correct block number. In NM, the DECtape flag is raised at each block number. In CM, the DECtape flag is raised only after the word count reaches zero. The current address is not

incremented during searching and the block number is placed in core memory at the location specified by the content of the CA. Data is transferred to or from the computer core memory from locations specified by the CA register; which is incremented by one before each transfer.

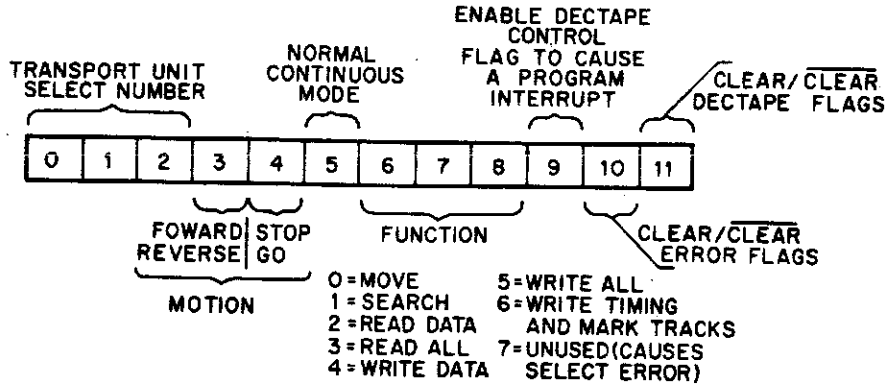


Figure 7-7 DECtape Status Register A-Bit Assignments

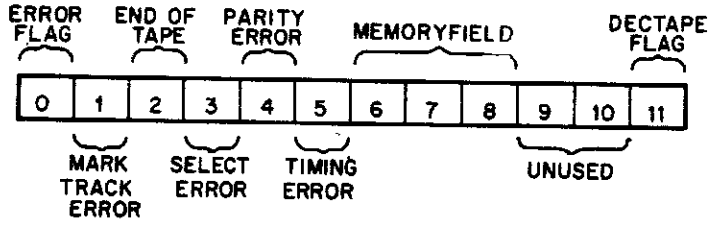


Figure 7-8 DECtape Status Register B-Bit Assignments

Each time the DECtape system is ready to transfer a 12-bit word, and when the start of the data position of the block is detected, the data flag is raised to initiate a data break request to the data break facility. Therefore, the main computer program continues running, but is interrupted approximately every 133⅓ μs for a data break to transfer a word. Transfers occur between DECtape and successive core memory locations specified by the CA. The initial transfer address minus one is stored in the CA by an initializing routine. The number of words transferred is determined by tape format in NM, or by tape format and the word count in CM. At the conclusion of the data transfer, the DT flag is raised and a program interrupt occurs. The interrupt subroutine checks the DECtape error bits to determine the validity of the transfer and either initiates a search for the next information to be transferred or returns to the main program.

During all normal writing transfers, a check character (the 6-bit logical equivalence of the words in the data block) is computed automatically by the control and is recorded automatically as one of the control words immediately following the data portion of the block. This same character is used during reading to determine that the data playback and recognition take place without error.

Any one of the eight tape transports may be selected for use by the program. After using a particular transport, the program can stop the transport currently being used and select another transport, or can select another transport while permitting the original selection to continue running. This is a particularly useful feature when rapid searching is desired, since several transports may be used simultaneously. Caution must be exercised however, for although the original transport continues to run, no tape-end detection or other sensing takes place. Automatic tape-end sensing that stops tape motion occurs in all functions, but only in the selected tape transport.

The following is a list of timing considerations for programmed operations. (n_s = the number of block numbers to be read in the search function and CM, counting through the one causing the word count overflow. Only the block number causing the word count overflow requests a program interrupt n_d = number of words transferred ÷ the number of words per block. If the remainder ≠ 0, use the next larger whole number. n_A = number of words transferred.)

Operation	Timing
Answer a data break request	Up to 66 μs ±30%
Word transfer rate	One 12-bit word every 133 μs ±30%
Block transfer rate	One 129-word block every 18.2 ms ±30%
Start time	375* ms ±20%
Stop time	375* ms ±20%
Turn around time	375* ms ±20%
Change function from search to read data for the current block after DT flag from block number	400 μs ±30%
Change function from search to write data for current block after DT flag from block number	400 μs ±30%
Change function from read data to search for the next block after DT flag from transfer completion.	1000 μs ±30%
Change function from write data to search for next block after DT flag from transfer completion	1000 μs ±30%
DECtape flag rises in continuous mode	
Move function	Never
Search function	(n_s) x (18.2 ms ±30%)
Read data function	(n_D) x (18.2 ms ±30%)
Read all function	(n_A) x (133 μs ±30%)
Write data function	(n_D) x (18.2 ms ±30%)
Write all function	(n_A) x (133 μs ±30%)
Write T & M function	(n_A) x (133 μs ±30%)
In normal mode	
Move function	Never
Search function	Every 18.2 ms ±30%
Read data function	Every 18.2 ms ±30%

*These times are typical, but are not accurately controlled.

Read all function	Every 133 μs ±30%
Write data function	Every 18.2 ms ±30%
Write all function	Every 133 μs ±30%
Write T & M function	Every 133 μs ±30%

Software — Four types of programs have been developed as DECtape software for the PDP-8/I [PDP-8/L]

a. Subroutines which the programmer may easily incorporate into a program for data storage, logging, data acquisition, data buffering (queuing), etc.

b. A library calling system for storing named programs on DECtape and a means of calling them with a minimal size loader.

c. System software which provides for storing, assembling and editing of programs on DECtape, thereby greatly increasing the versatility and flexibility of the PDP-8/I [PDP-8/L].

d. Programs for preformatting tapes controlled by the content of the switch register to write the timing and mark channels, to write block formats, to exercise the tape and check for errors, and to provide ease of maintenance.

Program development has resulted in a series of subroutines which read or write any number of DECtape blocks, read any number of 129-word blocks as 128 words (one memory page), or search for any block (used by read and write, or to position the tape). These programs are assembled with the user's program and are called by a JMS instruction. The program interrupt is used to detect the setting of the DECtape flag, thus allowing the main program to proceed while the DECtape operation is being completed. A program flag is set when the operation has been completed. Thus, the program effectively allows concurrent operation of several input/output devices along with operation of the DECtape system. These programs occupy two memory pages ($400_8 = 256_{10}$ words).

The library system has the following features: First, the computer state remains unchanged when it exits. Second, the library calls programs by name from the keyboard and allows for expansion of the program file stored on the tape. Finally, the library conforms to existing system conventions, namely, that all of memory except for the last memory page (7600_8-7777_8) is available to the programmer. The PDP-8/I [PDP-8/L] DECtape library system is loaded by a 17_{10}-instruction bootstrap routine that starts at address 7600_8. This loader calls a larger program into the last memory page, whose function is to preserve on the tape the content of memory from 6000_8 through 7577_8, and then load the INDEX program and the directory into those same locations. Since the information in this area of memory has been preserved, it can be restored when operations have been completed. The basic system tape contains the following programs:

a. INDEX: Typing this word causes the names of all programs currently on file to be typed out.

b. UPDATE: Allows the user to add a new program to the files. UPDATE queries the operator about the program's name, its starting address, and its location in core memory.

c. GETSYS: Generates a skeleton library tape on a specified DECtape unit.

d. DELETE: Causes a named file to be deleted from the tape.

Starting with the basic library tape, the user can build a complete file of his active programs and continuously update it. One of the uses of the library tape may be illustrated as follows:

The programmer may call the PDP-8/I [PDP-8/L] FORTRAN compiler from the library tape and with it compile the program, obtaining the object program. The FORTRAN operating system may then be called from the library tape and used to load the object program. At this time the library program UPDATE is called, the operator defines a new program file (consisting of the FORTRAN operating system and the object program), and adds it to the library tape. As a result, the entire operating program and the object program are now available on the DECtape library tape.

The DECtape system software is permanently stored on DECtape, from which it can be rapidly loaded. Any systems programs such as the assemblers (XPAL and XMACRO), the Symbolic Editor (XEDIT), or the Binary Loader (XLOAD) can be loaded in less than one minute.

The system software uses a standard DECtape format. There are 128 (200_8) words per block and 1464 (2701_8) blocks, so the user has the remaining 1336 blocks for rapid access storage of his own programs.

The primary advantages for users are:

a. Efficient use of high-speed transfer rates between DECtape and core memory.

b. Symbolic programs may now be stored, edited, and assembled on DECtape, greatly increasing the versatility and flexibility of the PDP-8/I [PDP-8/L].

c. The computational workload can be more than doubled, compared to high-speed paper tape systems.

User's programs are written exactly as before for assembly by the PAL or MACRO-8 Assemblers. Using the Symbolic Editor, source programs are typed directly onto DECtape. After assembly, fast symbolic debugging can be done with DDT-8 — after loading the program symbol table into DDT with the symbolic loader, XSYM.

The Binary Loader (XLOAD) can load the assembled binary program directly from the DECtape for program execution. Source files, symbol tables, and program listings can be stored on DECtape and listed later, if desired. A duplicating program, XDUP, is available for copying programs.

This DECtape system also includes system calls to load any program from DECtape, to update or delete source files, and to restore the system for use by another programmer.

Although the system operates with one DECtape, a two DECtape configuration is strongly recommended as it will permit duplication of programs and saving of back-up master tapes. In a single DECtape system, if the system library is accidentally destroyed, the stored data cannot be replaced immediately because there is no means of recovery.

The last group of programs, called DECTOG, is a collection of short routines controlled by the content of the switch register. It provides for the recording of timing and mark channels and permits block formats to be recorded for any block length. Patterns may be written in these blocks and then read and checked. Writing and reading is done in both directions and checked. Specified areas of tape may be "rocked" for specified periods of time. A given reel of tape may thus be thoroughly checked before it is used for data storage. These programs may also be used for maintenance and checkout purposes.

AUTOMATIC MAGNETIC TAPE CONTROL, TYPE TC58

Functional Description

The TC58 will control the operation of a maximum of eight digital magnetic tape transports, Types TU20C and TU20D. The TC58 interfaces to and uses the PDP-8/I 3-cycle data break facility (via the DM01 or DM04, if more than one device using the data break facility is part of the system), for data transfer directly between system core memory and magnetic tape. [When the TC58 control is used with the PDP-8/L computer, the following interfacing units are required: the KD8/L Data Break Facility and the DW08-A I/O Conversion Panel.] The tape transports offer industry-compatibility (or IBM-compatibility) in both 7- and 9-channel tape transport models with the following characteristics:

Transport	Tape Speed (ips)	Densities (bpi)
TU20D (7-channel)	45	200/556/800
TU20C (9-channel)	45	800

Transfers are governed by the in-memory WC and CA register associated with the assigned data channel (memory locations 7752_8 and 7753_8). Since the CA is incremented before each data transfer, its initial contents should be set to the desired initial address minus one. The WC is also incremented before each transfer and must be set to the two's complement of the desired number of data words to be transferred. In this way, the word transfer which causes the word count to overflow (WC becomes zero) is the last transfer to take place. The number of IOT instructions required for the TC58 is minimized by transferring all necessary control data (i.e., unit number, function, mode, direction, etc.) from the PDP-8/I [PDP-8/L] AC to the control using IOT instructions. Similarly, all status information (i.e., status bits, error flags, etc.) can be read into the AC from the control unit by IOT instructions.

During normal data reading, the control assembles 12-bit computer words from successive frames read from the information channels of the tape. During normal data writing, the control disassembles 12-bit words and distributes the bits so they are recorded on successive frames of the information channels.

Instructions

The commands for the magnetic tape control system are as follows:

Skip on Error Flag or Magnetic Tape Flag (MTSF)

Octal Code: 6701
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The status of the error flag (EF) and the magnetic tape flag (MTF) are sampled. If either or both are set to 1, the content of the PC is incremented by one to skip the next sequential instruction.
Symbol: If MTF or EF = 1, PC + 1 → PC

Skip on Tape Control Ready (MTCR)

Octal Code: 6711
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: If the tape control is ready to receive a command, the PC is incremented by one to skip the next sequential instruction.
Symbol: If Tape Control Ready, PC + 1 → PC

Skip on Tape Transport Ready (MTTR)

Octal Code: 6721
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The next sequential instruction is skipped if the selected tape transport is ready.
Symbol: If tape unit ready, PC + 1 → PC

Clear Registers, Error Flag and Magnetic Tape Flag (MTAF)

Octal Code: 6712
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears the status and command registers, and the EF and MTF if tape control is ready. If tape control is not ready, clears MTF and EF flags only.
Symbol: If tape control is ready, 0 → MTF, 0 → EF, 0 → command register
If tape control is not ready, 0 → MTF, 0 → EF

Inclusive OR Contents of Command Register (MTRC)

Octal Code: 6724
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Inclusively OR the contents of the command register into AC0-11.
Symbol: AC V command register → AC

Inclusive OR Contents of Accumulator (MTCM)

Octal Code: 6714
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Inclusively OR the contents of AC0-5, AC9-11 into the command register; jam transfer AC6-8 (command function).
Symbol: AC05, AC9-11 V command register → command register
AC6-8 → command register bits 6-8

Load Command Register (MTLC)

Octal Code: 6716
Event Time: 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Load the contents of AC0-11 into the command register
Symbol: AC0-11 → command register

Inclusive OR Contents of Status Register (None)

Octal Code 6704
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Inclusively OR the contents of the status register into AC0-11.
Symbol: Status Register V AC → AC0-11

Read Status Register (MTRS)

Octal Code: 6706

Event Time: 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Read the contents of the status register into AC0-11.
Symbol: Status Register → AC0-11

Mag Tape "GO" (MTGO)

Octal Code: 6722
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Set "GO" bit to execute command in the command register if command is legal.
Symbol: None

Clear the AC (None)
Octal Code: 6702
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clear the accumulator.
Symbol: 0 → AC

Although any number of tapes may be simultaneously rewinding, data transfer may take place to or from only one transport at any given time. In this context, data transfer includes these functions: read or write data, write EOF (end-of-file), read/compare, and space. When any of these functions are in process, the tape control is in the not-ready condition. A transport is said to be not ready when the tape is in motion, when transport power is off, or when it is off-line.

Data transmission may take place in either parity mode; add for binary, even for BCD. When reading a record in which the number of characters is not a multiple of the number of characters per word, the final characters come into memory left-justified.

Ten bits in the magnetic tape status register retain error and tape status information. Some error types are combinations, such as lateral and longitudinal parity errors (parity checks occur after both reading and writing of data), or have a combined meaning, such as illegal command, to allow for the maximum use of the available bits.

The magnetic tape status register reflects the state of the currently selected tape unit. Interrupts may occur only for the selected unit. Therefore, other units which may be rewinding, for example, will not interrupt when done.

Magnetic Tape Functions

For all functions listed below, upon completion of the data operation (after the end-of-record character passes the read head), the MTF (magnetic tape flag) is set, an interrupt occurs (if enabled), and errors are checked.

No Operation — NO OP command defines no function in the command register. A MTGO instruction with NO OP will cause an illegal command error (set EF).

Space — There are two commands for spacing records, SPACE FORWARD and SPACE REVERSE. The number of records to be spaced (two's complement) is loaded into the WC. The CA need not be set. The MTF (magnetic tape flag)

is set, and an interrupt occurs at WC overflow, EOF (end of file), or EOT (end of tape), whichever occurs first. When issuing a space command, both the density and parity bits must be set to the density and parity in which the records where originally written.

Load Point or Beginning of Tape (BOT) detection during a backspace terminates the function with the BOT bit set. If a SPACE REVERSE command is given when a transport is set at BOT, the command is ignored, the illegal command error and BOT bits are set, and an interrupt occurs.

Read Data — Records may be read into memory only in the forward mode. Both CA and WC must be initialized CA, to the initial core address minus one; WC, to the two's complement of the number of words to be read. Both identify and parity bits must be set.

If WC is set to less than the actual record length, only the desired number of words are transferred into memory. If WC is greater than or equal to the actual record length, the entire record is read into memory. In either case, both parity checks are performed, the MTF is set, and an interrupt occurs when the end-of-record (EOR) mark passes the read head. If either lateral or longitudinal parity errors or bad tape have been detected, or an incorrect record length error occurs (WC not equal to the number of words in the record), the appropriate status bits are set. An interrupt occurs only when the MTF is set.

To continue reading without stopping tape motion, MTAF (clear MTF) and MTGO instructions must be executed. If the MTGO command is not given before the shut down delay terminates, the transport will stop.

Write Data — Data may be written on magnetic tape in the forward direction only. For the WRITE DATA function, the CA and WC and density and parity bits must be initialized. WRITE DATA is controlled by the WC, such that when the WC overflows, data transfer stops, and the EOR character and IRG (interrecord gap) are written. The MTF is set after the EOR has passed the read head. To continue writing, a MTGO command must be issued before the shut down delay terminates. If any errors occur, the EF will be set when the MTF is set.

A special feature of this control is the write extended interrecord gap capability. This occurs on a write operation when command register bit 5 is set. The effect is to cause a 3-inch interrecord gap to be produced before the record is written. The bit is automatically cleared when the writing begins. This is very useful for creating a 3-inch gap of blank tape over areas where tape performance is marginal.

Write EOF — The WRITE EOF command transfers a single character (17_8) record to magnetic tape and follows it with an EOR character. CA and WC are ignored for WRITE EOF. The density bits must be set, and the command register parity bit should be set to even (BCD) parity. If it is set to odd parity, the control will automatically change it to even.

When the EOF marker is written, the MTF is set and an interrupt occurs. The tape transport stops, and the EOF status bit is set, confirming the writing of EOF. If odd parity is required after a WRITE EOF, it must be specifically requested through the MTLC command.

Read/Compare — The READ/COMPARE function compares tape data with core memory data. It can be useful for searching and positioning a magnetic tape to a specific record, such as a label or leader, whose content is known in core memory, or to check a record just written. READ/COMPARE occurs in the

forward direction only; CA and WC must be set. If there is a comparison failure, incrementing of the CA ceases, and the READ/COMPARE error bit is set in the status register. Tape motion continues to the end of the record; the MTF is then set and an interrupt occurs. If there has been a READ/COMPARE error, examination of the CA reveals the word that failed to compare.

Rewind — The high-speed REWIND command does not require setting of the CA or WC. Density and parity settings are also ignored. The REWIND command rewinds the tape to loadpoint (BOT) and stops. Another unit may be selected after the command is issued and the rewind is in process. MTF is set, and an interrupt occurs (if the unit is selected) when the unit is ready to accept a new command. The selected unit's status can be read to determine or verify that REWIND is in progress.

Continued Operation

To continue operating in the same mode, the MTGO instruction is given before tape motion stops. The order of commands required for continued operation is as follows:

a. MTCM, if the command is to be changed.

b. MTAF, will only clear MTF and EF flags since tape control will be in a not ready state.

c. MTGO, if MTCM requested an illegal condition, the EF will be set at this time.

To change modes of operation, either in the same or opposite direction, the MTCM command is given to change the mode and an MTGO command is given to request the continued operation of the drive. If a change in direction is ordered, the transport will stop, pause, and automatically start up again.

If the WRITE function is being performed, the only forward change in command that can be given is WRITE EOF.

If no MTGO instruction is given, the transport will shut down in the inter-record gap.

NOTE

No flags will be set when the control becomes ready or the transport becomes ready, except if the REWIND command is present in the command register and the selected drive reaches BOT and is ready for a new command.

If a WRITE (odd parity) command is changed to WRITE EOF, the parity is automatically changed to even.

NOTE

Even parity will remain in the command register unless changed by a new command instruction, MTLC, which clears and loads the entire command register.

Status or Error Conditions

Twelve bits in the magnetic tape status register indicate status or error conditions. They are set by the control and cleared by the program.

The magnetic tape status register (STR) bits are:

BIT*	FUNCTION (WHEN SET)
0	Error Flag (EF)
1	Tape rewinding
2	Beginning of tape (BOT)
3	Illegal command
4	Parity error (Lateral or Longitudinal)
5	End of file (EOF)
6	End of tape (EOT)
7	Read/compare error
8	Record length incorrect WC = 0 (long) WC ≠ (short)
9	Data request late
10	Bad tape
11	Magnetic tape flag (MTF) or job done

*The register bits are equivalent in position to the AC bits (i.e., STR 0 = AC 0, etc.).

MTF (STR11) — The MTF flag is set under the following conditions:

a. Whenever the tape control has completed an operation (after the EOR mark passes the read head).

b. When the selected transport becomes ready following a normal REWIND function.

These functions will also set the EF if any errors are present.

EOF (STR5) — EOF is sensed and may be encountered for those functions which come under the heading of READ STATUS FUNCTION; i.e., SPACE, READ DATA, or READ/COMPARE and WRITE EOF. When EOF is encountered, the tape control sets EOF = 1. MTF is also set; hence, an interrupt* occurs and the EOF status bit may be checked.

*All references to interrupts assume the tape flags have been enables to the interrupt (command register bit 9 = 1) and that the unit is selected.

EOT (STR6) and BOT (STR2) — EOT detection occurs during any forward command when the EOT reflective strip is sensed. When EOT is sensed, the EOT bit is set, but the function continues to completion. At this time the MTF is set (and EF is set), and an interrupt occurs.

BOT detection status bit occurs only when the BOT reflective strip is read on the transport that is selected.

When BOT detection occurs and the unit is in reverse, the function terminates. If a tape unit is at load point when a REVERSE command is given, an illegal command error bit is set causing an EF with BOT set. An interrupt then occurs.

Illegal Command Error (STR3) — The illegal command error bit is set under the following conditions:

a. A command is issued to the tape control with the control not ready.

b. A MTGO command is issued to a tape unit which is not ready, and the tape control is ready.

c. Any command which the tape control, although ready, cannot perform; for example:

(1) WRITE with WRITE LOCK condition

(2) 9-channel tape and incorrect density

(3) BOT and SPACE REVERSE

Parity (STR4) — Longitudinal and lateral parity checks will occur in both reading and writing. The parity bit is set for either lateral or longitudinal parity failure. A function is not interrupted, however, until MTF is set. Maintenance panel indicators are available to determine which type of parity error occurred.

Read/Compare Error (STR7) — When READ/COMPARE function is underway, STR7 is set to 1 for a READ/COMPARE error (see earlier portion of this section on READ/COMPARE for further details).

Bad Tape (STR10) — A BAD TAPE error indicates detection of a bad spot on the tape. Bad tape is defined as three or more consecutive missing characters followed by data, within the period defined by the shutdown delay. The error bit is set by the tape control when this occurs. MTF and interrupt do not occur until the end of the record in which the error was detected.

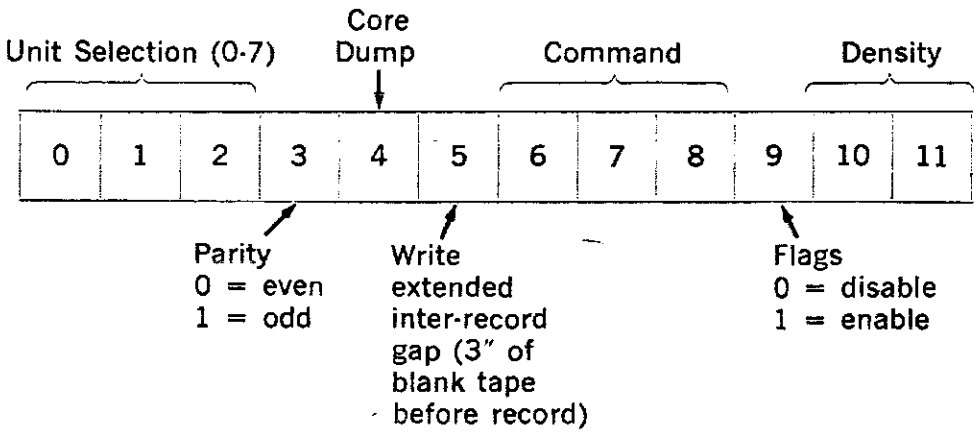
Tape Rewinding (STR1) — When a REWIND command has been issued to a tape unit and the function is underway, the tape rewinding bit is set in the control. This is a transport status bit, and any selected transport which is in a high-speed rewind will cause this bit to be set.

Record Length Incorrect (STR8) — During a READ or READ/COMPARE, this bit is set when the WC overflow differs from the number of words in the record. The EF flag is set.

Data Request Late (STR9) — This bit can be set whenever data transmission is in progress. When the data flag causes a data break cycle, the data must be transmitted before a write pulse or a read pulse occurs. If it does not, this error will occur, and data transmission will cease. The EF flag and bit 9 of the status register are set when the MTF is set.

Error Flag (STR0) — EF is set whenever any error status bit is present at the time that MTF is set. However, when an illegal command is given, the EF is set and the MTF is not set.

Command Register Contents



Unit Selection

Unit	Unit Selection Bits 0	1	2
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Density	Density Selection Density Bits
200 bpi	0 0
556 bpi	0 1
800 bpi	1 0
800 bpi 9 channel	1 1

Command Selection

COMMAND	BITS 6	7	8
NO OP	0	0	0
Rewind	0	0	1
Read	0	1	0
Read/Compare	0	1	1
Write	1	0	0
Write EOF	1	0	1
Space Forward	1	1	1
Space Reverse	1	1	1

Magnetic Tape Function Summary

Legend: CA = Current Address Register = 7752_8
WC = Word Count Register = 7753_8
F = Forward
R = Reverse
DS = Density Setting
PR = Parity Setting
EN = Enable Interrupt

Function	Characteristics	Status of Error Types
NO-OP	CA: Ignored WC: Ignored DS: Ignored PR: Ignored EN: Ignored	Illegal BOT Tape Rewinding
SPACE FORWARD	CA: Ignored WC: two's complement of number of records to skip DS: Must be set PR: Must be set EN: Must be set	Illegal EOF Bad Tape MTF, BOT, EOT
SPACE REVERSE	Same as Space Forward	Illegal EOF Bad Tape BOT MTF

READ DATA	CA: Core Address −1 WC: two's complement of number of words to be transferred DS: Must be set PR: Must be set EN: Must be set	Illegal EOF Parity Bad tape MTF EOT Data Request Late Record Length Incorrect
WRITE DATA	Same as READ DATA	Illegal EOT Parity MTF Bad Tape
WRITE EOF	CA: Ignored WC: Ignored DS: Must be set PR: Must be set EN: Must be set	Data Request Late Same as WRITE DATA plus EOF
READ/COMPARE	Same as READ DATA	Illegal EOF Read/Compare Error Bad Tape MTF EOT Data Late Record Length Incorrect
REWIND	CA: Ignored WC: Ignored DS: Ignored PR: Ignored EN: Must be set	Illegal Tape Rewinding MTF BOT

MAGNETIC TAPE TRANSPORT, TYPE TU20D (7-CHANNEL)

The TU20D is a digital magnetic tape transport designed to be compatible with the Type TC58 Magnetic Tape Control. The transport operates at a speed of 45 inches per second (ips) and has three selectable densities: 200, 556, and 800 bpi. The maximum transfer rate is 36,000 6-bit characters per second. Standard 7-channel, IBM-compatible tape format is used. The specifications for the unit are as follows:

Format: NRZI. Six data bits plus one parity bit.
End and loadpoint sensing compatible with IBM 729 I-VI.

Tape: Width of 0.5 in. Length of 2400 ft. (1.5 mil.) Reels are 10.5 in., IBM-compatible, with file protect (WRITE LOCK) ring.

Heads: Write-read gap of 0.300 in. Dynamic and static skew is less than 14 μs.

Tape Specifications: 45 ips. Start time is less than 5 ms. Start distance is 0.080 in. (+0.035, −0.025 in.). Stop time is less than 1.5 ms. Stop distance is 0.045 in. (±0.015 in.). Rewind −2400 feet in less than 3 min.

Density: 200, 556, and 800 bpi. Maximum transfer rate is 36 kHz.
Transport Mechanism: Pinch roller drive; vacuum column tension.

Controls: ON/OFF, ON LINE, OFF LINE, FORWARD, REVERSE, REWIND, LOAD, RESET.

Physical Specifications: Width of 22-1/4 in., depth of 27-1/6 in., height of 69-1/8 in. Weight — 600 lb.

Read (Read/Compare) Shutdown Delay: 3.6 ms.
Write Shutdown Delay: Approximately 4.5 ms.

MAGNETIC TAPE TRANSPORT, TYPE TU20C (9-CHANNEL)

The TU20C is a digital magnetic tape transport designed to be compatible with the TC58 Magnetic Tape Control. The transport operates at a speed of 45 (ips) and a density of 800 bpi. The maximum transfer rate is 36,000 8-bit characters per second. Standard 9-channel, IBM-compatible tape format is used. The specifications for the unit are as follows:

Format: NRZI. Eight data bits plus one parity bit.

End and loadpoint sensing compatible with IBM.

Tape: Width of 0.5 in. Length of 2400 ft. (1.5 mil). Reels are 10.5 in., IBM-compatible, with file protect (WRITE LOCK) ring.

Heads: Write-read gap of 0.150 in. Dynamic and static skew is less than 14 μs.

Tape Specifications: 45 ips. Start time is less than 5 ms. Start distance is 0.080 in. (+0.035, −0.025 in.) Stop time is less than 1.5 ms. Stop distance is 0.045 in. (+0.015 in.)

Density: 800 bpi. Maximum transfer rate is 36 kHz.
Transport Mechanism: Pinch roller drive; vacuum column tension.

Controls: ON/OFF, ON LINE, OFF LINE, FORWARD, REVERSE, REWIND, LOAD, RESET.

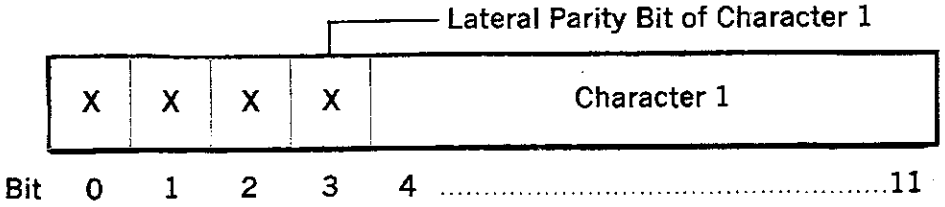
Physical Specifications: Width of 22-1/4 in., depth of 27-1/6 in., height of 69-1/8. Weight — 600 lb.

Read (Read/Compare) Shutdown Delay: 3.6 ms.
Write Shutdown Delay: Approximately 4.5 ms.

9-Track Operation

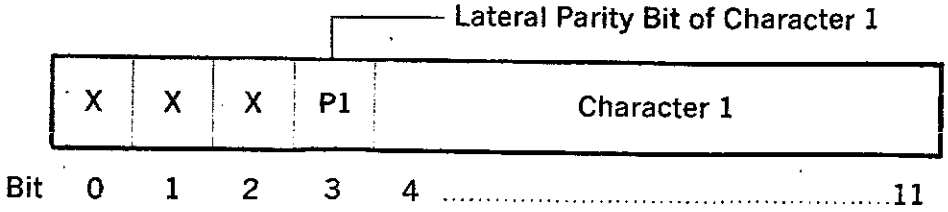
Nine and seven track transports may be intermixed on the TC58 control. When a transport is selected, it automatically sets the control for proper operation with its number of tracks.

Control of 9-track operation is identical to 7-track, except as noted below:
Write — A word in memory is written on tape in the following format.



X — These bits are ignored

Read — A word is read into memory from tape in the following format.

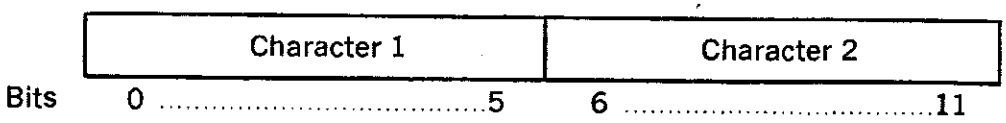


X — These bits are ignored

Read/Compare — A direct comparison of the characters on tape is made with those in memory. The parity bit is ignored, as are bits 0-3 in each memory word.

Core Dump Mode — This mode is used only with 9-track transports. It is entered by setting bit 4 of the command register.

Core dump mode permits the dumping of complete memory words in the form of two 6-bit characters. The format is:



This is accomplished by utilizing only 7 of the 9 tape tracks.

Tape written in core dump mode, must be read (read/compare) in the same mode. These operations are the same as for a 7-track transport.

INCREMENTAL MAGNETIC TAPE

Other magnetic tape options include the TR02 Tape Transport Control which is used with the TU22, TU25, or TU28 series of incremental write, synchronous read transports. The TR02, with either one or two of these transports, may be used with the PDP-8/I or PDP-8/L computers.

Incremental writing allows the user considerable flexibility in the acquisition of data, data reduction, and data storage. Data may be written one word at a time, eliminating the need for a buffer memory for storage of a block of data as is required with synchronous write units. Data may be transferred in entire or partial records, as the user deems necessary.

Tape format is either 7- or 9-track, dependent upon the transport used with the TR02. Tapes writen on the TR02 transport combination are compatible with industry-standard synchronous systems, allowing data acquired with this system to be utilized by another computer installation. Parity may be either even or odd, and is selected by two switches, one in the TR02 and one in the transport. (Both switches must be set to the same parity.) Another switch provides the option of reading tapes with 200, 556 or 800 bpi. The TR02 may be supplied as a single transport, or two transport unit.

Type TR02 Tape Transport Control

The four standard models of the TR02 are listed in Table 7-4.

Table 7-4. TR02 Tape Control Models

Control	Central Processor I/O Bus Polarity	Maximum Number of Transports
TR02-PA	Positive	1
TR02-PC	Positive	2
TR02-NA	Negative	1
TR02-NC	Negative	2

All models have 7/9 channel capacity

The TR02 control unit is designed to perform a programmed data transfer between the PDP-8/I [PDP-8/L] and an incremental-write and synchronous-read tape transport (7 or 9 channels). The control unit works directly off programmed I/O; the data break facility is not required and the interface requires no operator control.

The TR02 contains the electronics which decode the IOT instructions and generate the move commands. It also receives data from the transport to generate interrupt flags and to operate a status register. Level converters in the TR02 dynamically buffer the data between the central processor buffered accumulator (BAC) lines and the transport buffer during write, and between the transport buffer and the central processor accumulator input (AC) lines during read. The transport contains a static data buffer which stores data in either the read or write direction. The move commands are interpreted by the transport to provide the desired movement. Read/write selection is generated by the TR02 to provide the read/write command to the transport. The transport electronics generate all signals required to actually write data on the tape or read it from the tape, and generate the check characters for the interrecord gap or the end-of-file gap.

The TR02 can operate with either positive or negative central processor I/O busses, as selected by the logic modules installed. All models of the TR02 can operate with either 7- or 9-track tape transports.

The TR02 has an operating temperature range of 0°C to 50°C and requires the following power input specifications: 115 Vac (±10%), 50/60 Hz, 60W.

Optional Tape Transports

Table 7-5 lists the characteristics of the three optional types of tape transports used with the TR02 tape control.

Table 7-5. Summary of Tape Transport Characteristics

Incremental Writing Rate	— 700 steps/sec.
Synchronous Read Speed	— 25 in./sec. (200 and 556 bpi) or 15 in. per sec. (800 bpi).
Tape	— 0.5 in./1.5 mil (computer grade)
Reel Size	— 8.5 or 10.5 in. diameter (IBM compatible)

Mechanical Packaging		8.5 in.	10.5 in.
	Width	19.0 in.	19.0 in.
	Height	12.25 in.	24.5 in.
	Depth	11.7 in.	11.7 in. (from mounting surface 14.0 total in.).

Incremental Write Synchronous Read Magnetic Tape Transports

Transport Type	Density	Number of Tracks	Reel Size	Read Speed
TU22-A	200 bpi	7 track	10-1/2 in.	25 ips
TU25-A	556 bpi	7 track	10-1/2 in.	25 ips
TU28-A	800 bpi	7 track	10-1/2 in.	15 ips
TU28-C	800 bpi	9 track	10-1/2 in.	15 ips
TU22-E	200 bpi	7 track	8-1/2 in.	25 ips
TU25-E	556 bpi	7 track	8-1/2 in.	25 ips
TU28-E	800 bpi	7 track	8-1/2 in.	15 ips
TU28-H	800 bpi	9 track	8-1/2 in.	15 ips

Program Instructions

Operation of the tape transport is controlled by seven IOT instructions. Units controlling two transports have two identical sets of instructions, differentiated by the contents of bit 8 of the instruction code. The instructions are in the series of 670X for the first transport and 671X for the second. When two transports are used, the mnemonics for the second transport are differentiated by suffixing the letter A to the basic instruction.

The seven IOT instructions are selected by the last three bits of the IOT instruction (bits 9-11), which also select the generation of the IOP pulses. Each instruction is decoded as a unique instruction by the TR02; therefore, microprogramming of the instruction is not possible.

Skip on Read Done (IRS) (IRSA)

Octal Code: 6701/6711
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: When data is ready in the read buffer (RB) to be strobed into the AC, the PC is incremented by one to skip the next sequential instruction. The read done flag is cleared only if the skip occurs.
Symbol: If RB data is ready, PC + 1 → PC, RD flag, 0 → RD flag

Read Status Register (ISR) (ISRA)

Octal Code: 6702/6712
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the status register (STR) is read into AC0-8. The AC should be cleared before it is read by this instruction. (See Figure 7-9 and Table 7-6 for status conditions.)
Symbol: AC_{0-8} VSTC → AC_{0-8}

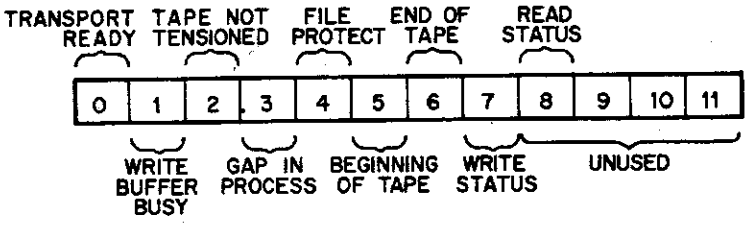


Figure 7-9. Read Status Register Conditions

Table 7-6. Conditions of Read Status Register

Bit	Functions
0	Transport Ready
1	Write Buffer Busy
2	Tape Not Tensioned
3	Gap in Process
4	File Protect (selected by ring on tape reel)
5	Beginning of Tape
6	End of Tape
7	Write Status
8	Read Status

Logic 1 in AC = function true

Skip on Write Done (IWS) (IWSA)

Octal Code: 6703/6713
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: If the write done flag is set, the next instruction is skipped and the write done flag is cleared.
Symbol: If Write Done flag = 1, PC + 1 → PC, 0 → Write Done

Move Commands (IMC) (IMCA)

Octal Code: 6704/6714
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The move command decoded from AC0-2 is generated. This instruction also clears the read done, write done, and gap detect flags. The indicated flag is set when the command has been executed.

Command	Command Selection AC Bits 0	1	2	Octal Code	Flag
REWIND	0	0	0	0000	Write Done
WRITE INTERRECORD GAP	0	0	1	1000	Write Done
WRITE END-OF-FILE GAP	0	1	0	2000	Write Done
SET UP TO WRITE AFTER READ	0	1	1	3000	Write Done
LOAD FORWARD READY TO READ	1	0	0	4000	Gap Detect
LOAD FORWARD READY TO WRITE	1	0	1	5000	Write Done
READ FORWARD	1	1	0	6000	Gap Detect
BACK SPACE ONE RECORD	1	1	1	7000	Gap Detect

Symbol: None

Skip on Gap Detect (IGS) (IGSA)

Octal Code: 6705/6715
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: If the gap detect flag is set, the next instruction is skipped and the gap detect flag is cleared.
Symbol: If Gap Detect flag = 1, PC + 1 → PC, 0 → Gap Detect flag

Write AC Into Tape Buffer and Generate Write Step (IWR) (IWRA)

Octal Code: 6706/6716
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The contents of the AC are loaded into the tape input data buffer (WB) and a write step command is generated. The write done flag is set when writing is completed (1.43 ms maximum).
Symbol: AC6-11 → WB (7 track) or AC4-11 → WB (9 track)
Write command → transport

Read Buffer Into AC (IRD) (IRDA)

Octal Code: 6707/6717
Event Time: 1, 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The AC is cleared and the content of the read buffer (RB) is loaded into the AC. Data bits are transferred into AC6-11 (7 track) or AC4-11 (9 track). Parity error is transferred into AC0 and is 0 if there is no parity error.

The maximum time available for character processing is a function of density and tape speed as follows:

Density	Tape Speed	Time
200 bpi	25 ips	100 μs
556 bpi	25 ips	36 μs
800 bpi	15 ips	42 μs

Symbol: 0 → AC, RB, PE V AC → AC

SECTION 7-7 RANDOM ACCESS DISK DEVICES

TYPE DF32 DISK FILE AND CONTROL AND TYPE DS32 EXPANDER DISK FILE

The DF32 Disk File is a fast, low-cost, random-access, bulk-storage device and control for use with the PDP-8/I [PDP-8/L] computer. [When the DF32/DS32 are being used with the PDP-8/L, the following interface options are also required: the KD8/L Data Break Facility and the DW08-A I/O Conversion Panel.] Operating through the 3-cycle Data Break Facility, the DF32 provides 32,768 13-bit words (12 bits plus parity) of storage, and is economically expandable to 131,072 words when using the DS32 Expander Disk.

Transfer rate of the DF32 is 66 μs per word; average access time is 16.67 ms for 60 Hz power (20 ms with 50 Hz power).

Two basic assemblies comprise the DF32: the storage unit with read/write electronics, and computer interface logic. The storage unit contains a nickel-cobalt-plated disk, driven by a hysteresis synchronous motor. Data is recorded on a single disk surface by 16 fixed-position read/write heads. A photo-reflective marker is used on the disk's outer perimeter to denote beginning and end of timing and address tracks.

Disk motor and shaft, read/write data heads, timing and address heads, and photocell assembly are mounted on a 19-inch relay rack assembly, which permits easy access to the unit by sliding the unit in and out of a standard Digital Equipment Corporation cabinet.

Instructions

The commands for the disk system are as follows:

Clear Disk Memory Address Register (DCMA)

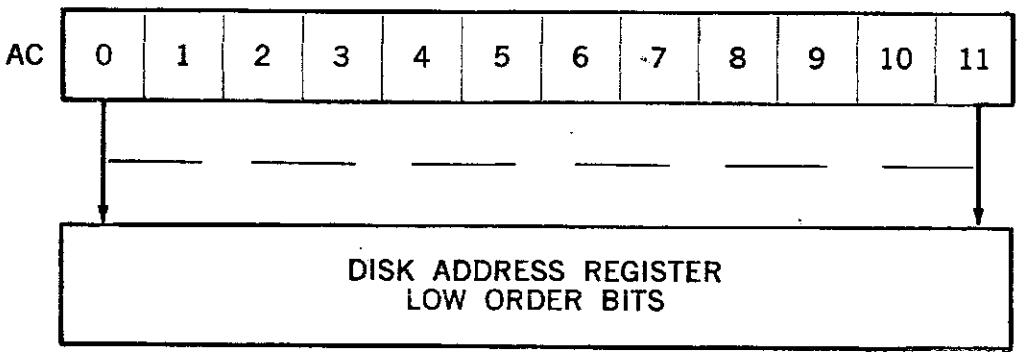
Octal Code: 6601
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears memory address register, parity error and completion flags. This instruction clears the disk memory request flag and interrupt flags.
Symbol: 0 → completion flag
0 → error flag

Load Disk Memory Address Register and Read (DMAR)

Octal Code: 6603
Event Time: 1, 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The contents of the AC are loaded into the disk memory address register and the AC is cleared. This IOT initiates reading of information from the disk into the specified core location. Clears parity error and completion flags.
Symbol: AC 0-11 → DMA 0-11
1 → read
0 → completion flag
0 → error flag

Load Disk Memory Address Register and Write (DMAW)

Octal Code: 6605
Event Time: 1, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The contents of the AC are loaded into the disk memory address register and the AC is cleared. This disk then begins to write information into the disk from the specified core location. Clears parity error and completion flags. Data break must be allowed to occur within 66 μs after issuing this instruction.



Symbol: AC 0-11 → DMA 0-11
1 → write
0 → completion flag
0 → error flag

Clear Disk Extended Address Register (DCEA)

Octal Code: 6611
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears the disk extended address and memory address extension register.
Symbol: 0 → disk extended address register
0 → memory address extension register

Skip on Address Confirmed Flag (DSAC)

Octal Code: 6612
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Skips next instruction if address confirmed flag is a 1. Flag is set for 16 μs whenever the address on the disk equals the contents of the disk address registers. AC is cleared.
Symbol: 0 → AC
If address confirmed flag = 1, then PC + 1 → PC

Load Disk Extended Address (DEAL)

Octal Code: 6615
Event Time: 1, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The disk extended address and memory address extension regis-

ters are cleared and loaded with the track address data held in the AC. The previous contents of these registers, plus the photocell mark, and three error flags, are ORed into the AC. (See DEAC instruction.)
Symbol: AC 6-8 → EA 3-1 Core Memory Extension
AC 1-5 → EMA 5-1 Disk Address Extension 32K, 64K, 96K, 128K
AC0, 9-11 used in DEAC instruction

Read Disk Extended Address Register (DEAC)

Octal Code: 6616
Event Time: 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clear the AC, then load the contents of the disk extended address register into the AC to allow program evaluation. Skip the next instruction if address confirmed flag is a 1.
Symbol: 32K, 64K, 120: EMA 5-1 → AC 1-5
Extended break address EA 3-1 → AC 6-8
Photocell sync mark → AC0 (available 200 μs)
Data Request Late flag → AC9
Nonexistent or write lock switch on → AC10
Parity Errors → AC11

NOTE

Write lock switch status is true only when disk module contains a write command. The nonexistent disk condition will appear following the completion of a data transfer during read, where the address acknowledged was the last address of a disk and the next word to be addressed falls within a nonexistent disk. The completion flag for this data transfer is set by the non-existent disk condition 16 μs following the data transfer.

Skip On Zero Error Flag (DFSE)

Octal Code: 6621
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Skips the next instruction if parity error, data request late, and write lock switch flag are all zero. Indicates no errors.
Symbol: If parity error flag = 0 and if data request late flag = 0 and if write lock switch flag = 0, then
PC + 1 → PC

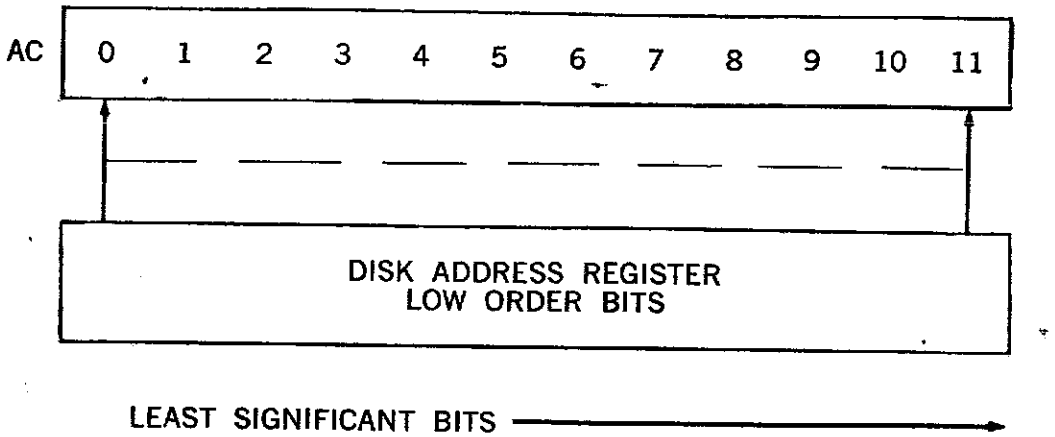
Skip on Data Completion Flag (DFSC)

Octal Code: 6622
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Skips the next instruction if the completion flag is a 1; indicating data transfer is complete.
Symbol: If completion flag = 1, PC + 1 → PC

Read Disk Memory Address Register (DMAC)

Octal Code: 6626
Event Time: 2, 3

Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears the AC, then loads the contents of the disk memory address dress register into the AC to allow program evaluation. During read, the final address will be the last one transferred.
Symbol: DMA 0-11 → AC 0-11



Three-cycle data break locations: Work Count address is 7750 (field 0), Current Address is 7751 (field 0).

Three maintenance IOTs are also used by the DF32. These IOTs are used to simulate certain pulses within the disk control for static logic tests. Since they all use device code 63, this code should not be used by other peripheral devices when a DF32 is part of the system.

NOTE

For the DEAL and DEAC instructions, refer to the diagrams shown below:

Bits 1-5 (DEAL, DEAC Inst.)	Accumulator (Low Order 12 Bits) 0-11 of DMAW or DMAR	Disk Address (17 Bit)
Field Bits 6-8 (DEAL, DEAC Inst.)	Cell 7751 (Current Address)	Current Address (Memory) Address (15 Bit)

The computer can handle 12 bits; therefore, the high order bits for disk and memory address are manipulated by the DEAL and DEAC instructions. Low order bits are manipulated in the AC.

The disk address is a 17-bit value. Bit 1 of the DEAL and DEAC instructions is the most significant bit. The memory address is a 15-bit value. Bit 6 of the DEAL and DEAC instructions is the most significant bit.

Note that the word count 7750 is loaded with the two's complement of the number of words to be transferred and that the disk address is loaded with the desired starting address. The memory of current address (7751) is loaded with the desired address minus one.

Software

The DF32 Disk System, available with either the PDP-8/I or the PDP-8/L, is a

fast, convenient keyboard-oriented monitor which will enable the user to efficiently control the flow of programs through the computer. This system is modular and open-ended, allowing the user to build the software components required in his environment. The user may specify the system device (disk or DECtape), the amount of core, number of disks available, and the number, name, and size of his resident system program.

TYPE RF08 DISK FILE AND CONTROL AND TYPE RS08 EXPANDER DISK FILE

The RF08 control and the RS08 disk combine to provide fast, low-cost, random access, bulk storage for the PDP-8/I [PDP-8/L] computers. One RF08/RS08 provides 262,144 13-bit words of storage. Up to four RS08 disks can be added to the RF08 control for a total of 1,048,576 words of storage. Data is recorded on a single disk surface by 128 fixed read/write heads.

Data transfer is accomplished through the three-cycle break system of the computer and its associated required options, which are the same as for the DF32/DS32 system. Fast track-switching time permits spiral read or write. Data may be read or written in blocks of from 1 to 4096 words. Transfers across disks are handled automatically by the control unit. Table 7-7 lists the RF08/RS08 specifications.

Table 7-7. RF08/RS08 Specifications

Disks	Four RS08s may be controlled by one RF08 for 1,048,576 words.	
Storage Capacity	Each RS08 stores 262,144 13-bit words (12 plus one even parity bit)	
Data Transfer Path	3-Cycle Break	Address Locations 7750 Word Count 7751 Current Address
Data Transfer Rate	60 Hz Power 16.0 μs per word	50 Hz Power 19.2 μs per word
Minimum Access Time	258 μs	320 μs
Average Access Time	16.9 ms	20.3 ms
Maximum Access Time	33.6 ms	40.3 ms
Program Interrupt	33 ms Clock Flag Data Transmission Complete Flag Error Flag	
Write Lock Switches	Eight switches per disk capable of locking out any combination of eight 16,384 word blocks in addresses 0 to 131,071.	
Data Tracks	128	
Words Per Track	2048	
Recording Method	NRZ1	
Density	1100 bpi Maximum	
Timing Tracks	3 plus 3 spare (spares can be used to recover data on disk)	

Table 7-7. RF08/RS08 Specifications (Cont)

Operating Environment	Recommended temperature 65° to 90°F.
Vibration/Shock	Good isolation is provided. To prevent data errors, extreme vibrations should be avoided while the RS08 is transferring information.
Heat Dissipation	RF08: 150W RS08: 300W
AC Power Requirements	115/230 ± 10% Vac, single phase, 50 ± 2 or 60 ± 2 Hz, 5A (maximum) for logic power. (Logic power for one RF08 and up to four RS08s is provided by one DEC Type 705B Power Supply) Additional line current is required for RS08 disk motor as shown below.
RS08 Motor Power Requirements	Motor start, 5.5A for 20 ± 3s. Motor run, 4.0A continuous @ 115 Vac. (A stepdown autotransformer is provided for 230 Vac operation).
Line Frequency Stability	Maximum line frequency drift 0.1 Hz/s. A constant frequency motor-generator set or static ac/ac inverter should be provided for installation with unstable power sources.
Motor Bearing Life	Expected operating life of at least 20,000 hours, under standard computer operating environment.
Reliability	Six recoverable errors and one nonrecoverable error in 2×10^9 bits transferred. A recoverable error is defined as an error that occurs only once in four successive reads. All other errors are nonrecoverable. On-off cycling of the RS08 is not recommended. For this reason, the RS08 motor control operates independently of the computer power control.
Cabinet	A dedicated cabinet is designed to accommodate one RF08, up to two RS08s and power supply. Two additional RS08s can be mounted in a second cabinet. Other equipment should not be mounted in disk cabinets.
Shipping Information	Weight of RF08, one RS08, power supply and cabinet: 590 lb (crated) 500 lb (uncrated) Weight of RF08, two RS08, power supply and cabinet: 690 lb (crated) 600 lb (uncrated) (The RF08/RS08 are shipped mounted in cabinets)

Programming Instructions

The programming instructions for the RF08/RS08 differ slightly from those provided in the DF32/DS32 description. The extended address capability and associated instructions (DCEA, DEAL, and DEAC) are replaced, in sequence, by interrupt enable and memory address extension register instructions (DCIM, DIML, and DIMA).

Clear Disk Interrupt Enable and Core Memory Address Extension Register (DCIM)

Octal Code: 6611
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clear the disk interrupt enable (DIE) and core memory address extension (MAE) registers.
Symbol: 0 → DIE, 0 → MAE

Load Interrupt Enable and Memory Address Extension Register (DIML)

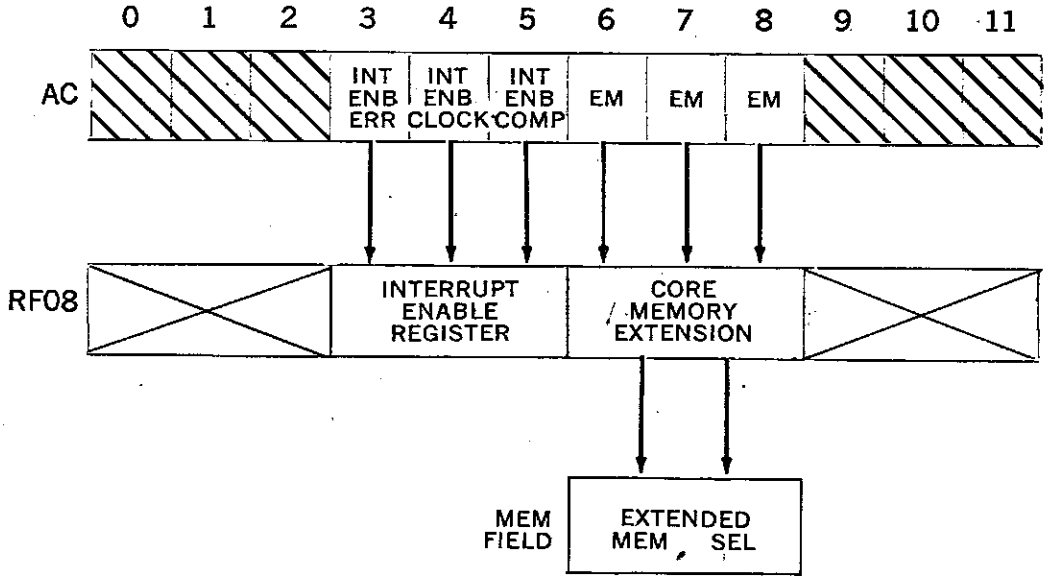
Octal Code: 6615
Event Time: 1, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clear the interrupt enable (IE) and MAE, then load the interrupt enable and memory address extension registers with data held in the AC. Then clear AC.

NOTE

Transfers cannot occur across memory fields. Attempts to do so will cause the transfer to "wrap around" within the specified memory field.

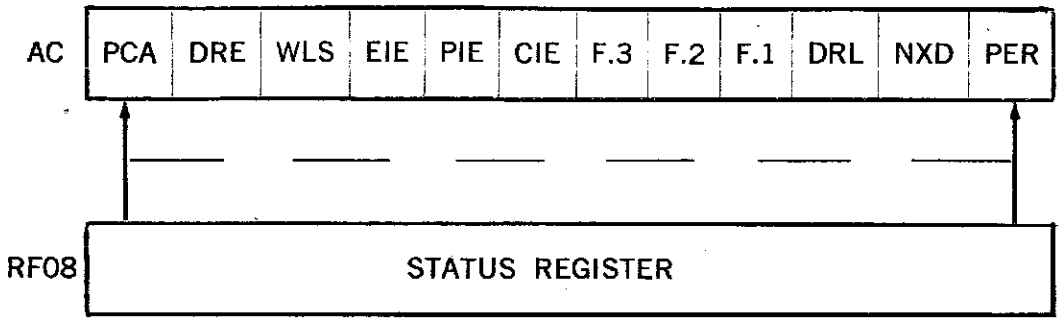
Symbol: 0 → IE, 0 → MAE
AC 3-15 → IE, AC 6-8 → MAE
0 → AC

AC TO DISK STATUS REGISTER



Load Interrupt and Extended Memory Address (DIMA)

Octal Code: 6616
Event Time: 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clear the AC. Then load the contents of the status register (STR) into the AC to allow program evaluation.



AC Bit	Abbr.	Description
0	PCA	Photocell Sync Mark (available 100 μs status)
1	DRE	Data Request Enable (maintenance only status)
2	WLS	Write Lock Status
3	EIE	Error Interrupt Enable
4	PIE	Photocell Interrupt Enable
5	CIE	Completion Interrupt Enable
6-8	F	(FIELD) Core Memory Extension Fields
9	DRL	Data Request Late
10	NXD	Nonexistent Disk
11	PER	Parity Error
Symbol:	0 → AC	
	STR → AC	

In addition to these changes in instructions, the RF08/RS08 utilizes six additional instructions: DFSE, DISK, DCXA, DXAL, DXAC, and DMMT.

Skip on Disk Error (DFSE)

Octal Code: 6621
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Skip next instruction if there is parity error, data request late, write lock status, or nonexistent disk flag set.
Symbol: Parity error, data request late, write lock status, or nonexistent disk flags are set, PC + 1 → PC.

Skip Error or Completion Flag (DISK)

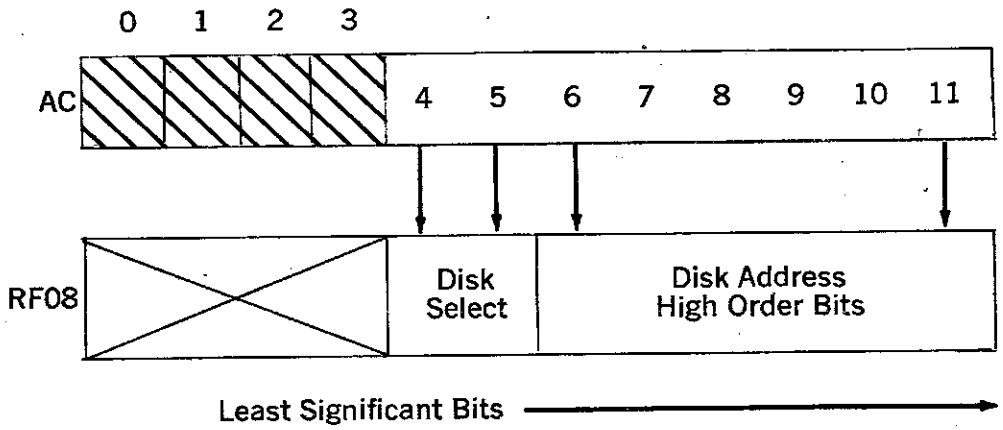
Octal Code: 6623
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: If either the error or data completion flag (or both) is set, the next instruction is skipped.
Symbol: If PER or Data Complete, PC + 1 → PC

Clear High Order Address Register (DCXA)

Octal Code: 6641
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clear the high order 8-bit disk address register (DAR).
Symbol: 0 → DAR

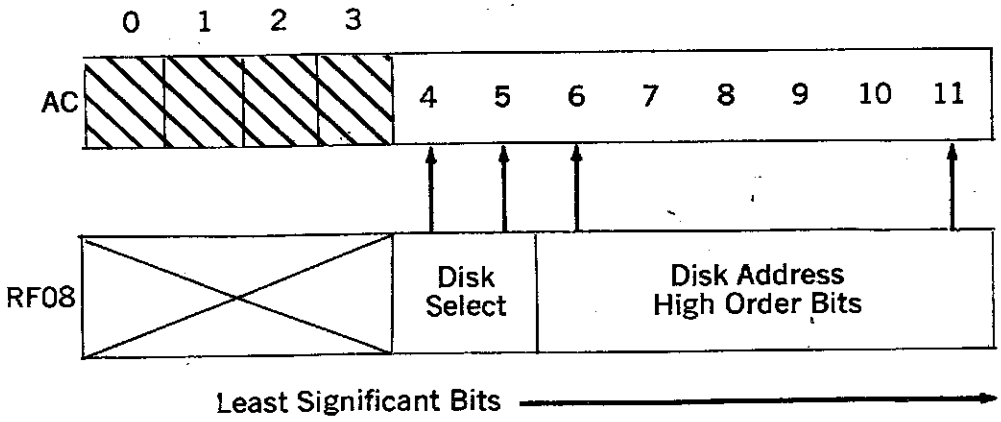
Clear and Load High Order Address Register (DXAL)

Octal Code: 6643
Event Time: 1, 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clear the high order 8 bits of the DAR. Then load the DAR from data stored in the AC. Then clear AC.
Symbol: 0 → DAR high order 8 bits,
AC → DAR,
0 → AC



Clear Accumulator and Load DAR Into AC (DXAC)

Octal Code: 6645
Event Time: 1, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clear the AC; then load the contents of the high order 8-bit DAR into the AC.
Symbol: 0 → AC,
DAR high order 8 bits → AC



Initiate Maintenance Register (maintenance purposes only) (DMMT)

Octal Code: 6646
Event Time: 2, 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: For maintenance purposes only with the appropriate maintenance cable connections and the disk disconnected from the RS08 logic, the following standard signals may be generated by IOT 6646 and associated AC bits. AC is cleared and the maintenance register (MAIR) is initiated by issuing an IOT 6601 command.

AC (1) Track A Pulse
AC (1) Track B Pulse
AC (1) Track C Pulse
AC (1) DATA PULSE (DATA HEAD #0)
AC (1) 1 Photocell
AC (1) 1 DBR
Setting DBR to a 1 causes data break request in computer.

Symbol: AC → MAIR

Three-cycle data break locations: word count address is 7550 (field 0), current address is 7751 (field 0).

DF32 Programming Compatibility

The IOT instructions 660X and 6622 are identical in every respect to the DF32 instruction; i.e., the same operations are performed. The 661X and 662X instructions differ only in the following:

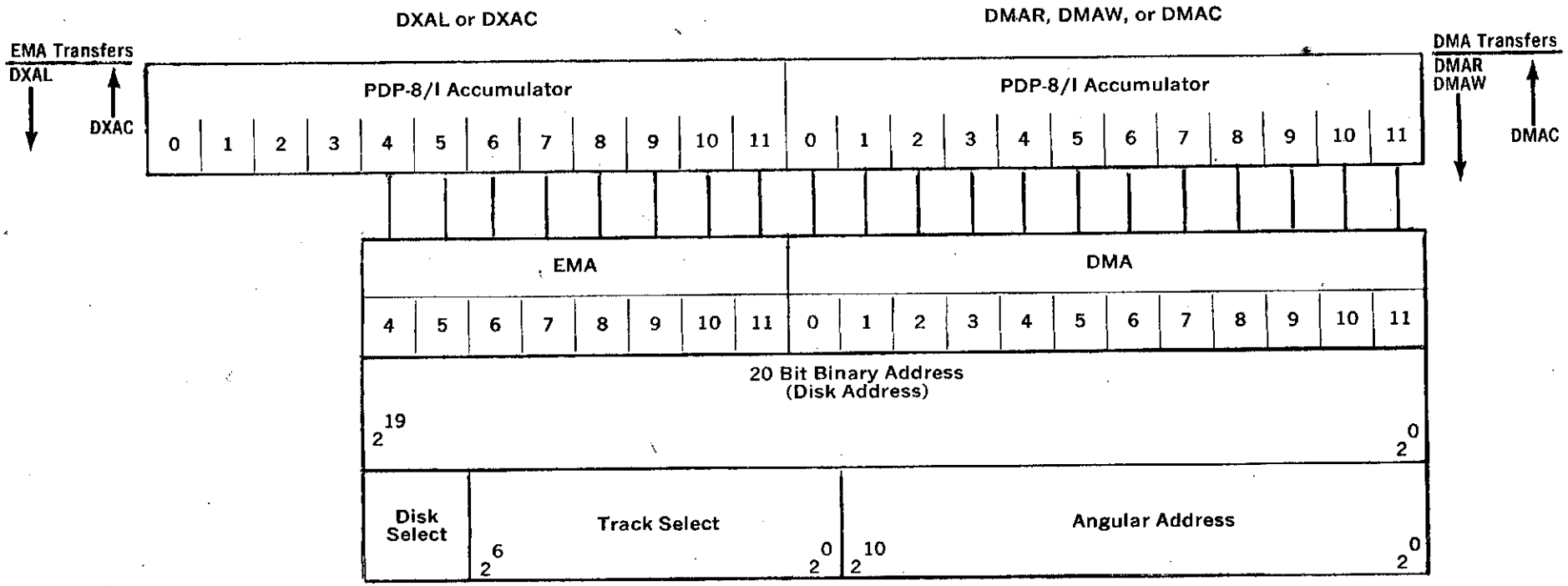
a. IOT 6615 does not transmit the extended disk address bits for addressing over 32K; instead, AC 3-5 are assigned to enable or disable conditions on the program interrupt line. The AC is cleared upon execution of this instruction.

b. IOT 6616 no longer reads back the extended address bits by 1 through 5 into the AC. These bits are assigned to examine the status of interrupt enable. In addition, AC2 indicates the status of write lock and AC10 shows only nonexistent disk conditions. AC1 shows the conditon of data request enable used for maintenance purposes.

c. IOT 6621 has been changed to skip on error rather than no-error. Nonexistent disk has been included as an error skip condition.

d. IOT 6623 (DISK) is a new skip instruction that will skip on either error or completion flags or both.

The DF32 maintenance instruction IOT 663X is not assigned to the RF08 system.


DXAL or DXAC
DMAR, DMAW, or DMAC
EMA Transfers
DXAL
DXAC
PDP-8/I Accumulator
0 1 2 3 4 5 6 7 8 9 10 11
PDP-8/I Accumulator
0 1 2 3 4 5 6 7 8 9 10 11
DMA Transfers
DMAR
DMAW
DMAC
EMA
4 5 6 7 8 9 10 11
DMA
0 1 2 3 4 5 6 7 8 9 10 11
20 Bit Binary Address
(Disk Address)
2^19
2^0
Disk Select
Track Select
2^6
2^0
Angular Address
2^10
2^0

Software

A sample of a typical I/O routine for the RF08/RS08 is as follows:

```
0200  4777           JMS I   (DISKIO)
0201  0000  FUNCT,    0              /X0=READ, X1=WRITE (X=0=7 MEMORY FIELD)
0202  0000  WDCT,     0              /+ WORD COUNT
0203  0000  CORE,     0              /CORE LOCATION
0204  0000  DSKHI,    0              /HIGH ORDER 8 BITS
0205  0000  DSKLOW,   0              /LOW ORDER 12 BITS
0206  5020           JMP ERROR              /ERROR RETURN (AC=ERROR CONDITION)
                                     /NORMAL RETURN (AC=0)

0207  0000  DISKIO,   0
0210  7300           CLL CLA
0211  1607           TAD I DISKIO
0212  6615           DIML           /LOAD EXTENDED MEMORY BITS
0213  1607           TAD I DISKIO
0214  0376           AND (7
0215  7640           SZA CLA

0216  7126           STL RTL        /+2
0217  1375           TAD (3
0220  1374           TAD (6600
0221  3236           DCA RORW               /6603=READ, 6605=WRITE
0222  2207           ISZ DISKIO
0223  1607           TAD I DISKIO
0224  7041           CIA
0225  3773           DCA I (7750    /STORE=WORD COUNT
0226  2207           ISZ DISKIO
0227  1607           TAD I DISKIO
0230  3772           DCA I (7751)   /LOAD CORE ADDRESS
0231  2207           ISZ DISKIO
0232  1607           TAD I DISKIO
0233  6643           DXAL           /LOAD HIGH ORDER 9
                                     /BITS OF DISK ADDRESS,

0234  1607           TAD I DISKIO
0235  2207           ISZ DISKIO

0236  0000  RORW, .   0              /READ OR WRITE

0237  6623           DISK           /DONE?
0240  5237           JMP .-1        /NO
0241  6621           DFSE           /YES, ERROR?
0242  2207           ISZ DISKIO     /SKIP TO NORMAL RETURN
0243  5607           JMP I DISKIO   /RETURN


      6615  DIML = 6615
      6623  DISK = 6623
      6643  DXAL = 6643
      6621  DFSE = 6621
      0020  ERROR = 20


0372  7751
0373  7750
0374  6600
0375  0003
0376  0007
0377  0207


CORE     0203
DFSE     6621
DIML     6615
DISK     6623
DISKIO   0207
DSKHI    0204
DSKLOW   0205
DXAL     6643
ERROR    0020
FUNCT    0201
RORW     0236
WDCT     0202
```

SECTION 7-8 ANALOG INPUT SUBSYSTEMS

TYPE AF01A ANALOG-TO-DIGITAL SYSTEM

The Type AF01A General-Purpose, Multiplexed Analog-to-Digital (A/D) Converter combines a versatile, multipurpose converter with a multiplexer to provide fast automatic multichannel scanning and conversion capability. It is intended for use in computer systems in which sampling and processing of analog data from sensors or other external signal sources is desired. The Type AF01A option is used with the PDP-8/I [PDP-8/L and DW08-A I/O Conversion Panel], to multiplex up to 64 analog signals and to convert the signals to binary numbers. Analog data on each of 64 channels can be accepted and converted into 12-bit digital numbers 420 times per second.*

Switching point accuracy is ±0.025%, with an additional quantization error of half the least significant bit (LSB). If less resolution and accuracy is required, all 64 channels can be scanned and the analog signals on them converted into 6-bit digital numbers 1420 times each second.**

Switching point accuracy in this case is ±1.6%, again with the additional quantization error of half the digital value of the LSB.

*Conversion rate = $[(35 + 2)\ (10^{-6})\ (64)]^{-1}$ = 420 cycles/sec.

**Conversion rate = $[(9 + 2)\ (10^{-6})\ (64)]^{-1}$ = 1420 cycles/sec.

A/D Converter Specifications

The AF01A has a successive approximation converter that measures a 0 to −10 analog input signal and provides a binary output indication of the input signal amplitude. The characteristics of the A/D converter are as follows:

CHARACTERISTICS	SPECIFICATIONS
Accuracy and Conversion Times	See Table 7-9 (includes all linearity and temperature errors)
Converter Recovery Time	Zero
Input and Input Impedance	0V to −10V at 10 megohms standard. Input scaling may be specified using the amplifier or sample and hold options (See Table 7-8)
Input Loading	1 μA and 125 pF for 0V to −10V input signal.
Output	Binary number of 6 to 12 bits, with negative numbers represented in 2's complement notation. A 0V input gives a 4000_8; a −5V input a 0000_8 and a −10V (minus 1 LSB) input gives 3777_8.

Provision is made for using the Type AH02 Sample and Hold option between the multiplexer output and A/D converter input to reduce the effective aperture to less than 150 ns. The AH02 option may also be used to scale the signal input to accept ±10V, ±5V, or 0V to +10V. The AH03 amplifier may be substituted for the AH02 to accomplish the same signal scaling without reducing the effective aperture. The AH02 and AH03 options may be used together to simultaneously obtain high input impedance and small aperture.

Multiplexer Specifications

The multiplexer can include from 1 to 16 Type A121 Multiplexer Switch Modules. Each module contains four single-pole, high-speed, insulated-gate FET switches with appropriate gating. The A121 switches are arranged as a 64-channel group of series-switched, single-pole switches with a separate continuous ground wire for each signal input. The switched signal input wire and the continuous ground for each channel are run as twisted pairs to the input connectors mounted on the rear panel. The continuous grounds for all channels are terminated at the high quality ground of the AF01A system. Specifications (measured at input connector) are as follows and as contained in Tables 7-8 and 7-9:

CHARACTERISTICS	SPECIFICATIONS
Input Operating Signal Voltages	+10V to −10V
Current	1 mA
On Resistance	500Ω (max)
Voltage Offset	0
"Off Leakage"	0.010 μA (max)
Capacitance	10 pF (max)
Speed 10% Input to within 1 LSB of output	2 μs
Operate Time	The time required to switch from one channel to another is 2 μs to within 1 LSB of the final voltage. This time is preset within the control and starts when a set, clear or index command is received.

Operation

The AF01 system may be operated in either the random or sequential address modes. In the random address mode, the control routes the analog signal from any selected channel to the A/D converter input. In the sequential address mode, the multiplexer control advances its channel address by one each time an index command is received. After indexing through the maximum number of channels implemented, the address is returned to 0. When using sequential operation, the conditioning levels for random addressing are ignored.

The multiplexer switch setting time is preset within the control to initiate the conversion process automatically after a channel has been selected in either the random or sequential address mode. A separate A/D convert I/O transfer command may also initiate one or more conversions on a currently selected channel.

A/D conversion times are increased by 2 μs when multiplexer channels are switched to allow for settling time of the analog signal at the multiplexer output. Conversion times are increased an additional 3 μs when AH03 is used. These times are added to the conversion times shown in Table 7-9 under selected channel conversion time, which is the only time required for each successive conversion on a selected channel.

When the AH02 Sample and Hold option is required, the multiplexer switch setting time and the sample and hold acquisition time are overlapped. The total conversion and switching time is increased by 10 μs. (See AH02 specifications.)

TABLE 7-8 INPUT SIGNAL SCALING

CONFIGURATION	GAIN	INPUT SIGNAL	INPUT IMPEDANCE	BINARY OUTPUT	OPTION DESIGNATION
Standard		0	10 meg.	4000_8	STD
		−5	10 meg.	0000_8	
		−10	10 meg.	3777_8	
Sample and Hold	−1	+5	10K	3777_8	AH02
	−1	0	10K	0000_8	
	−1	−5	10K	4000_8	
Sample and Hold	−½	+10	10K	3777_8	AH02
	−½	0	10K	0000_8	
	−½	−10	10K	4000_8	
Amplifier	+1	+5	>100 meg.	4000_8	AH03
	+1	0	>100 meg.	0000_8	
	+1	−5	>100 meg.	3777_8	
Amplifier	+½	+10	>100 meg.	4000_8	AH03
	+½	0	>100 meg.	0000_8	
	+½	−10	>100 meg.	3777_8	
Amplifier and Sample and Hold	−1 +5 +10 or 0 or 0 −½ −5 −10		>100 meg.	3777_8	AH03 and AH02
			>100 meg.	0000_8	
			>100 meg.	4000_8	

Note: Unipolar signals (0 to +5, or 0 to +10V) may also be specified with either the AH03 or AH02 option.

TABLE 7-9 SYSTEM CONVERSION CHARACTERISTICS**

WORD LENGTH (NO. OF BITS)	MAX SWITCHING POINT ERROR*	SELECTED CHANNEL (A/D) CONVERSION TIME (μs)	RANDOM OR SEQUENTIAL (MPX. & A/D) CONVERSION TIME (μs)**	W/AH03 AMP. (MPX. & A/D) CONVERSION TIME (μs)**	W/AH02 SAMPLE & HOLD (MPX. & A/D) CONVERSION TIME (μs)**	W/AH03 & AH02 (MPX. & A/D) CONVERSION TIME (μs)**
6	±1.6%	9.0	11.0 (9.5)	14.0 (11.0)	19.0 (14.0)	21.0 (18.0)
7	±0.8%	10.5	12.5 (11.0)	15.5 (12.5)	20.5 (15.5)	22.5 (19.5)
8	±0.4%	12.0	14.0 (12.5)	17.0 (14.0)	22.0 (17.0)	24.0 (21.0)
9	±0.2%	13.5	15.5 (14.0)	18.5 (15.5)	23.5 (18.5)	25.5 (22.5)
10	±0.1%	18.0	20.0 (18.5)	23.0 (20.0)	28.0 (23.0)	30.0 (27.0)
11	±0.05%	25.0	27.0	30.0	35.0	37.0
12	±0.025%	35.0	37.0	40.0	45.0	47.0

*±½ LSB for quantizing error.

**If system is to operate at less than 10 bits continuously, conversion times may be reduced to times shown in parentheses.

A/D Converter/Multiplexer Controls

The A/D Converter Multiplexer Controls are listed in Table 7-10

TABLE 7-10 A/D CONVERTER/MULTIPLEXER CONTROLS

Designation	Function
WORD LENGTH	Rotary switch used to select digital word length or conversion accuracy. Refer to Table 7-9 for corresponding conversion times.
POWER ON/OFF	Applies 117 Vac power to internal power supplies.
CLR	Clear multiplexer channel-address registers; i.e., selects analog channel 0 for conversion.
INDEX	Advances multiplexer channel-address register by one each time it is depressed, enabling manual addressing of channels (up to 64) in sequential mode. Returns address to zero when maximum value is reached.
ADC	Starts conversion of the analog voltage on the selected channel to a binary number when depressed.
A/D CONVERTER	Indicates binary contents of A/D converter register.
MULTIPLEXER	Indicates binary contents of multiplexer channel-address register.
POWER	Indicates ON/OFF status.

Programming

Programmed control of the converter/multiplexer by the PDP-8/I [PDP-8/L] is accomplished with the IOT instructions listed below. The computer selects the converter/multiplexer with two device selection codes, depending upon whether conversion of multiplexing functions is being selected; 53_8 and 54_8. The converter/multiplexer interprets the device selection code to enable execution of the IOP command pulse generated by the IOT instruction.

Skip on A/D Flag (ADSF)

Octal Code: 6531
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The A/D converter flag is sensed and, if it contains a binary 1 (indicating that the conversion is complete), the content of the PC is incremented by one and the next instruction is skipped.
Symbol: If A/D Flag = 1, then PC + 1 $\rightarrow$ PC

Convert Analog Voltage to Digital Value (ADCV)

Octal Code: 6532
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The A/D converter flag is cleared, the analog input voltage is converted to a digital value, and the A/D converter flag is set to 1. The time is, from start of conversion to setting of the flag, is a function of the accuracy and word length switch setting as listed in Table 7-9. The number of binary

bits in the digital-value word and the accuracy of the word is determined by the preset switch position.
Symbol: 0 → A/D flag at start of conversion, then
1 → A/D flag when conversion is done.

Read A/D Converter Buffer (ADRB)
Octal Code: 6534
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The converted number contained in the converter buffer (ADCB) is transferred into the AC left justified, unused bits of the AC are left in a clear state, and the A/D converter flag is cleared. This command must be preceded by a CLA instruction.
Symbol: ADCB Vac → AC
0 → A/D Converter Flag

Clear Multiplexer Channel (ADCC)
Octal Code: 6541
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The channel address register (CAR) of the multiplexer is cleared in preparation for setting of a new channel.
Symbol: 0 → CAR

Set Multiplexer Channel (ADSC)
Octal Code: 6542
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The channel address register of the multiplexer is set to the channel specified by AC6-11. A maximum of 64 single-ended input channels can be used.
Symbol: AC6-11 → CAR

Increment Multiplexer Channel (ADIC)
Octal Code: 6544
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the channel address register of the multiplexer is incremented by one. If the maximum address is contained in the register when this command is given, the minimum address (00) is selected.
Symbol: CAR + 1 → CAR

The converter/multiplexer may be operated by the computer program in either the random or sequential addressing mode. In the random addressing mode, the analog channel is selected arbitrarily by the program for digitizing and the resultant binary word is read into the accumulator. A sample program for the random addressing mode is as follows:

```
TAD ADDR    /GET CHANNEL ADDRESS
ADSC        /AND SEND TO MULTIPLEXER
ADCV        /CONVERT A TO D
```

CLA	/CLEAR ACCUMULATOR
ADSF	/SKIP ON A/D DONE FLAG
JMP	/WAIT FOR FLAG
ADRB	/AND READ INTO AC

In the sequential address mode, the program advances the multiplexer channel-address register to the next channel and generates an ADCV command.

Should the converter/multiplexer be operated in the interrupt mode, the computer will be signaled each time that a binary word is ready, enabling the system to use processor time more efficiently.

Amplifier Sample and Hold Options for AF01A

AH03 Amplifier Option — The AH03 consists of a DEC amplifier (part no. 1505379) mounted on an A990 Amplifier Board with appropriate scaling networks and gain trim, and balance potentiometers.

CHARACTERISTICS	SPECIFICATIONS
Open loop gain	$2 + 10^6$
Rated output voltage	±11V (@ 10 ma)
Frequency response	
Unity Gain, small signal	10 mHz
Full output voltage	300 kHz
Slewing rate	30v/μs
Overload recovery	200 μs
Input voltage offset	Adjustable to 0
Avg vs temp	20 μV/°C
Vs supply voltage	15 μV/%
Vs time	10 μV/day
Input current offset	±0.002 μA
Avg vs temp	0.0004 μA/°C
Vs supply voltage	0.00015 μA/°C
Input impedance	
Between input	6 megohms
Common mode	500 megohms
Input voltage	±10V
Max common mode	±10V
Common mode rejection	20,000
Power	
Voltage	±15V
Current at rated load	3 mA

AH02 Sample and Hold Option — A400 (standard gain options)

Acquisition time to 0.01% (full scale step)	< 12 μs

Aperture time	< 150 ns
Hold inaccuracy (droop)	< 1mV/ms
Temperature coefficient	0.1 mV/ms/°C
Gain (negative)	−1.0
Input range (volts)	±5.0V ±10.0V
Impedance	≥ 10 Kohms ≥10 Kohms
Output voltage	±10V
Impedance	≤1.0 ohm

AC01A Sample and Hold Option

The AC01A Sample and Hold Control is used in conjunction with an analog-to-digital converter. It provides skewless sampling of analog data from up to eight analog sources. Typical of varied applications for this device is the sampling of data from seven-track analog tape recorders.

Included with the AC01A is a dual 15V, 1.5A, floating, regulated power supply. This supply is intended to power the sample and hold modules only. Logic voltage must be supplied from another source.

The option contains control logic which interprets programmed commands, and controls the switching of up to eight AH02 Sample and Hold options.

Specifications — The following specifications apply to the AC01A:

8 Channels
±10V (max) input voltage
10K (min) input impedance
12 μs (max) track time to within 0.025% F.S.
150 ns (max) aperture time
≤1V/s droop
±10V (max) ouput voltage
10 mA (max) output current

Programming — The following instructions are used with the AC01A:

Random Hold (HRAN)

Octal Code: 6571
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The contents of AC_{3-5} are transferred to the channel address register (CHAR). The 3-bit code is decoded to address any of the 8 channels.
Symbol: AC 3-5 → CHAR 1 → Hold FF

Simultaneous Hold (HSIM)

Octal Code: 6572
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Simultaneously placed all eight channels into the hold mode
Symbol: 1 → Hold FF's

Sample (SAMP)
Octal Code: 6574
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Places all eight channels into the sample (or track) mode.
Symbol: 0 → Hold FF's

TYPES AF02A AND AF03A ANALOG-TO-DIGITAL SYSTEMS

Many functional, operating, and programming similarities exist between the AF02A and the AF03A; therefore both systems will be included in this discussion, with the differences between the two systems being noted where applicable.

Leading Particulars
Table 7-11 describes the leading particulars of both the AF02A and the AF03A systems.

TABLE 7-11. AF02A and AF03A Leading Particulars

FUNCTION	AF02A	AF03A
Multiplexer unit required	AM02	AM03
Controller required	AM08	AM08
Type of input unit	A122 single-ended	A111 3-wire floating
Analog-to-digital converter computer interface	ADC-1 and ADC-8	ADC-1 and ADC-8
Channel capacity	1024 (max)	1024 (max)
Input Operating Voltage	High level —	Lowel level —
Range	(1V — 10V F.S.)	(10 mV — 10V F.S.)
Typical New Channel Selection Time	2.5 μs	4 ms
Type of multiplexer switches	FET	Relay

The AF02A and AF03A may be used directly with tht PDP-8/I computer. [When used with the PDP-8/L computer, the DW08-A I/O Conversion Panel is required].

The AF02A system functions as a 6-12 bit ADC system with multiplexer control for 4 to 1024 channels of single-ended, high-level, analog inputs. Channels are implemented with additional Type A122 modules. The system consists of an ADC-1 6-12 bit analog-to-digital converter, an ADC-8 interface, an AM08 multiplexer control for 1024 channels, and an AM02 multiplexer and analog input connects. Standard options for the AF02A include the AH02, AH03, AC01A (all of which are discussed in the AF01A portion of this section), and an A122 4-channel FET multiplexer switch.

The AF03A system functions as a 6-12 bit ADC system with multiplexer control for 2 to 1024 low- or high-level, 3-wire, differential analog inputs. This system also uses the ADC-1 converter, ADC-8 interface and the AM08 multiplexer

The AF03A system functions as a 6-12 bit ADC system with multiplexer control for 2 to 1024 low- or high-level, 3-wire, differential analog inputs. This system also uses the ADC-1 converter, ADC-8 interface and the AM08 multiplexer control; however, it uses the AM03A (in lieu of the AM02A) multiplexer with analog input connectors. The standard options for the AF03A are similar to these used with the AF02A, except that the A122 is replaced with the A111 which is a 2-channel guarded relay multiplex module and an optional AG01 differential amplifier. An optional AG02 6-range programmable gain differential amplifier is available.

Subsystem Unit Descriptions

The AM08 interfaces with the computer and is connected to the I/O bus with standard bus cables. The AM08 can operate up to 8 AM02A multiplexer mounting panels or up to 16 AM03A multiplexer mounting panels. The AM08 controls selection of signal switching paths in both the low- and high-level multiplexers. These multiplexers are used to route individual analog source signals, in large multiples, to intermediate amplifiers prior to their conversion to digital signals.

The AM02A high-level signal multiplexers comprise one Type 1943 Standard Mounting Panel and one analog-signal connector panel that routes individual signal sources to the input switching circuits and one common signal output from panel-mounted switching modules. The AM02A is used when less than 128 high-level analog signals must be converted to their digital equivalents.

The AM03A low-level signal multiplexer comprises one Type 1943 Standard Mounting Panel and one analog-signal connector panel that routes individual signal sources to the input switching circuits and one common output that routes all signals from the panel-mounted switching modules.

Both the AM02A and the AM03A multiplexers are used in conjunction with the AM08 multiplexer control and are intended for use in analog-signal switching systems where many analog signal sources must be sampled. These analog signal sources are sampled, either at random or sequentially, for conversion to their digital equivalents for subsequent use as data inputs for the computer.

Signal multiplexers are driven by control signals from the multiplexer control. Each switching module in the multiplexer receives one or more analog signals from the signal source through preamplifiers or directly from the source transducers. Each transducer converts any measurable physical variable to a continuous electrical signal that is coupled to one of the switches in the signal-switching module contained in the multiplexer. The common output from any multiplexer is applied to either an AG01, AG02 differential amplifier or AH03 for subsequent buffering or amplification.

The primary purpose of the differential amplifiers is to convert double-ended inputs to signal-ended outputs with high noise rejection, signal buffering, signal attenuation, or signal amplification.

The AG01 has a high common-mode rejection obtained from a double-ended input. This amplifier has a manually adjustable gain ranging from unity to 1000. The gain can be obtained with a bandwidth control that permits bandwidth selection in four decades, from DC to 10 kHz. The AG02 differs from the AG01 in that it has six discrete programmable-gain settings between unity and 1000. The AG02 gain is controlled by the computer program.

The ADC-1 A/D converter has convenience switches mounted on the control and indicator panel: a power switch, to control the AC power to the system and an ADC pushbutton switch to manually initiate conversion. A rotary word-length switch is provided to select the word length and, thus, the conversion

accuracy and conversion time. The panel also has 12 indicators to display the contents of the A/D converter buffer, and a power on-off indicator.

Programming

Both the AF02A and the AF03A use the following instructions: ADSF (octal code 6531), ADCV (octal code 6532), ADRB (octal code 6534), ADCC (octal code 6541), ADSC (octal code 6542), and ADIC (octal code 6544). These codes have been described in detail for the AF01A. When the AF02A includes the AC01A sample and hold unit, instructions HSIM (octal code 6572) and SAMP (octal code 6574) also apply. These codes have been described in detail for the AC01A.

TYPE AD08-A AND AD08-B ANALOG-TO-DIGITAL CONVERTER AND MULTIPLEXER

Type AD08-A A/D Converter

The AD08-A A/D converter is an I/O device used with the PDP-8/I [PDP-8/L and DW08-A I/O Conversion Panel], for high-speed, analog-to-digital conversions. The AD08-A is capable of converting analog signals from 0V to +10V in amplitude to a 10-bit digital word. The AD08-A uses the technique of successive approximations for data conversion with an accuracy of 0.1% ± ½ least significant bit (LSB) for quantizing error. Dc power for both the digital logic and the analog logic is supplied with the AD08-A.

General Specifications — The specifications for the AD08-A include:

Analog input voltage (standard)	0V to +10V full scale
with amplifier (option)	±10V max. full scale
with sample and hold (option)	±10V max. full scale
Input impedance (standard)	1000 ohms
with amplifier (option)	≥ 10,000 ohms (inverting) ≥ 100 Mohms (non-inverting)
with sample and hold (option)	≥ 10,000 ohms
Digital output	Parallel Binary 1 = −3V 0 = 0V
Number notation	2's Complement 0V = 0000 +5V = 4000 +10V = 7774
Word Length	10 bits fixed
Accuracy	0.1% of full scale ± ½ LSB
Aperture time (standard)	Same as conversion time
with sample and hold (option)	150 ns
Acquisition time	
with sample and hold (option)	12 μs 100 kHz
Conversion rate	10 μs
Resolution	1 part in 1024 (10 mV)

Power Requirements and Environmental Considerations

Warm-up time	5 min.
Temperature Coefficient	0.5 mV/°C
Operating Temperature	0 to 50°C
Input Power	115V 60 Hz 50 W

The AH02 option may be used preceding the ADC input to reduce the effective aperture to less than 0.15 μs. The AH02 may also be used to scale the signal input to accept ± 10V, ± 5V, or 0V to −10V. The AH03 option may be substituted for the AH02 option to accomplish the same signal scaling, without reducing the effective aperture.

The AH02 and AH03 options may also be used in combination to obtain both high input impedance and small aperture. Power for the amplifier and/or sample-and-hold options is contained in the converter.

Programming — The AD08-A uses the following program instructions:

ADSF (Octal Code 6531), ADCV (Octal Code 6532), and ADRB (Octal Code 6534). These codes have been described in detail for the AF01A.

Type AD08-B A/D Converter and Multiplexer

The AD08-B is an I/O device used with the PDP-8/I [PDP-8/L and DW08-A I/O Conversion Panel] for high-speed conversion of up to 16 time-shared analog voltages.

Multiplexer channel selection occurs via program control, by providing a channel address to the multiplexer control. Two methods of selection are available: one uses an instruction to select a specific channel, and the other uses an instruction to index the address already contained in the multiplexer address register. In the latter case, the multiplexer returns to zero upon command to advance from the final channel. Sequenced operation may be short-cycled when the number of channels used is less than maximum.

An automatic conversion is generated once each time the multiplexer is cleared, indexed, or randomly selected. When a conversion is initiated, the A/D converter converts the selected analog voltage to a 10-bit binary number. The selected input may be sampled as many times as desired between address changes. A status flag indicates the end of each conversion.

Provision is included for the addition of an AH02 or AH03 option. These options are used when it is desired to sample fast time-varying waveforms or for scaling input signal voltages.

General Specifications — The specifications for the AD08-B include:

Basic System

a. 10-bit A-D Converter with input buffer amplifier
b. Multiplexer control for up to 16 channels of analog information (the number of A121 multiplexer switches is optional)
c. Complete interfacing to the PDP8, PDP-8/S, or PDP-8/L computers with necessary I/O cables
d. Input analog connectors (two W023)
e. Necessary power supplies

Options

a. A121 Multiplexer Switch (four channels to a module)
b. AH02 Sample and Hold Option*
c. AH03 Amplifier Option*

*Standard available input voltages are ±10V, 0V to +10V, 0V to −10V, ±5V, 0V to +5V, 0V to −5V. (All gains have been preset at the factory.)

General Performance Specifications (measured at input connectors, W023s):

	Standard	With AH02 Sample and Hold Option	With AH03 Amplifier Option
Input Signal (max)	0V to +10V	±10V	±10V
Input Impedance (typical)	100 Mohm	100 Mohm	100 Mohm
Accuracy	0.1% of full scale ±½ LSB		
Word Length	10 bits fixed		
Resolution	1 part in 1024 (9.8 mV per step)		
Conversion Time (max)	10 μs	10 μs	10 μs
Aperture Time (max)	10 μs	150 μs	10 μs
Acquisition Time (typical) to 1 LSB (1 ohm source)	5 μs	15 μs	8 μs
Droop (max)	N/A	1V/s	N/A
Recovery Time	None	10 μs	None
Operating Temperature	0°C to +50°C		
Temperature Coefficient	0.5 $\frac{mv}{C°}$		
Warm up time, to 0.1%	15 minutes		
Output notation	2's Complement 0V = 0000 +5V = 4000 +10 (−1 LSB) = 7774		
Input Voltage	115 Vac 47-420 Hz 30 W		

Programming — The AD08-B operates under direct control of the central processor. With the exception of the one automatic conversion initiated when switching the multiplexer, all conversions must be generated by the program.

Whenever the multiplexer is switched for any reason, an automatic conversion is initiated. The timing for this automatic conversion allows sufficient time for the switch and amplifier or sample and hold option to settle to within 1 LSB of the newly acquired signal.

The AD08-B operates through the program interrupt. The only flag is A/D done which is set at the end of conversion and remains set until another conversion is initiated or until the A/D buffer is read.

The AD08-B A/D converter option uses the same program instructions as used by the AF01A option. These instruction codes are discussed in this section under the AF01A portion.

TYPE AF04A GUARDED SCANNING INTEGRATING DIGITAL VOLTMETER

Description

The Type AF04A is a Guarded Scanning Integrating Digital Voltmeter system, with wide dynamic range and high common-mode rejection, and is fully capable of expansion to 1000 channels. The AF04A is used with PDP-8/I [PDP-8/L and DW08-A I/O Conversion Panel] systems to multiplex up to 1000 3-wire analog channels into six decimal digit integrating digital voltmeter (IDVM). Each digit is BCD coded for input and display by the IDVM. Full scale ranges are from ±10 mV to ±300V, with automatic ranging, 300 percent overranging, and a usable 5μV resolution. Guarded input construction and active integration assist in attaining an effective common-mode rejection of greater than 140 dB at all frequencies. (Normal-mode rejection is infinite at multiples of power line frequency.)

This system is ideally suited for data acquisition or process monitoring where a wide range of signals requires large dynamic range. The 10mV range has 0.001 percent resolution, and, coupled with a common-mode noise rejection greater than 140 dB at all frequencies, allows accurate direct measurement of thermocouples, strain gauges, load cells, and other low-level transducers without additional amplification.

The AF04A IDVM, operated under program control, is capable of either random channel selection or sequential channel selection. The computer selects either program-controlled ranging (for fastest speed) or auto-ranging, as well as the integration time of the integrating digital voltmeter.

The digitized data, as well as the current channel address, is read by the computer in either two or three bytes.

A decimal display of the digitized value, including sign and decimal location, is continuously displayed on the front panel. The current channel number is also displayed. Front-panel controls on the IDVM allow for manual setting of all the programmed functions. A front-panel control allows continuous display of the internal secondary standard, which can be prewired to a particular channel for reference checking during normal operation. The AF04A may be manually controlled, completely independent of the computer.

Specifications

Characteristic	Specification
Full scale ±	10mV, 100mV, 1V, 10V, 100V, 300V, and automatic ranging
Over-ranging	300% on all but highest range
Maximum Input Voltage	300V
Resolution	5μV (usable), 0.1μV (LSB)
Accuracy (overall worst case with daily calibration temperature)	±0.004% of reading ±0.01% of full scale ±5μV
Stability (RMS full scale and zero drift)	±0.006%/day
Temperature coefficient	±0.003% of reading/°C
Full scale	±0.002% of full scale/°C
Zero	(±0.006% of full scale /°C on 10mV and 1V range)
Line voltage stability	±0.0005%/10% change

Maximum common-mode voltage	±300V from power line ground
Common-mode rejection (166.6ms integration period and 1000-ohm source unbalance)	>140 dB at all frequencies
Normal-mode rejection	Infinite at multiples of line frequency
Input impedance	
10, 100, 1000 mV ranges	1000 megohms/V
10, 100, 300V ranges	10 megohms
Internal secondary standard	
Value	±1.000V
Accuracy	±0.002% traceable to National Bureau of Standards
Stability	±0.005%/month
Temperature coefficient	negligible

Selected Resolution

	0.001%		0.01%		0.1%	
DC Voltage Range	Maximum Reading	Resolution	Maximum Reading	Resolution	Maximum Reading	Resolution
10mV	30.0000mV	0.1μV	030.000mV	1μV	0030.00mV	10μV
100mV	300.000mV	1μV	0300.00mV	10μV	00300.0mV	100μV
1000mV	3000.00mV	10μV	03000.0mV	100μV	003000.mV	1mV
10V	30.0000V	100μV	030.000V	1mV	0030.00V	10mV
100V	300.000V	1mV	0300.00V	10mV	00300.0V	100mV
1000V*	0300.00V	10mV	00300.0V	100mV	000300.V	1V

1000V range is scanner-limited to 300V peak maximum

Scanning Speed

(Programmed Range)

Resolution	Integration Time	Total Time	Speed Scanning
0.1%	1.6 ms	20 ms	50 ch/s
0.01%	16.6 ms	40 ms	25 ch/s
0.001%	166.6 ms	188 ms	5 ch/s

Scanning Speed (Auto-Range) — Add 6-36 ms depending on per-channel voltage span.

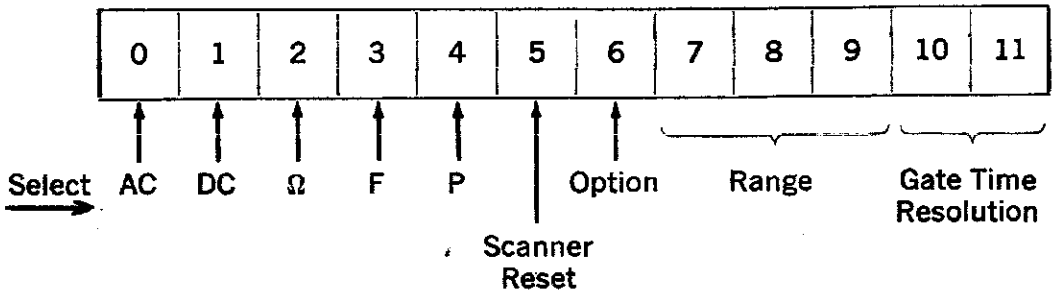
Instructions

The IOT commands associated with the scanning IDVM are designed to minimize the computer overhead associated with this option, while retaining maximum program controlled flexibility. The IOT instructions are:

Select Range and Gate (VSEL)

Octal Code: 6542
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The contents of the accumulator are transferred to the AF04A control register.
Symbol: C(AC) → C(VCR)

Control Word 1 (from PDP-8/I [PDP-8/L])

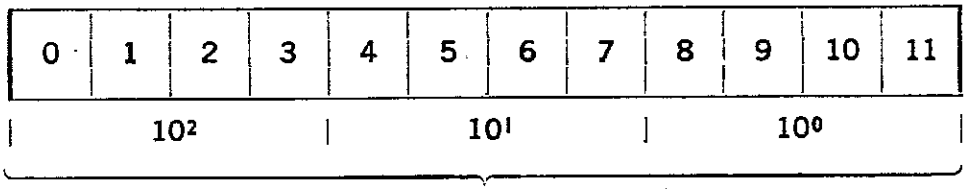


Control Word 1 used only if a range change is required.

Select Channel and Convert (VCNV)

Octal Code: 6541
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The contents of the accumulator are transferred to the AF04A channel address register. The analog signal on the selected channel is automatically digitized.
Symbol: C(AC) → C(VAR)

Control Word 2 (from PDP-8/I) [PDP-8/L]



Index Channel and Convert (VINX)

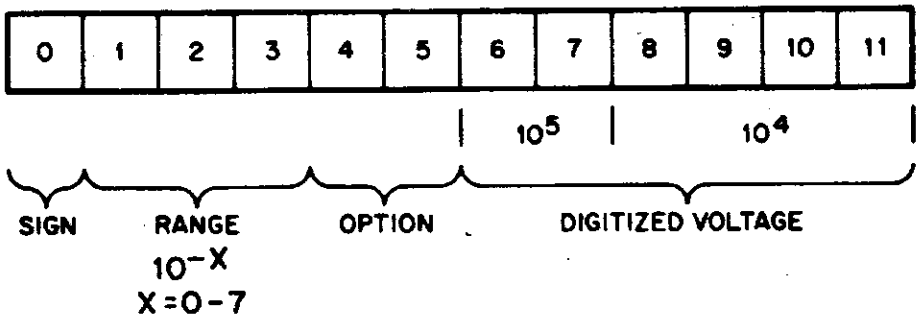
Octal Code: 6544
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The last channel address is incremented by one and the analog signal on the selected channel is automatically digitized. The contents of the control register are unchanged.
Symbol: VAR + 1 → VAR

Skip on Data Ready (VSDR)

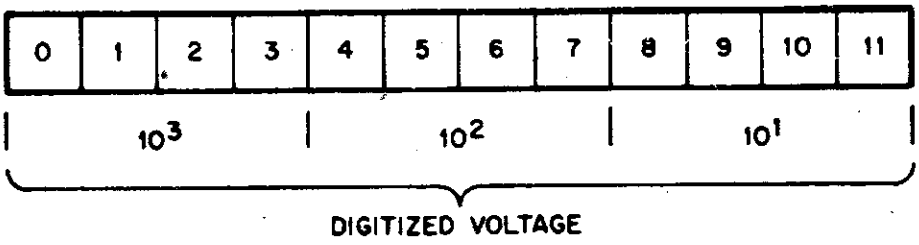
Octal Code: 6561
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: When the scanning voltmeter has selected a channel and digitized the analog signal, a data ready flag is set. This instruction is used to test for the data ready flag.
Symbol: If Flag = 1, then PC + 1 → PC

Read Data and Clear Flag (VRD)
Octal Code: 6562
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the selected byte of the IDVM output word is transferred to the accumulator and the data ready flag is cleared. The first data available after the flag is set is always byte 1. Subsequent bytes are program-selected using the byte advance command.
Symbol: C(VOR) → C(AC)

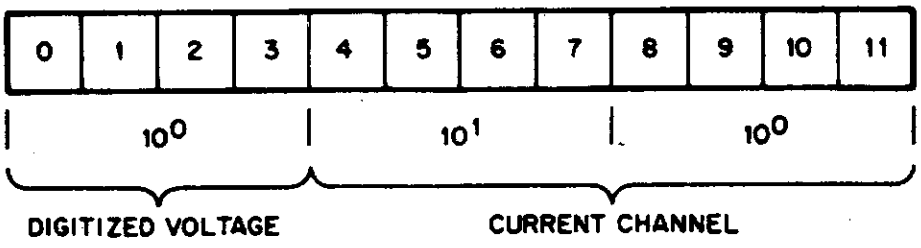
Data Word (to PDP-8/I [PDP-8/L] Byte 1



Data Word (to PDP-8/I [PDP-8/L]) Byte 2



Data Word (to PDP-8/I [PDP-8/L]) Byte 3



All address and digitized data are in 8-4-2-1 BCD format.

Byte Advance (VBA)
Octal Code: 6564
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]

Execution Time: 4.25 μs
Operation: The total data word from the AF04A is 36-bits long. The first data word after the flag is set, is always the 12 most significant bits. The BYTE ADVANCE command requests the next 12 most significant bits. When the data is available, the data ready flag is set again. To select the 12 least significant bits, a second BYTE ADVANCE command is required. When the data is available, the data ready flag is set again.
Symbol: C(VOR0-12) → C(VOR13-23) or
C(VOR13-23) → C(VOR24-35)

Sample Current Channel (VSCC)

Octal Code: 6571
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The analog signal on the current channel is digitized. This command is not required except when multiple samples are required on any channel. (Using this command on a preselected channel saves up to 10 ms per sample.)
Symbol: None

FREQUENCY AND PERIOD MEASUREMENT OPTIONS FOR AF04A

A separate input permits the IDVM to be used as a frequency counter capable of counting to 2 mHz with selectable gate times of 1, 10, and 100 ms, providing measurement resolution of 10 Hz. Increased accuracy at low frequencies (to 10 kHz with automatic 250% over-ranging) is accomplished with the period-measurement mode. This mode counts an internal frequency source for 1, 10, or 100 periods of the frequency being measured, thereby providing increased full-scale accuracy. Period readout is in milliseconds. Frequency and voltage measurements may be made within one scanning cycle by grouping all frequency inputs in one master or slave scanner and all voltage inputs in another master or slave scanner. The output of one scanner may then be connected to the frequency-input connector of the IDVM and the output of the other scanner to the voltage input. One of the optional control word bits is used to program the IDVM for frequency or period measurements.

Specifications

Frequency Measurements
Range: 10 Hz to 2 mHz
Sensitivity: 100 mV rms or −1V pulses, at least 0.3μs wide at 50% points. 100V rms maximum working voltage.
Input Impedance: 22KΩ shunted by less than 1000 pF, including internal cabling.
Accuracy: ±1 count + time base accuracy.
Time Base: 100 kHz crystal oscillator with initial accuracy of ±0.0005%, long-term stability ±0.001%/wk; temp. coefficient ±0.0002%/°C.
Period Measurements
Range: 1, 10, and 100 period average. Input frequency from 10 Hz to 25 kHz sine wave or 0.1 pps to 25,000 pps.
Sensitivity: 100 mV rms or −1V pulses, at least 0.3μs wide at 50% points. 100V rms maximum working voltage.
Input Impedance: 22kΩ shunted by less than 1000 pF, including internal cabling.
Accuracy: ±1 count + time base accuracy + trigger error. Trigger error <±0.03% for 100 mV rms sine wave with 40 dB signal-to-noise ratio.
Time Base: 100 kHz crystal oscillator with initial accuracy of ±0.0005%, long-term stability ±0.0001%/wk; temp. coefficient ±0.0002%/°C.

Selected Resolution

Selected Resolution	0.001%		0.01%		0.1%	
Function	Maximum Reading	Resolution	Maximum Reading	Resolution	Maximum Reading	Resolution
Frequency	2000.00kHz	10Hz	02000.0kHz	100Hz	002000kHz	1kHz
Period	99.9999msec	0.1μs	999.999msec	1.0μs	9999.99msec	10μs

Additional AF04A Options

A Type AF04X Expansion Mounting Panel is available which provides an additional 200 channels. For each 10 channels implemented, the Type AF04S 10-Channel Guarded Reed Relay Multiplexer Switch is required.

These options, as well as those listed below, may be obtained by contacting the nearest office of Digital Equipment Corporation:

Frequency (period) measurements
AC/ohms/DC Converter
Time-of-day clock
Thumb-wheel data entry panel
Thermocouple reference junctions
Extended scanner for more than 1000 channels
Special cabinet with roll-out drawer chassis accessibility.

SECTION 7-9 DIGITAL-TO-ANALOG CONVERTERS

TYPE AA01A DAC

The AA01A general-purpose DAC converts 12-bit binary output values to analog output values. When the AA01A DAC is used with the PDP-8/L computer, the DW08-A I/O Conversion Panel must also be used as an interface between the DAC and the computer. The basic converter consists of three channels, each containing a 12-bit buffer and a DAC. Inputs to all three channels are received in parallel from the computer AC bus. A loading instruction determines which of the three channels is to be loaded with a 12-bit character. Appropriate precision reference voltages are provided for the converter circuits to provide precise voltage outputs.

An IOT instruction simultaneously selects one of the three converter channels and transfers a 12-bit character into the channel buffer. The channel converter operates continuously on the content of the associated channel buffer to provide an output voltage that is an analog value of the buffered character.

The AA01A DAC options can be specified in a wide range of basic configurations; e.g., one to three channels, with or without output operational amplifiers, and with internally or externally supplied precision reference voltages. Configurations with double buffer registers in each channel are also available. Double buffering allows data to be stored for transfer at a later time to the AA01A.

Each single-buffered converter channel is operated by a discrete instruction. Device selection codes 55, 56, and 57 are assigned to the DAC, enabling selection of nine single-buffered channels or various configurations of single- and double-buffered channels. The instruction for outputting data to the DAC is:

Load Digital-to-Analog Converter 1(DAL1)

Octal Code: 6551 (where 55 selects channel 1)
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The character in the accumulator is loaded into the channel 1 buffer. The DAC then converts the buffered value to the analog equivalent.
Symbol: AC → DAC1

NOTE:

Similar instructions for DAL2 and DAL3 load respective DACs.

The analog output voltage of the DAC ranges from ground (0V) to −9.9976V (other voltages are available by use of AH03 output operational amplifiers. All computer output characters are assumed to be 12 bits in length with negative numbers represented in two's complement notation. A character code of 4000_8 yields an analog output of 0V; a code of 0000_8 yields an output of −5V; and a code of 777_8 yields an output of −10V (nominal), minus the analog value of the least significant bit. Output accuracy is ±0.0125% of full scale and resolution is 0.025% of full scale value. Response time, measured directly at the DAC output, is 3 μs for a change of a full-scale step minus one least significant bit.

TYPE AA05/AA07 DIGITAL-TO-ANALOG CONVERTER/EXPANSION UNIT

The AA05 Digital-to-Analog Converter (DAC) and control provides housing, power, and control for up to 24 10-bit DAC modules. The AA07 Expansion Unit extends the capacity of the system to 64 channels of DAC.

Each conversion channel may use any of the four printed circuit card DAC modules. These modules include two single-buffered units, Types A608 and A609, and two double-buffered units, Types A610 and A611. A608 is a single-buffered, 10-bit DAC, with unipolar output (0V to +10V). Type A609 is a single-buffered, 10-bit DAC with bipolar output and variable offset. A610 and A611 are similar to A608 and A609, respectively, except that the former are double-buffered units.

The principal power supply furnishes all power for up to 64 DAC modules, with the exception of the −10V reference power. Reference power is furnished by the Type H706 Reference Power Supply which is optional to the AA05/AA07 unit. A maximum of five H706 supplies can be allocated to the various DAC channels, two of which are in the AA05 and three of which are in the AA07.

The AA05/AA07 DAC and expansion unit is used with the PDP-8/I [PDP-8/L] computers to control up to 64 DAC channels. [When the AA05 DAC is used with the PDP-8/L computer, the DW08-A I/O Conversion Panel must also be used as an interface between the DAC and the computer.] Both the DAC address and the digital word to be converted are program-controlled as two I/O data words for 12-bit computers. The DAC address is stored in the AA05 and remains there until changed by the program for fast updating of any channel.

Six indicators on the front panel of this device indicate the binary address of the DAC channel currently being addressed. All data bits and I/O transfer commands are buffered to present a minimum load to the computer bus even with 64 DACs in use. The AA07 expansion assembly allows expansion to 64 single- or double-buffered DACs.

The AA05/AA07 consists of a 10-bit buffer register, level converters, a precision divider network, and a current-summing amplifier capable of driving large external loads. Provisions are made for double-buffering and bipolar output voltage where required. A precision reference voltage, supplied externally by the H706 power supply, ensures greater efficiency and optimum scale-factor matching in multiple-channel systems.

Software differences exist between the AA05/AA07 and the AA01/AA04 DAC systems. Where the AA01/AA04 employs only one basic instruction for each of three channels (DAL1, DAL2, or DAL3), the AA05/AA07 DAC utilizes four separate instructions. These instructions clear the DAC address register, transfer the contents of AC_{0-9} to the input register of the selected DAC, and update all double-buffered channels (if applicable).

SECTION 7-10 COMMUNICATIONS SUBSYSTEMS

The communications subsystems available as options to the PDP-8/I and the PDP-8/L include the DP01AA Synchronous Modem Interface and the PT08 Asynchronous Serial Line Interface. For the PDP-8/I, a complete communications system, designated the 680/I, can be built up by using the DL8/I and DC08 options. For the PDP-8/L the communications subsystem option DC02A Multiple Asynchronous Line Unit control is available.

MODEL DP01AA SYNCHRONOUS MODEM INTERFACE

The DP01AA is an interface between the computer and a full- or half-duplex serial synchronous modem having interface characteristics compatible with EIA RS-232-B standards. An XOR (exclusive OR) option, Type DP01E, adds a non-memory reference XOR command to the computer, to facilitate special character detection and longitudinal message parity generation.

Functional Description

The DP01AA consists of two independent channels for interfacing to a synchronous modem. One channel is designated as the send channel and the other is designated as the receive channel. Basically, this unit converts parallel characters to a serial bit stream for the send channel and, conversely, converts serial bit streams to parallel for the receive channel.

In synchronous communication systems, continuity in character flow has to be maintained. To ease the software burden, input and output are double-buffered so that response time is equal to a character length instead of a bit length.

Character synchronization between the DP01AA and the distant station is established by a unique sync character. Once a sync character is detected, the receiving channel assembles every 6, 7, 8 or 9 successive bits to form a character.

Data Format

Serial data is transmitted and received continuously once synchronization is achieved. The transmission format consists of sync characters (three in succession are recommended), followed by the characters which make up the text of the message. Character lengths of 6, 7, 8 or 9 bits are selected by a pre-wired patch plug, to allow the DP01AA to communicate with remote sites at various word lengths. Sync characters provide a frame reference at the start of every message to enable a receiving terminal to determine which successive bits in the incoming serial stream make up each character. When the DP01AA

receive channel recognizes a sync character, it assembles every N bits in a buffer (N = 6, 7, 8, or 9). Sync characters employed by the DP01AA are:

Bits Per Character	Sync Characters
6	010 110
7	0 010 110
8	10 010 110
9	010 010 110

Specifications

Specifications for the DP01AA are listed in Table 7-12.

Table 7-12 DP01AA Specifications

Requirement	Specification
Type of Channel	Serial synchronous, half- or full-duplex
Speed	Function of modem being used
Interface	Conforms to EIA standard RS-232-B
Data Format	6-, 7-, 8-, or 9-bit serial characters selected by pre-wired plug-in connector; unique sync character code for each character length; least significant bit is transmitted and received first. (Sync codes are listed above.)
Data Sets	Partial list of data sets which are compatible with the DP01AA includes: Type — Speed (Baud) Bell 201A — 2K Bell 201B — 2400 Bell 205B — 600, 1200, 1800 Rixon FM-12 — 1200 Rixon Sebit 48 — 4800 GE TDM Series — 2400 Lenkurt 26C — 150-2400
Operating Function	Character length-selection connector is installed in control logic assembly to determine whether 6-, 7-, 8-, or 9-bit characters will be supplied.

Programming

The following IOT commands are assigned to the DP01AA:

Skip on Transmit Flag (STF)

Octal Code: 6611
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Causes the program to skip the next instruction if the transmit flag is in the 0 state. When the transmit flag is in the 1 state, the transmit buffer register (TB) is ready to accept another character.
Symbol: Transmit flag = 0, PC + 1 → PC
Transmit flag = 1, PC + 1, TB ready to receive next character.

Clear Transmit Flag (CTF)

Octal Code: 6602
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Resets the transmit flag. If transmit active flag is set, CTF also causes the program to skip the next instruction.
Symbol: 0 → transmit flag
If transmit active flag = 0, PC + 1 → PC

Transmit A Character (TAC)

Octal Code: 6601
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Causes the contents of the AC (6, 7, 8, or 9 bits right-justified) to be transferred into the TB.
Symbol: AC → TB

Clear Idle Mode (CIM)

Octal Code: 6604
Event Time: 4
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Resets the transmit logic idle mode (IM) flip-flop.
Symbol: 0 → IM

Set Idle Mode (SIM)

Octal Code: 6614
Event Time: 4
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Sets the transmit idle mode (IM) flip-flop.
Symbol: 1 → IM

Skip on Receive Flag (SRF)

Octal Code: 6651
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Causes the program to skip the next instruction if the receive flag is zero. The flag is set when a received character is ready for transfer to the AC and the flag is cleared when an RRB instruction is issued.
Symbol: Receive flag = 0, PC + 1 → PC

Read Receive Buffer (RRB)

Octal Code: 6612
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Transfers the contents of the receive buffer (RB) (6, 7, 8, or 9 bits right-justified) to the computer AC. RRB also resets the receive flag.
Symbol: RB → AC
0 → Receive flag

Skip on Receive End Flag (SEF)

Octal Code: 6621

Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Causes the program to skip the next instruction if the receive end flag is a zero. (The receive end flag flip-flop is set when the receive logic has stopped receiving serial data from the communications equipment due to termination of the serial clock receive pulse train.)
Symbol: Receive end flag = 0, PC + 1 → PC

Clear End Flag (CEF)

Octal Code: 6622
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears the Receive End flag.
Symbol: 0 → Receive End flag.

Set Ring Enable (SRE)

Octal Code: 6624
Event Time: 4
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Sets the ring enable (RE) flip-flop to a one which permits the ring flag to request a program interrupt.
Symbol: 1 → RE
Ring flag requests program interrupt

Clear Ring Enable (CRE)

Octal Code: 6644
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears the ring enable (RE) flip-flop.
Symbol: 0 → RE

Skip on Ring Indicator (SRI)

Octal Code: 6631
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Causes the program to skip the next instruction if the ring flag is a 0. The ring flag is set (1) when a ring input is received.
Symbol: Ring flag = 0, PC + 1 → PC

Clear Ring Flag (CRF)

Octal Code: 6632
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears the ring flag.
Symbol: 0 → Ring flag

Set Terminal Ready (STR)

Octal Code: 6634
Event Time: 3

Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Sets the terminal ready (TR) flip-flop to the 1 state. This causes the terminal ready lead to the modem to be set to the ON state.
Symbol: 1 → TR

Clear Terminal Ready (CTR)

Octal Code: 6642
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Resets the ring enable (RE) flip-flop to the 0 state. This causes the terminal ready lead to the modem to be set to the OFF state.
Symbol: 0 → RE

Skip on Data Set Ready (SSR)

Octal Code: 6641
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Causes the program to skip the next instruction if the data-set-ready lead from the modem is in the ON state.
Symbol: Communications equipment ready? If yes, PC + 1 → PC

Clear Receiver Active (CRA)

Octal Code: 6652
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears the receive active (RA) flip-flop, taking the receive logic out of the active state. This inhibits any more receive flags until a new sync character is received.
Symbol: 0 → RA

The following command pertains to the exclusive OR buffer (XOR) option:

Clear XOR Buffer (COB)

Octal Code: 6661
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Clears the XOR buffer.
Symbol: 0 → XOR

Idle/Non-Idle Modes

In the normal non-idle mode of operation, selected by the CIM command, the DP01AA will stop transmitting (hold the transmitted data line in the binary 1 condition and set the request-to-send lead to the modem to the OFF state), if a new character is not available for transmission (indicate skip on clear transmit flag). An alternate idle mode can also be set up by the SIM command. In the transmit idle mode, when a new character is not available, an idle character transmission is repeated (the last character that was transmitted).

Clear Receive Active

In a number of applications, it is desirable for the remote station or terminal to idle the transmission line by sending idle characters to the DP01AA between valid data messages. The clear receive action commands put the receive logic out of sync, thus requiring a new sync character to be seen before causing any

receive flags to be set. This feature prevents idle characters from causing program interrupts.

XOR Options

This option (Type DP01E) adds a non-memory reference XOR (Exclusive OR) command to the computer, to facilitate special character detection and longitudinal message parity generation.

Inclusive OR Buffer (IOB)

Octal Code: 6664
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Transfers 1s from the AC to the buffer register (BR).
Symbol: AC V 1s BR

Read OR Buffer (ROB)

Octal Code: 6662
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Transfers the buffer register (BR) content to the AC.
Symbol: 1 V AC_{im}

Exclusive OR Buffer (XOB)

Octal Code: 6654
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: Causes an Exclusive OR of the AC with the buffer register (BR).
Symbol: AC V BR
The result is in the Buffer register.

Maintaining Transmit Data Stream

Since the DP01AA operates in a bit-synchronous mode, data must be made available for transmission as it is needed. Since every N bits are assembled to form a character, an extra or dropped bit will cause a faulty character at the receiving station. The DP01A safeguards against interruptions in the character pattern by double buffering and by program interrupt requests for new characters.

As the last bit of each character is transmitted, the character in the transmit buffer register is automatically loaded into the output shift register, and a "transmit flag" is set, causing a program interrupt. The program responds with a TAC instruction that loads a new character into the transmit buffer register. This method allows a full character time for the program to update the buffer character. The transmit flag is reset by a CTF instruction after the character has been delivered.

TYPE PT08 ASYNCHRONOUS SERIAL LINE INTERFACE

The PT08 is a serial-to-parallel, parallel-to-serial converter which provides full-duplex communication between an asynchronous channel and a PDP-8/I or PDP-8/L computer. Two basic configurations are offered: PT08B (one full-duplex channel) and PT08C (two full-duplex channels). Systems may be expanded up to five duplex channels by stacking PT08 units.

The PT08B and C are designed to supply transmit and receive keying current

that is intended for use with 20 mA, dc-keyed devices. Digital Equipment Corporation's Model 33 or 35 teleprinter units have been modified to be compatible with the PT08. Devices equivalent to the modified teleprinter units are also compatible with the PT08.

Any PT08 channel can be modified by the PT08F option for compatibility with EIA standard RS-232-B interface logic levels. Another option, the PT08X, can be installed in any channel for customer selection of character format and speed. With the PT08F and PT08X options combined, the bit rate can be increased to 100K baud for driving medium- to high-speed asynchronous modems. This combination can be used for an economical inter-computer communication channel or for interfacing to special equipment with unique asynchronous speeds and character formats. (Modem interfaces are for data transmission only.)

Specifications

Performance specifications are summarized in Table 7-13.

Table 7-13. PT08 Specifications

Characteristics	Specifications
Speed	110 baud is standard; up to 100K (software limited) with PT08X option.
Character Format	Standard: 8 character bits; 2-unit stop element. PT08X Option: 5 or 8 character bits, 1- or 1.5-unit stop element at user's request.
Operating Mode	Full duplex.
Interface	Standard: Supplies transmit and receive keying current that is intended for use with 20 mA, dc-keyed devices. PT08F Options: Provides interface that conforms to EIA RS-232-B devices.
Transmission Distance	1500 ft. maximum (environment dependent) for local terminals. EIA interface transmission distance is limited only by characteristics of modem and associated communication facility. A 25-ft. cable to the modem is supplied.

Figure 7-10 illustrates the various PT08 equipment configurations for both the standard system expansion and interface provisions.

Programming

The following IOT instructions test for character-ready conditions and transfer assembled characters to and from the computer's accumulator. The same basic commands are used for all channels, with individual channels assigned different device selection codes. The device codes for the basic PDP-8/I teleprinter are 03 and 04; for PT08 channel 1, the devices codes are 40 and 41; etc. (see Table 7-14 for complete listing of PT08 device codes). The basic mneumonic plus the PT-number designator identifies the mneumonic for the specific channel.

PT08F Option — This option consists of an adapter which can convert any PT08 channel to compatible interfacing with the EIA RS-232-B (Dataphone or equivalent) modems. The modems often used with the PT08 consist of the Dataphone/teleprinter. A significant difference between the characteristics of the teleprinter used with the PT08 and that used with the PT08F option are

that the teleprinter used with the PT08B or C receive station is clutched to the PT08; whereas, the receive (or read) section of the teleprinter is not clutched to the PT0F when that option is used. This unit does not control other modem leads such as: request to send on half-duplex facilities. The terminal-ready signal is forced on.

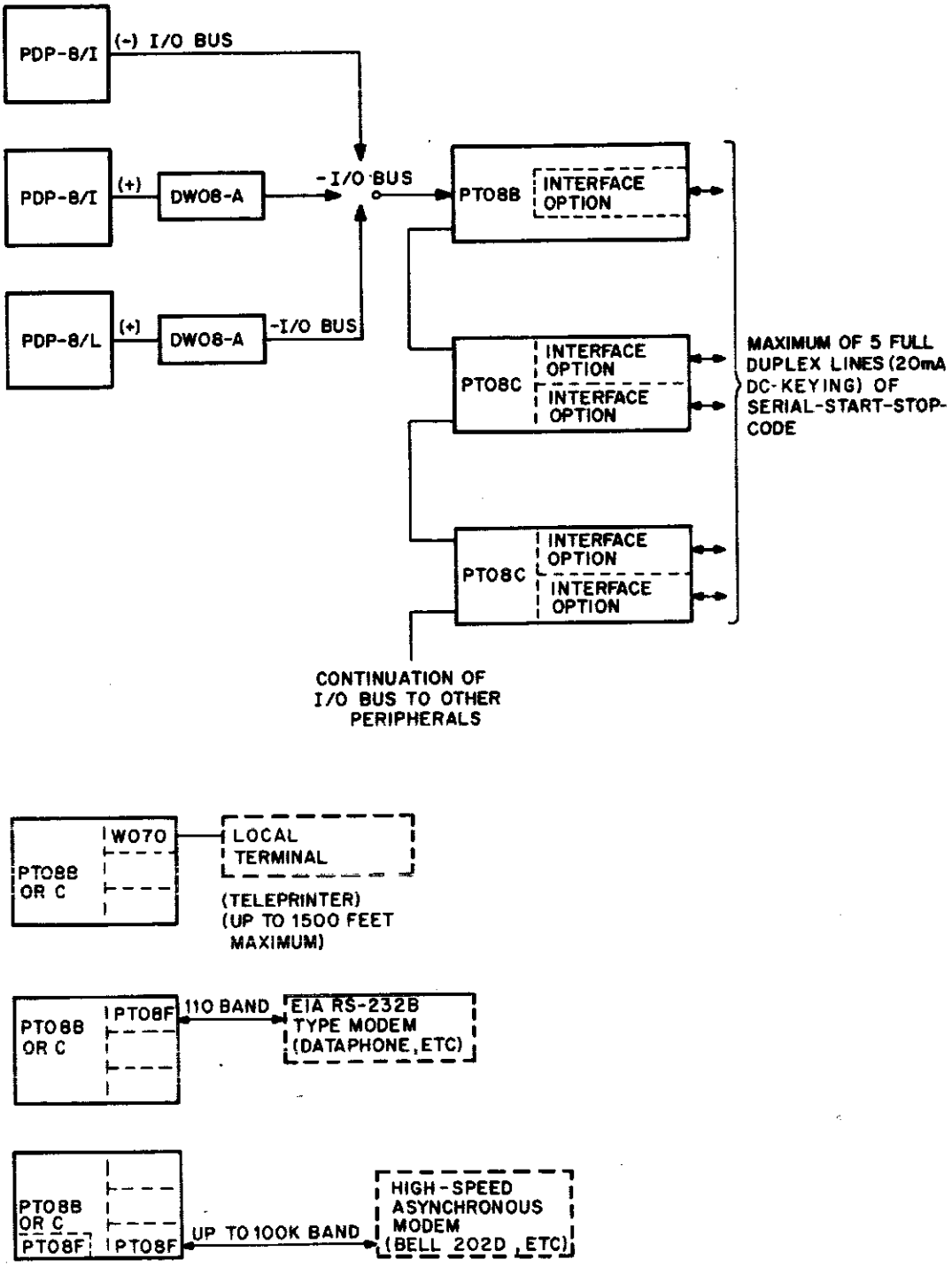


Figure 7-10. PT08 Equipment Configurations

PT08X Option — This option can be installed in any channel for user's selection of character format and speed.

Instructions

The following instructions are used with the PT08:

Table 7-14. PT08 Device Codes

BASIC MNEUMONIC	CHANNEL NUMBER 1	2	3	4	5
KSF	6401 KSFPT1	6421 KSFPT2	6441 KSFPT3	6461 KSFPT4	6111 KSFPT5
KCC	6402 KCCPT1	6422 KCCPT2	6442 KCCPT3	6462 KCCPT4	6112 KCCPT5
KRS	6404 KRSPT1	6424 KRSPT2	6444 KRSPT3	6464 KRSPT4	6114 KRSPT5
KRB	6406 KRBPT1	6426 KRBPT2	6446 KRBPT3	6466 KRBPT4	6116 KRBPT5
TSF	6411 TSFPT1	6431 TSFPT2	6451 TSFPT3	6471 TSFPT4	6121 TSFPT5
TCF	6412 TCFPT1	6432 TCFPT2	6452 TCFPT3	6472 TCFPT4	6122 TCFPT5
TPC	6414 TPCPT1	6434 TPCPT2	6454 TPCPT3	6474 TPCPT4	6124 TPCPT5
TLS	6416 TLSPT1	6436 TLSPT2	6456 TLSPT3	6476 TLSPT4	6126 TLSPT5

Skip on Receive Flag (KSF)

Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Causes the program to skip the next instruction if the receive flag is set, indicating that an assembled character is ready.
Symbol: If receive flag = 1, PC + 1 → PC

Clear Receive Flag and AC (KCC)

Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clears the accumulator and the receive flag.
Symbol: 0 → AC, 0-RF

Read Receive Buffer (Static) (KRS)

Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Transfers an assembled character from the receive buffer to AC 4-11. Does not reset AC or receive flag.
Symbol: RB → AC 4-11

Read Receive Buffer (Dynamic) (KRB)

Event Time: 2, 3
Indicators: Iot, Fetch, Pause

Execution Time: 4.25 μs
Operation: Performs the functions of KCC and KRS together, so that the receive flag and AC are cleared before data is transferred from the receive buffer to the AC.
Symbol: 0 → AC, 0 → RF
RB → AC 4-11

Skip on Transmit Flag (TSF)

Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Causes the program to skip the next instruction if the transmit flag is set, indicating that the transmit buffer is ready for another character.
Symbol: If transmit flag = 1, PC + 1 → PC

Clear Transmit Flag (TCF)

Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Resets the transmit flag.
Symbol: 0 → TF

Load Transmit Character (TPC)

Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Loads the transmit buffer from AC 4-11 and initiates transmission of a character.
Symbol: AC 4-11 → TB

Load Transmit Sequence (TLS)

Event Time: 2, 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Performs the functions of TCF and TPC together.
Symbol: 0 → TF
AC 4-11 → TB

Maximum Data Rates

In transmitting the PT08 provides a full character cycle for the program to deliver new data. In receiving one bit-time is required to read in necessary data. However, for maximum data transfer rates, the time at which data transfer can occur is limited to an aperture equal to the stop bit time plus half a bit time. This response time is measured from the beginning of a stop bit (the time at which the transmit or receive flag is reset), and the midpoint of the next character's start bit. If the program fails to respond within this time, a character is lost. Timing is illustrated in Figure 7-11.

MODEL DC02 MULTIPLE ASYNCHRONOUS LINE UNIT

The DC02 option for the PDP-8/L allows the user to add from one to four serial-to-parallel, parallel-to-serial asynchronous channels. The DC02 is a pre-wired option in the BA08 Peripheral Expansion Unit and is used in the PDP-8/L computer. The DC02 option consists of: the DC02A Serial Line Interface control and from one to four DC02D Line Interface Units (each full-duplex). The DC02D Line Interface Units are designed to supply transmit and receive keying current that is intended for use with 20 mA dc-keyed devices. Digital Equipment

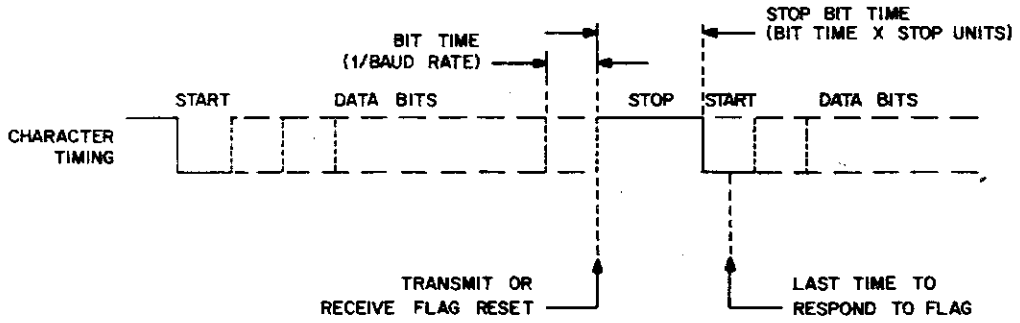


Figure 7-11. PT08 Program Response Time

For example, at 110 baud (9.09 ms bit time), response time is:
Stop bit time + half a data bit time
= 2 X 9.09 + 9.09/2
= 22.725 ms

Note that the number of bits per character need not be considered.

Corporation's Model 33 or 35 teleprinter units have been modified to be compatible with the DC02A. Equivalent devices to the modified teleprinter units are also compatible with the DC02A.

Any DC02A channel can be modified by the DC02DA Format Speed option to specify the speeds of lines 1 and 2 or lines 3 and 4 and the format of each line unit. A Type BC01A-25 Modem Interface Adapter is available for modifying any DC02A for compatibility with EIA standard RS-232-B interface logic levels. With the DC02DA and BC01A-25 options combined, the bit rate can be increased to 125K baud for driving medium- to high-speed asynchronous modems (modem interfaces are for data transmission only).

DC02 Specifications

Characteristics	Specifications
Speed	110 baud is standard; up to 100K (software limited) with DC02DA option.
Character Format	Standard: 8 character bits; 2-unit stop element. DC02DA option: 5 or 8 character bits 1- or 1.5-unit stop element at user's request.
Operating Mode	Full duplex.
Interface	Standard: Supplies transmit and receive keying current that is intended for use with 20 mA, dc-keyed devices.
Transmission Distance	1500 ft. maximum (environment dependent) BC01A Option: Provides interface that conforms to EIA RS-232-B devices. for local terminals. EIA interface transmission distance is limited

only by characteristics of modem and associated communication facility. A 25-ft. cable to the modem is supplied.

DC02A Control Instructions

The following instructions are used with the DC02A control:

Multiple Teleprinter Flag (MTPF)

Octal Code: 6113
Event Time: 1, 2
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Transfer status of teleprinter flags to AC 0-3.
Symbol: Teleprinter flags → AC 0-3

Multiple Interrupt (MINT)

Octal Code: 6115
Event Time: 3
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Interrupt on if AC 11 is set (interrupt request if any flags).
Symbol: If AC 11 = 1, Interruption

Multiple Teleprinter (MTON)

Octal Code: 6117
Event Time: 1, 2, 3
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Transfer AC 0-3 to selection register (SR) (select stations when bit is set).
Symbol: AC 0-3 → SR

Multiple Teleprinter Keyboard Flag (MTKF)

Octal Code: 6123
Event Time: 1, 2
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Transfer status of keyboard flags (KFs) to AC 0-3.
Symbol: KFs → AC 0-3

Multiple Interrupt and Skip (MINS)

Octal Code: 6125
Event Time: 1, 4
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Skip if the interrupt request is active (if interrupt is on and any flag is raised).
Symbol: If interrupt request is active, PC + 1 → PC

Multiple Teleprinter Read Status (MTRS)

Octal Code: 6127
Event Time: 1, 2, 3
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Transfer the status of the selection register (SER) to AC 0-3.
Symbol: SER → AC 0-3

DC02D Instructions

Decoder instructions for the DC02D are in two basic groups: receiver and transmitter. The receiver instructions are 6111, 6112, 6114, and 6116; the transmitter instructions are 6121, 6122, 6124, and 6126.

Multiple Keyboard Skip on Flag (MKSF)

Octal Code: 6111
Event Time: 1
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Skip the next instruction if the keyboard flag is set.
Symbol: If KF = 1, PC + 1 → PC

Multiple Clear Keyboard Flags (MKCC)

Octal Code: 6112
Event Time: 2
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Clear the keyboard and reader flags and enable the reader on the teleprinter; clear AC.
Symbol: 0 → KF, 0 → RF, 0 → AC

Multiple Keyboard Read and Shift (MKRS)

Octal Code: 6114
Event Time: 3
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Transfer the shift register contents to AC 4-11.
Symbol: SR → AC 4-11

Multiple Load Keyboard Sequence (MKRB)

Octal Code: 6116
Event Time: 2, 3
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Clear the keyboard and reader flags, clear AC; transfer the shift register contents to AC 4-11 (MKCC and MKRS combined).
Symbol: 0 → KF
0 → RF
0 → AC
SR → AC 4-11

Multiple Teleprinter Skip on Flag (MTSF)

Octal Code: 6121
Event Time: 1
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Skip the next instruction if the teleprinter flag is set.
Symbol: If TF = 1, PC + 1 → PC

Multiple Clear Teleprinter Flag (MTCF)

Octal Code: 6122
Event Time: 2
Indicators: IR = 6, F
Execution Time:4.25 μs

Operation: Clear the teleprinter flag.
Symbol: 0 → TF

Multiple Load Teleprinter and Print (MTPC)

Octal Code: 6124
Event Time: 3
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Load AC 4-11 into the shift register (begin print/punch).
Symbol AC 4-11 → SR

Multiple Load Teleprinter Sequence (MTLS)

Octal Code: 6126
Event Time: 2, 3
Indicators: IR = 6, F
Execution Time: 4.25 μs
Operation: Clear the teleprinter flag and load AC 4-11 into the shift register (MTCF and MTPC combined).
Symbol: 0 → TF
AC 4-11 → SR

680/I DATA COMMUNICATIONS SYSTEM

The 680/I Data Communications System is designed as an economical solution to servicing a number of low speed asynchronous communication lines. The system is physically capable of servicing up to 128 full- or half-duplex lines, with a series of line interface options that provide physical and electrical compatibility between the 680/I system line units and the various communication facilities. The 680/I system can accommodate all peripherals which are available on the PDP-8/I.

The basic 680/I system consists of a PDP-8/I with a DL8/I option (which converts the PDP-8/I to a 680/I), which adds two instructions to the PDP-8/I giving it the ability to assemble a character in core memory by sampling the state of a serial asynchronous data stream (such as that generated by a Teletype). This line scanning capability is multiplexed by a DC08A option which is physically capable of servicing up to 128 lines. Line servicing capability is implemented in a module two fashion using dual Type M750 Serial Line Units. The end result is the efficient use of a very economical computer to replace line sampling and character assembly functions which are normally done by dedicated hardware.

For a detailed description of how the 680/I works and its programming characteristics refer to the Digital Equipment Corporation publication entitled, "What is the 680/I-CAN-2."

Specifications

Table 7-15. 680/I Specifications

Characteristics	Specifications
Speed	0 to 150 baud
Character Format	5-, 7-, or 8-level code (supported by DEC software).
Operating Mode	Full duplex (half-duplex operation is available).

Numbers of Channels	128 full duplex; expansion of a system is modular on a per-line basis.
Interface	20 mA dc switching for local connection to Teletype Model 33, 35, or equivalent.
	20 to 60 mA polar or neutral World Trade Telegraph lines.
	RS-232 interface devices for Dataphone service.
Transmission Distance	1500 ft. for local terminals.
	Built-in adjustments permit operation over several miles of dc-keyed telegraph wire lines.
	EIA RS-232-B interface: 25-ft. cable supplied. Distance beyond data set limited only by communication medium.

Following is a list of IOT commands which are associated with the 680/I and its related DC08 options. This information is for programming reference only and the appropriate application note should be referenced for a detailed discussion of a unit's operating characteristics.

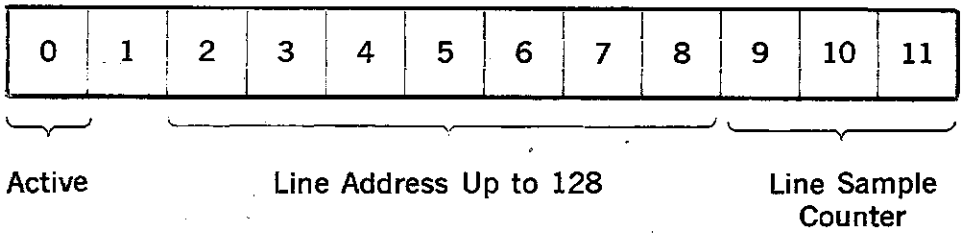
The DC08F is an EIA Modem Interface Connector Panel that may be used to interface and control as many as 64 103A, 103E, or the equivalent, Dataphone lines to the DC08A: Line control is provided by one DC08G for every two data lines. The DC08H is an Automatic Callup Unit which provides automatic callup interface for 10 data lines on a per-line basis, using a DC08J module set.

680/I Instructions

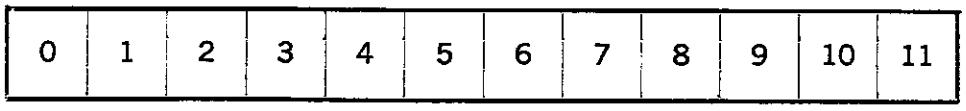
Sample Line and Shift Contents (TTI)

Octal Code: 6402
Event Time: 2
Indicators: Iot, Fetch
Execution Time: 3.4 μs + 1.5 μs if CAW is fetched.
Operation: Causes a JMS to be executed (N+3) if the R Register (RR) does not equal 0 and either the line hold bit of the selected line (specified by bits 2-8 of the LSW) is in the 1 state, or as a result of a jamming of the line state into and shifting the CAW; bit 11 of the CAW is a 1.

LSW Format



CAW Format



State of the line is shifted into bit 0 and all bits are shifted right by one bit position. Bit 11 is lost. The TTI instruction specifies the line to be sampled by bits 2-8 of the LSW. The effect of the TTI is dependent on the state of the active bit (bit 0 of the LSW) and line sample counter (bits 9-11 of the LSW).
Symbol: None

Clear Link and Shift Link and Accumulator (TTO)

Octal Code: 6404
Event Time: 3
Indicators: Iot, Fetch
Execution Time: 1.5 μs
Operation: Clears the link and shifts the link and accumulator one bit position to the right. Bit 11 of the accumulator is shifted into the line unit specified by the line register. The previous contents (1 bit) of the selected line unit is lost.
Symbol: 0 → L,
L shifted 1 bit to the right
AC shifted 1 bit to the right

Clear Line Register (TTCL)

Octal Code: 6411
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Sets the contents of the line register to 0.
Symbol: 0 → LR

Shift Data Into Line Register (TTSL)

Octal Code: 6412
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Contents of AC 5-11 are ORed into the line register.
Symbol: AC 5-11 V LR

Clear and Shift Data into Line Register (TTLL)

Octal Code: 6413
Event Time: 1, 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: The line register is cleared and the contents of AC 5-11 are transferred (ORed) into the line register. This is a microprogram of TTCL and TTSL.
Symbol: 0 → LR, AC 5-11 V LR

Shift Data Into Accumulator (TTRL)

Octal Code: 6414
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Contents of the line register are ORed into AC 5-11. The AC must be 0 for a true transfer.
Symbol: LR V AC 5-11

Increment Line Register (TTINCR)

Octal Code: 6401
Event Time: 1

Indicators: Iot, Fetch
Execution Time: 4.25 μs
Operation: Causes contents of the line register to be incremented by 1. This command, when microprogrammed with a TTO, causes the increment to occur after the TTO command is executed.
Symbol: LR + 1

Increment R Register (TTRINC)

Octal Code: 6461
Event Time: 1
Indicators: Iot, Fetch
Execution Time: 4.25 μs
Operation: This command causes the contents of the R Register to be incremented by 1. Because it is loaded with a two's complement number, the result is a subtract. This instruction can be microprogrammed with TTRR.
Symbol: RR + 1

Read R Register (TTRR)

Octal Code: 6464
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Read contents of R Register into AC 7-11. Contents of AC must be 0s before issuing this instruction. This instruction, when microprogrammed with TTINCR causes the incremented results to be read into the AC.
Symbol: RR → AC 7-11

Clear R Register (TTCR)

Octal Code: 6471
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Causes R Register to be set to 0.
Symbol: 0 → RR

Shift Data Into R Register (TTLR)

Octal Code: 6472
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Contents of AC 7-11 are ORed into the R Register.
Symbol: AC 7-11 V RR

Clear and Shift Data Into R Register (TTLDR)

Octal Code: 6473
Event Time: 1, 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clears the R Register and transfers contents of AC 7-11 into the R Register. This is a microprogram of TTCR and TTLR.
Symbol: 0 → RR, AC 7-11 V RR

Clock 1 On (T1on)

Octal Code: 6424
Event Time: 3
Indicators: Iot, Fetch, Pause

Execution Time: 4.25 μs
Operation: Clock control instruction which enables clock 1 to set its flag at the predetermined clock rate. The flag in the 1 state causes a program interrupt when the interrupt is enabled. This instruction also sets the flag to the 0 state.
Symbol: None

Clock 1 Off (T1off)
Octal Code: 6422
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clock control instruction which inhibits clock 1 from setting its flag. This instruction also sets the flag to the 0 state.
Symbol: None

Clock 1 Skip (T1skip)
Octal Code: 6421
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clock control instruction which causes the program to skip the next instruction if clock flag 1 is in the 1 state. To clear the flag, either T1on or T1off can be used.
Symbol: None

Clock 2 On (T2on)
Octal Code: 6434
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1on, except enables clock 2.
Symbol: None

Clock 2 Off (T2off)
Octal Code: 6432
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1off, except inhibits clock 2.
Symbol: None

Clock 2 Skip (T2skip)
Octal Code: 6431
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1skip, except flag 2 is in the 1 state.
Symbol: None

Clock 3 On (T3on)
Octal Code: 6434
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1on, except enables clock 3.
Symbol: None

Clock 3 Off (T3off)
Octal Code: 6432
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1off, except inhibits clock 3.
Symbol: None

Clock 3 Skip (T3skip)
Octal Code: 6441
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1skip, except flag 3 is in the 1 state.
Symbol: None

Clock 4 On (T4on)
Octal Code: 6454
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1on, except enables clock 4.
Symbol: None

Clock 4 Off (T4off)
Octal Code: 6452
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1off, except inhibits clock 4.
Symbol: None

Clock 4 Skip (T4skip)
Octal Code: 6451
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Same as T1skip, except flag 4 is in the 1 state.
Symbol: None

DC08F Instructions

Skip on Ring Flag (SRF)
Octal Code: 6701
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: The next instruction is skipped when the ring flag is set to a 1.
Symbol: Ring flag = 1, PC + 1 → PC

Read Ring Scanner (RRS)
Octal Code: 6702
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Contents of the ring scanner (RS) are ORed into AC 5-11.
Symbol: RS V AC 5-11

Enable Data Flip-Flop (EDF)
Octal Code: 6704
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Enables AC decoding and data set interrupts (set F-enable flip-flop).
Symbol: 1 → FE

Skip on Carrier Clag (SCF)
Octal Code: 6714
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: The next instruction is skipped when the carrier flag is set to a 1.
Symbol: Carrier flag = 1, PC + 1 → PC

Read Carrier Scanner (RCS)
Octal Code: 6714
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Contents of the Carrier Scanner (CS) are ORed into AC 5-11 and Carrier status is ORed into AC0.
Symbol: CS V AC 5-11, CST V AC0

Disable Dataphone (DDF)
Octal Code: 6712/6472
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: When DC08-FX option (normally supplied) is in the system, this instruction disables interrupts from DC08-FX, disables the carrier scanner, and increments the scanner (octal code = 6472). When DC08-FX option is not supplied, the dataphone features are disabled (reset F-enable flip-flop) (octal code — 6712).
Symbol: None

Clear Terminal Ready Flip-Flop (CTR)
Octal Code: 6721
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clear the terminal ready flip-flop of the selected line (modem).
Symbol: 0 → TR

Clear Request (CRS)
Octal Code: 6722
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clears the request-to-send (RTS) flip-flop of the selected line (modem).
Symbol: 0 → RTS

Clear Carrier Flag (CCF)
Octal Code: 6724

Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clear carrier flag of the line selected by AC and restart the carrier scanner.
Symbol: 0 → CF, restart Carrier Scanner

Set Terminal Ready (STR)

Octal Code: 6731
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Set terminal ready (TR) of selected line (modem).
Symbol: 1 → TR

Set Request (SRS)

Octal Code: 6732
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Set request-to-send (RTS) of the selected line (modem).
Symbol: 1 → RTS

Clear Ring Flag (CRF)

Octal Code: 6734
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clear the ring flag and restart the ring scanner.
Symbol: 0 → RF, restart Ring Scanner

DC08H and J Instructions

Clear Digit Flag (CDF)

Octal Code: 6741
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clears the digit flag flip-flop.
Symbol: 0 → DF

Skip on Status Flag (SSF)

Octal Code: 6751
Event Time: 1
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: The status flag is set if:

1. Request to call (RTC) and modem power is OFF or data line is occupied.
2. Call request (CRO) is ON and modem power fails (power turned off or cable disconnected).
3. If data set status turn ON.
4. If abandon call and retry turns ON.

Symbol: 1 → SF

Clear Call Request (CCR)
Octal Code: 6752
Event Time: 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clears the call request (CR) flip-flop.
Symbol: 0 → CR

Skip on Digit Flag (SDF)
Octal Code: 6753
Event Time: 1, 2
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: The next instruction is skipped if the digit flag is set (PND — Present Next Digit — is turned ON).
Symbol: If DF = 1, PC + 1 → PC

Load A Unt (LAU)
Octal Code: 6754
Event Time: 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: The unit (or line) number of the desired ACU is specified by loading AC8-11 with the desired number and issuing a LAU. The unit number of the desired ACU must be set before checking status or issuing control instructions.
Symbol: Number (N) → AC 8-11

Clear Status and Read (CSR)
Octal Code: 6755
Event Time: 1, 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: Clears the status flag, if set, and transfers the status of PWI, DLO, PND, ACR, and DSS to the AC. (NOTE 1: If the EIA converter does not exist for the selected line, all 0s will be transferred to the AC. NOTE 2: The CSR instruction transfers status to the AC regardless of the state of the status flag.)

The status flag is set if:

1. If PWI turns OFF while dialing.
2. DSS turns ON.
3. ACR turns ON.
4. Request to transmit instruction (RTC) is ANDed with PWI OFF or DLO ON.

Symbol: 0 → SF, PNI → AC
DLO → AC
PND → AC
ACR → AC
DSS → AC

Request to Transmit (RTC)
Octal Code: 6756
Event Time: 2, 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs

Operation: Sets the call request (CR) to the selected line if the data line is not occupied (DLO OFF) and the 801 Calling Unit has power (PWI) ON. If the RTC instruction is issued, and DLO is ON or PWI is OFF, call request will not be set and a status interrupt occurs.
Symbol: If DLO = OFF and PNI = ON, CR → selected line

Load A Digit (LAD)
Octal Code: 6757
Event Time: 1, 2, 3
Indicators: Iot, Fetch, Pause
Execution Time: 4.25 μs
Operation: The digit to be presented to the ACU is specified by loading the AC8-11 bits with the desired digit and issuing the LAD instruction. The LAD instruction also sets dial request if the call request is set.
Symbol: Desired digit (D) → AC 8-11

SECTION 7-11 LINE PRINTER OPTION

TYPE 645 AUTOMATIC LINE PRINTER AND CONTROL

The line printer can print 132 characters per/line at a rate of 300 lines/minute. Each character is selected from the set of 64 available, by a 6-bit binary code (Appendix C lists the ASCII character specified for each code). Each 6-bit code is loaded separately into a core storage printing buffer (LPB) from AC_{6-11}. To load the LPB requires 132 load instructions. A print command causes the characters specified by the LPB to be printed on one line. After the last character in a line is printed, the LPB is cleared automatically.

The line printer can load characters into the LPB at a 10 μs rate, clear the LPB in 3 to 6 ms, and move paper at the rate of one line every 18 ms. When transfer of one code into the LPB is completed, the line printer done flag raises to indicate that the printer is ready to receive another code. When printing of the last character of the LPB is completed, the line printer done flag raises and causes a program interrupt to request reloading of the LPB. A line printer error flag raises and causes a program interrupt if the line printer detects an inoperative condition (printer power off, control circuits not reset, paper supply low, etc.) when the printer is cleared, the buffer loaded, or a print command is given.

A 3-bit format register (FR) in the printer is loaded from bits AC_{9-11} during a print command. This register selects one of eight channels of a perforated tape in the printer to control spacing of the paper. The tape moves in synchronism with the paper until a hole is sensed in the selected channel to halt paper advance. A recommended tape has the following characteristics:

FR CODE (Octal)	Paper Spacing	Tape Track
0	1 line	2
1	2 lines	3
2	3 lines	4
3	6 lines	5
4	11 lines (1/6 page)	6
5	22 lines (1/3 page)	7
6	33 lines (1/2 page)	8
7	top of form	1

The IOT instructions which command the line printer are:

Skip on Line Printer Error (LSE)
Octal Code: 6651
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the line printer error flag is sensed and, if it contains a binary 1 (indicating that an error has been detected), the content of the PC is incremented by one so that the next sequential instruction is skipped.
Symbol: If Line Printer Error Flag = 1, then PC + 1 → PC

Clear Printer Buffer (LCB)
Octal Code: 6662
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The line printer buffer is cleared.
Symbol: 0 → LPB

Load Printer Buffer (LLB)
Octal Code: 6654
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The next location in the printer buffer is loaded from the content of AC6-11, then the AC is cleared. This instruction clears the line printer done flag which will then be set to 1 upon completion of the transfer (within 10 μs).
Symbol: AC6-11 → LPB, then 0 → AC

Skip on Line Printer Done Flag (LSD)
Octal Code: 6661
Event Time: 1
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The content of the line printer done flag is sensed and, if it contains a binary 1, the content of the PC is incremented by one so that the next sequential instruction is skipped.
Symbol: If Line Printer Done Flag = 1, then PC + → PC

Clear Line Printer Flags (LCF)
Octal Code: 6652
Event Time: 2
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The line printer done and error flags are cleared.
Symbol: 0 → Line Printer Done Flag
0 → Line Printer Error Flag

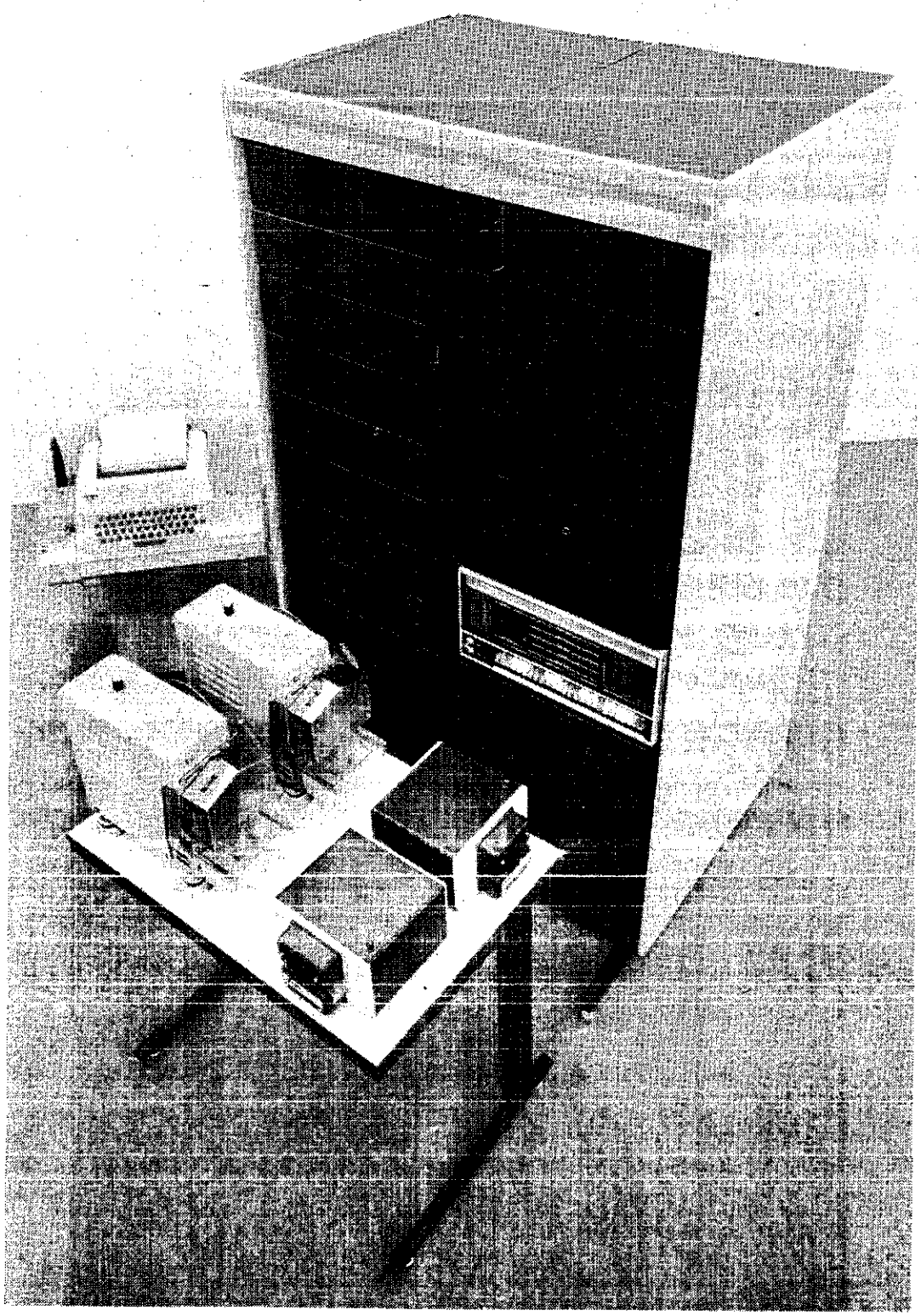
Print Line Print (LPR)
Octal Code: 6664
Event Time: 3
Indicators: Iot, Fetch, Pause [IR = 6, F]
Execution Time: 4.25 μs
Operation: The line printer format register (FR) is cleared, then loaded from the content of AC9-11 and the AC is cleared. The line contained in the printer

buffer (LPB) is printed. Paper is advanced in accordance with the selected channel of the format tape, if the content of AC8 is a 1. If AC8 is a 0, paper advance is inhibited. After printing, the line printer buffer is left in a cleared state.

Symbol: 0 → FR
AC9-11 → FR
0 → AC
The content of half of the LPB is printed. If AC8 = 1, then advance paper according to format tape channel (FR).

The following routine demonstrates the use of these commands in a sequence which prints an unspecified number of 132-character lines. This sequence assumes that the printer is not in operation, that the paper is manually positioned for the first line of print, and that 1-character words are stored in sequential core memory locations beginning at 2000. The PRINT location starts the routine.

```
PRINT,    LCB          /INITIALIZE PRINTER BUFFER
          CLA
          TAD LOC      /LOAD INITIAL CHARACTER ADDRESS
          DCA 10       /STORE IN AUTO-INDEX REGISTER
LRPT,     TAD CNT      /INITIALIZE CHARACTER COUNTER
          DCA TEMP
LOOP,     LSD          /WAIT UNTIL PRINTING BUFFER READY
          JMP LOOP
          LCF          /CLEAR LINE PRINTER FLAG
          TAD I 10     /LOAD AC FROM CURRENT CHARACTER
                       /ADDRESS
          LLB          /LOAD PRINTING BUFFER
          ISZ TEMP     /TEST FOR 132 CHARACTERS LOADED
          JMP LOOP
          TAD FRM      /LOAD SPACING CONTROL AND
          LPR          /PRINT A LINE
          JMP LRPT     /JUMP TO PRINT ANOTHER LINE
LOC,      1777         /INITIAL CHARACTER ADDRESS -1
CNT,      -204         /CHARACTER COUNTER — 132 DECIMAL
TEMP,     0            /CURRENT CHARACTER ADDRESS
FRM,      10           /SPACING CONTROL AND FORMAT
```



TYPESET-8, the PDP-8 computerized typesetting system, is being used by newspapers and commercial typesetters all over the world. The system takes unjustified and unhyphenated paper tape and converts it to justified, hyphenated tape for use with hot metal and photocomposition machines.

CHAPTER 8
INPUT/OUTPUT FACILITIES

All notations in brackets [ ] indicate data for the PDP-8/L computer only.

Since the processing power of a computer depends upon the range and number of peripheral devices that can be connected to it, the PDP-8/I [PDP-8/L] has been designed to interface readily with a broad variety of external equipment. The following chapters of this handbook define the interface characteristics of the computer to allow the user to design and implement any electrical interfaces required to connect devices to the PDP-8/I [PDP-8/L]. Chapters 9 and 10 provide functional descriptions of the logic circuit elements involved in programmed data transfers and data break transfers, respectively. Chapter 11 gives detailed information on digital logic circuits used for computer interfacing. Chapter 12 describes the design techniques, and connections of interface equipment. Chapter 13 presents the information for planning the installation of basic PDP-8/I [PDP-8/L] facilities.

The simple I/O technique of the PDP-8/I, [PDP-8/L], the availability of DEC's FLIP CHIP logic circuit modules, and DEC's policy of giving assistance wherever possible allow inexpensive, straightforward device interfaces to be realized. Should questions arise relative to the computer interface characteristics, the design of device interfaces using DEC modules, or installation planning, customers are invited to telephone the main plant in Maynard, Massachusetts, or any of the sales offices. Digital Equipment Corporation makes no representation that the interconnection of its circuit modules in the manner described herein will not infringe on existing or future patent rights. Nor do the descriptions contained herein imply the granting of license to use, manufacture, or sell equipment constructed in accordance therewith.

The basic PDP-8/I [PDP-8/L] contains a processor and core memory composed of Digital's M Series TTL circuit modules. These circuits have an operating temperature exceeding the limits of 32°F to 130°F (0°C to 40°C), so no air conditioning is required at the computer site. Standard 115V, 60-Hz power operates an internal solid-state power supply that produces all required voltages and currents. The high-capacity, high-speed I/O capabilities of the PDP-8/I [PDP-8/L] allow it to operate a variety of peripheral devices in addition to the standard teletype keyboard/printer, tape reader, and tape punch. DEC options, consisting of an interface and normal data processing equipment, are available for connecting into the computer system. These options include a random-access disk file, card equipment, line printers, magnetic tape transports, magnetic drums, analog-to-digital converters, CRT displays, and digital plotters, and data communication systems.

The PDP-8/I [PDP-8/L] system can also accept other types of instruments or hardware devices that have an appropriate interface. Up to 61 devices requiring one device code each can be connected to the computer. One device using the data break facility can be connected directly to the PDP-8/I [PDP-8/L] or up to seven such devices can be connected through a Type DM01 or DM04. Data Multiplex interfacing any device to the computer requires no modifications to the processor and can be achieved in the field.

Control is necessary to determine when an information exchange is to take place between the PDP-8/I [PDP-8/L] and peripheral equipment and to indicate the location(s) in the computer memory which will transfer the data. Either the computer program or the device external to the computer can exercise this control. Transfers controlled by the computer, hence under control of its stored program, are called programmed data transfers. Transfers made at times controlled by the external devices through the data break facility are called data break transfers.

PROGRAMMED DATA TRANSFERS

The majority of I/O transfers occur under control of the computer program. To transfer and store information under program control requires about six times as much computer time as under data break control. In terms of real time, the duration of a programmed transfer is rather small, due to the high speed of the computer, and is well beyond that required for laboratory or process control instrumentation.

To realize full benefit of the built-in control features of the PDP-8/I [PDP-8/L], programmed I/O transfers should be used in most cases. Controls for devices using programmed data transfers are usually simpler and less expensive than controls for devices using data break transfers. Using programmed data transfer facilities, simultaneous operation of devices is limited only by the relative speed of the computer with respect to the device speeds, and the search time required to determine the device requiring service. Analog-to-digital converters, digital-to-analog converters, digital plotters, line printers, message switching equipment, and relay control systems typify equipment using only programmed data transfers.

DATA BREAK TRANSFERS

Devices which operate at very high speeds or which require very rapid responses from the computer use the data break facilities. Use of these facilities permits an external device, almost arbitrarily, to transfer words from the computer core memory, bypassing all program control logic. Because the computer program has no cognizance of transfers made in this manner, programmed checks of input data are made prior to use of information received in this manner. The data break is highly effective for devices that transfer large amounts of data in block form; e.g., random-access disk file, high-speed magnetic tape systems, high-speed drum memories, or CRT display systems containing memory elements.

CHAPTER 9
PROGRAMMED DATA TRANSFERS

All notations in brackets [ ] indicate data for the PDP-8/L computer only.

The PDP-8/I and PDP-8/L bus structures are nearly identical. As has already been discussed under the KA8/I option, the PDP-8/I is available with either a negative I/O bus (standard PDP-8/I) or a positive I/O bus (PDP-8/I with KA8/I). The bus for the latter configuration (PDP-8/I with KA8/I) is identical to the standard positive bus of the PDP-8/L. [The PDP-8/L is not available with a negative bus.] Four extra signals on the PDP-8/I positive bus connect to options unavailable with the PDP-8/L.

In the descriptions which follow, all information pertains to both positive and negative busses. Where differences occur, they will be appropriately noted.

The majority of I/O transfers take place under program control, taking advantage of control elements built into the computer. Although programmed transfers take more computer and actual time than do data break transfers, the timing discrepancy is insignificant, considering the high speed of the computer with respect to the speed of most peripheral devices. The maximum data transfers rate for programmed operations of 12-bit words is 138 kHz [134 kHz] when no status checking, end transfer check, etc. is done. This speed is well beyond the normal rate required for typical laboratory or process control instrumentation.

The PDP-8/I [PDP-8/L] is a parallel-transfer machine that distributes and collects data in bytes of up to 12 bits. All programmed data transfers take place through the accumulator, the 12-bit arithmetic register of the computer. The computer program controls the loading of information into the AC for an output transfer, and the storing of information in core memory from the AC for an input transfer. Output information in the AC is power-amplified and supplied to the interface connectors for bussed connection to many peripheral devices. Then the program-selected device can sample these signal lines to strobe AC information into a control or information register. Input data pulses from I/O devices are received at the AC via the interface connectors. These pulses are produced by gating circuits within the program-selected device. Command signals, generated by the device, flow to the I/O skip facility (IOS) to sample the condition of I/O device flags. The IOS allows branching of the program based on the condition or availability of peripheral equipment, effectively making programmed decisions to continue the current program or jump to another part of the program (such as a subroutine) that services an I/O device.

The bussed system of I/O data transfers imposes the following requirements on peripheral equipment:

a. Each device samples the select code generated by the computer during IOT instructions and, when selected, is capable of producing sequential IOT command pulses in accordance with computer-generated IOP pulses. Circuits which perform these functions in the peripheral device are called the device selector (DS).

b. Each device, receiving output data from the computer, contains gating circuits at the input of a receiving register which are capable of strobing the AC signal information into the register when triggered by a common

pulse from the DS. Gating is also recommended at the input to the peripheral device to minimize loading on the BAC signal lines. (Positive bus only.)

c. Each device supplying input data to the computer contains gating circuits at the output of the transmitting register which are capable of sampling the information in the output register and supplying a pulse to the computer input bus when triggered by a command pulse from the DS.

d. Each device should contain a busy/done flag (flip-flop) and gating circuits which can pulse the computer I/O skip bus upon command from the DS. When the flag is set in the binary 1 state, the device is ready to transfer another byte of information.

Figure 9-1 illustrates the information flow within the computer which effects a programmed data transfer with I/O equipment. All instructions stored in core memory as a program sequence are read into the memory buffer register (MB) for execution. The transfer of the operation code in the three most-significant bits (bits 0, 1, and 2) of the instruction into the instruction register (IR) takes place, and it is decoded to produce the appropriate control signals. The computer, upon recognition of the operation code as an IOT instruction, enters a 4.25 μs expanded-computer-cycle and enables the IOP generator to produce time-sequenced IOP pulses as determined by the three least-significant bits of the instruction (bits 9, 10, and 11) in the MBO. These IOP pulses and the buffered output of the select code from bits 3 through 8 of the instruction word in the MB are bussed to device selectors in all peripheral equipment. Figure 9-2 indicates the timing of the programmed data transfers and Figure 9-3 illustrates the decoding of the IOT instruction.

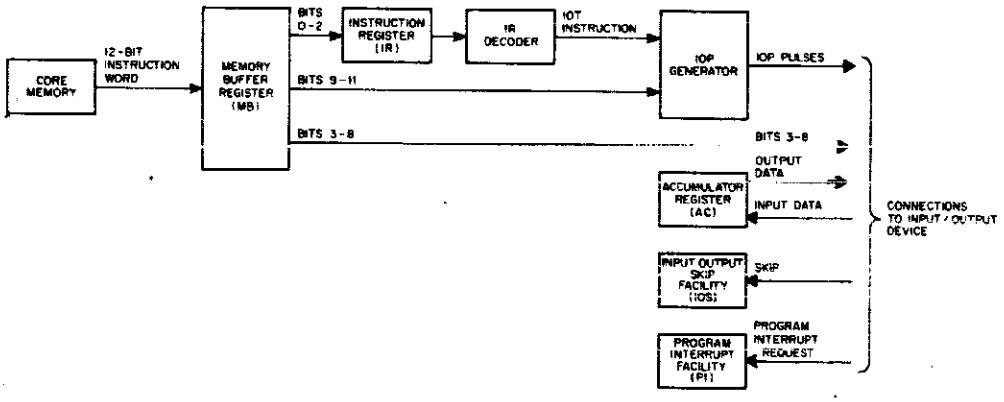


Figure 9-1. Programmed Data Transfer Interface Block Diagram

Devices which either (1) require immediate service from the computer program, (2) take an exorbitant amount of computer time to discontinue the main program until transfer needs are met, can activate the program interrupt (PI) facility. In this mode of operation, the computer can initiate operation of I/O equipment and continue the main program until the device requests servicing. A signal input to the PI requesting a program interrupt causes the conditions of the main program to be stored and initiates a subroutine to service the device. At the conclusion of this subroutine, the main program is reinstated until another interrupt request occurs.

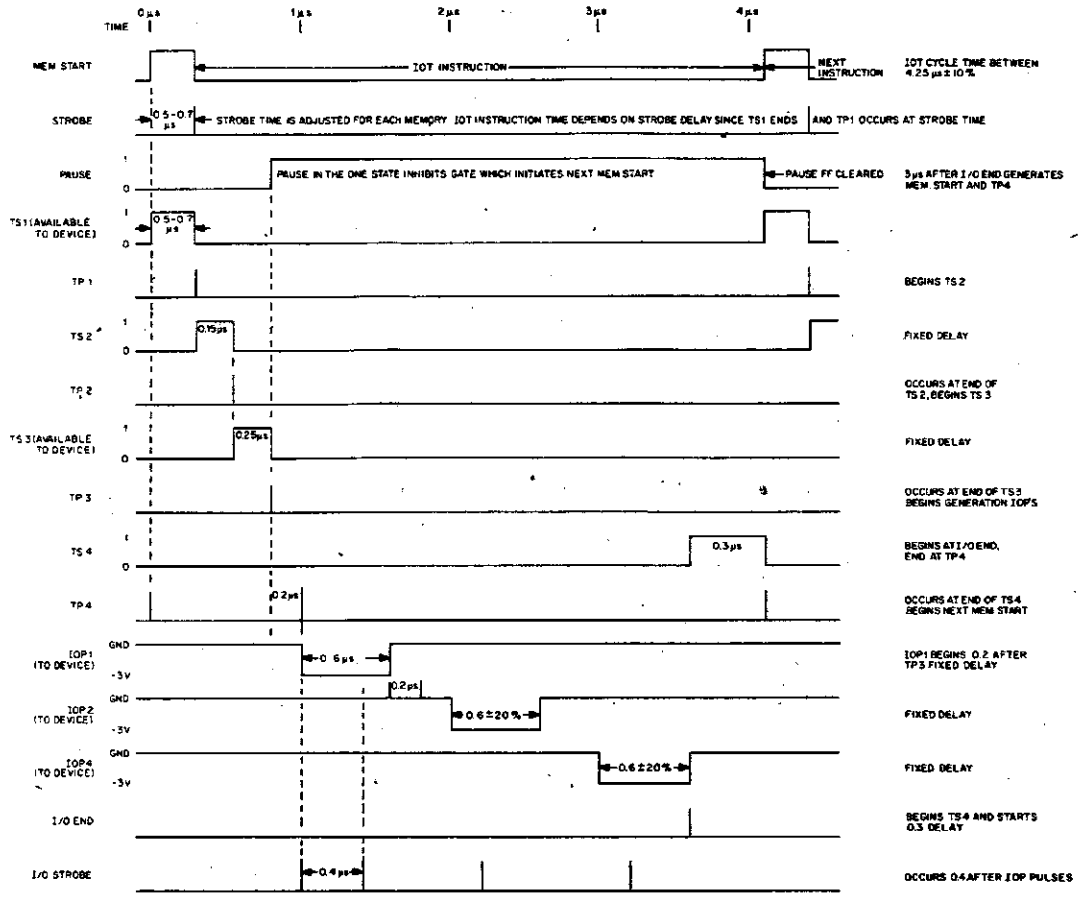


Figure 9-2. Programmed Data Transfer Timing Diagrams

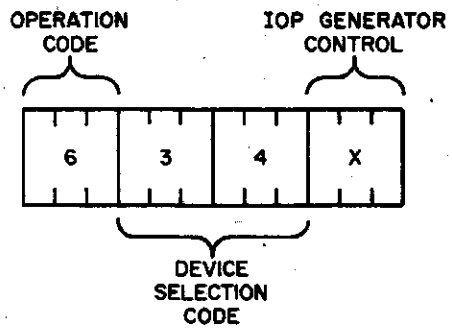
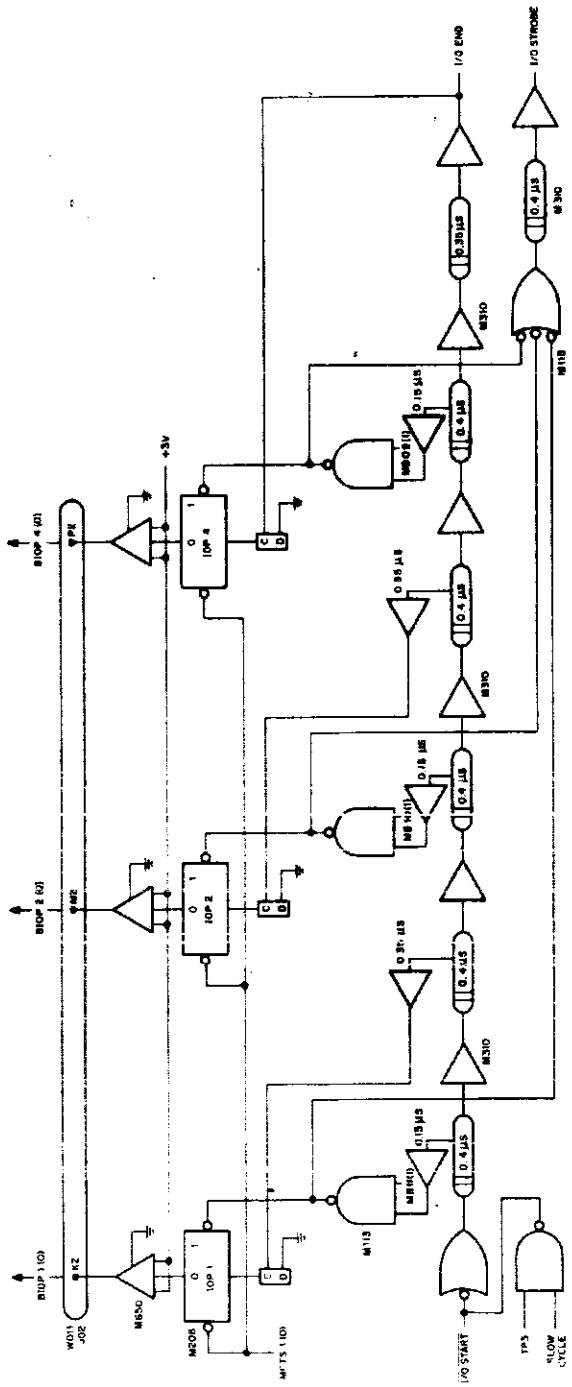


Figure 9-3 Typical IOT Instruction Decoding

TIMING AND IOP GENERATOR

When the IR decoder detects an operation code of 6_8, it identifies an IOT instruction and generates a slow cycle pulse. (IOT instructions which enable and disable the program interrupt facility, and those associated with the MC8/I Memory Extension Control and the DL8/I Data Line Interface inhibit generation of the slow cycle signal; the IOP generator is not enabled, so normal timing pulses and special device selectors execute these instructions. [On the PDP-8/L, there is no distinction between internal and external I/O functions, so all I/O functions cause the slow cycle.])

The slow cycle signal is AND with TP3 to generate $\overline{\text{I/O START}}$ and set the PAUSE flip-flop. The logic circuits of the IOP generator are shown in Figure 9-4 and consist of a three-stage delay chain. $\overline{\text{I/O START}}$ is inverted to drive a pulse amplifier, which may set the IOP1 flip-flop, and to begin the first stage of delay. After 0.2 μs, the same pulse which may set the IOP2 flip-flop clears the IOP1 flip-flop; similarly the IOP2 flip-flop is reset by IOP4. A final delay of 0.5 μs occurs before IOP4 is cleared and I/O END is generated to restart normal timing.



[NOTE: PDP-8/L logic is similar, and produces same results.]

Figure 9-4 IOP Generator Logic

The pulses which set the IOP1, IOP2, and IOP4 flip-flops are ANDed with MB11 (1), MB10 (1), and MB09 (1), respectively, to determine which flip-flops will be set and which pulses shall be generated for a particular instruction. For each pulse which is generated, an I/O STROBE pulse follows 0.3 μs later.

NOTE

All cycle times have a tolerance of 20% [12%].

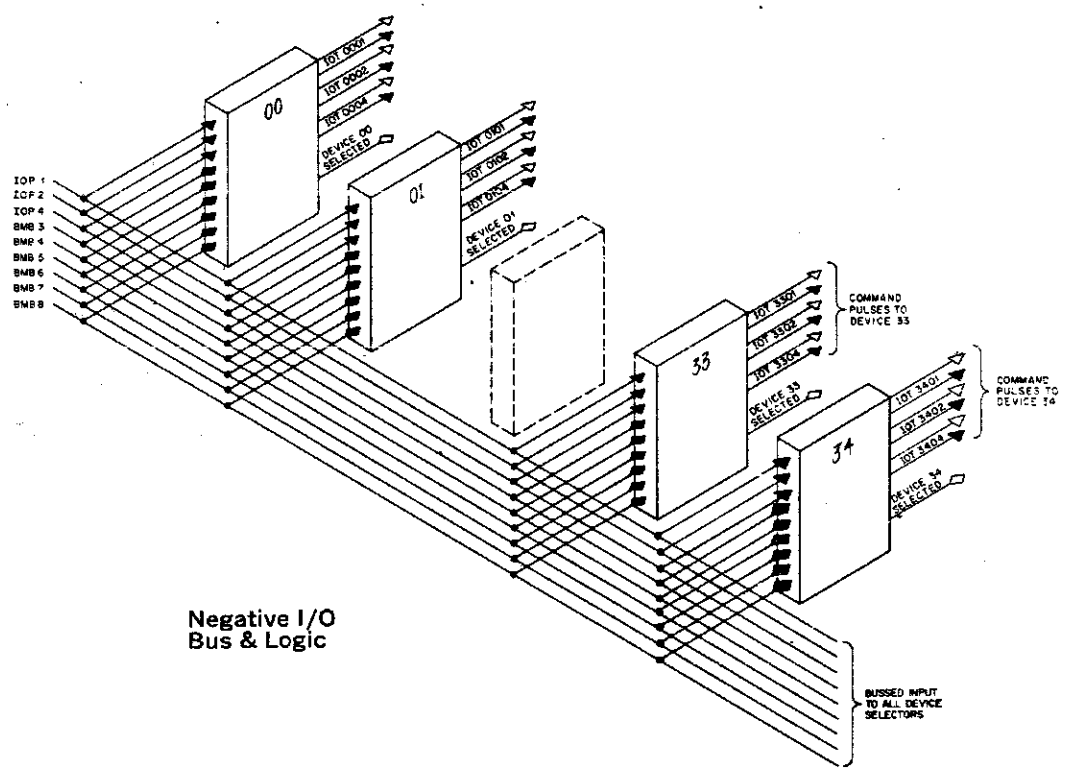
Instruction bit, IOP pulse, IOT pulse, and event time correspondence is as follows:

Instruction BIT	IOP Pulse	IOT Pulse	Event Time	Used Primarily For
11	IOP 1	IOT 1	1	Sampling Flags, Skipping.
10	IOP 2	IOT 2	2	Clearing Flags, Clearing AC.
9	IOP 4	IOT 4	3	Reading, Loading, and Clearing buffers.

DEVICE SELECTOR (DS)

Bits 3 through 8 of an IOT instruction serve as a device or subdevice select code. Processor bus drivers buffer both the 1 and 0 output signals of MB 3-8 and distribute them to the interface connectors for bussed connection to all device selectors. Each DS is assigned a select code and is enabled only when the assigned code is present in the MB. When enabled, a DS regenerates IOP pulses as IOT command pulses and transmits these pulses to skip, input, or output gates within the device and/or to the processor to clear the AC.

Each group of three command pulses requires a separate DS channel (W103 module for negative bus and M103 for positive bus) and each DS channel requires a different select code (or I/O device address). One I/O device can, therefore, use several DS channels. Note that the processor produces the pulses identified as IOP1, IOP2, and IOP4 and supplies them to all device selectors. The device selector produces pulses IOT1, IOT2, and IOT4 which initiate a transfer or effect some other control. Figure 9-5 shows generation of command pulses by several DS channels.



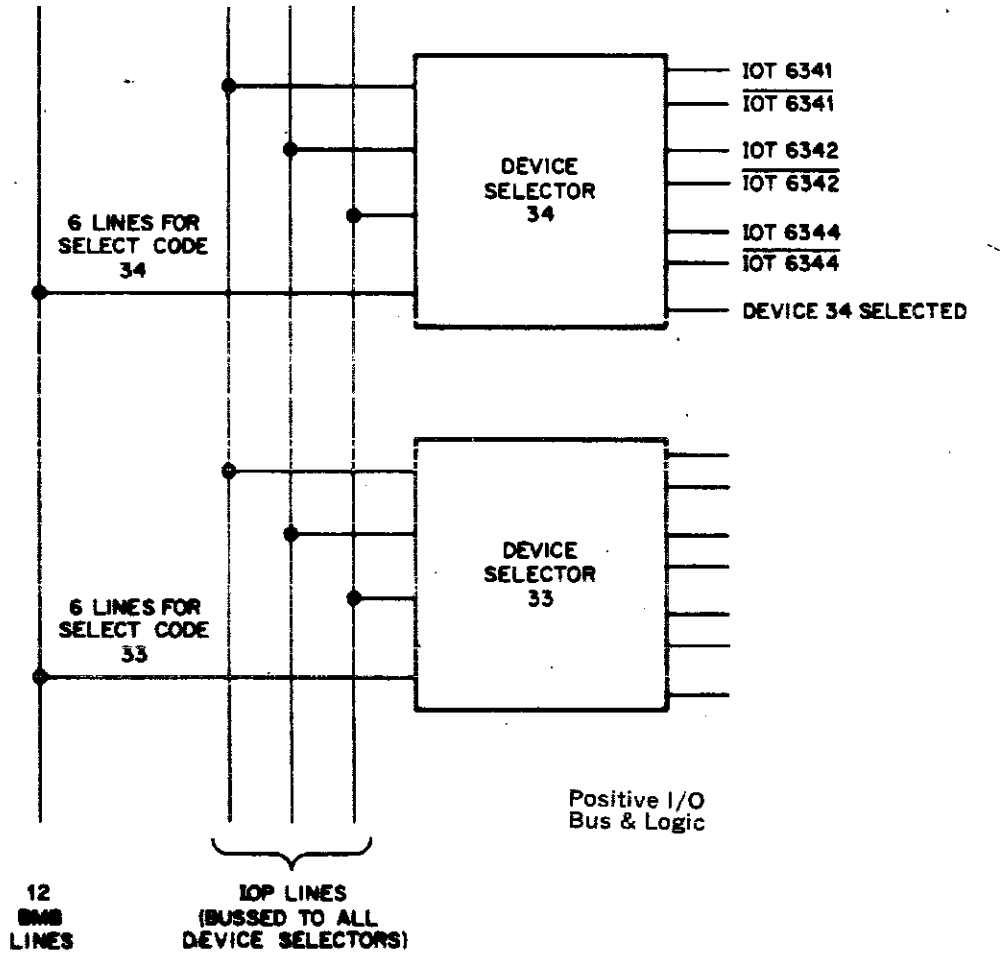


Figure 9-5. Generation of IOT Command Pulses by Device Selectors

The logical representation for a typical channel of the DS, using channel 34, is shown in Figure 9-6. A 6-input NAND gate, wired to receive the appropriate signal outputs from MB3-8 for select code 34, activates the channel. In the W103 module, the NAND gate (used on the negative bus) contains 14 diode

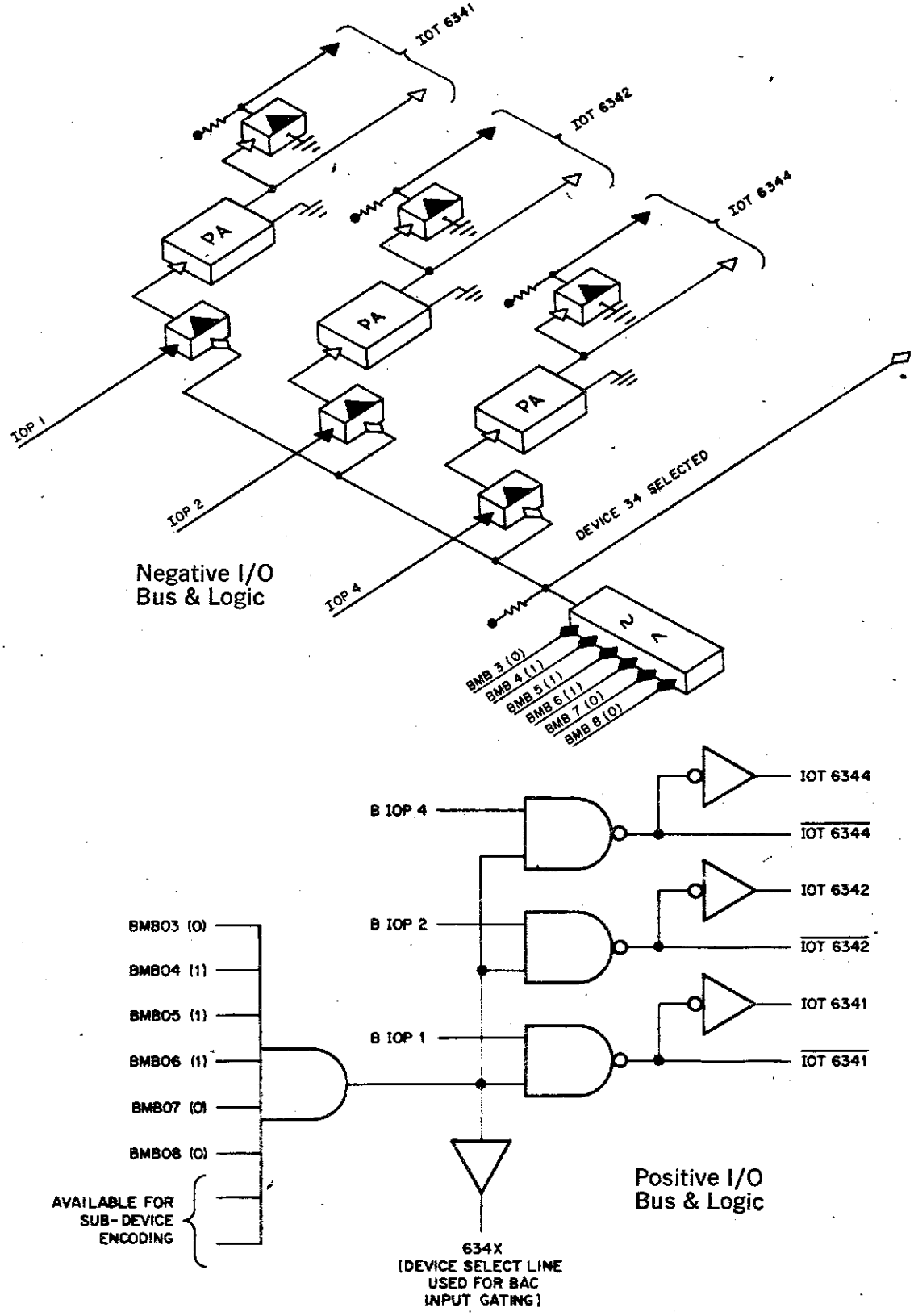


Figure 9-6. Typical Device Selector (Device 34)

input terminals; 12 of these connect to the complementary outputs of MB3-8 and 2 are open to receive subdevice or control-condition signals, as needed. Either the 1 or the 0 signal from each MB bit is disconnected by removing the appropriate diode from the NAND gate when establishing the select code. In the M103 module (used on the positive bus), the appropriate MB levels are wired to the 6 DS inputs. The ground-level output of the NAND gate indicates when the IOT instruction selects the device, and can enable circuit operations within the device. This output also enables three gating inverters, allowing them to trigger a pulse amplifier if an IOP pulse occurs. The output from each pulse amplifier is an IOT command pulse which is identified by the select code and the number of the initiating IOP pulse. Three inverters receive the IOT pulses to produce complementary IOT output pulses. A pulse amplifier module or bus device modules can be connected in each channel of the DS to provide greater output drive or to produce pulses of the specific duration required by the selected device.

INPUT/OUTPUT SKIP (IOS)

An IOT pulse can be generated and used to test the condition or status of a device flag and, based on the test results, to continue or to skip the next sequential instruction. This operation is performed by a 2-input AND gate in the device connected as shown in Figure 9-7. One input of the skip gate receives the status level (flag output signal), the second input receives an IOT pulse, and, when the skip conditions are fulfilled, the output drives the computer IOS bus to ground. When the IOS bus is driven to ground, the content of the program counter is incremented by 1 to advance the program count without executing the instruction at the current program count. In this manner,

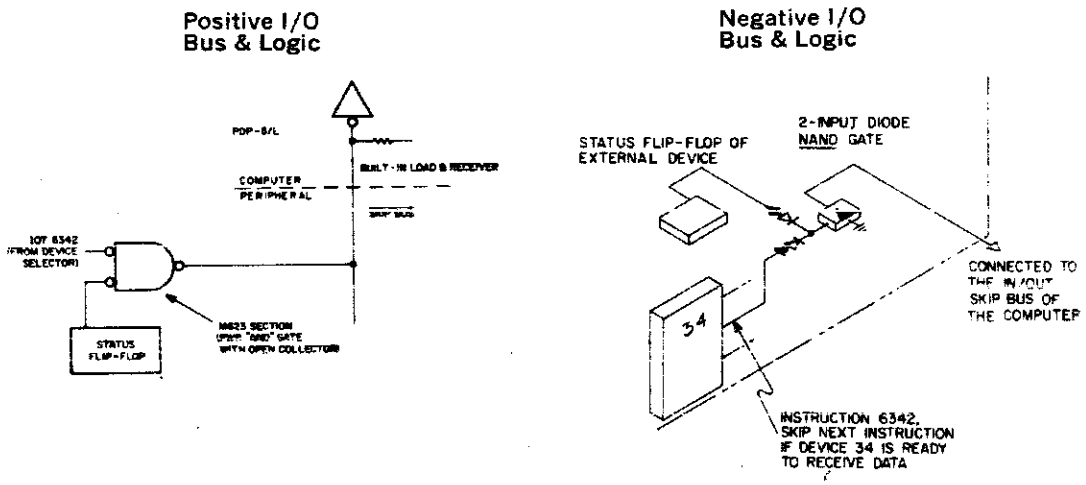


Figure 9-7. Use of IOS to Test the Status of an External Device

an IOT instruction can check the status of an I/O device flag and skip the next instruction if the device requires servicing. Programmed testing in this manner allows the routine to jump out of sequence to a subroutine that services the device tested.

Assuming that a device is already operating, a possible program sequence to test its availability is as follows:

Address	Instruction	Remarks
	.	
	.	
	.	
	.	
100,	6342	/SKIP IF DEVICE 34 IS READY
101,	5100	/JUMP .−1
102,	5XXX	/ENTER SERVICE ROUTINE FOR /DEVICE 34
	.	
	.	
	.	
	.	

When the program reaches address 100, it executes an instruction skip with 6342. The skip occurs only if device 34 is ready when the IOT 6342 command is given. If device 34 is not ready, the flag signal disqualifies the skip gate, and the skip pulse does not occur. Therefore, the program continues to the next instruction, which is a jump back to the skip instruction. In this example, the program stays in this waiting loop until the device is ready to transfer data, at which time the skip gate in the device is enabled and the skip pulse is sent to the computer IOS facility. When the skip occurs, the instruction in location 102 transfers program control to a subroutine to service device 34.

This subroutine can load the AC with data and transfer it to device 34, or can load the AC from a register in device 34 and store it in some known core memory address.

ACCUMULATOR

The binary 1 output signal of each AC flip-flop, buffered by a bus driver, is available at the interface connectors. These computer data output lines are bus-connected to all peripheral equipment receiving programmed data output information from the PDP-8/I [PDP-8/L]. A terminal leading to each AC flip-flop is connected to the interface connectors for bussing to all peripheral equipment supplying programmed data input to the computer. A pulse that drives the AC input bus terminal to ground is sampled by the computer and this causes loading of the corresponding AC flip-flop. Output and input connections to the accumulator appear in Figure 9-8.

Figure 9-8 illustrates the twelve bits of the accumulator and the link bit. The status of the link bit is not available for transfers with peripheral equipment (unless it is rotated into the AC). A bus driver continuously buffers the output signal from each AC flip-flop. These buffered accumulator (BAC) signals are available at the interface connectors.

INPUT DATA TRANSFERS

When it is ready to transfer data into the PDP-8/I [PDP-8/L] accumulator, the device sets a flag connected to the IOS. The program senses the ready status of the flag and issues an IOT instruction to read the content of the external device's buffer register into the AC. If the AC is not cleared before the transfer and, if the AC-clear signal is not asserted, the resultant word in the AC is the inclusive OR of the previous word in the AC and the word transferred from the device buffer register.

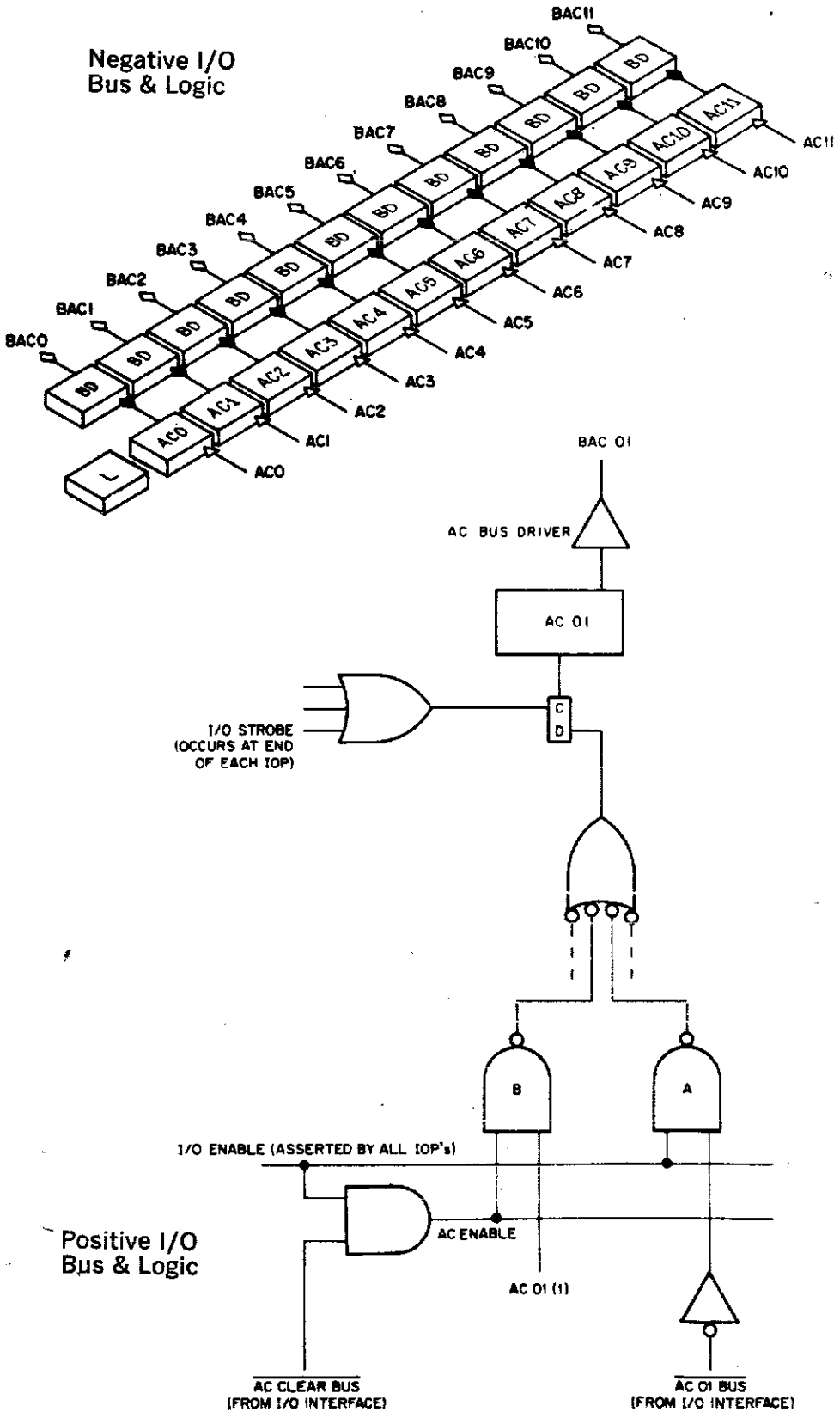


Figure 9-8. Accumulator Input and Output

Figure 9-9 shows that the accumulator has an input bus for each bit flip-flop. Setting a 1 into a particular bit of the accumulator necessitates grounding of

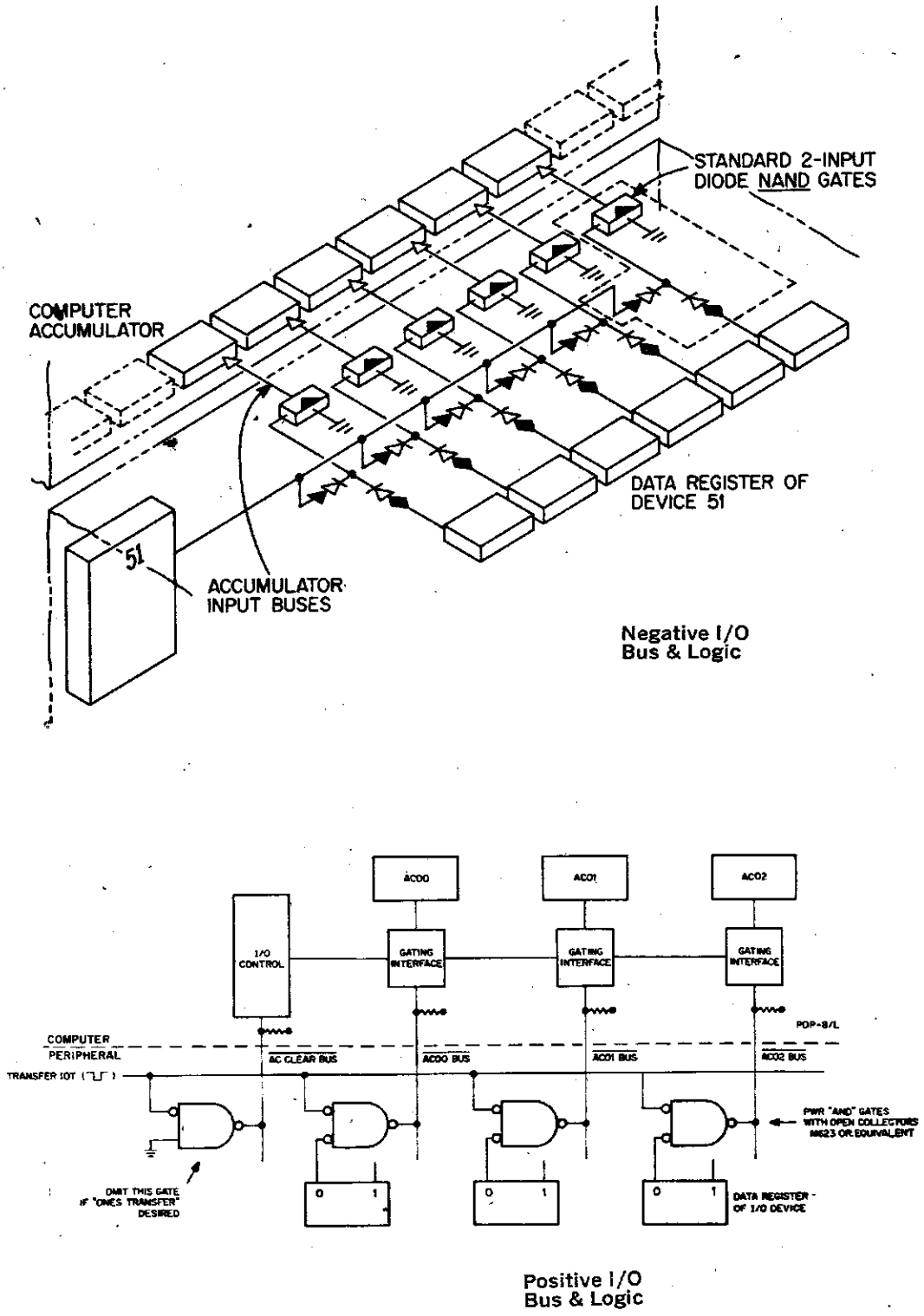


Figure 9-9. Loading Data into the Accumulator from an External Device

the interface input bus by a transistor with an open collector. Typical gates are provided in this section. In this illustration, the 2-input AND gates set various bits of the accumulator. In this case an IOT pulse is AND-combined with the flip-flop state of the external device to conditionally set 1s into the accumulator. (The program must include a clear-AC command prior to loading in this manner; otherwise an inclusive OR takes place between the previous content of the accumulator and the content of the register being read.)

Following the transfer (possibly in the same instruction), the program can issue a command pulse to initiate further operation of the device and/or clear the device flag.

AC CLEAR

A separate input line may be asserted (grounded) to cause clearing of the AC with an IOT. In the PDP-8/I and PDP-8/L (but not in their predecessors), this line may be asserted at the same time as the AC input bits to effect a transfer from the peripheral to the AC without the need for prior initialization of the AC. The user is cautioned against permanently grounding this line, since many IOTs would then cause unwanted clearing of the AC.

OUTPUT DATA TRANSFERS

The AC is loaded with a word (e.g., by a CLA TAD instruction sequence); then the IOT instruction is issued to transfer the word into the control or data register of the device by an IOT pulse (e.g., IOP 2), and operation of the device is initiated by the same or another IOT pulse (e.g., IOP 4). The data word transferred in this manner can be a character to be operated upon, or can be a control word sampled by a status register to establish a control mode.

Loading considerations require that, on the positive bus, the BAC interface lines be gated by the select code at each device to prevent excessive loading. A special module, the M101, is provided for this purpose. See Chapter 11 for more details.

Since the BAC interface bus lines continually represent the status of the AC flip-flops, the receiving device can strobe them to sense the value of the accumulator. In Figure 9-10 a strobe pulse samples six bits of the accumulator to conditionally set an external 6-bit data register. Since this is not a jam transfer, it is first necessary to clear the external data register before setting 1s into it. (The availability of the D-type flip-flops for use on the positive bus often makes prior clearing of devices on this bus unnecessary.) The read-in gates driving the external data register are part of the external device and are not supplied by the computer. The data register can contain any number of flip-flops up to a maximum of twelve. (If more than twelve flip-flops are involved, two or more transfers must take place.) The clear pulse and the strobe pulse shown in Figure 9-10 must occur when the data to be placed in the external data register is held in the accumulator. These pulses, therefore, must be under computer control to effect synchronization with the operation or program of the computer.

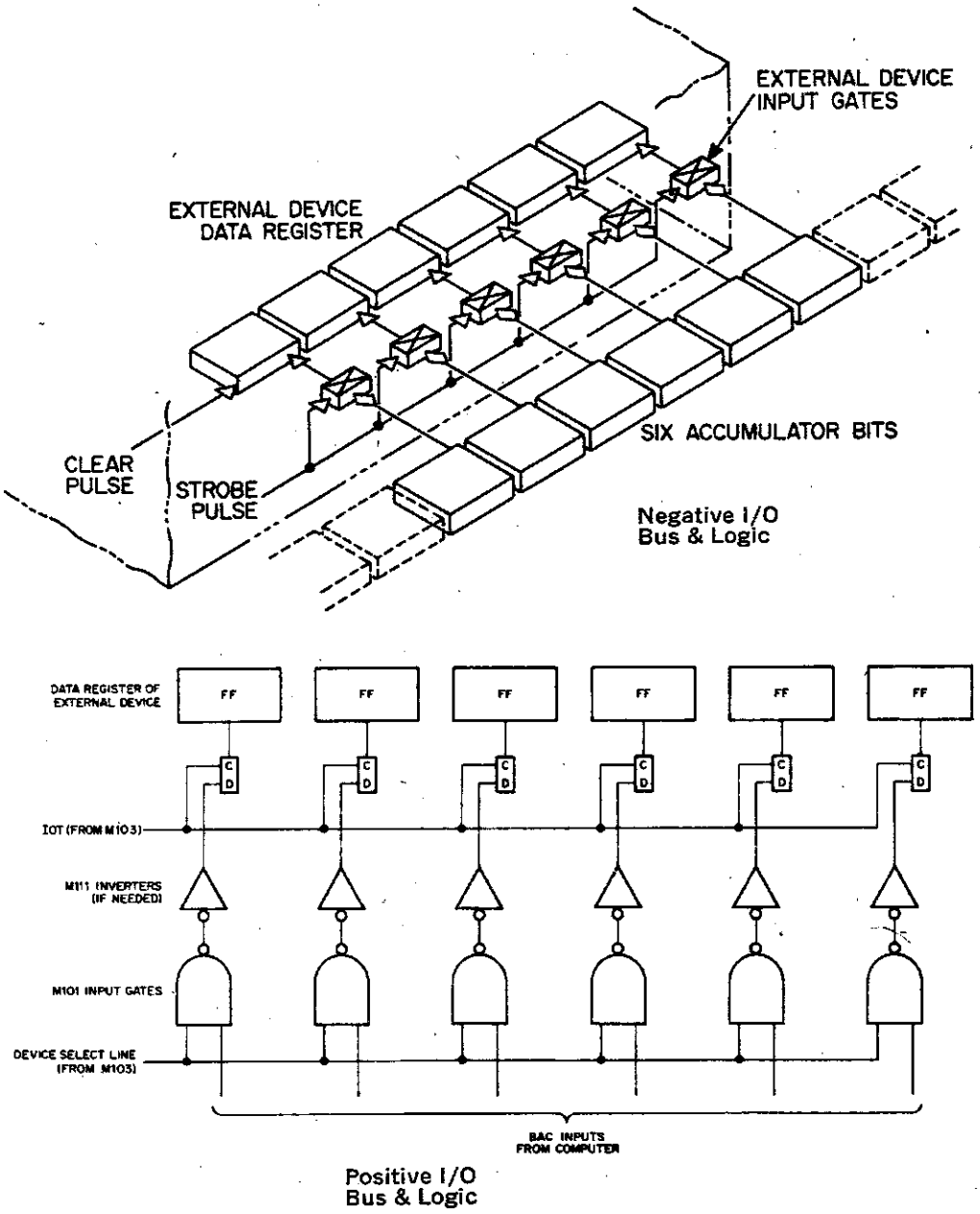


Figure 9-10. Loading a 6-Bit Word into an External Device from the Accumulator

Figure 9-11 illustrates the use of two of the pulses being gated by the device selector coded for "34." Pulse IOT1 clears the data register and IOT4 strobes the data from the accumulator into the data register. Note that the processor produces the IOP1, IOP2 and IOP4 pulses and supplies them to all device selectors. The program-selected DS produces IOT1, IOT2, and IOT4 pulses which initiate a transfer or effect some control. As indicated in Figure 9-11, this particular system adds two new microinstructions to the PDP-8/I repertoire. One generates a pulse to clear the data register of device number 34. The other microinstruction produces a pulse to load the data register of device number 34 with the content of the accumulator.

IOT cycle timing is shown in Figure 9-2. Note that the AC bus drivers are quiescent for more than 400 ns before the IOP1 pulse occurs. Since the FLIP CHIP DCD gates, often used on the negative bus, require a 400 ns set-up time, the IOP1 pulse can be used to load the content of the AC into an external buffer register having input DCD gates. It is more customary, however, to use IOP1 to reset the external buffer and to use IOP2 or IOP4 to load 1s into the buffer. Any IOP can be used with the IOS facility. Gating delay within the peripheral from IOPs to any input signals being returned to the computer, should be 100 μs or less.

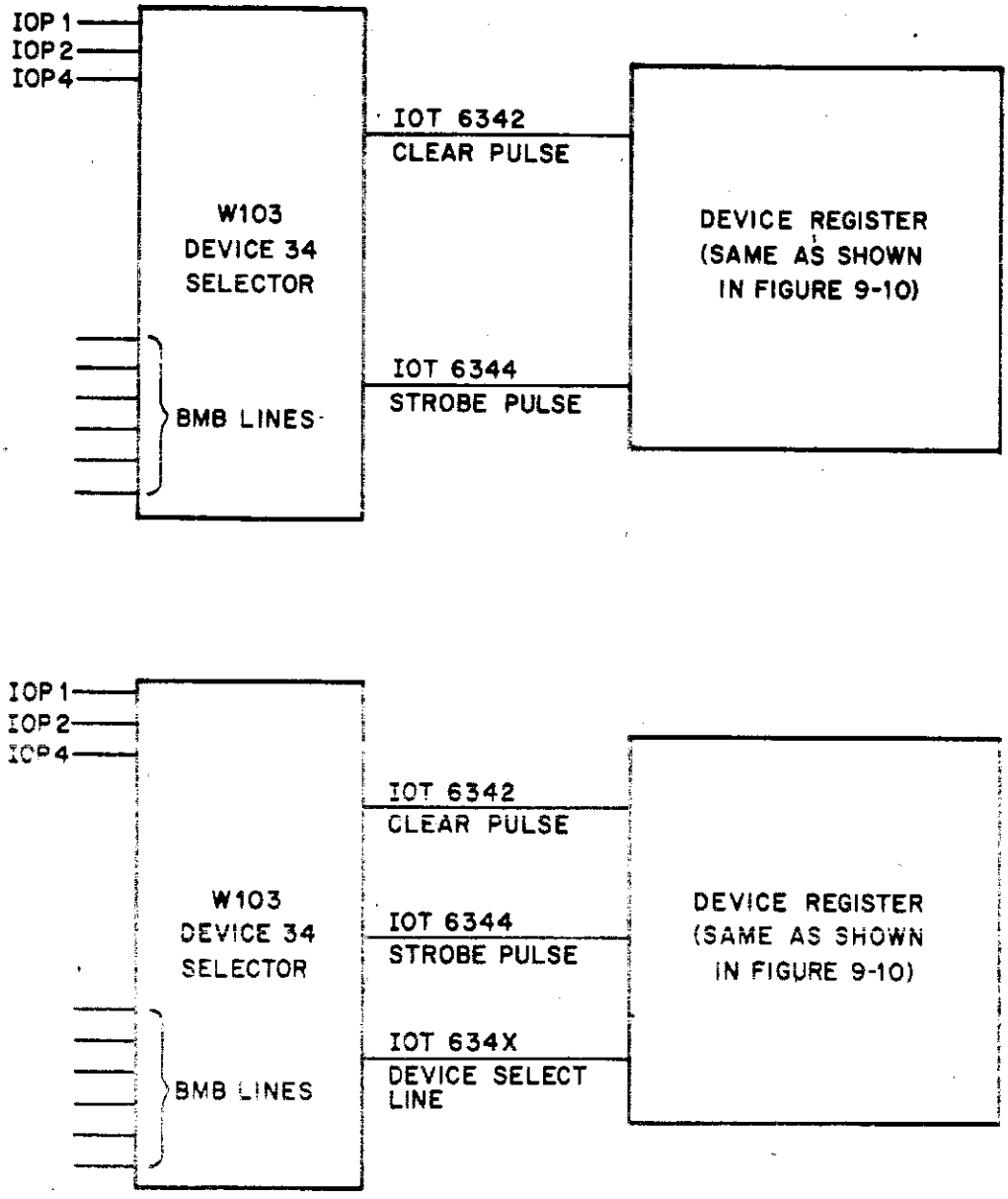


Figure 9-11. Use of a Device Selector for Activating and Controlling an External Device

PROGRAM INTERRUPT (PI)

When a large amount of computing is required, the program should activate an I/O device and then continue the main program, rather than wait for the device to become ready to transfer data. The program interrupt facility, when enabled by the program, relieves the main program of the need for repeated flag checks by allowing the ready status of the I/O device flags to automatically cause a program interrupt. When the program interrupt occurs, program control transfers to a subroutine that determines which device requested the interrupt and initiates an appropriate service routine.

In the example shown in Figure 9-12, a flag signal from a status flip-flop operates a gate with no collector load. When the status flip-flop indicates the need for device service, the gate requests a program interrupt by driving the program-interrupt-request bus to ground. If only one device is connected to the PI facility when an interrupt occurs, program control can be transferred directly to a routine that services the device. This operation occurs as follows:

```
Tag     Address     Instruction          Remarks
        1000            .          /MAIN PROGRAM
        1001            .          /MAIN PROGRAM CONTINUES
        1002            .          /INTERRUPT REQUEST OCCURS
                    INTERRUPT OCCURS
        0000            .          /PROGRAM COUNT (PC = 1003)
                                   /IS STORED IN 0000
        0001        JMP SR         /ENTER SERVICE ROUTINE
                    .
                    .
SR      2000                       /SERVICE SUBROUTINE FOR
                                   /INTERRUPTING DEVICE,
                                   /INCLUDING SEQUENCE TO SAVE
                                   /AND RESTORE AC, AND
                                   /L IF REQUIRED.
        3002        ION            /TURN ON INTERRUPT
        3003        JMP I 0000     /RETURN TO MAIN PROGRAM
                    .
                    .
                    .
        1003            .          /MAIN PROGRAM CONTINUES
        1004            .
```

In most PDP-8/I [PDP-8/L] systems, numerous devices are connected to the PI facility, so the routine beginning in core memory address 0001 must determine which device requested an interrupt. The interrupt routine determines the device requiring service by checking the flags of all equipment connected to the PI. The routine then transfers program control to a service routine for the first device encountered that has its flag in the state required to request a program interrupt. In other words, when program interrupt requests can originate in numerous devices, each device flag connected to the PI must also be connected to the IOS.

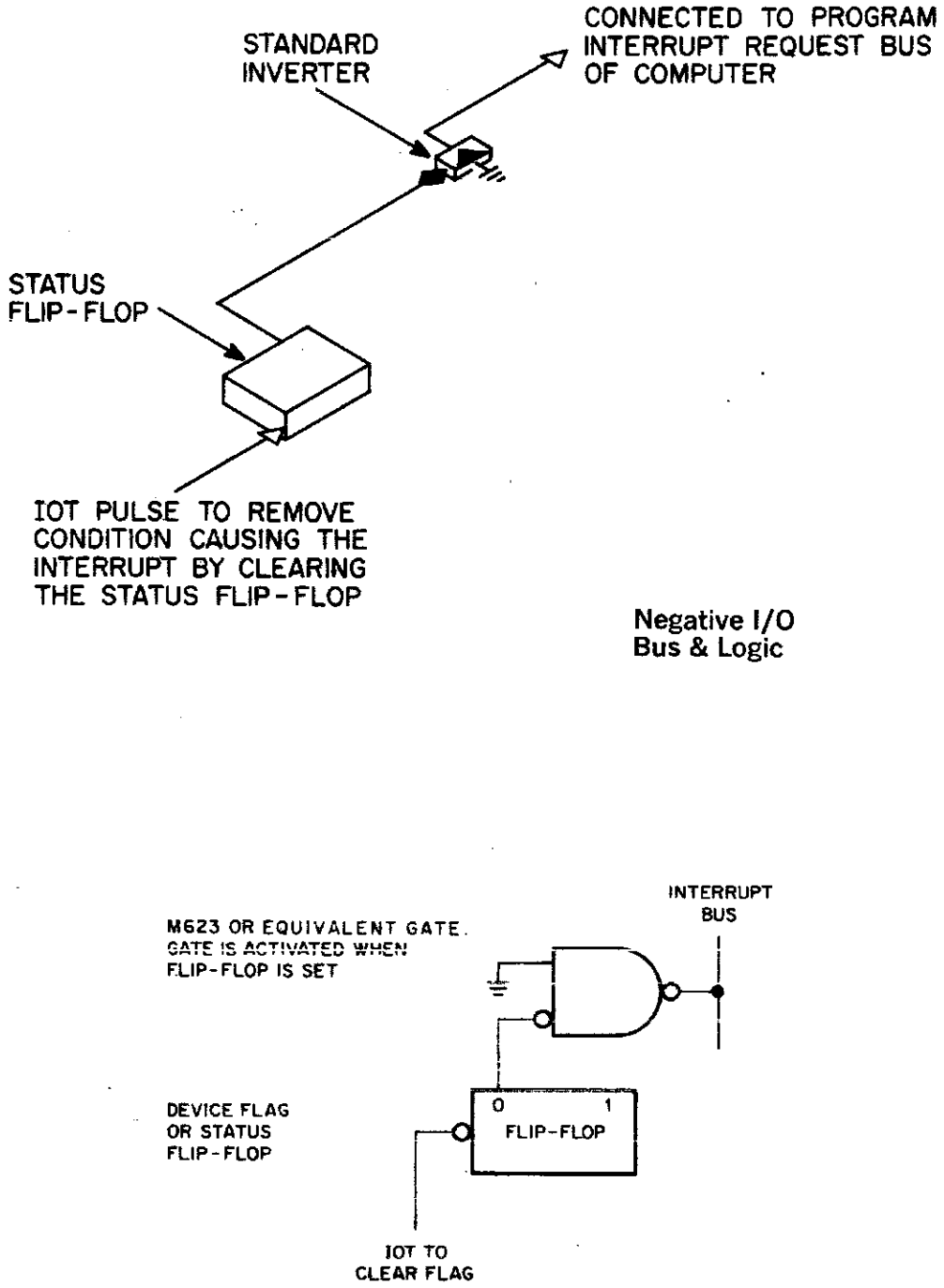


Figure 9-12. Program Interrupt Request Signal Origin

MULTIPLE USE OF IOS AND PI

In common practice, more than one device is connected to the PI facility. In the basic PDP-8/I and PDP-8/L, the teletype flags are already connected. Therefore, since the computer receives a request that is the inclusive OR of requests from all devices connected to the PI, the IOS must identify the device making the request. When a program interrupt occurs, a routine is entered from address 0001 to sequentially check the status of each flag connected to the PI and to transfer program control to an appropriate service routine for the device whose flag is requesting a program interrupt. Figure 9-13 shows IOS and PI connections for typical devices.

The following program example illustrates how the program interrupt routine determines the device requesting service:

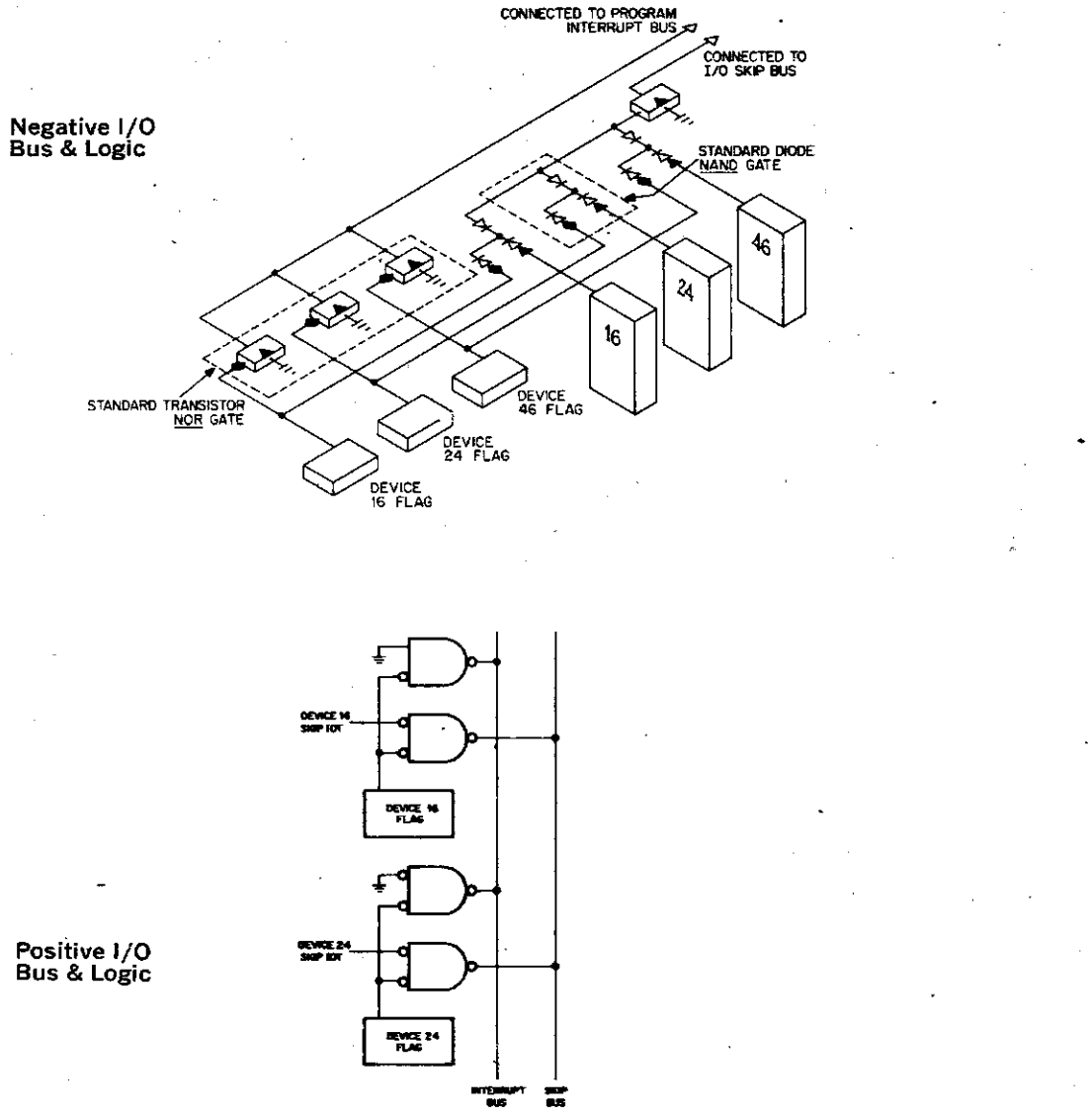


Figure 9-13. Multiple Inputs to IOS and PI Facilities

Tag	Address	Instruction	Remarks
	1000	.	/MAIN PROGRAM
	1001	.	/MAIN PROGRAM CONTINUES
	1002	.	/INTERRUPT REQUEST OCCURS
			INTERRUPT OCCURS
	0000		/STORE PC (PC = 1003)
	0001	JMP FLG CK	/ENTER ROUTINE TO DETERMINE
			/WHICH DEVICE CAUSED INTERRUPT
FLG CK		IOT 6341	/SKIP IF DEVICE 34 IS REQUESTING
		SKP	/NO — TEST NEXT DEVICE
		JMP SR34	/ENTER SERVICE ROUTINE 34
		IOT 6441	/SKIP IF DEVICE 44 IS REQUESTING
		SKP	/NO — TEST NEXT DEVICE
		JMP SR44	/ENTER SERVICE ROUTINE 44
		IOT 6541	/SKIP IF DEVICE 54 IS REQUESTING
		SKP	/NO — TEST NEXT DEVICE
		JMP SR54	/ENTER SERVICE ROUTINE 54
		.	
		.	

Assume that the device that caused the interrupt (e.g., device 44) is an input device (e.g., tape reader). The following example of a device service routine might apply.

Tag	Instruction	Remarks
SR44,	DAC TEMP	/SAVE AC
	IOT XX	/TRANSFER DATA FROM DEVICE
		/BUFFER TO AC
	DAC I 10	/STORE IN MEMORY LIST
	ISZ COUNT	/CHECK FOR END
	SKP	/NOT END
	JMP END	/END. JUMP TO ROUTINE TO HANDLE
		/END OF LIST CONDITIONS
	.	
	.	
	.	
		/RESTORE L AND EPC IF REQUIRED
	TAD TEMP	/RELOAD AC
	ION	/TURN ON INTERRUPT
	JMP I 0	/RETURN TO PROGRAM

The subroutine above assumes that locations 10 and COUNT are used solely by device 44, and that they have been properly initialized. If the device that caused the interrupt was essentially an output device (receiving data from computer), the IOT — then — DAC I 10 sequence might be replaced by a TAD I 10 — then — IOT sequence.

CHAPTER 10
DATA BREAK TRANSFERS

All notations in brackets [ ] indicate data for the PDP-8/L computer only. The data break facility allows I/O devices to transfer information directly with the PDP-8/I [PDP-8/L] core memory on a cycle-stealing basis. Up to seven devices can connect to the data break facility through the optional Type DM01 Data Multiplexer used with the negative bus. (The optional Type DM04 Data Multiplexer used with the positive bus can only connect (individually) up to three peripheral devices; however, up to three DM04s can be used simultaneously.) The data break is particularly well-suited for devices which transfer large amounts of information in block form.

Peripheral I/O equipment operating at high speeds can transfer information with the computer through the data break facility more efficiently than through programmed means. The maximum transfer rate of the data break facility is over 7.9 million bps [7.5 million bps for PDP-8/L]. Information flow to effect a data break transfer with an I/O device appears in Figure 10-1.

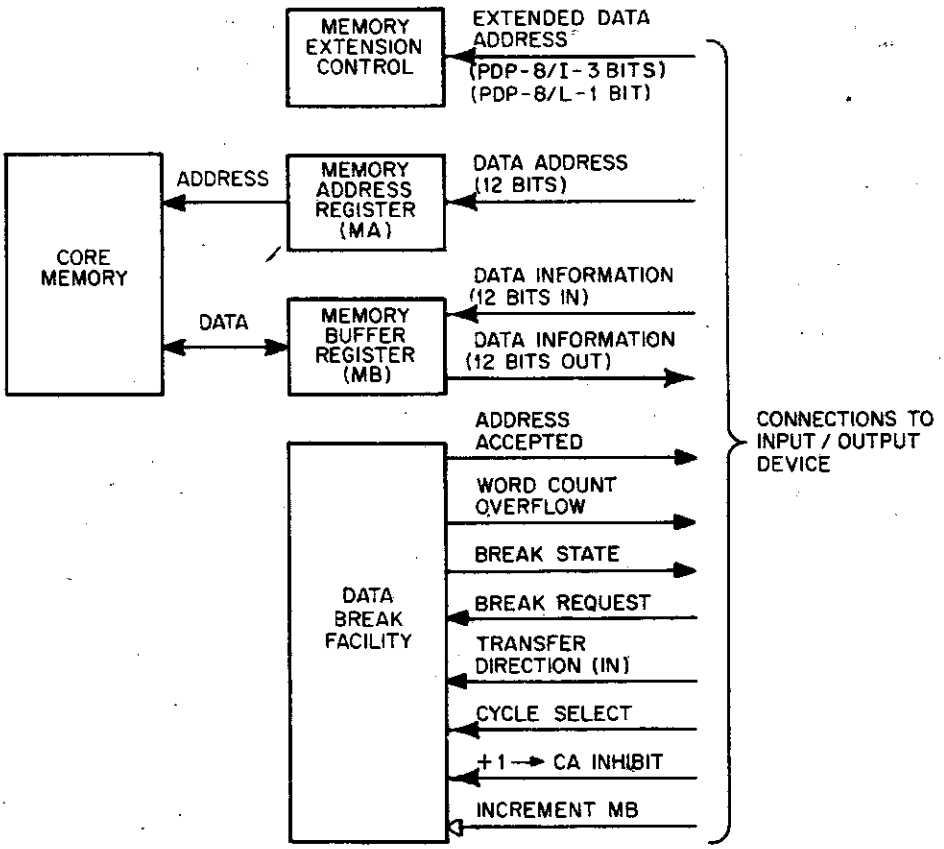


Figure 10-1. Data Break Transfer Interface Block Diagram

In contrast to programmed operations, the data break facilities permit an external device to control information transfers. Since the external device receives its initialization and control signals from the computer, programmed-data transfer, as well as data break, logic is usually required on such devices. Therefore, data-break device interfaces require more control logic circuits, causing a higher cost than programmed-transfer interfaces.

Data breaks are of two basic types: single-cycle and three-cycle. In a single-cycle data break, registers in the device (or device interface) specify the core memory address of each transfer and count the number of transfers to determine the end of data blocks. In the three-cycle data break, two computer core memory locations perform these functions, simplifying the device interface by omitting two hardware registers.

In general terms, to initiate a data break transfer of information, the interface control must do the following:

a. Specify the affected address in core memory.

b. Provide the data word by establishing the proper logic levels at the computer interface (assuming an input data transfer), or provide read-in gates and storage for the word (assuming an output data transfer).

c. Provide a logical signal to indicate direction of data word transfer.

d. Provide a logical signal to indicate single-cycle or three-cycle data break operation.

e. Request a data break by supplying the proper signal to the computer data break facility.

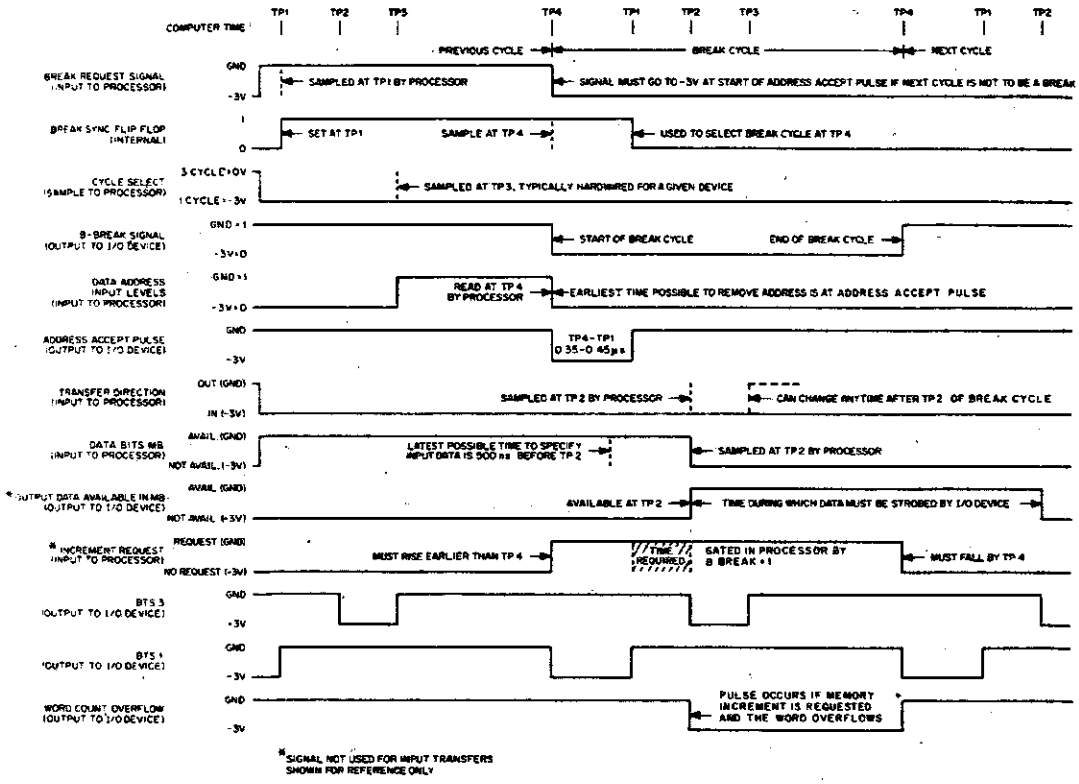
SINGLE-CYCLE DATA BREAKS

Single-cycle data breaks are used for input data transfers to the computer, output data transfers from the computer, and memory increment data breaks. Memory increment is a special output data break in which the content of a memory address is read, incremented by 1, and rewritten at the same address. It is useful for counting iterations or external events without disturbing the computer program counter (PC) or AC registers.

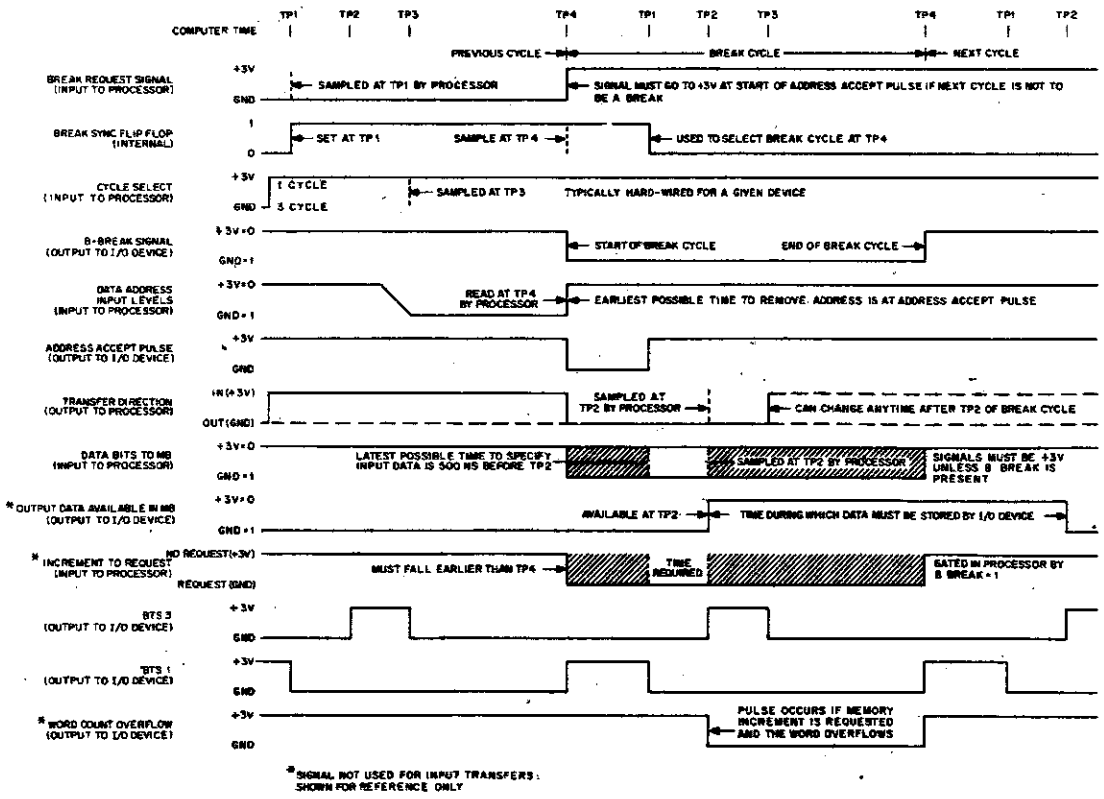
Input Data Transfers

Figure 10-2 illustrates the timing sequence of an input transfer data break. The address to be affected in core is normally provided in the device interface in the form of a 12-bit flip-flop register (data break address register), which has been present by the interface control by programmed transfer from the computer.

External registers and control flip-flops supplying information and control signals to the data break facility and other PDP-8/I [PDP-8/L] interface elements are shown in Figure 10-3. The input data register (DR) holds the 12-bit data word to be written into the computer core memory location specified by the address contained in the address register (AR). Appropriate output terminals of these registers are connected to the computer to supply ground potential to designate binary 1s. Since most devices that transfer data through the data break facility are designed to use either single-cycle or three-cycle breaks, but not both, the cycle-select signal can usually be supplied from a stable source (such as a ground connection or a −3V clamped load resistor), rather than from a bistable device as shown in Figure 10-3.



Negative I/O Bus & Logic



Positive I/O Bus & Logic

Figure 10-2. Single-Cycle Data Break Input Transfer Timing Diagram

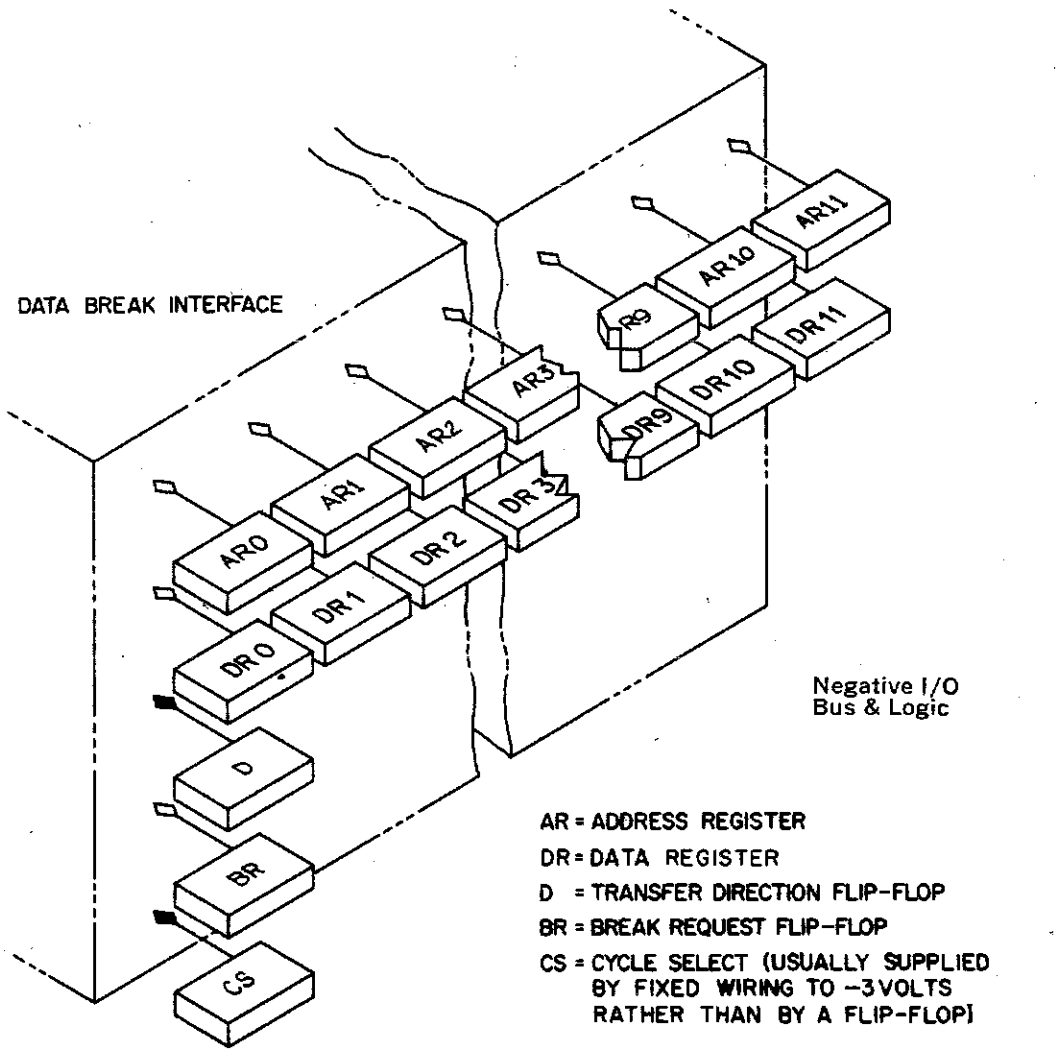


Figure 10-3. Device Interface Logic for Single-Cycle Data Break Input Transfer

Other portions of the device interface, not shown in Figure 10-3, establish the data word in the DR, set the address into the AR, set the direction flip-flop to indicate an input data transfer, and control the break request flip-flop. These operations can be performed simultaneously or sequentially, but all transients should occur before the data break request is made. Note that the device interface need supply only static levels to the computer, minimizing the synchronizing logic circuits necessary in the device interface.

When the data break request arrives, the computer completes the current instruction, generates an address-accepted pulse (at TP4 of the cycle preceding the data break) to acknowledge receipt of the request, then enters the Break state to effect the transfer (see Chapter 5 of this handbook for more details on data break operations performed by the computer). The address-accepted pulse can be used in the device interface to clear the break request flip-flop, increment the content of the address register, etc. If the break-request signal is removed before TP1 time of the data break cycle, the computer performs the transfer in one 1.5-μs cycle and returns to programmed operation. Devices using the positive bus must gate their data bits with B Break to ensure proper operation when used with the DM04 multiplexer.

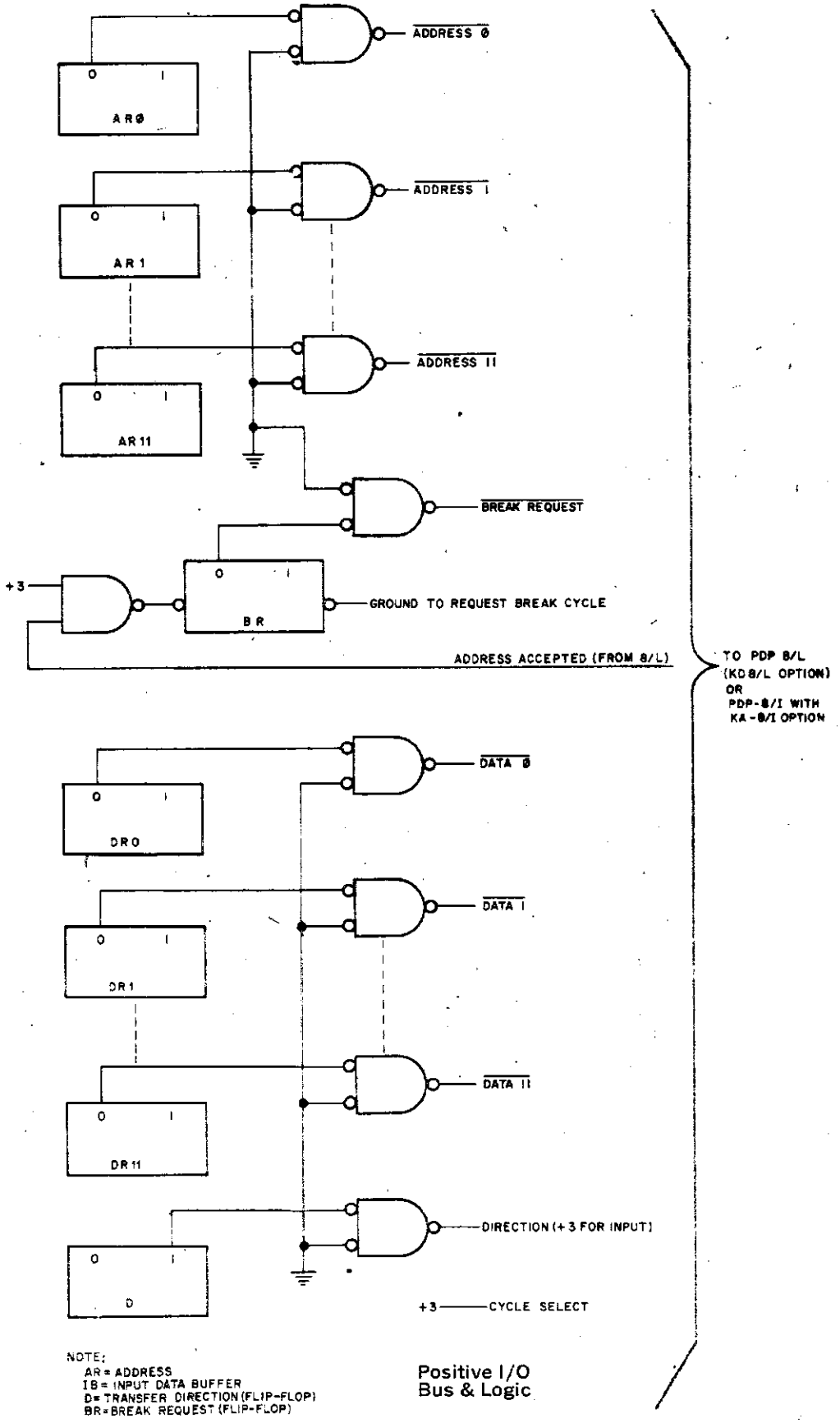


Figure 10-3. Device Interface Logic for Single-Cycle Data Break Input Transfer

Output Data Transfers

Timing of operations occurring in a single-cycle data break is shown in Figure 10-4. Basic logic circuits for the device interface used in this type of transfer are shown in Figure 10-5. Address and control signal generators are similar to those discussed previously for input data transfers, except that the transfer-direction signal must be at ground potential to specify the output transfer of computer information. An output data register (OB) is usually required in the device interface to receive the computer information. The device, and not the computer, controls strobing of data into this register. The device must supply strobe pulses for all data transfers out of the computer (programmed or data break), since circuit configuration and timing characteristics differ in each device.

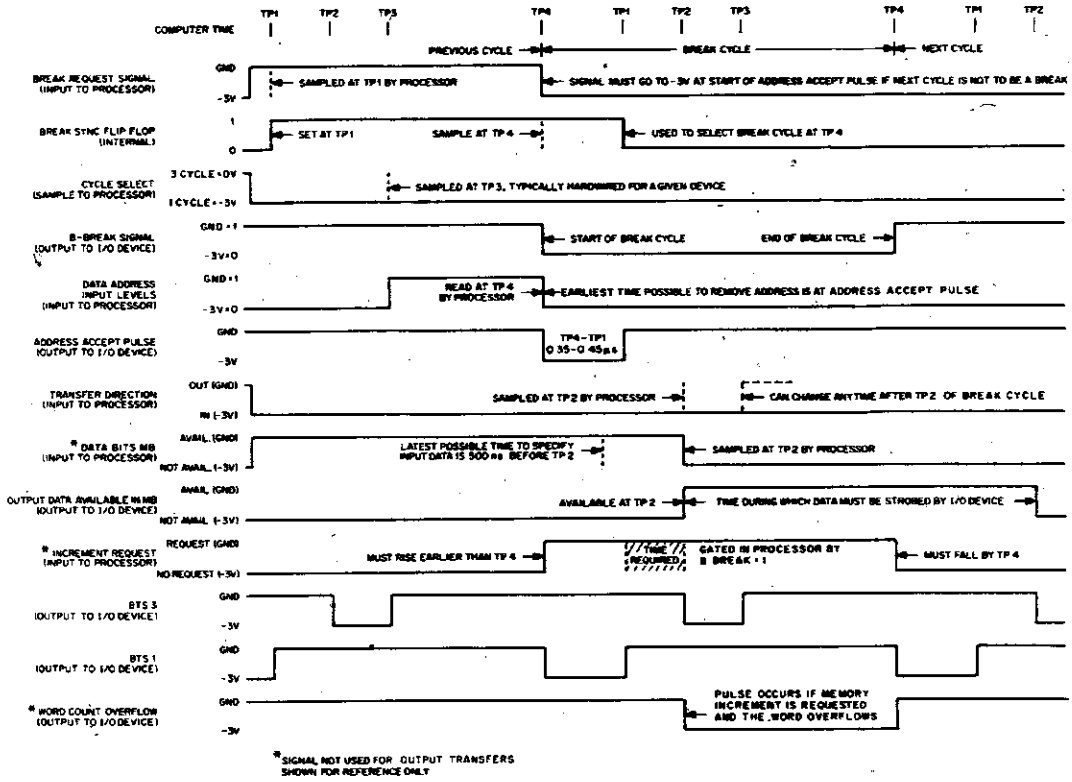
When the data break request arrives, the computer completes the current instruction and generates an address-accepted pulse as in Input Data Transfers. At time TP4, the address supplied to the PDP-8/I [PDP-8/L] is loaded into the MA, the Break state is entered, and the MB is cleared. Not more than 950 ns after TP4 (at time TP2), the content of the device-specified core memory address is read and available in the MB. (This word is automatically rewritten at the same address during the last half of the break cycle and is available for programmed operations when the data break is complete.) Data-bit signals are available as static levels of ground potential for binary 1s and −3V for binary 0s on the negative bus. On the positive bus, output data bits are available as static levels of +3V for binary 1s, and 0V, for binary 0s. The MB is changed only at time TP2 of each computer cycle, so the data word is available in the MB for approximately 1.5 μs to be strobed by the device interface.

Generation of the strobe pulse by the device interface can be synchronized with computer timing through use of timing pulses BTS1 or BTS3, which are available at the computer interface. In addition to a timing pulse (delayed or used directly from the computer), generation of this strobe pulse should be gated by condition signals that occur only during the break cycle of an output transfer. Figure 10-6 shows typical logic circuits to effect an output data transfer. In this example, the B-break signal and an inverted transfer-direction signal are combined in a diode NAND gate to a condition a diode-capacitor-diode (DCD) gate. A buffered BTS1 pulse triggers the DCD gate to produce the strobe pulse.

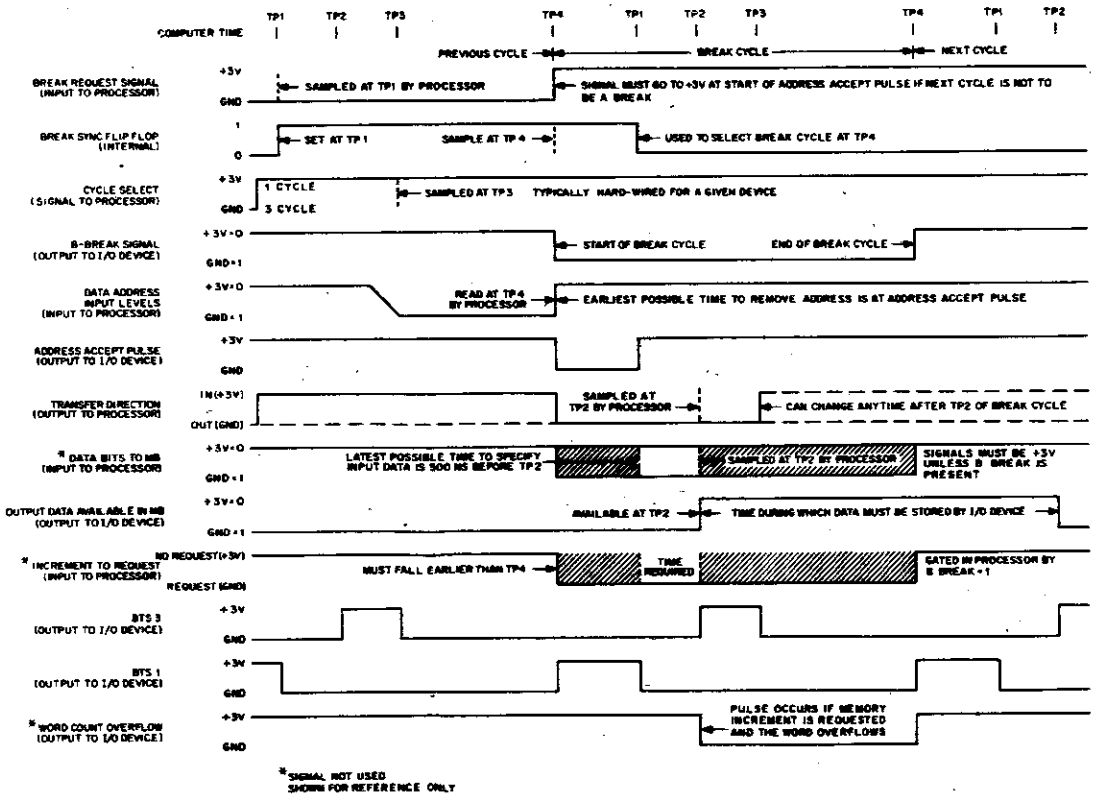
The BTS1 pulse determines the timing of the transfer in this example, since the input of the output buffer register has DCD gates. Conventional DCD gates require a minimum set-up time of 400 ns, which is adequately provided between the time when data is available in the MB and time TS1.

A similar diagram for use on the positive bus is also shown.

By careful design of the input and output gating, one register can serve as both the input and the output buffer register. Most DEC options using the data break facility have only one data buffer register with appropriate gating to allow it to serve as an output buffer when the transfer direction is out of the computer, and as an input buffer when the transfer direction is into the computer.

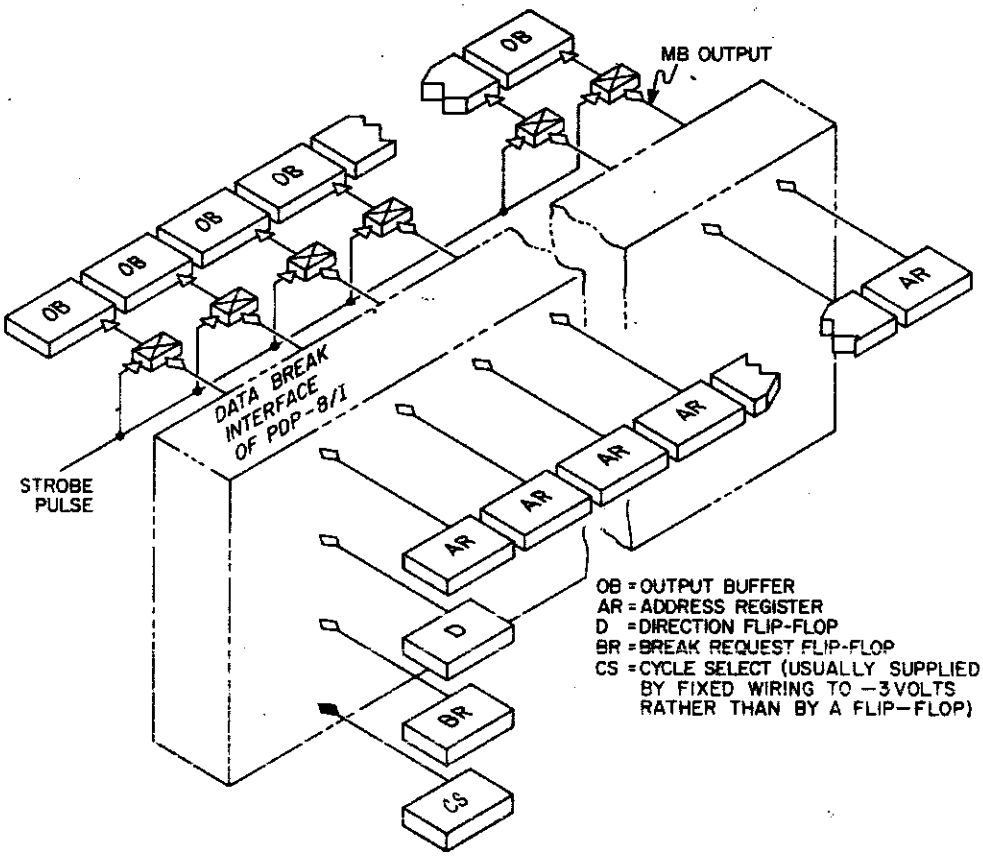


Negative I/O Bus & Logic



Positive I/O Bus & Logic

Figure 10-4. Single-Cycle Data Break Output Transfer Timing Diagram



Negative I/O
Bus & Logic

Figure 10-5. Device Interface Logic for Single-Cycle Data Break Output Transfer

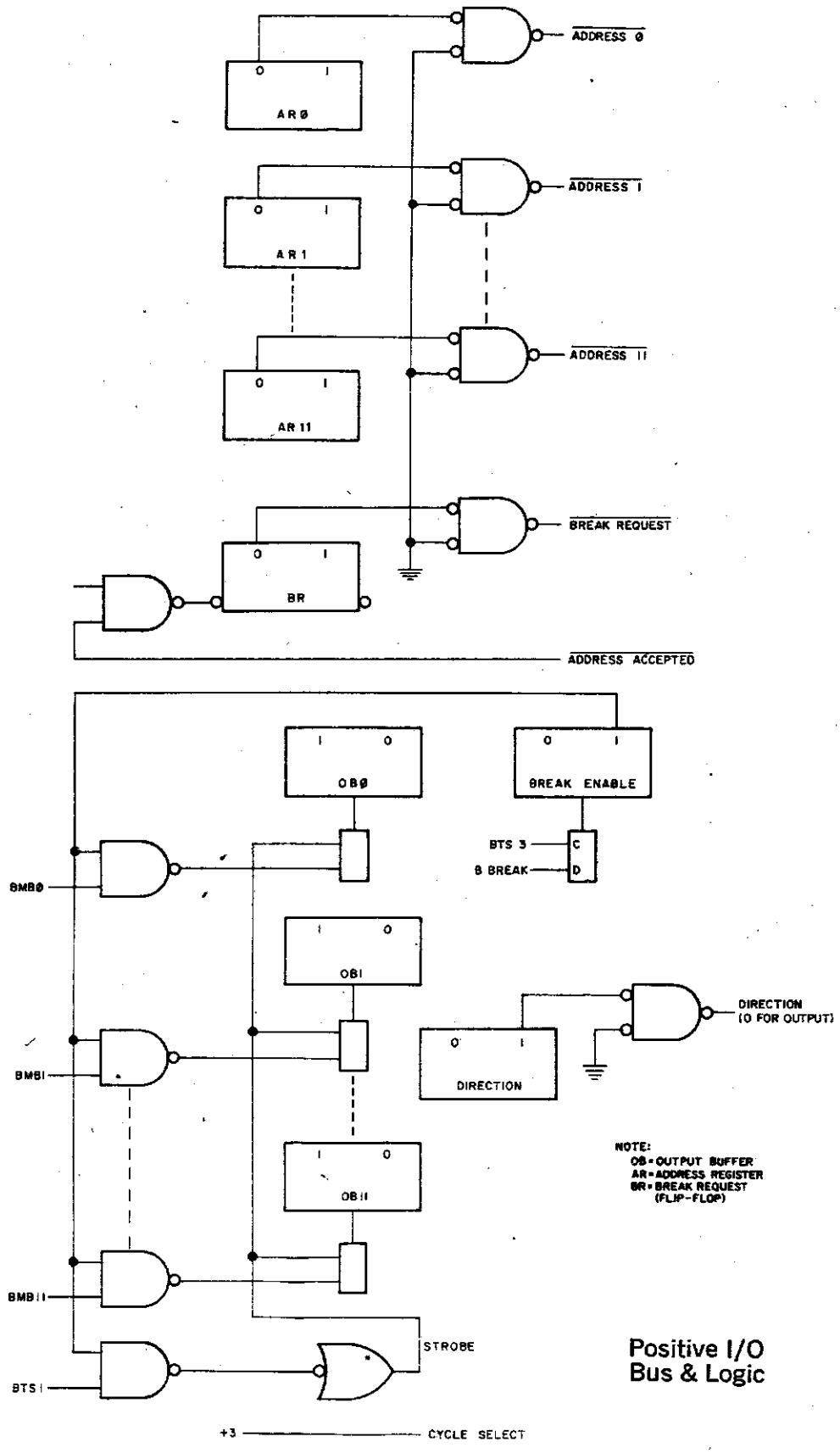


Figure 10-5. Device Interface Logic for Single-Cycle Data Break Output Transfer

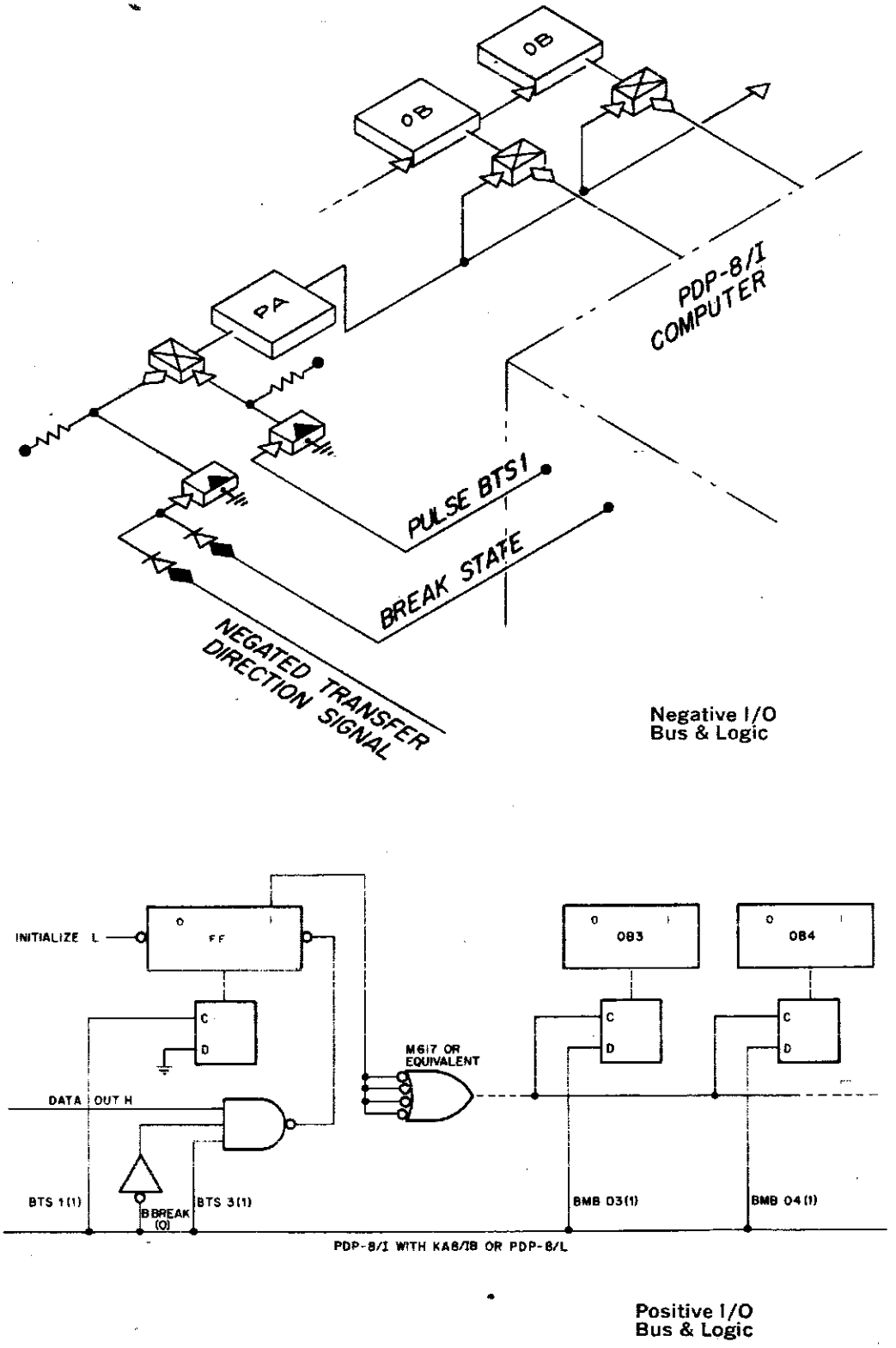


Figure 10-6. Device Interface for Strobing Output Data

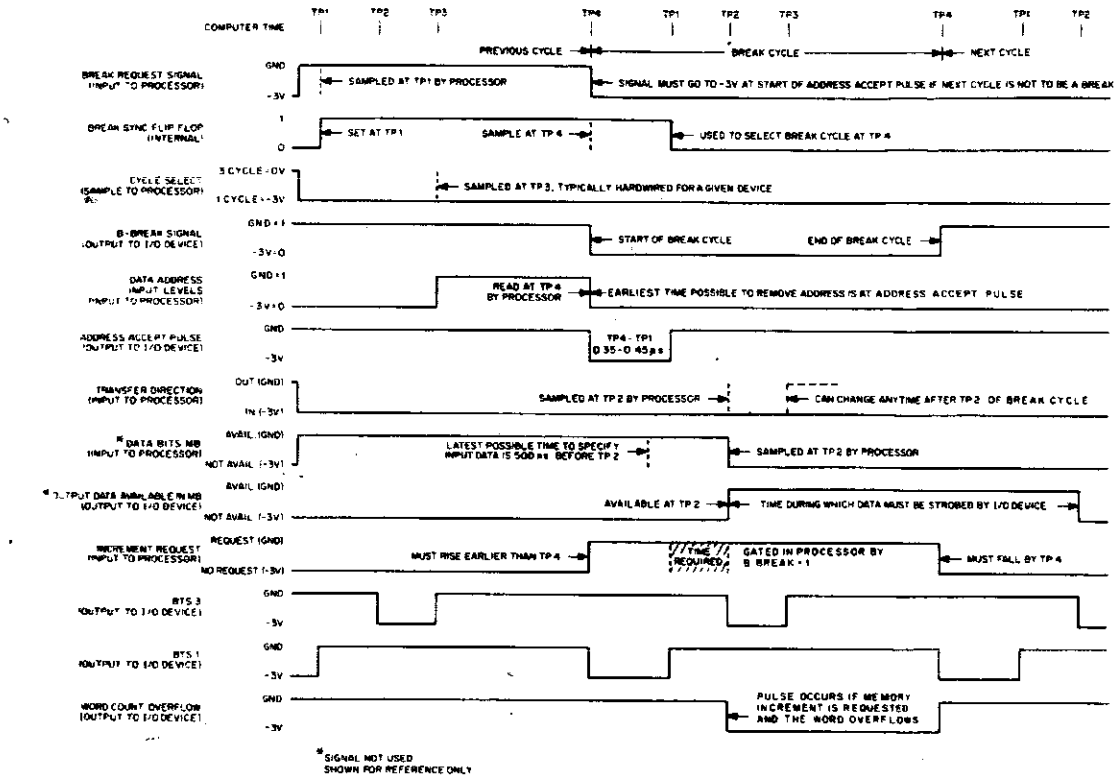
Memory Increment

In this type of data break, the content of core memory at a device-specified address is read into the MB, incremented by 1, and rewritten at the same address within one 1.5 μs [1.6 μs] cycle. This feature is particularly useful in building a histogram of a series of measurements, such as in pulse-height analysis applications. For example, in a computer-controlled experiment that counts the number of times each value of a parameter is measured, a data break can be requested for each measurement, and the measured value can be used as the core memory address to be incremented (counted).

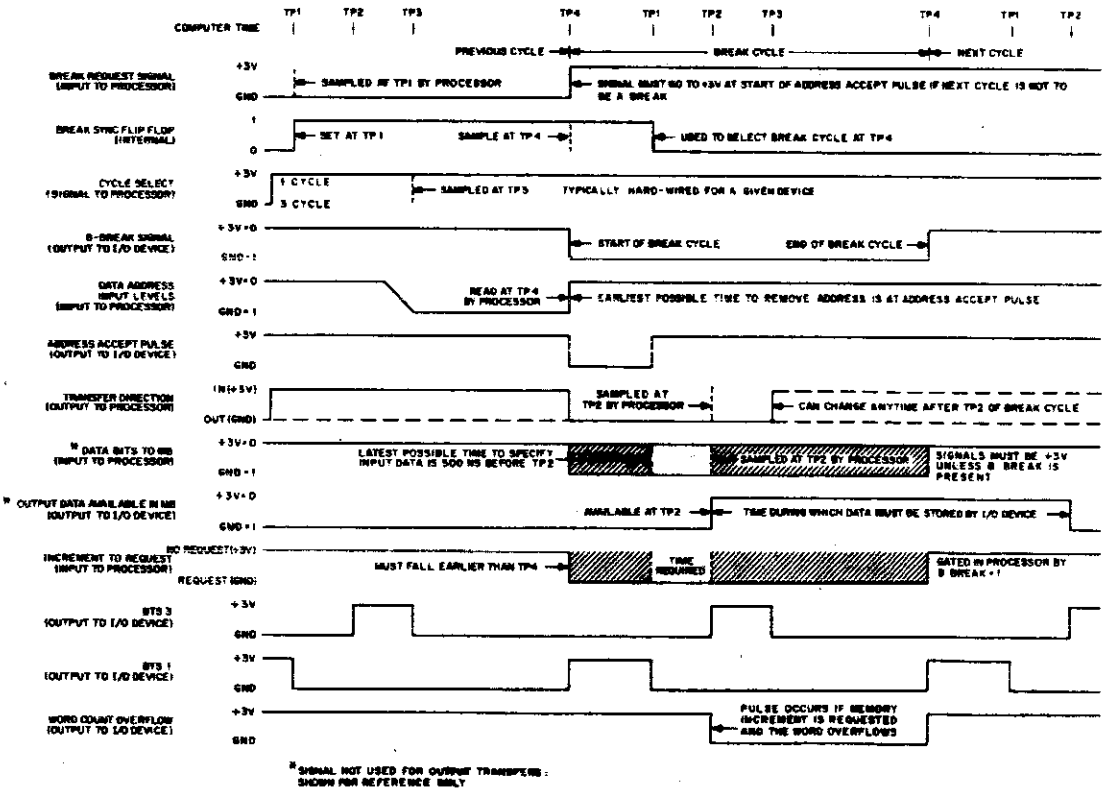
Signal interface for a memory increment data break is similar to an output transfer data break, except that the device interface generates an increment-MB signal and does not generate a strobe pulse (no data transfer occurs between the PDP-8/I [PDP-8/L] and the device). Timing of memory increment operations appears in Figure 10-7 and an example of the logic circuits used by a device interface appears in Figure 10-8.

An interface for a device using memory increment data breaks must supply 12 data-address signals, a transfer-direction signal, a cycle-select signal, and a break-request signal to the computer's data break facility, as is detailed in the description of an output transfer data break. In addition, a ground potential increment-MB signal must be provided at least 250 ns before time TP2 of the break cycle. This signal can be generated in the device interface by AND-combining the B-break computer output signal, the output transfer condition of the transfer-direction signal, and the condition signal in the device that indicates that an increment operation should take place. When the computer receives this increment-MB signal, it forces the MB control element to generate a count-MB signal at time TP2 to increment the content of the MB.

The device interface logic, shown in Figure 10-8, samples the word count overflow signal to determine if word count overflows when the data word is incremented. If overflow occurs, this logic requests a program interrupt to allow the program to take some appropriate action, such as incrementing a core memory for numbers above 4096, stopping the test to compile the data gathered to the current point in the operation, reinitializing the addressing, etc. The logic in the figure uses the select code of programmed data transfer operation to skip on the overflow condition to determine the cause of a program interrupt, to clear the overflow flip-flop, and to clear the device flag. Note that the devices that use data break transfers almost always use programmed data transfers to start and stop operation of the device, to initialize registers. etc., and do not rely on data break facilities alone to control their operations.

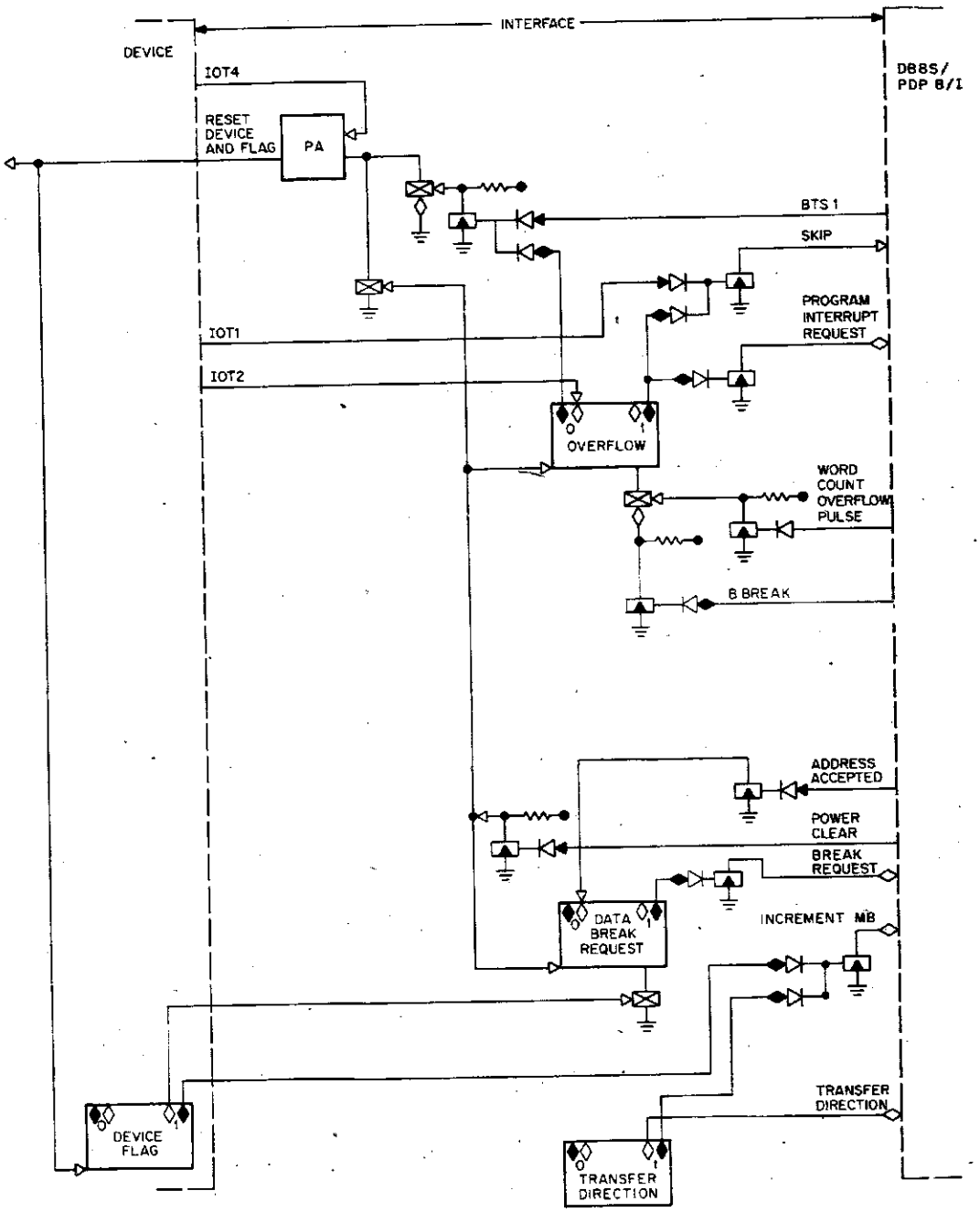


Negative I/O Bus & Logic



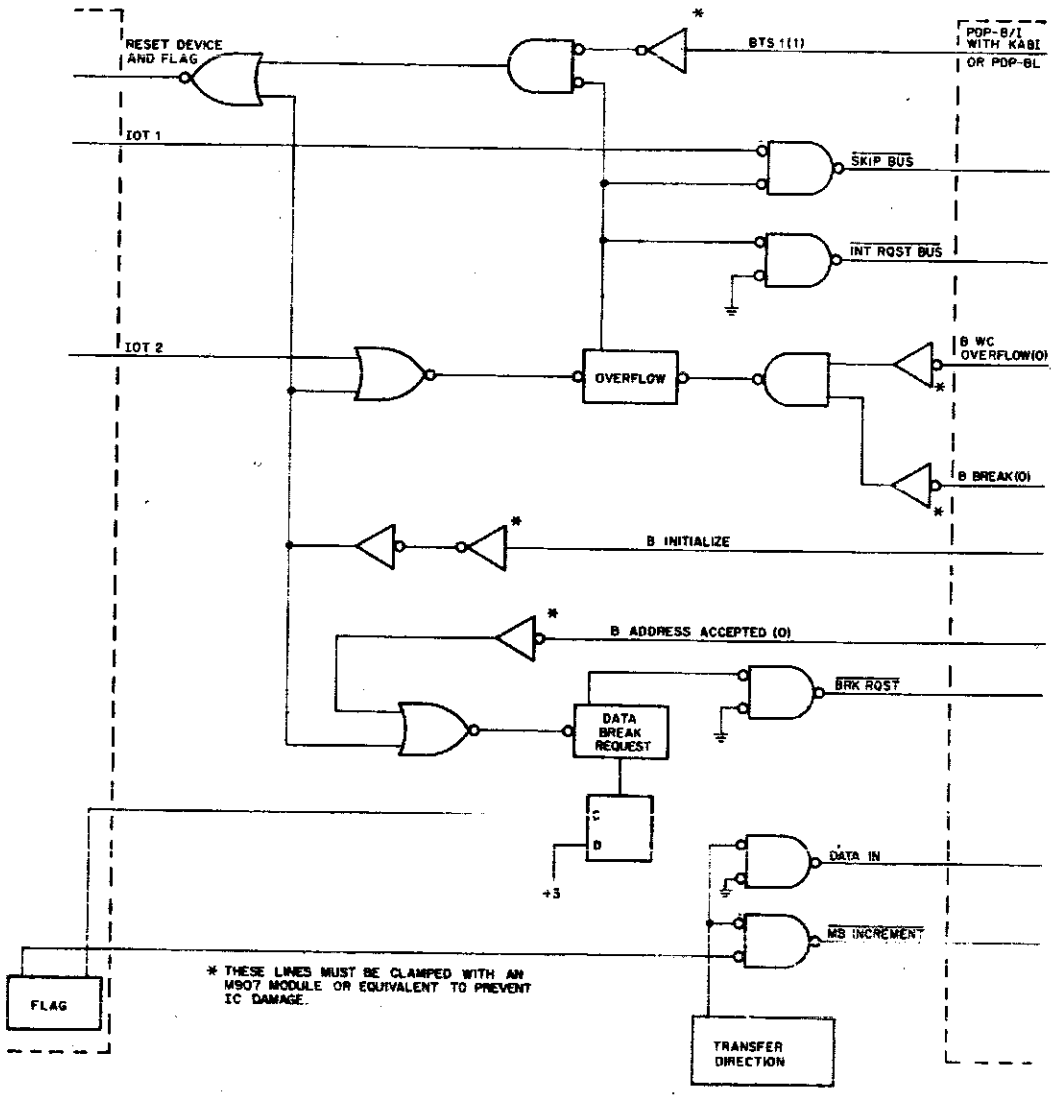
Positive I/O Bus & Logic

Figure 10-7 Memory Increment Data Break Timing Diagram



Positive I/O
Bus & Logic

Figure 10-8. Device Interface Logic for Memory Increment Data Break



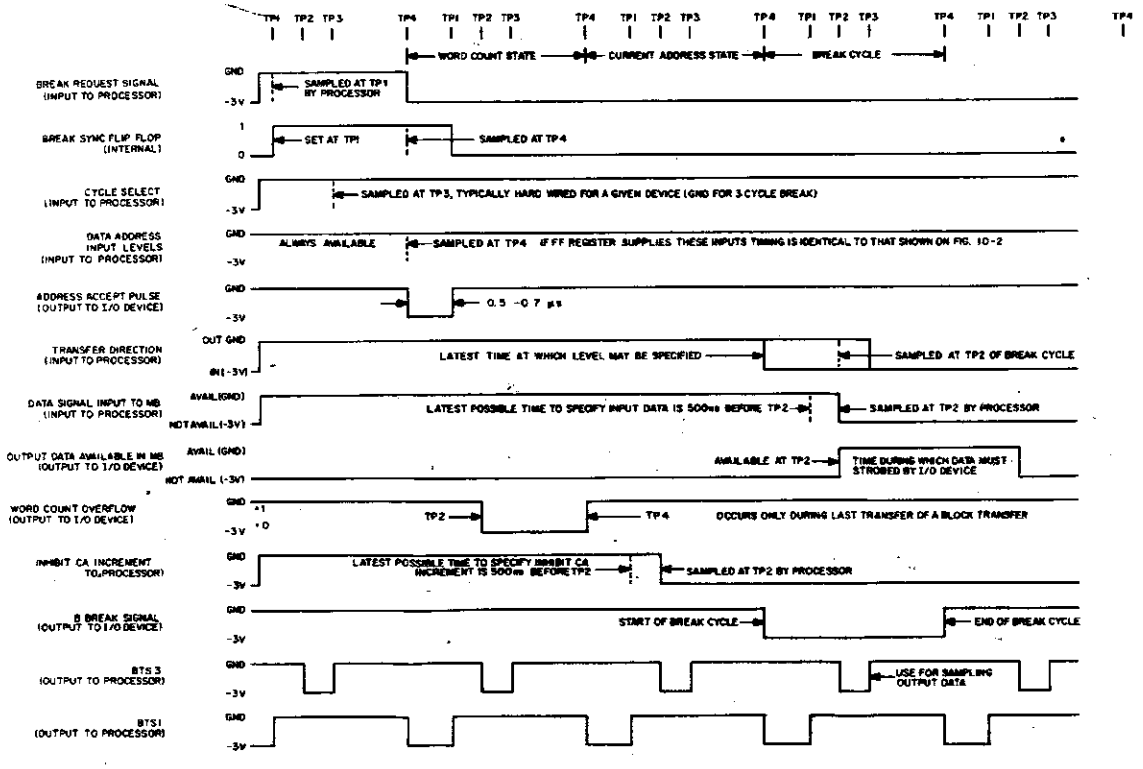
Negative I/O
Bus & Logic

Figure 10-8. Device Interface Logic for Memory Increment Data Break

THREE-CYCLE DATA BREAKS

Timing of input or output 3-cycle data breaks is illustrated in Figure 10-9. The 3-cycle data break uses the block transfer control circuits of the computer. The block transfer control provides an economical method of controlling the flow of data at high speeds between PDP-8/I [PDP-8/L] core memory and fast peripheral devices, e.g., drum, disk, magnetic tape and line printers, allowing transfer rates in excess of 220 kHz [208 kHz].

The 3-cycle data break facility provides separate current address and word count registers in core memory for the connected device, thus eliminating the necessity for flip-flop registers in the device control. When several devices are connected to this facility, each is assigned a different set of core locations for word count and current address, allowing interlaced operations of all devices as long as their combined rate does not exceed 220 kHz [208 kHz]. The device specifies the location of these registers in core memory; thus, the software remains the same regardless of what other equipment is connected to the machine. Since these registers are located in memory, they may be



Negative I/O
Bus & Logic

Figure 10-9. Three-Cycle Data Break Timing Diagram

loaded and unloaded directly without the use of IOT pulses. In a procedure where a device requests to transfer data to or from core memory, the 3-cycle data break facility performs the following sequence of operations:

a. An address is read from the device to indicate the location of the word count register. This address is always the same for a given device; thus, it can be wired in and does not require a flip-flop register.

b. The content of the specified address is read from memory and 1 is added to it before it is rewritten. If the content of this register becomes 0 as a result of the addition, a WC-overflow pulse will be transmitted to the device. To transfer a block of n words, this register is loaded with -n during programmed initialization of the device. After the block has been fully transferred, this pulse is generated to signify completion of the operation.

c. The next sequential location is read from memory as the current address register. Although the content of this register is normally incremented before being rewritten, an increment CA-inhibit (+1 ➔ CA Inhibit) signal from the device may inhibit incrementation. To transfer a block of data beginning at location A, this register is program-initialized by loading it with A-1.

d. The content of the previously read current address is transferred (after incrementing) to the MA to serve as the address for the data transfer. This transfer may be in either direction, in a manner identical to the single-cycle data break system.

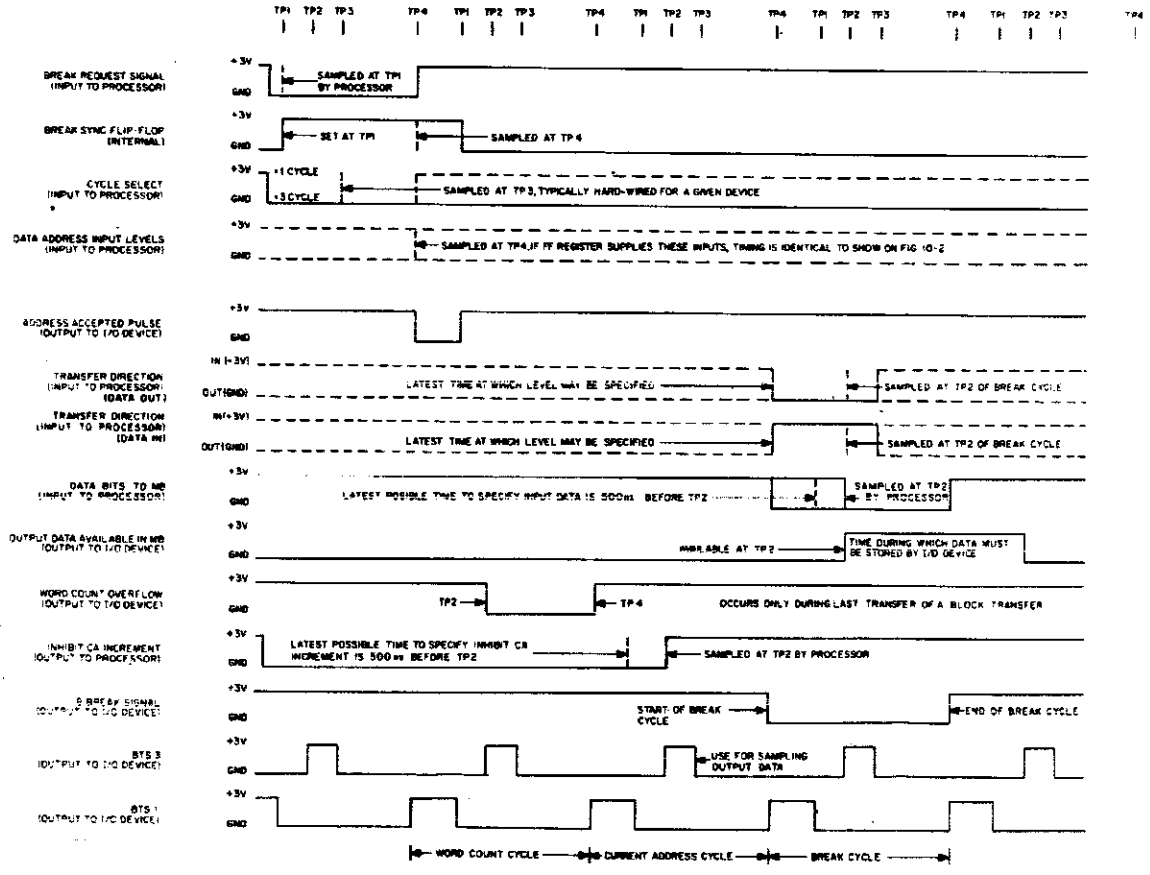


Figure 10-9. Three-Cycle Data Break Timing Diagram

The 3-cycle data break facility uses many of the gates and transfer paths of the single-cycle data break system, but does not preclude the use of single-cycle data break devices. Any combination of 3-cycle and single-cycle data break devices can be used in one system, as long as a multiplexer channel is available for each. Two additional control lines are provided with the 3-cycle data break. These are:

a. Word Count Overflow. A level change from GND to −3V, from time TP2 to time TP4, is transmitted to the device when the word count becomes equal to 0.

b. Increment CA Inhibit. When ground potential exists, this device-supplied signal inhibits incrementation of the current address word.

In summary, the 3-cycle data break is entered similarly to the single-cycle data break, with the exception of supplying a ground level cycle-select signal to allow entry of the Word Count state to increment the fixed core memory location containing the word count. The device requesting the break supplies this address, as in the single-cycle data break, except that this address is fixed and can be supplied by wired ground and −3V or +3V signals, rather than from a register. Following the Word Count state, a Current Address state is entered in which the core memory location following the word count address is read, incremented by one, restored to memory, and used as the transfer address (by MB being transferred to MA). Then the normal Break state is entered to effect the transfer.

GLC-8 is a computer-based system that will service 20 or more gas chromatographs simultaneously. It is the least expensive, most sophisticated system of its type available.

CHAPTER 11
DIGITAL LOGIC CIRCUITS

The digital logic circuits in this chapter are used to interface I/O devices to the PDP-8/I or PDP-8/L computer using Digital Equipment Corporation FLIP CHIP Modules. Logic Handbooks published by DEC, describe hundreds of FLIP CHIP Modules, with their component circuits, associated accessories, hardware, Power supplies, and mounting panels. Before beginning on interface design for a special I/O device, the designer should study the Logic Handbooks carefully.

The basic logic circuits used for interfacing with these computers are: AND, OR, NAND, NOR, Flip-Flop, Single-Shot, Schmitt Trigger, Inverter, Amplifier, and Bus Drivers. A brief discussion of these functions and their logic symbology follows.

The symbology employed with the PDP-8 family of computers and M-series modules is similar to MIL-STD-806B. This chapter describes DEC symbology with definitions of logic functions, graphic representations of the functions, and examples of their application. A Table of Combinations is also shown.

LOGIC SYMBOLS

The following list of logic symbols has truth tables that show graphic representations of the logic functions. In the truth tables, the letter H stands for HIGH (±3V) and the letter L stands for LOW (0V). Examples of DEC symbology are shown along with figures and truth tables.

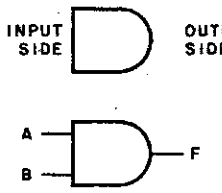
State Indicator

The presence of the small circle symbol at the input(s) of a function indicates that an L input signal activates the function. The absence of this small circle indicates that an H input signal activates the function. Similarly, a small circle at the output of a function indicates that the output terminal of the activated function is relatively low and the absence of the circle indicates the output is relatively high.

State Indicator Absent

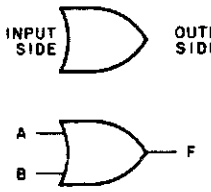
a. AND — The symbol in Figure 11-1 represents the AND function. The output (F) is high only if both inputs (A and B) are high.

b. OR — The symbol in Figure 11-2 represents the OR function. The OR output (F) is high if any input (A or B) is high.



INPUT		OUTPUT
A	B	F
L	L	L
L	H	L
H	L	L
H	H	H

Figure 11-1. Symbol, AND Function



INPUT		OUTPUT
A	B	F
L	L	L
L	H	H
H	L	H
H	H	H

Figure 11-2. Symbol, OR Function

State Indicator Present

a. NAND — The symbol in Figure 11-3 represents one version of the NAND function. The output (F) is low only when all inputs (A, B and C) are high. NAND logic is the major gate configuration in the PDP-8 family.

INPUT			OUTPUT
A	B	C	F
L	L	L	H
L	L	H	H
L	H	L	H
L	H	H	H
H	L	L	H
H	L	H	H
H	H	L	H
H	H	H	L

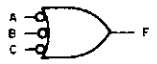
Figure 11-3. Symbol, NAND Function

b. NOR — The symbol in Figure 11-4 represents one version of the NOR function. The output (F) is low if one or more of the inputs (A, B or C) is high.

INPUT			OUTPUT
A	B	C	F
L	L	L	H
L	L	H	L
L	H	L	L
L	H	H	L
H	L	L	L
H	L	H	L
H	H	L	L
H	H	H	L

Figure 11-4. Symbol, NOR Function

c. NOR — The symbol in Figure 11-5 represents another version of the NOR function. The output (F) is high if one or more of the inputs (A, B or C) is low. The NOR version for this function is identical to one version of the NAND function, shown in Figure 11-3.



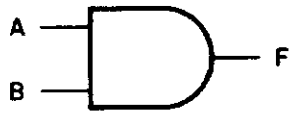
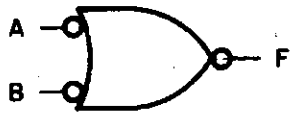
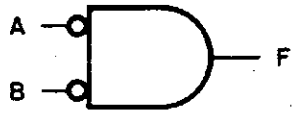
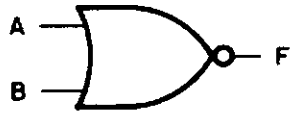
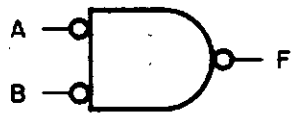
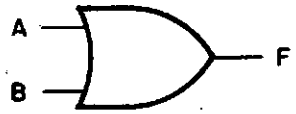
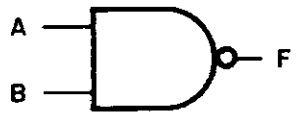
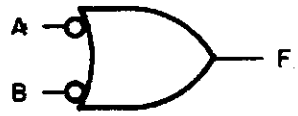
INPUT			OUTPUT
A	B	C	F
L	L	L	H
L	L	H	H
L	H	L	H
L	H	H	H
H	L	L	H
H	L	H	H
H	H	L	H
H	H	H	L

Figure 11-5. Symbol, NOR Function

Table of Combinations

Table 11-1 illustrates the applications, functions, and truth tables of two variables and their equivalents, as well as their relationship to DEC logic.

Table 11-1. Table of Combinations

AND	OR	A	B	F
		H	H	H
		H	L	L
		L	H	L
		L	L	L
		H	H	L
		H	L	L
		L	H	L
		L	L	H
		H	H	H
		H	L	H
		L	H	H
		L	L	L
		H	H	L
		H	L	H
		L	H	H
		L	L	H

Flip-Flop

The symbol in Figure 11-6 shows the flip-flop function. The pins are numbered counterclockwise on a standard flip-flop. The flip-flop has four possible inputs: sets(s), reset(r), data(d), and clock(c); and two data outputs, logic 0 (low) and logic 1 (high). If the data input is high and a pulse is applied to the clock input, the flip-flop will set to the logic 1 state. If the data input is low and a pulse is applied to the clock input, the flip-flop will reset to the logic 0 state. When the data input is shown with a small circle (redefined flip-flop), the opposite of the above is true; that is, if the data input is high and a pulse is applied to the clock input, the flip-flop will go to its logic 0 or reset state, etc. Note that the pins are numbered clockwise on a redefined flip-flop. Normally, the set and reset inputs are high, and a change from high-to-low state at either of these inputs causes the flip-flop to set or reset, respectively.

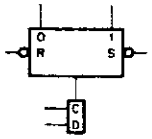


Figure 11-6. Symbol, Flip-Flop Function

Single-Shot Function

The symbol in Figure 11-7 shows the single-shot (SS) function. The output signal shape, amplitude, duration, and polarity are determined by the circuit characteristics of the SS device. When it is not activated, the single-shot device is in either a 0 or 1 state. When the input is pulsed by a high-to-low level change, the 1 output goes high and remains high, and the 0 output goes low and remains low for the specific time of the device (normally 50 ms).

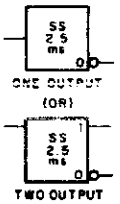


Figure 11-7. Symbol, Single-Shot Function

Schmitt Trigger

The symbol in Figure 11-8 shows the Schmitt trigger (ST) function. The Schmitt trigger is either in a 0 or a 1 state normally (inactivated). When the input signal crosses a predetermined voltage threshold, the Schmitt trigger changes to its opposite state and remains there until the input signal falls below the threshold.

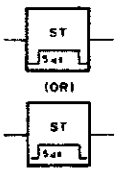


Figure 11-8. Symbol, Schmitt Trigger Function

General Logic Symbols

The symbol in Figure 11-9 is used for functions not specified elsewhere. An example of this symbology is the inhibit driver used in the PDP-8 family.

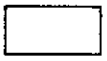


Figure 11-9. Symbol, General Logic

Amplifier

The symbol in Figure 11-10 shows a linear or nonlinear, current or voltage amplifier. This symbol is used to represent level changers, inverters, emitter followers, and lamp drivers.

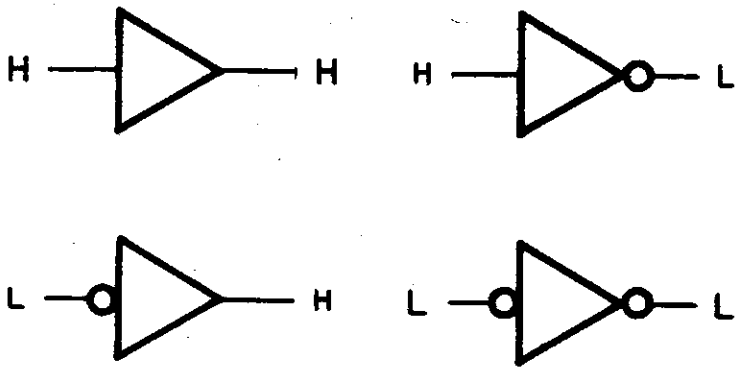


Figure 11-10. Symbol, Amplifier

Time Delay

The symbol in Figure 11-11 shows a time-delay device. The time-delay duration is specified inside the symbol unless there are two or more outputs. When there are two or more outputs, the delay time of each output is marked adjacent to that output.

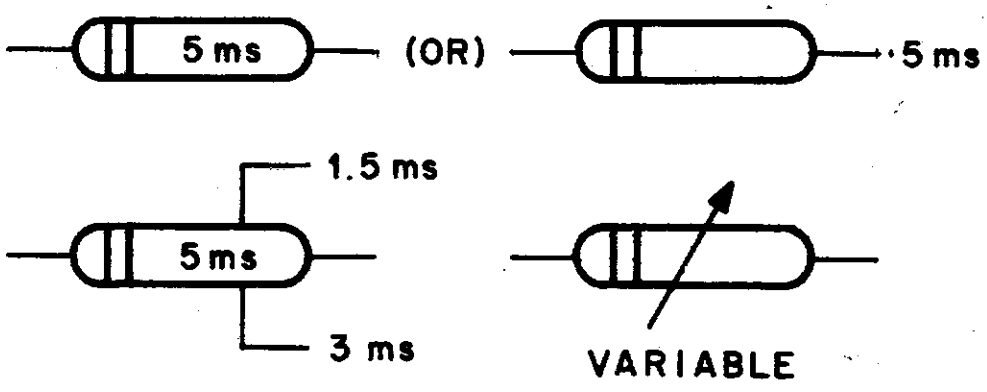


Figure 11-11. Symbol, Time Delay Function

DEC Symbology (Excluding M-Series FLIP CHIP Logic)

The PDP-8/I (negative bus) is shown in DEC logic symbology; therefore, since this symbology is used in this handbook, DEC logic symbols are included and are shown in Figure 11-12.

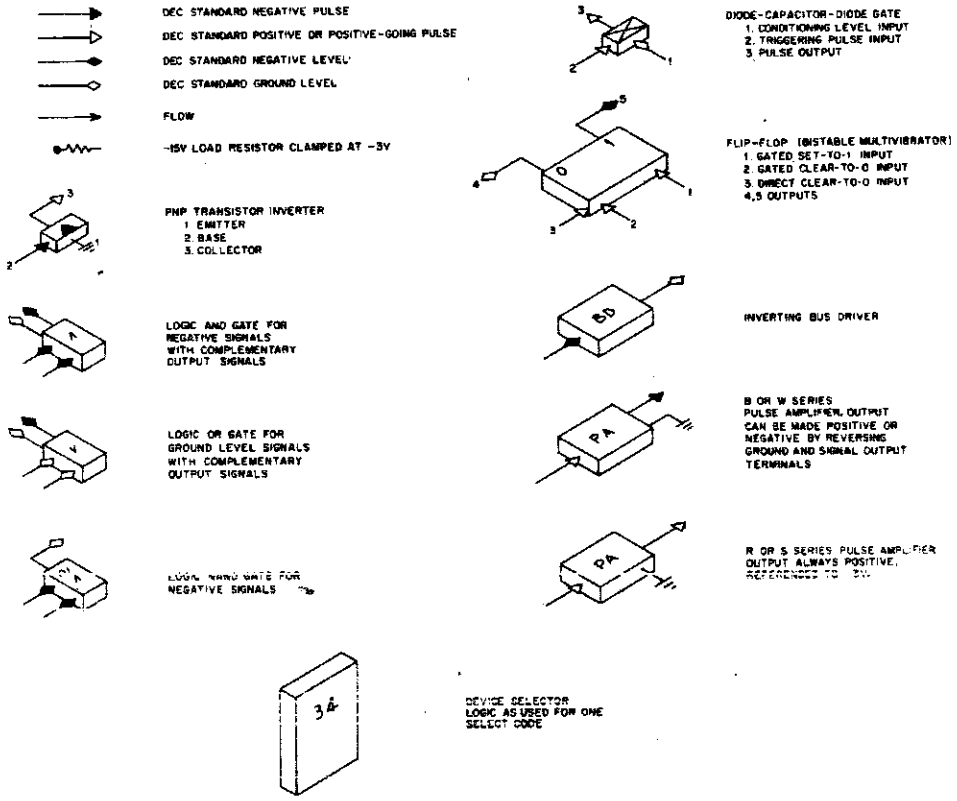


Figure 11-12. DEC Logic Symbols

CHAPTER 12
INTERFACE

All notations in brackets [ ] indicate data for the PDP-8/L computer only.

GENERAL

Since the processing power of a computer depends largely upon the type and number of peripheral I/O devices that can be interfaced with it, the PDP-8/I and PDP-8/L have been designed for easier interface with a large variety of these I/O devices. This chapter deals with and defines the electrical characteristics of computers, thus allowing the user to design and implement any interfaces which might be required to connect I/O devices.

The simple I/O technique of the computers, the availability of DEC's FLIP CHIP logic circuit modules, and DEC's policy of giving assistance wherever possible, allow inexpensive, straightforward I/O device interfaces to be constructed. Should questions arise relative to the computer interface characteristics, the design of interfaces using DEC modules or installation planning, customers are invited to telephone any of the sales offices or the main plant in Maynard, Massachusetts. Digital Equipment Corporation makes no representation that the interconnection of its circuit modules in the manner described herein will not infringe on existing or future patent rights. Nor do the descriptions contained herein imply the granting of licenses to use, manufacture, or sell equipment constructed in accordance therewith.

NOTE: Chapters 9 and 10 of this Handbook must be read and thoroughly understood before attempting any interface design.

SECTION 12-1 INTERFACE SIGNALS

SIGNAL SUMMARY

All interface connections to the PDP-8/I [PDP-8/L] are made at assigned module receptacle connectors in the mounting frame. Capital letters designate horizontal rows of modules within a mounting from top to bottom. Module receptacles are numbered from left to right as viewed from the wiring side (right to left from the module side). Terminals of a connector or module are assigned capital letters from top to bottom, omitting G, I, O, and Q. Double-sided connectors or modules are used with suffix number 1 designating the left side and suffix number 2 designating the right side.

Module receptacles and associated interface signal connections are listed in Table 12-1.

Table 12-1
Interface Signal Connections

PDP-8/I Interface Signals

RECEPTACLE	SIGNAL USE
J01	BAC0-8 outputs
J02	BAC9-11, BIOP, BIOP2, BIOP4, BTS1, BTS3, and B initialized outputs
J03	BMB0-5 outputs
J04	BMB6-11 outputs
J05	AC0-8 inputs

Table 12-1
Interface Signal Connections (cont.)

RECEPTACLE	SIGNAL USE
J06	AC9-11, SKIP, CLEAR AC, LINE MUX, INTERRUPT REQUEST inputs, B TTINST, B RUN, outputs
J07	DATA ADD 0-8 inputs
J08	DATA ADD 9-11 inputs, B ADD ACCEPTED, B BREAK outputs, and INCREMENT MB
J09	DATA 0-8 inputs
J10	DATA 9-11 inputs, BWC OVERFLOW, CYCLE SELECT, and INCREMENT CA, EXT DATA ADD 0, 1, 2 (on late model machines)
J11	EXT DATA ADD 0, 1, 2 inputs
PDP-8/L Interface Signals	
D36	BAC 0-11, BIOP1, BIOP2, BIOP4, BTS1, BTS3, and B INITIALIZE outputs
D35	BMB 0-11 outputs
D34	AC0-11, SKIP, INTERRUPT REQUEST, and CLEAR AC inputs, B RUN output
C36	DATA ADD 0-11, BREAK REQUEST, DATA IN inputs; B BREAK, B ADD ACCEPTED, B INITIALIZE outputs
C35	DATA 0-11, CYCLE SELECT, INCREMENT CA, EXT DATA ADD inputs, BWC OVERFLOW output

Terminals C, F, J, L, N, R, and U of these receptacles are grounded within the computer and terminals D, E, H, K, M, P, S, T, and V carry signals. These terminals mate with Type W011 or W031 Signal Cable Connectors. [Terminals A1, C1, F1, K1, N1, R1, T1, C2, F2, J2, L2, N2, R2, and U2 of these receptacles are grounded within the computer, and terminals B1, D1, E1, H1, J1, L1, M1, P1, S1, D2, E2, H2, K2, M2, P2, S2, T2, and V2 carry signals. Terminal A2 and B2 are used normally for power. No connection should be made to these two pins. These receptacles mate with either M903 or M904 Cable Connectors.]

Interface connection to the PDP-8/I [PDP-8/L] can be established for all peripheral equipment by making series cable connections between devices. In this manner only one set of cables is connected to the computer and two sets are connected to each device; one receiving the computer connection from the computer itself or the previous device; and one passing the connection to the next device. Where physical location of equipment does not make series bus connections feasible, or when cable length becomes excessive, additional interface equipment must be provided near the computer.

All logic signals passing between a positive bus PDP-8/I [PDP-8/L] and input/output equipment are positive voltage levels or positive signals, allowing direct TTL logic interface with appropriate diode clamp protection. Signals passing between a negative bus PDP-8/I and input/output equipment are standard negative DEC levels or standard DEC pulses.

For the positive bus computer (PDP-8/I with the KA8/IB Positive I/O bus option installed, and the PDP-8/L), the positive levels and pulses change from ground potential (0V to 0.4V) to (+2.4V to +3.6V) and vice-versa.

For the negative bus PDP-8/I computer, the standard DEC levels are either ground potential (0.0V to −0.3V), designated by an open diamond (— ◇); or −3V (−3.0V to −4.0V), designated by a solid diamond (— ◆). Standard pulses in the positive direction are designated by an open triangle (— ▷) and negative pulses are designated by a solid triangle (— ▶).

INPUT/OUTPUT INTERFACE SIGNAL DESCRIPTIONS

The words "input" and "output", enclosed in parentheses, indicate signal flow into or out of the computer, respectively.

Buffered Accumulator Bits (BAC 00-11) (Output)

The output data path for programmed I/O and interrupt service. These lines change as a function of contents of the accumulator (whether IOT instruction is being executed or not). A pulse, generated by IOT decoding, loads contents of these lines into external registers.

Buffered Memory Bits (BMB 00-11) (Output)

These are probably the 12 most important bits in the system. BMB03-08 are brought out buffered from both sides of the MB flip-flops. Thus, it is easy to construct gates based on various combinations of these bits. Bits BMB00-02 and BMB09-11 are used primarily for data break, although BMB09-11 are also useful for extra decoding when microprogramming is not required.

IOPs (Output)

When an IOT is detected, the computer sequentially samples the state of MB11, MB10, and MB09 flip-flops, in that order. If any of these bits are set, pulses appear at the lines designated IOP1, IOP2, and IOP4, respectively. IOPs are approximately 600 ns wide and 800 ns apart in both the PDP-8/I and PDP-8/L computers. It should be noted that MB is stable during the IOP interval. The MB is loaded approximately 0.5 us before IOP1; however, users should not rely on this time for DCD gate setting time because of cable delays and the fact that future machines may have different timing.

AC Inputs (Input)

These signals cause data to be read into the AC. The data path for input programmed transfers, these lines, like all input lines, are designed to be operated as wired OR circuits. That is, these lines are loaded in the central processor and are driven to ground in the peripheral(s) by circuits with no collector loads.

Clear Accumulator (Input)

In the PDP-8/I and PDP-8/L, this line may be asserted at the same time as the AC inputs are asserted. The result is a jam-transfer of data from AC input lines to AC. The Clear Accumulator line may also be asserted by a separate IOP if desired. The result is then 0 → AC.

Skip (Input)

This signal causes an extra PC increment.

Interrupt Request (Input)

This signal causes the execution of a JMS to location 0 if the interrupt system has been enabled.

B Run (Output)

This signal signifies that the computer is executing instructions.

Power Clear or Initialize (Output)

A level which appears at the time the machine is turned on and also whenever the START switch is depressed. It is used for clearing I/O devices (especially flags and flip-flops controlling mechanical motion) and for placing the I/O devices into a known state.

Timing Pulses (Output)

These pulses indicate where the machines are in the memory cycle. The presence of BTS1 indicates that the computer is starting a memory cycle; in the middle of the cycle (between read and write) the timing pulse is BTS3. These signals are not usually necessary for interfacing, but are used primarily by the DM01 and DM04 data multiplexer for synchronizing several data break devices to the same computer.

BTT Inst., TT Data

Special signals used only by the 680/I Data Communication System.

Data Address (Inputs)

These signals are used by the break system. The data presented to these lines defines the address to which the break will occur (if a single-cycle break is being requested), or the address of Word Count information (if a 3-cycle break is being requested).

Data Bits (Inputs)

These signals are used to transfer data to the MB during an input data break. (NOTE: In positive bus machines, the gates driving the lines must be gated with B Break, otherwise the DM04 Multiplexer will not function properly).

Break Control Signals (Input and Output)

Break control signals include input break request, input transfer direction, input cycle select, input increment CA, input increment MB, output address accepted, output word count overflow, and output B break.

An input break signal causes the machine to initiate a 1- or 3-cycle data break. Signal is synchronized by the computer and must be removed upon receipt of the address-accepted pulse, unless a second data break cycle is desired. Break cycle(s) are entered following the execution of the current instruction.

An input transfer direction signals indicate the direction of data flow desired. Grounding this line causes data flow from the computer to the peripheral. This signal must be present during the B cycle. This line is labeled "Data In" in the PDP-8/I with the positive bus KA8/I, and in the PDP-8/L.

An input cycle select signal specifies 1- or 3-cycle data breaks. Grounding this line causes the break to be a 3-cycle break. Timing considerations for this signal are identical to those for the Break Request signal. This line is labeled "3-Cycle" in positive-bus machines (PDP-8/I with KA8/I option and PDP-8/L).

An input increment CA line is normally asserted, causing the accessed current address register to be incremented before use each time a 3-cycle data break is requested. Grounding this line inhibits the incrementing gates. If used, the signal must be present during the CA cycle. Since CA is not directly available, this signal should be established at or before-accepted time and may be removed when B Break is asserted.

An input INCREMENT MB signal is normally negated and is gated with the B Break signal at the MB incrementing gates. Asserting this signal during a break cycle (with data direction out), causes the contents of the accessed location to be incremented.

Output Address Accepted pulse indicate that the contents of the data address lines have been strobed into the computer memory address register and that the computer has recognized the break request. Break request and cycle-select signals are no longer required for the data break presently in progress and

should be removed unless a second data break is desired.

Output Word Count Overflow is a pulse which indicates that a 3-cycle data break is in progress, and that the computer is executing a word count cycle in which the word count is being incremented from 7777 to 0000. This signal is used to signify to the peripheral that the desired number of transfers will have been completed at the end of the current data break cycle. A pulse also occurs on this line if an INCREMENT MB signal causes an overflow.

The output B Break signal is asserted for one memory cycle. It signifies the data exchange memory cycle (the true Break cycle) is in progress. At the end of this cycle, the computer fetches the next instruction unless another data break request has been made at the beginning of this cycle.

Extended Data Address Input and Data Field Output

When the Memory Extension Control, Type MC8/I [MC8/L] is in the computer system, devices using the data break facility must supply a 12-bit data address and a 3-bit [1-bit] extended data address. Conversely, the programmed transfer of an address to a register in a device that uses the data break occurs as a 12-bit word from the accumulator and a 3-bit [1-bit] data field extension from the MC8/I [MC8/L]. The extended data address signal must be ground potential to designate a binary 1 and −3V [+3V] to designate a binary 0. During a 3-cycle break, WC and CA cycles always occur in field 0. Only the B cycle occurs in the field specified by the extended data address.

B Run Output Signal

The output of the RUN flip-flop flows to devices through the interface circuits. For the negative bus PDP-8/I, this signal is at −3V when the computer is performing instructions and is at ground potential when the program halts. For the positive-bus PDP-8/I (with the KA8/I option installed) and the PDP-8/L, the signal is at ground potential when the computer is performing instructions and is at +3V when the program halts. Magnetic tape and DECtape equipment use this signal to stop transport motion when the PDP-8/I [PDP-8/L] halts, preventing the tape from running off the end of the reel.

BTS1 and BTS3 Output Pulses

Two buffered timing pulse signals, designated BTS1 and BTS3, are supplied to I/O devices. These signals can synchronize operations in external equipment with those in the computer. The BTS1 and BTS3 pulse signals are derived from the TS1 and TS3 signals generated by the timing signal generator of the PDP-8/I [PDP-8/L]. The Type M650 Negative Output (level) Converter standardizes the TS1 and TS3 pulses in the negative-bus PDP-8/I as negative pulses. [The Type M660 Positive Level Driver standardizes the TS1 and TS3 pulses in both the positive bus PDP-8/I and the PDP-8/L as positive pulses.]

Initialize Output Pulses

The Initialize pulses generated and used within the PDP-8/I [PDP-8/L] are made available at the interface connections. External equipment uses these pulses to clear registers and control logic during the power turn-on period. Use of Initialize pulses in this manner is valid only when the logic circuits cleared by the pulses are energized before or at the same time the POWER switch is turned on. Operating the KEY START switch also generates the Initialize pulses. Two B Initialize signals are provided in later PDP-8/Is and in the PDP-8/L at the interface connections (see Table 12-10), one each for I/O and Data Break. The load on these two lines should be equalized by using B INITIALIZE-2 for break devices, but the two lines should not be connected together.

Figure 12-1 and Table 12-2 illustrate and define the input and output lines between the PDP-8/I and the I/O device on the programmed I/O bus. Similarly, Figure 12-2 and Table 12-3 illustrate and define the input and output lines between the PDP-8/I and the I/O device on the data break bus.

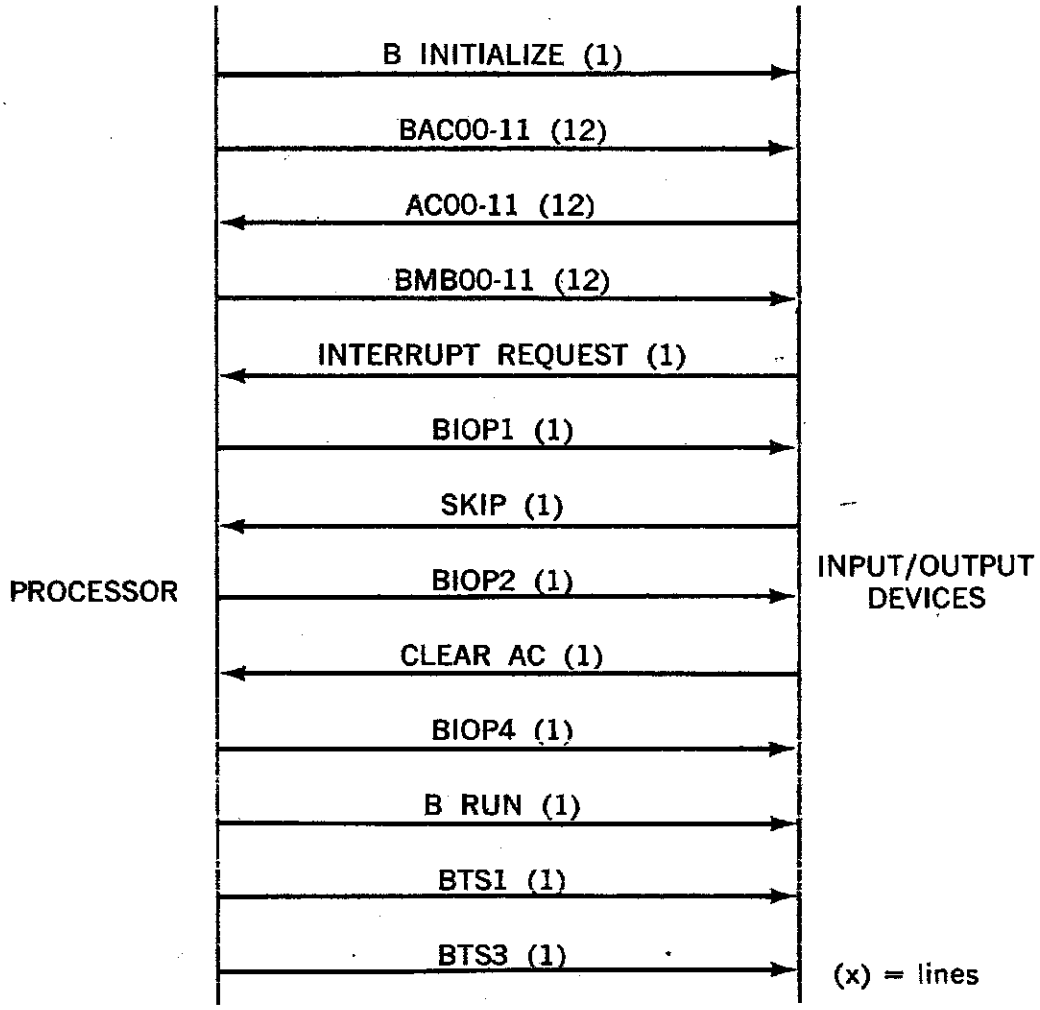


Figure 12-1 PDP-8/I and PDP-8/L PROGRAMMED I/O BUS

Table 12-2
Programmed I/O Bus, PDP-8/I and PDP-8/L

LINE	DESCRIPTION
B Initialize	This line is asserted when the processor is initially powered up or when the start key is depressed. Usually (Output) performs housekeeping on all peripheral devices. Example: Reset all flip-flops on power up.
AC00-11	These lines carry information from the peripheral device to the accumulator. (Input)
BAC 00-11	These lines carry information from the accumulator to the peripheral devices. (Output)
BMB 00-11	These lines carry the device identifier code, a unique address to which only one unique device will respond. They also carry data out of the computer during a break cycle. (Output)
INTERRUPT REQUEST	This line is activated by the device flag and, when asserted, causes the processor to JMS to location 0 of memory field 0 and disable the interrupt system. (Input)
BIOP 1	This line, when active, is ordinarily used to test device flags. (Output)
SKIP	This line when active during an IOP, will set the Skip flip-flop in the processor.
BIOP 2	This line, when active, is ordinarily used to clear the device flag and/or cause the device to operate. (Output)
CLEAR AC	This control line, when asserted, changes the mode of I/O input transfer to a jam transfer. (Input)
BIOP 4	When active, is ordinarily used to effect data transferred to or from the peripheral devices. (Output)
B Run	When active, signals peripheral devices that the processor is executing instructions. (Output)
BTS1 and BTS3	These lines are used to sync the peripheral devices, particularly break devices, to the processor. (Output)

NOTE 1: All lines coming from the peripheral devices to the processor are considered inactive when a voltage potential is applied to them. All lines are loaded within the processor.

NOTE 2: Most lines going to the peripheral devices from the processor are considered inactive when there is no potential applied to them.

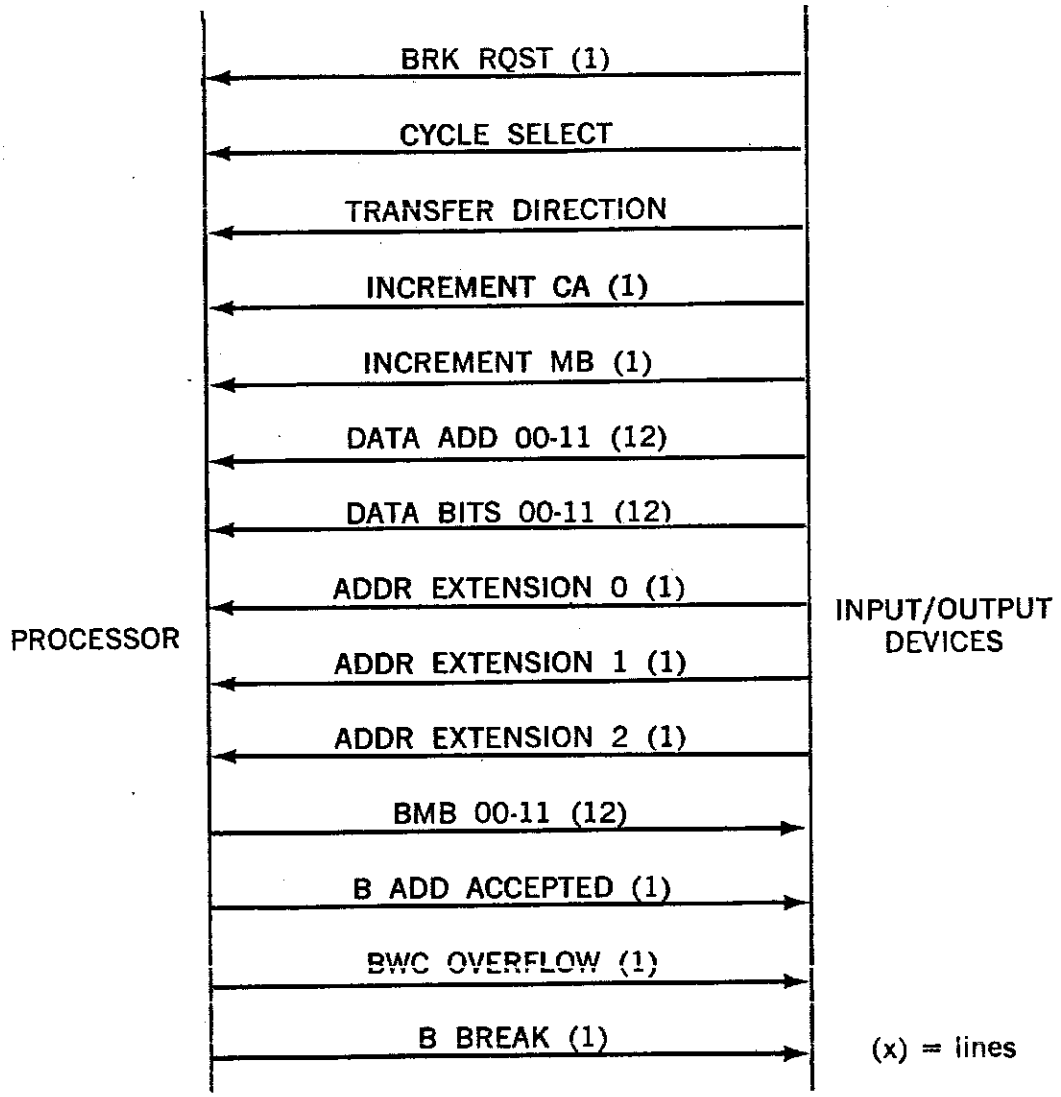


Figure 12-2 PDP-8/I and PDP-8/L Data Break Bus

Table 12-3
Data Break Bus, PDP-8/I and PDP-8/L

LINE	DESCRIPTION
BRK RQST	When at 0V, indicates to the processor that a data break is being requested. (Input)
CYCLE SELECT	Informs the processor that either a single cycle data break (voltage potential), or a 3-cycle data break (0V) is requested. (Input)
TRANSFER DIRECTION	This line, when at a voltage potential, indicates a read operation and, when at 0V, indicates a write operation. (Input)
INCREMENT CA	When at 0V, this line will not allow the current address counter to be incremented. (Useful when searching for file numbers.) (Input)
INCREMENT MB	When at 0V, this line allows the memory buffer contents to be incremented by 1. Useful in pulse height analysis. (Input)
DATA ADD 00-11	These lines are pre-wired at the peripheral for 3-cycle data break or are driven by buffers from a flip-flop register for a 1-cycle data break. They specify the address of Word Count (3-cycle break) or the Break address (1 cycle break). Ground (0V) = 1. (Input)
DATA BITS 00-11	These lines are connected to the MB and insert information into a memory location. Line conditions are the same as AC 00-11. (Input)
BMB 00-11	These lines are connected to the MB and contain information extracted from a memory location. (Input)
B ADDRESS ACCEPTED	When active, this line contains a pulse beginning at TP4 of the first cycle of a data break. It signifies to the peripheral that the computer has recognized the break request. (Output)
BWC OVERFLOW	When active, this line indicates to the I/O device that the word count location in memory has become 0. The line becomes active at TP2 of the word count cycle and lasts to the end (TP4) of the current machine cycle to indicate to the I/O device to stop data transfers. A pulse also appears on this line if overflow occurs on a memory buffer increment. (Output)
B BREAK	When active, indicates to the device that the Break cycle (the last of the 3 cycles when a 3-cycle break is requested) has started. The device then sends or receives data information, depending on the direction of the transfer. (Output)
ADDR EXTENSION lines 0, 1, and 2	When the MC8/I Memory Extension Control and the Data Break Facility are used, these three bits serve as a 3-bit address extension. [Only one of these lines is used by the PDP-8/L.] (Input)

Input/Output Signal Connections

Tables 12-4 through 12-9 list the cable connections for the PDP-8/I and PDP-8/L computers. Tables 12-4 and 12-5 list the input and output (respectively) cable connections for the negative bus PDP-8/I. Assertion of the signals are represented by the symbol column in the tables.

Tables 12-6 and 12-7 list the input and output (respectively) cable connections for the positive bus PDP-8/I; and tables 12-8 and 12-9 list the input and output (respectively) cable connections for the PDP-8/L.

Table 12-4
PDP-8/I Negative Bus Input Signals

Signal	Symbol	Interface Connection	Module Terminal	Module Type
AC00		J05D2	J13A1	M506
AC01		E2	D2	M506
AC02		H2	F1	M506
AC03		K2	K2	M506
AC04		M2	M1	M506
AC05		P2	R2	M506
AC06		S2	J14A1	M506
AC07		T2	D2	M506
AC08		V2	F1	M506
AC09		J06D2	K2	M506
AC10		E2	M1	M506
AC11		H2	R2	M506
SKIP		K2	J15A1	M506
Interrupt Request		M2	D2	M506
Clear AC		P2	F1	M506
Data Add 00 (1)		J07D2	J16A1	M506
Data Add 01 (1)		E2	D2	M506
Data Add 02 (1)		H2	F1	M506
Data Add 03 (1)		K2	K2	M506
Data Add 04 (1)		M2	M1	M506
Data Add 05 (1)		P2	R2	M506
Data Add 06 (1)		S2	J17A1	M506
Data Add 07 (1)		T2	D2	M506
Data Add 08 (1)		V2	F1	M506
Data Add 09 (1)		J08D2	K2	M506
Data Add 10 (1)		E2	M1	M506
Data Add 11 (1)		H2	R2	M506

Table 12-4
PDP-8/I Negative Bus Input Signals (cont.)

Signal	Symbol	Interface Connection	Module Terminal	Module Type
BRK RQST		K2	J15M1	M506
Transfer Direction		*M2	J15R2	M506
Increment MB		T2	J20A1	M506
Data Bit 00 (1)		J09D2	J18A1	M506
Data Bit 01 (1)		E2	D2	M506
Data Bit 02 (1)		H2	F1	M506
Data Bit 03 (1)		K2	K2	M506
Data Bit 04 (1)		M2	M1	M506
Data Bit 05 (1)		P2	R2	M506
Data Bit 06 (1)		S2	J19A1	M506
Data Bit 07 (1)		T2	J19D2	M506
Data Bit 08 (1)		V2	F1	M506
Data Bit 09 (1)		D2	K2	M506
Data Bit 10 (1)		E2	M1	M506
Data Bit 11 (1)		**H2	R2	M506
Cycle Select		K2	J20D2	M506
Increment CA		M2	F1	M506
Addr Extension 0		V2/J11H2	K2	M506
Addr Extension 1		T2/J11E2	M1	M506
Addr Extension 2		S2/J11D2	R2	M506
Line MUX		J06V2	J15K2	M506

*Direction is into PDP-8/I when signal is −3V, out of PDP-8/I when ground potential (0V).

**3-cycle when at ground potential.
1-cycle when at −3V level.

Table 12-5
PDP-8/I Negative Bus Output Signals

Signal	Symbol	Interface Connection	Module Terminal	Module Type
BAC 00 (1)		J01D2	H07D2	M651
BAC 01 (1)		E2	K2	M651
BAC 02 (1)		H2	S2	M651
BAC 03 (1)		K2	H08D2	M651
BAC 04 (1)		M2	K2	M651
BAC 05 (1)		P2	S2	M651
BAC 06 (1)		S2	H09D2	M651
BAC 07 (1)		T2	K2	M651
BAC 08 (1)		V2	S2	M651
BAC 09 (1)		J02D2	H10D2	M651
BAC 10 (1)		E2	K2	M651
BAC 11 (1)		H2	S2	M651
BIOP1		K2	H11D2	M651
BIOP2		M2	K2	M651
BIOP4		P2	S2	M651
BTS3		S2	H12D2	M651
BTS1		T2	K2	M651
B Initialize		V2	S2	M651
BMB 00 (1)		J03D2	H13D2	M651
01 (1)		E2	K2	M651
02 (1)		H2	S2	M651
03 (0)		K2	H14D2	M651
03 (1)		M2	K2	M651
04 (0)		P2	S2	M651
04 (1)		S2	H15D2	M651
05 (0)		T2	K2	M651
05 (1)		V2	S2	M651
06 (0)		J04D2	H16D2	M651
06 (1)		E2	K2	M651
07 (0)		H2	S2	M651
07 (1)		K2	H17D2	M651
08 (0)		M2	K2	M651
08 (1)		P2	S2	M651
09 (1)		S2	H18D2	M651
10 (1)		T2	K2	M651
BMB 11 (1)		V2	S2	M651
B Break		J08P2	H20D2	M651
B Add Accepted		J08S2	H20K2	M651
BWC Overflow (0)		J10P2	H19S2	M651
B Run		J06S2	H19D2	M651
BTTINST		J06T2	H19K2	M651

Table 12-6.
Positive 8/I Bus Input Signals

(NOTE: All of these signals are asserted at ground.)

Signals	Interface Connection	Module Terminal	Module Type
AC 00 BUS	J05D2	J13A1	M516
AC 01 BUS	J05E2	J13D2	M516
AC 02 BUS	J05H2	J13F1	M516
AC 03 BUS	J05K2	J13K2	M516
AC 04 BUS	J05M2	J13M1	M516
AC 05 BUS	J05P2	J13R2	M516
AC 06 BUS	J05S2	J14A1	M516
AC 07 BUS	J05T2	J14D2	M516
AC 08 BUS	J05V2	J14F1	M516
AC 09 BUS	J06D2	J14K2	M516
AC 10 BUS	J06E2	J14M1	M516
AC 11 BUS	J06H2	J14R2	M516
SKIP BUS	J06K2	J15A1	M516
INT RQST BUS	J06M2	J15D2	M516
AC CLEAR CONT. BUS	J06P2	J15F1	M516
LINE MUX	J06V2	J15K2	M516
Data Add 00	J07D2	J16A1	M516
Data Add 01	J07E2	J16D2	M516
Data Add 02	J07H2	J16F1	M516
Data Add 03	J07K2	J16K2	M516
Data Add 04	J07M2	J16M1	M516
Data Add 05	J07P2	J16R2	M516
Data Add 06	J07S2	J17A1	M516
Data Add 07	J07T2	J17D2	M516
Data Add 08	J07V2	J17F1	M516
Data Add 09	J08D2	J17K2	M516
Data Add 10	J08E2	J17M1	M516
Data Add 11	J08H2	J17R2	M516
BRK RQST	J08K2	J15M1	M516
DATA IN	J08M2	J15R2	M516
MB INCREMENT	J08T2	J20A1	M516
DATA 00	J09D2	J18A1	M516
DATA 01	J09E2	J18D2	M516
DATA 02	J09H2	J18F1	M516
DATA 03	J09K2	J18K2	M516
DATA 04	J09M2	J18M1	M516
DATA 05	J09P2	J18R2	M516
DATA 06	J09S2	J19A1	M516
DATA 07	J09T2	J19D2	M516
DATA 08	J09V2	J19F1	M516
DATA 09	J19D2	J19H2	M516
DATA 10	J10E2	J19M1	M516
DATA 11	J10H2	J19P2	M516
3 CYCLE	J10K2	J20D2	M516
CA INCREMENT	J10M2	J20F1	M516
EXT DATA ADD 0	J11H2, J10V2	J20K2	M516
EXT DATA ADD 1	J11E2, J10T2	J20M1	M516
EXT DATA ADD 2	J11D2, J10S2	J20R2	M516

Table 12-7
Positive 8/I Bus Output Signals

(NOTE: All of these signals are asserted at +3V.)

Signals	Interface Connection	Module Terminal	Module Type
BAC 00 (1)	J01D2	H07D2	M661
BAC 01 (1)	J01E2	H07K2	M661
BAC 02 (1)	J01H2	H07S2	M661
BAC 03 (1)	J01K2	H08D2	M661
BAC 04 (1)	J01M2	H08K2	M661
BAC 05 (1)	J01P2	H08S2	M661
BAC 06 (1)	J01S2	H09D2	M661
BAC 07 (1)	J01T2	H09K2	M661
BAC 08 (1)	J01V2	H09S2	M661
BAC 09 (1)	J02D2	H10D2	M661
BAC 10 (1)	J02E2	H10K2	M661
BAC 11 (1)	J02H2	H10S2	M661
BIOP1 (1)	J02K2	H11D2	M660
BIOP2 (1)	J02M2	H11K2	M660
BIOP4 (1)	J02P2	H11S2	M660
BTS3 (1)	J02S2	H12D2	M660
BTS1 (1)	J02T2	H12K2	M660
B INITIALIZE 1, 2	J02V2, J08V2	H12S2	M660
BMB 00 (1)	J03D2	H13D2	M661
BMB 01 (1)	J03E2	H13K2	M661
BMB 02 (1)	J03H2	H13S2	M661
BMB 03 (0)	J03K2	H14D2	M661
BMB 03 (1)	J03M2	H14K2	M661
BMB 04 (0)	J03P2	H14S2	M661
BMB 04 (1)	J03S2	H15D2	M661
BMB 05 (0)	J03T2	H15K2	M661
BMB 05 (1)	J03V2	H15S2	M661
BMB 06 (0)	J04D2	H16D2	M661
BMB 06 (1)	J04E2	H16K2	M661
BMB 07 (0)	J04H2	H15S2	M661
BMB 07 (1)	J04K2	H17D2	M661
BMB 08 (0)	J04M2	H17K2	M661
BMB 08 (1)	J04P2	H17S2	M661
BMB 09 (1)	J04S2	H18D2	M661
BMB 10 (1)	J04T2	H18K2	M661
BMB 11 (1)	J04V2	H18S2	M661
B RUN (0)	J06S2	H19D2	M661
B TT INST	J06T2	H19K2	M661
B WC OVERFLOW (0)	J10P2	H19S2	M661
B ADD ACCEPTED (0)	J09S2	H20K2	M661
B BREAK (0)	J09P2	H20D2	M661

Table 12-8
Positive 8/L Bus Input Signals

(NOTE: All of these signals are asserted at ground.)

Signals	Interface Connection	Module Terminal	Module Type
AC 00 BUS	D34B1	D29K2, D13E2	M906, M111
AC 01 BUS	D34D1	D29M2, D13H2	M906, M111
AC 02 BUS	D34E1	D29P2, D13K2	M906, M111
AC 03 BUS	D34H1	D29S2, D13M2	M906, M111
AC 04 BUS	D34J1	A32A1	M516
AC 05 BUS	D34L1	A32D2	M516
AC 06 BUS	D34M1	A32F1	M516
AC 07 BUS	D34P1	A32K2	M516
AC 08 BUS	D34S1	A32M1	M516
AC 09 BUS	D34D2	A32R2	M516
AC 10 BUS	D34E2	A33A1	M516
AC 11 BUS	D34H2	A33D2	M516
I/O SKIP	D34K2	A33F1	M516
INT RQST	D34M2	A33K2	M516
AC CLEAR	D34P2	A33M1	M516
DATA ADD 00	C36B1	B32B1, D11P1	M906, M113
DATA ADD 01	C36D1	B32D1, D11P2	M906, M113
DATA ADD 02	C36E1	B32E1, D11T2	M906, M113
DATA ADD 03	C36H1	B32H1, B12J1	M906, M111
DATA ADD 04	C36J1	B32J1, D13A1	M906, M111
DATA ADD 05	C36L1	B32L1, D13J1	M906, M111
DATA ADD 06	C36M1	B32M1, D13P2	M906, M111
DATA ADD 07	C36P1	B32P1, D13S2	M906, M111
DATA ADD 08	C36S1	B32S1, D13U2	M906, M111
DATA ADD 09	C36D2	B32D2, A34A1	M906, M111
DATA ADD 10	C36E2	B32E2, A34C1	M906, M111
DATA ADD 11	C36H2	B32H2, A34D1	M906, M111
BRK RQST	C36K2	D29T2, BIIT2	M906, M216
DATA IN	C36M2	B32M2, AIOM2	M906, M160
MEMORY INCREMENT	C36T2	A33R2	M516
DATA 00	C35B1	B33B1, A34F1	M906, M111
DATA 01	C34D1	B33D1, A34J1	M906, M111
DATA 02	C34E1	B33E1, A34L1	M906, M111
DATA 03	C34H1	B33H1, A34N1	M906, M111
DATA 04	C34J1	B33J1, A34R1	M906, M111
DATA 05	C35L1	B33L1, A34V1	M906, M111
DATA 06	C34M1	B33M1, A34E2	M906, M111
DATA 07	C34P1	B33P1, A34H2	M906, M111
DATA 08	C34S1	B33S1, A34K2	M906, M111
DATA 09	C34D2	B33D2, A34M2	M906, M111
DATA 10	C34E2	B33E2, A34P2	M906, M111
DATA 11	C34H2	B33H2, A3452	M906, M111
3 CYCLE	C34K2	B33K2, A34U2	M906, M111
CA INCREMENT	C34M2	B33M2, A12L2	M906, M160

Table 12-9
Positive 8/L Bus Output Signals

(NOTE: All of these signals are asserted at +3V.)

Signals	Interface Connection	Module Terminal	Module Type
BAC 00 (1)	D36B1	D29B1, D27D1	M906, M623
BAC 01 (1)	D36D1	D29D1, D27E1	M906, M623
BAC 02 (1)	D36E1	D29E1, D27K1	M906, M623
BAC 03 (1)	D36H1	D29H1, D27L1	M906, M623
BAC 04 (1)	D36J1	D29J1, D27R1	M906, M623
BAC 05 (1)	D36L1	D29L1, D27S1	M906, M623
BAC 06 (1)	D36M1	D29M1, D27H2	M906, M623
BAC 07 (1)	D36P1	D29P1, D27J2	M906, M623
BAC 08 (1)	D36S1	D29S1, D27N2	M906, M623
BAC 09 (1)	D36D2	D29D2, D27P2	M906, M623
BAC 10 (1)	D36E2	D29E2, D27U2	M906, M623
BAC 11 (1)	D36H2	D29H2, D27V2	M906, M623
BIOP1 (1)	D36K2	C29D2	M660
BIOP2 (1)	D36M2	C29K2	M660
BIOP4 (1)	D36P2	C29S2	M660
BTS3 (1)	D36S2	C30D2	M660
BTS1 (1)	D36T2	C30K2	M660
B INITIALIZE-1	D36V2	C30S2	M660
B INITIALIZE-2	C36V2	B32V2, C27V2	M906, M623
BMB 00 (1)	D35B1	D30B1, D28D1	M906, M623
BMB 01 (1)	D35D1	D30D1, D28E1	M906, M623
BMB 02 (1)	D35E1	D30E1, D28K1	M906, M623
BMB 03 (0)	D35H1	D30H1, D28L1	M906, M623
BMB 03 (1)	D35J1	D30J1, D28R1	M906, M623
BMB 04 (0)	D35L1	D30L1, D28S1	M906, M623
BMB 04 (1)	D35M1	D30M1, D28H2	M906, M623
BMB 05 (0)	D35P1	D30P1, D28J2	M906, M623
BMB 05 (1)	D35S1	D30S1, D28N2	M906, M623
BMB 06 (0)	D35D2	D30D2, D28P2	M906, M623
BMB 06 (1)	D35E2	D30E2, D28U2	M906, M623
BMB 07 (0)	D35H2	D30H2, D28V2	M906, M623
BMB 07 (1)	D35K2	D30K2, C27D1	M906, M623
BMB 08 (1)	D35M2	D30M2, C27E1	M906, M623
BMB 08 (1)	D35P2	D30P2, C27K1	M906, M623
BMB 09 (1)	D35S2	D30S2, C27L1	M906, M623
BMB 10 (1)	D35T2	D30T2, C27R1	M906, M623
BMB 11 (1)	D35V2	D30V2, C27S1	M906, M623
B RUN	D34S2	D29V2, C27H2	M906, M623
B WC OVERLOW (0)	C35P2	B33P2, C27N2	M906, M623
B ADD ACCEPTED (0)	C36S2	B32S2, C27V2	M906, M623
B BREAK (0)	C36P2	B32P2, C27P2	M906, M623

INPUT/OUTPUT CABLES

The PDP-8/I uses two basic types of I/O buses, positive and negative. The basic PDP-8/I has a negative I/O bus, but can be converted to a positive I/O bus by adding the optional KA8/IB Positive I/O Bus option. The PDP-8/L has a positive I/O bus only.

Overall bus lengths should be as short as possible; bus (cable) lengths are defined on page 248 of this chapter.

The negative-bus PDP-8/I has an 11-cable interface, 6 cables for programmed I/O devices and 5 cables for the Data Break Devices. These cables are single-sided connector cables carrying 9 signals each (see Figure 12-3a).

The only difference between the negative bus PDP-8/I and the positive bus PDP-8/I is that the latter contains only 4 single-sided-connector cables for Data Break Devices. Since the basic computer is the same, when the system is wired for positive bus, the fifth cable output is jumpered to the last of the 4 cable connectors. Interfacing cables for the positive bus PDP-8/I are single-sided-to-double-sided wye cables. The double-sided connectors are used on the I/O device and the single-sided ends, are inserted in the computer interface connectors (see Figure 12-3b). Table 12-10 lists the part numbers for the single-sided-to-double-sided interface cables.

The PDP-8/L differs from both versions of the PDP-8/I in that the former's interface connector cables are double-sided-to-double-sided. Thus, programmed I/O devices require 3 double cables and 2 double cables are required for Data Break Devices (see Figure 12-3c). Table 12-11 lists the part numbers for the double-sided-to-double-sided interface cables.

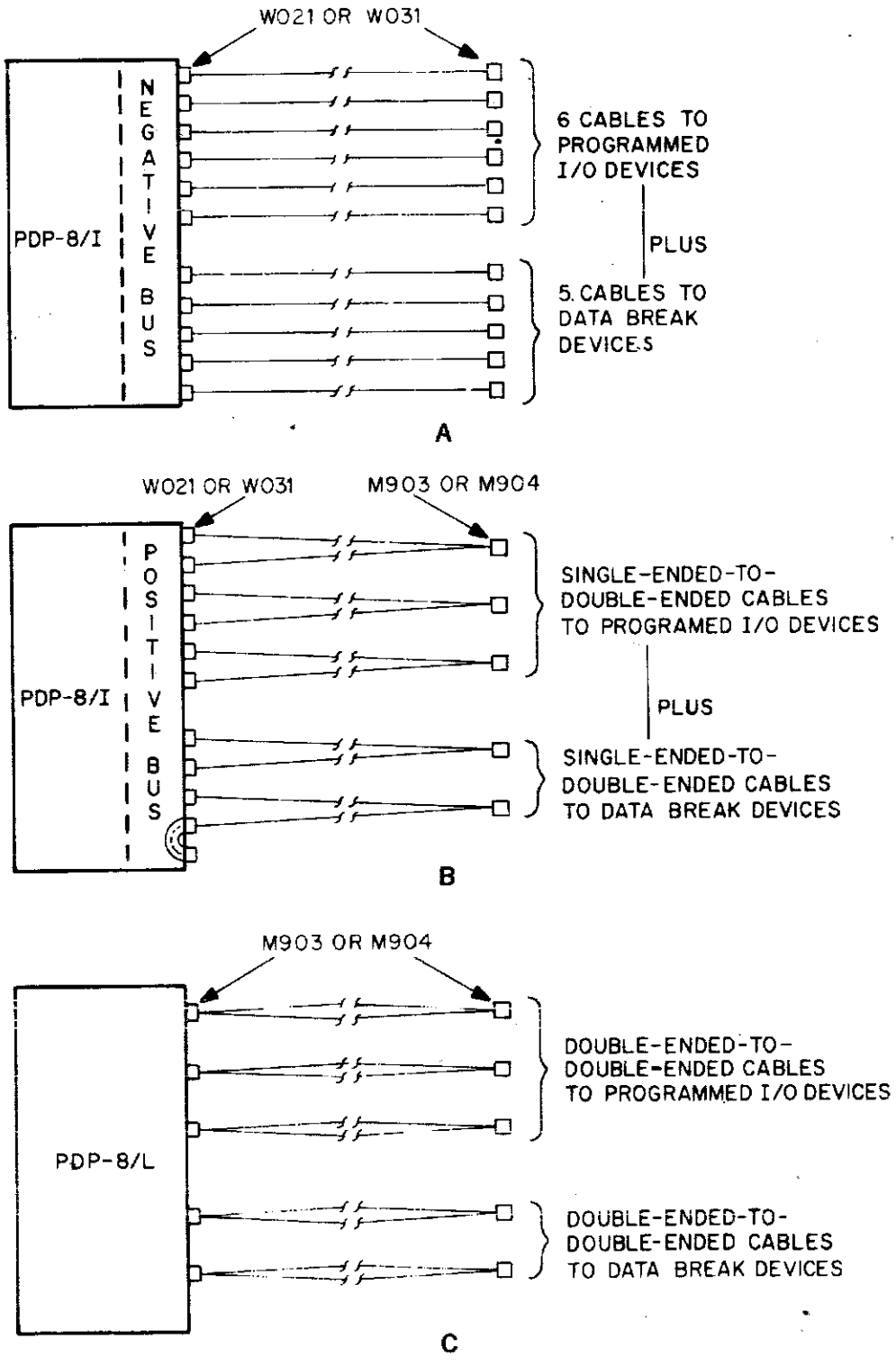


Figure 12-3. Interface Cable Connection Configuration Diagram

Table 12-10
Single-sided-to-Double-sided Interface Cables

Cable Length	Coax No.	Shielded Mylar No.
1 ft.	BC08D-1	BC08C-1
2 ft.	BC08D-2	BC08C-2
3 ft.	BC08D-3	BC08C-3
4 ft.	BC08D-4	BC08C-4
5 ft.	BC08D-5	BC08C-5
6 ft.	BC08D-6	BC08C-6
7 ft.	BC08D-7	BC08C-7
8 ft.	BC08D-8	BC08C-8
9 ft.	BC08D-9	BC08C-9
10 ft.	BC08D-10	BC08C-10

Table 12-11
Double-sided-to-double-sided Interface Cables

Cable Length	Coax No.	Shielded Mylar No.
1 ft.	BC08B-1	BC08A-1
2 ft.	BC08B-2	BC08A-2
3 ft.	BC08B-3	BC08A-3
4 ft.	BC08B-4	BC08A-4
5 ft.	BC08B-5	BC08A-5
6 ft.	BC08B-6	BC08A-6
7 ft.	BC08B-7	BC08A-7
8 ft.	BC08B-8	BC08A-8
9 ft.	BC08B-9	BC08A-9
10 ft.	BC08B-10	BC08A-10
3.5 ft.		BC08A-3A

SECTION 12-2 INTERFACE DESIGN

GENERAL CABLING RULES AND SUGGESTIONS

Round and flat coaxial cable are electrically interchangeable and may be intermixed in a system. Round coax is preferable for interconnecting free-standing cabinets, since it is more resistant to abuse from using personnel (e.g., computer operators, programmers, technicians, etc.)

Indiscriminate intermixing of shielded mylar and coaxial cable is not advised for use on PDP-8 family computers. For consistency and economy, Digital Equipment Corporation recommends all cables be shielded mylar (mylar cables with alternate solid mylar), unless they are used to interconnect free-standing cabinets or to gain maximum length. However, no more than one change from mylar to coax or from coax to mylar is permitted over the length of a bus.

Table 12-12 lists the various equipment and the associated cable restrictions/information

Table 12-12 Equipment Cabling Restrictions/Information

EQUIPMENT	RESTRICTION/INFORMATION
1. Data Break Bus	The maximum length of coaxial cable that may be used is 30 feet. The maximum length of shielded mylar cable that may be used is 25 feet.
2. Programmed I/O Bus	The maximum length of coaxial cable that may be used is 50 feet. The maximum length of shielded mylar cable that may be used is 45 feet.
3. DM01 Data Channel Multiplexer	When inserted on the data break bus, the DM01 represents 10 feet of cable; only 20 feet of coaxial cable may be used or only 15 feet of shielded mylar cable may be used for the remaining total bus length.
4. DM04 Data Channel Multiplexer	When inserted on the data break bus, the DM04 represents 5 feet of cable; only 25 feet of coaxial cable may be used or only 20 feet of shielded mylar cable may be used for the remaining total bus length.

NOTE

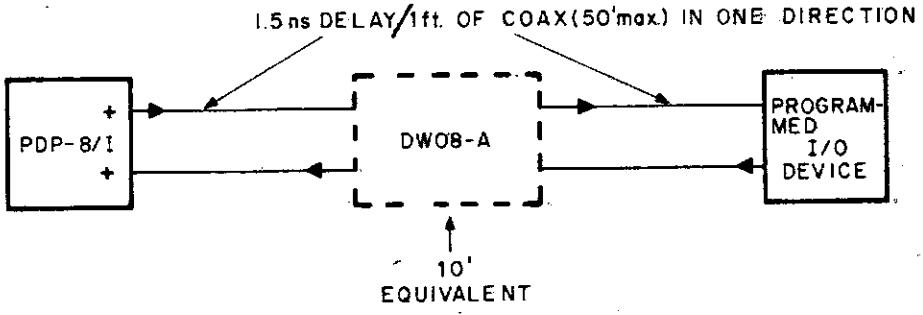
These restrictions for the DM01 and DM04 refer to the sum cable lengths from the processor to the DM and from the DM to the most distant data break device

5. DW08 (A or B) I/O Converter Panel

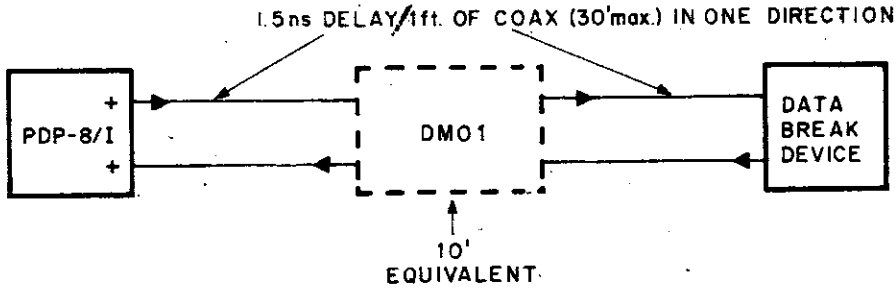
When inserted on the I/O bus, the DW08 represents 10 feet of cable. The system now has two buses, and for one of these buses (the one originating in the computer), the maximum length of coax or shielded mylar is as stated in 1 and 2 above.

For the converted bus, the sum of the length of cable from the computer to the DW08, and from the DW08 to the most distant peripheral is 10 feet shorter than the figures stated in 1 and 2 above.

GENERAL NOTE: In order to avoid time delay problems, the maximum cable lengths given must not be exceeded. Peripheral gating time from IOP to SKIP, IOP to AC input signals, and IOP to AC CLEAR should be limited to 100 ns. Figure 12-4 illustrates a computation of a complete sample system, computing the time delays encountered from the computer through the cables and devices to the peripherals.



Since DW08-A is used with programmed I/O device, max. cable length = 50' ∴ 40' of cable + 10' of equivalent = 50', @ 1.5 ns = 75 ns of delay. ∴ Time from transmission of IOP to receipt of IOP skip back into the computer is approximately 150 ns.



Since DM01 and Data Break device are used, max. cable length = 30' ∴ 20' of cable + 10' of equivalent = 30', @ 1.5 ns = 45 ns of delay. ∴ Time from transmission of IOP to receipt of IOP skip back into the computer is approximately 90 ns.

Figure 12-4 Computation of Sample System Cabling

INTERFACE TIMING CRITERIA

Interface timing criteria must be determined when interfacing a peripheral device to a PDP-8/I or PDP-8/L. There are certain basic premises involved with interrupt processing which must be understood. These are:

a. The interrupt feature must be turned on via the ION instruction in order for the device to be allowed to interrupt the processor.

b. In order to honor the interrupt, the central processor must have completed the instruction it is presently doing.

c. When an interrupt request is honored, the hardware of the machine will force a JMS instruction to location 0 in memory field 0 and will also disable the interrupt system.

d. An interrupt servicing routine must be resident in memory and the starting address of this routine must be defined in the memory location immediately following location 0.

The longest time required to honor an interrupt request is approximately the time duration of the slowest instruction. Thus, for a PDP-8/I without the EAE option (or for the PDP-8/L), this time would be 4.5 μs: (the time required to complete a 3-cycle instruction) and for the PDP-8/I with the EAE option, the time would be 11 μs. These times assume an interrupt request was made just after the processor entered the Fetch state.

Two examples are provided to illustrate the use of interface times:

Example 1 PDP-8/I without EAE option

50.0 μs	Time between interrupts
$\underline{-4.5}$ μs	Maximum processor time before interrupt
45.5 μs	
$\underline{-1.5}$ μs	for hardware JMS to location 0
44.0 μs	Maximum time allowed for servicing before possible errors arise.

Example 2 PDP-8/I with EAE option

50.0 μs	Time between interrupts
$\underline{-9.0}$ μs	Maximum processor time before interrupt (with EAE option installed) (24-bit long shift)
41.0 μs	
$\underline{-1.5}$ μs	for hardware JMS to location 0
39.5 μs	Maximum time allowed for servicing before possible errors arise.

Break synchronization timing may be calculated similarly, using the following numbers:

Machine	Max. time before Address Accepted	Instruction Sequence Producing This Time
PDP-8/I without EAE	8.05 μs	IOT followed by 3-cycle instruction
PDP-8/L	8.35 μs	IOT followed by 3-cycle instruction
PDP-8/I with EAE	15.8 μs	Two sequential long shifts of 24 bits each

LOADING RULES

In the design and construction of interface circuits the following must be taken into consideration: fan-out, fan-in, grounding, time delays, cabling, and clamping.

Fan-out is the output loading, in terms of unit loads, a gate is capable of driving. Typical fan-out is ten unit loads. A unit load is defined as follows: For the logical zero state 1.6 mA (maximum); for the logical one state 40 mA (maximum).

Fan-in is the input loading, in terms of unit loads, caused by a gate on the output of the source driving it. Typical fan-in (input loading) is one unit load.

Grounding. A good ground system is very essential to reliable logic operation. In systems using digital-to-analog interface the digital system ground should be connected to the analog system ground at a single point, often at the analog-to-digital interface and a good earth ground provided at this point. Sometimes conduit grounding is sufficient but it is generally safer, and oftentimes necessary, to ground the system to a good earth ground through #4 gage copper wire, or equivalent, to a steel beam or a cold water pipe.

Time Delays and Cabling. DEC logic generates waveforms with rising edges containing frequencies of over 100 MHz. At these frequencies the inductance, mutual inductance, capacitance, and transmission line properties of I/O busses and interface cabling become noticeable. To avoid potential problems, the following rules and guides are provided.

a. The propagation delay of typical wiring is 1.5 ns/ft (4.5 ns/m). Although this delay is usually small in comparison with gate delays, it is often significant when overshoot and reflections are considered.

b. The current carrying capacity of a wire is only V/Z_0 until the wave (change) has propagated along the wire three times (4.5 ns/ft, 13.5 ns/m). Typical wiring has a characteristic impedance (Z_0) of approximately 150 ohms, so that the current available at the end of the wire for rising waveforms is only 20 mA until reflections propagated, regardless of the source current available.

c. The inductance and capacitance of wiring combine to produce high frequency ringing on the transitions of waveforms. This ringing can be controlled by either resistively terminating the line with approximately 100 ohms, if the circuit will drive it, or with the DEC level terminator circuit incorporated into the G796, G704, and other modules.

d. The mutual inductance and capacitance of the wiring also causes high-frequency cross-talk which may produce false operation of the logic. This can be reduced by one of the following ways: minimizing the number of high-frequency signal components, by clipping or clamping high-frequency ringing with a level terminator circuit; or by wiring with short wires and/or twisted pairs, thereby reducing coupling.

Clamping is used to prevent the excursion of the output or input voltages beyond certain predetermined limits. This is sometimes necessary to prevent false triggering of gates.

I/O BUS MODULES

Computer I/O bus modules consist of the M506, M651, M111, M906, M516, M660, M661, M113, and M623 modules. Interface signals to the PDP-8/I (negative bus) computer use the M506 module; interface signals to the PDP-8/I (positive bus) use the M516 module; and interface signals to the PDP-8/L use either a combination of M111 and M906 modules or the M516 module. Interface signals from the PDP-8/I (negative bus) computer use the M651 module; interface signals from the positive-bus PDP-8/I computer use either the M660 or M661 modules; and interface signals from the PDP-8/L computer originate from a combination of M623 and M906 modules for data signals and M660 modules for timing signals.

M506 Negative Input Converter (See Figure 12-5) — The M506 inverter module accomplishes level conversion from the negative potential received from the peripheral devices to the positive potential required for the positive logic functioning within the PDP-8/I. This module contains six non-inverting signal converters which are used to interface the negative logic levels or pulses to positive logic levels of +3V and ground.

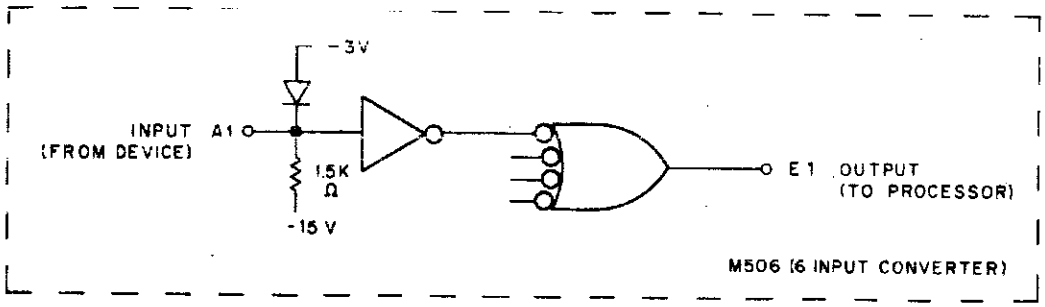


Figure 12-5 Typical M506 Negative Input Converter

M651 Negative Output Converter (see Figure 12-6) — The M651 contains three non-inverting signal converters which are used to interface the positive logic levels or pulses to DEC negative logic levels of −3V and ground. These converters provide current drive at a low output impedance so that unterminated cables or wires can be driven with a minimum of ringing and reflections. A positive AND condition at the input gate produces a ground output.

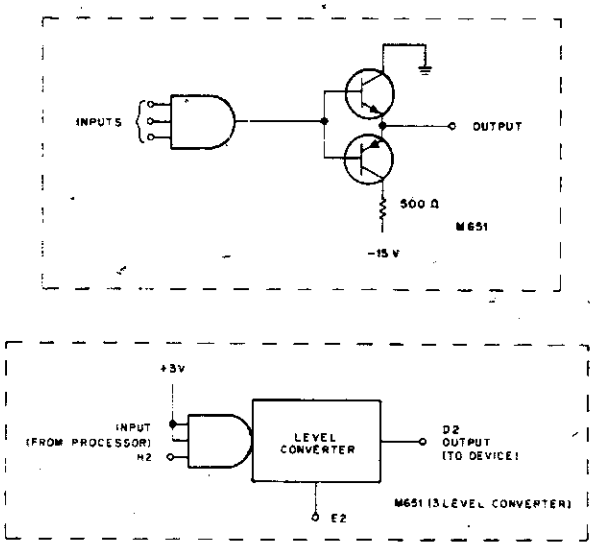


Figure 12-6 Typical M651 Negative Output Converter

M111/M906 and M113/M906 Positive Input Circuits (See Figure 12-7) — The M111 or M113 Inverter module is used in conjunction with the M906 Cable Terminator module which clamps the input to prevent excursions beyond +3V and ground in the PDP-8/L. The M906 also provides the pull-up resistors to +5V.

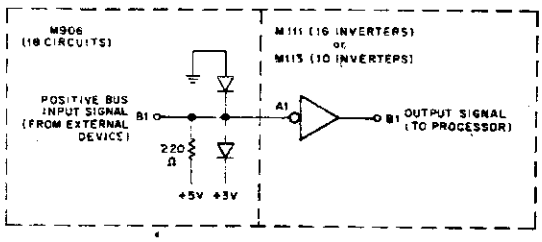


Figure 12-7 Typical M111/M906 Positive Input Circuit

M516 Positive Bus Receiver Input Circuit (See Figure 12-8) — Six 4-input NAND gates with overshoot and undershoot clamp on one input of each gate. Pull-up resistors connected to +5V are also provided.

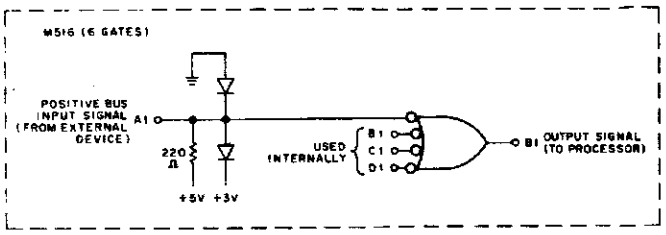


Figure 12-8 Typical M516 Positive Bus Receiver Input Circuit

M623/M906 Positive Output Circuit (See Figure 12-9) — The M623 Bus Driver module contains 6 2-input AND gates bus drivers for driving of the positive input bus. Used in conjunction with the M906 Cable Terminator module, the output is clamped to prevent excursions beyond +3V and ground. Output can drive +5 mA at the high level and sink 20 mA at the low level (for the PDP-8/L only).

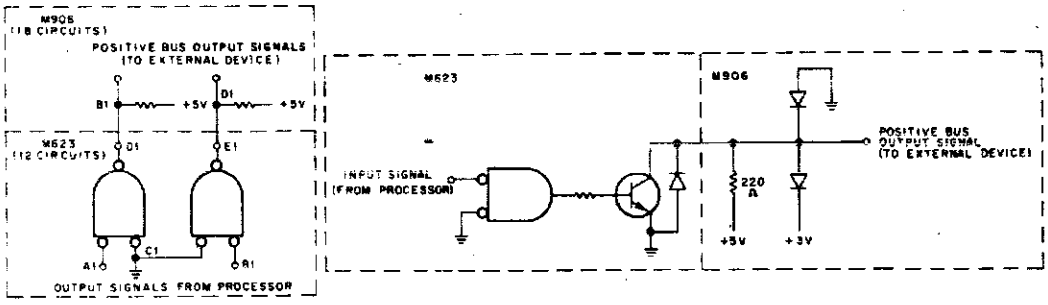


Figure 12-9 Typical M623/M906 Positive Output Circuit

M660 Positive Level Driver (See Figure 12-10) — The M660 Cable Driver consists of three circuits, each of which can drive 100-ohm terminated cable with M Series levels or pulses whose duration is greater than 100 ns. The output can drive +5 mA at the high level and sink 20 mA at the low level, in addition to termination current required by the G717 termination module. The M660 module is used in the PDP-8/I (positive bus) and the PDP-8/L for the following output signals: BIOP1, BIOP2, BIOP4, BTS3, BTS1, and B INITIALIZE [B INITIALIZE-1].

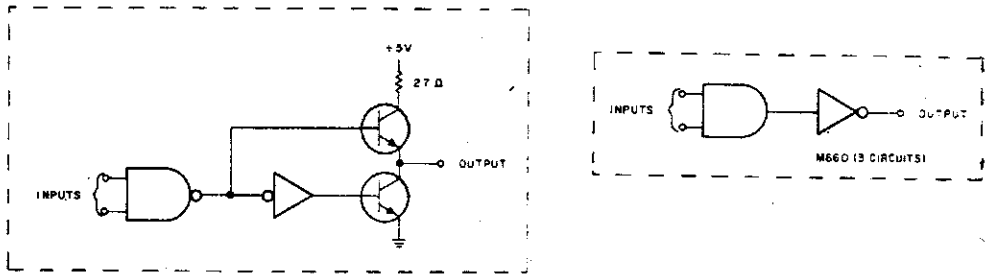


Figure 12-10 Typical M660 Positive Level Driver

M661 Positive Level Driver (ee Figure 12-11) — This circuit consists of a transistor and load resistor capable of sinking 20 mA to ground and capable of supplying 5 mA at +3V. It must not be terminated. The M661 module is used in the positive bus PDP-8/I to interface all output signals other than those handled by the M660 module (See Table 12-8).

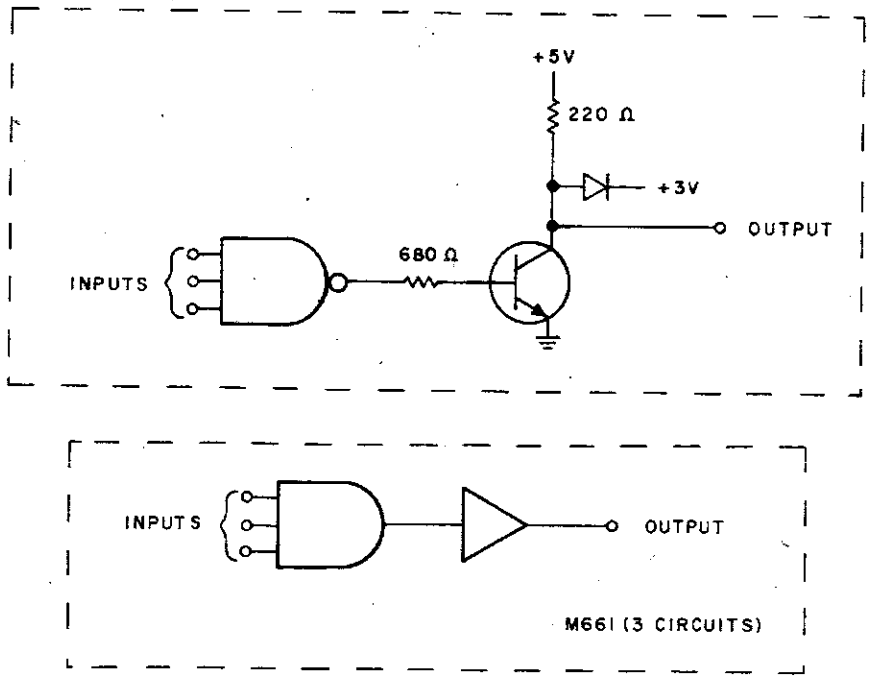


Figure 12-11 Typical M661 Positive Level Driver

PERIPHERAL EQUIPMENT INTERFACE MODULES.

Two FLIP CHIP modules are of particular interest in the design of equipment to interface to the positive bus computer. Complete details on these and other FLIP CHIP modules can be found in the Digital Logic Handbook.

M103 Device Selector — (See Figure 12-12)

The M103 selects an input/output device according to the code in the instruction word (being held in the memory buffer during the IOT cycle). M103 module includes diode protection clamps on input lines so that it may be used directly on the positive bus computer.

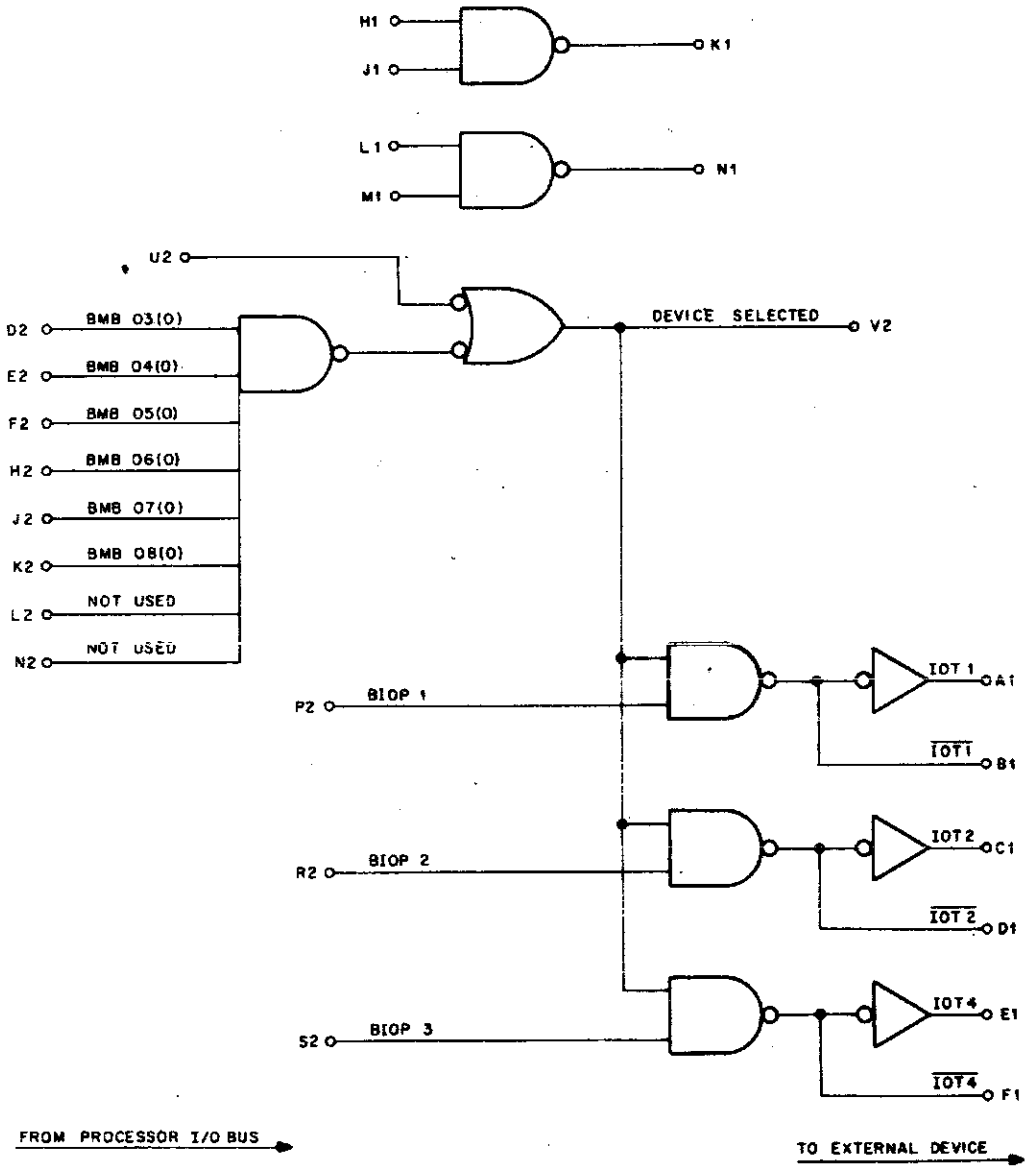


Figure 12-12 M103 Device Selector Logic Circuit

M101 — Bus Data Interface — (See Figure 12-13)

Fifteen two-input NAND gates with one input of each gate tied to a common line. For use in strobing data off of the positive bus computer I/O bus. M101 module includes diode protection clamps on input lines so that it may be used directly on the positive bus computer positive bus.

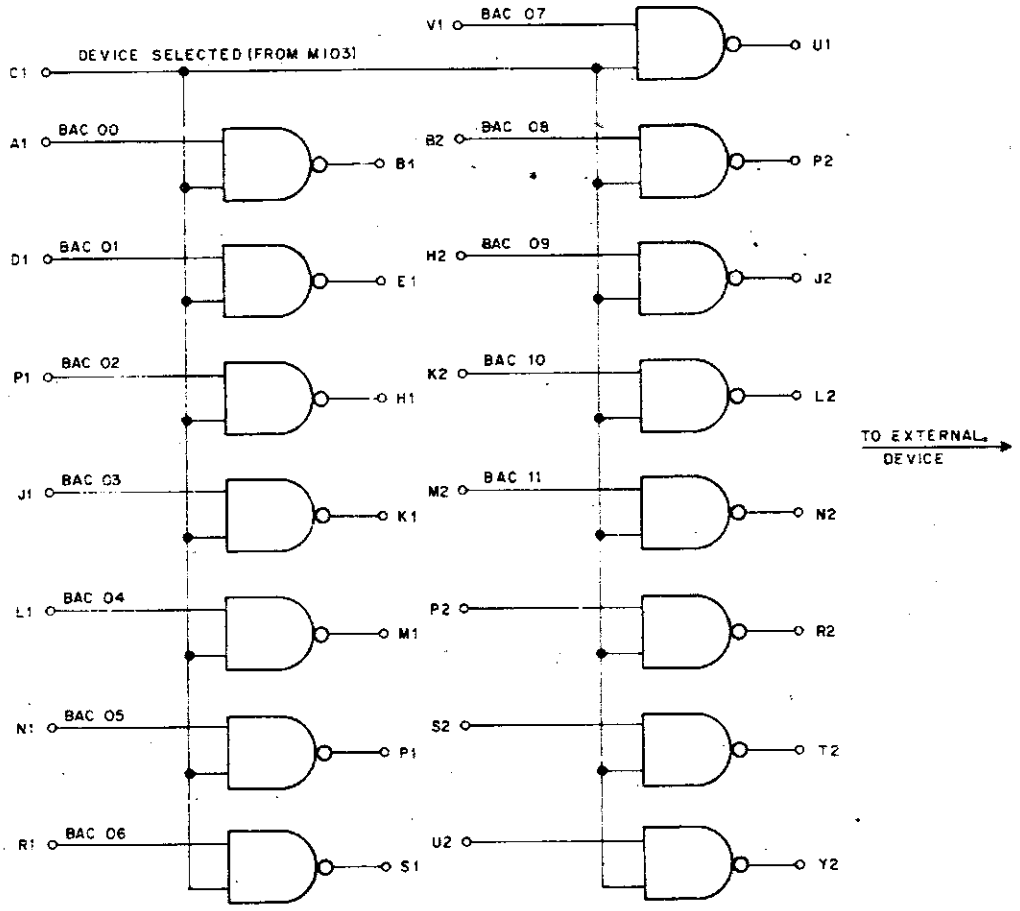


Figure 12-13 M101 — Bus Data Interface Logic Circuit

M Series Module Summary

The following is a list of M Series modules available from Digital Equipment Corporation that can be used in designing special interfaces and special devices. The majority of these modules are described in the Digital Logic Handbook. For those that cannot be found in the Handbook, contact the nearest Digital representative.

M Series Module Summary

Type	Function	Description
M002	15 Loads	15 +3 V sources each capable of driving 10 unit loads. Can be used for tying off unused inputs.
M040	Solenoid Driver	Output ratings of −70 V and 0.6 A allow these 2 drivers to be used with a variety of medium current loads.
M050	50 ma Indicator and Driver	Output ratings of −20 V and 50 mA. Allow any of the 12 circuits on this module to drive a variety of incandescent lamps. These drivers can also be used as slow speed open collector PNP level shifters to −3 V system.
M101	Bus Data Interface	15 two-input NAND gates with one input of each gate tied to a common line. For use in strobing data from the PDP-8/I or PDP-8/L I/O bus. Pins are compatible with M111.
M103	Device Selector	Similar to W103 only designed for use with PDP-8/I and PDP-8/L options. Output pulses are not regenerated but only buffered.
M111	Inverter	16 inverter circuits with a fan-in of 1 unit load and fan-out of 10 unit loads.
M112	NOR Gate	10 positive NOR gates with a fan-in of 1 unit load and fan-out of 10 unit loads.
M113	10 2-Input NAND Gates	10 2-input positive NAND gates with a fan-in of 1 unit load and fan-out of 10 unit loads.
M115	8 3-input NAND Gates	8 3-input positive NAND gates with a fan-in of 1 unit load and a fan-out of 10 unit loads.
M117	6 4-input NAND Gates	6 4-input positive NAND gates with a fan-in of 1 unit load and a fan-out of 10 unit loads.
M119	3 8-Input NAND Gates	3 8-input positive NAND gates with a fan-in of 1 unit load and a fan-out of 10 unit loads.
M121	AND/NOR Gates	6 gates which perform the positive logic function AB + CD. Fan-in on each input is 1 unit load and gate fan-out is 10 unit loads.
M141	NAND/OR Gates	12 2-input positive NAND gates which can be used in a wired OR manner. Gates are grouped in a 4-4-3-1 configuration with a fan-in of 1 unit load and a fan-out which depends on the number of gates ORed together.
M160	Gate Module	3 general purpose multi-input gates which can be used for system input selection. Fan-in is 1 unit load and fan-out is 10 unit loads.
M161	Binary to Octal/Decimal Decoder	A binary to 8-line or BCD to 10-line decoder. Gating is provided so that up to 6 binary bits can be decoded using only M161's. Accepts a variety of BCD codes.
M161	Parity Circuit	2 circuits each of which can be used to generate even or odd parity signals for four bits of binary input.
M169	Gating Module	4 circuits that can be used for input selection. Each circuit is of an AND/OR configuration with 4 2-input AND gates.

Type	Function	Description
M202	Triple J. K. Flip-Flop	3 J. K. flip-flops with multiple input AND gates on J and K. Versatile units for many control or counter purposes. All direct set and clear inputs are available on module pins.
M203	Set-Reset Flip-Flops	8 single-input set-reset flip-flops for use as buffer storage. Each circuit has a fan-in of 1 unit load and a fan-out of 10 unit loads.
M204	Counter-Buffer	4 J-K flip-flops which can be interconnected as a ripple or synchronous counter or used as general control elements.
M206	Six Flip-Flops	6 D-type flip-flops which can be used in shift registers, counters, buffer registers, and general purpose control functions.
M207	Flip-Flops	6 single-input J- and K-type flip-flops for use in shift registers, ripple counters, and general purpose control functions.
M208	Buffer Shift Register	An internally connected 8-bit buffer or shift register. Provisions are made for gated single-ended parallel load, bipolar parallel output, and serial input.
M211	Binary Up/Down Counter	A 6-bit binary up/down ripple counter with control gates for direction changes via a single control line.
M212	6-Bit Shift Register	An internally connected left-right shift register. Provisions are made for gated single-ended parallel load, bipolar parallel output, and serial input.
M213	BCD Up/Down Counter	One decade of 8421 up or down counting is possible with this module. Provisions are made for parallel loading, bipolar output, and carry features.
M230	Binary to BCD Shift Register Converter	One decade of a modified shift register which allows high speed conversion (100 ns per binary bit) of binary data to 8421 BCD code. System use of this module requires additional modules.
M302	One Shot Delay	2-pulse or level-triggered one-shot delays with output delay adjustable from 50 ns to 7.5 ms. Fan-in is 2 unit loads and fan-out is 25 unit loads.
M310	Delay Line	Fixed tapped delay line with delay adjustable in 50 ns increments from 50 ns to 500 ns. Two digital output amplifiers and one driver are included.
M360	Variable Delay	Continuously variable delay line with a range of 50 ns to 500 ns. Module includes delay line drivers and digital output amplifiers.
M401	Clock	A gateable RC clock with both positive and negative pulse outputs. The output frequency is adjustable from 10 MHz to below 100 Hz.
M405	Crystal Clock	Stable system clock frequencies from 1 kHz to 10 MHz are available with this module. Frequency drift at either the positive or negative pulse output is less than 0.01% of the specified frequency.
M410	Reed Clock	A stable low frequency reed control clock similar to the M452. Stability in the range 10°C to 70°C is better than 0.15%. For use with communications systems and available with only standard teletype and data set frequencies.

Type	Function	Description
M452	Variable Clock	Provides 880Hz, 440Hz, and 220Hz square waves necessary for clocking and M706 and M707 in a 110-baud teletype system.
M501	Schmitt Trigger	Provides regenerative characteristics necessary for switch filtering, pulse shaping, and contact closure sensing. This circuit can be AND/OR expanded.
M502	Negative Input Converter	Pulses as short as 35 ns can be level shifted from −3 V systems to standard M Series levels by the two circuits in this converter. This module can also drive low impedance terminated cables.
M506	Negative Input Converter	This converter will level shift pulses as short as 100 ns from −3 V systems to M Series levels. Each of the 6 circuits on this module provide a low impedance output for driving unterminated long lines.
M507	Bus Converter	6 inverting level shifters which accept −3 V and GND, as inputs and have an open collector NPN transistor at the output. Output rise is delayed by 100 ns for pulse spreading.
M516	Positive Bus	6 4-input NOR gates with overshoot and undershoot clamps on one input of each gate. In addition, one input of each gate is tied to +3 V with the lead brought out to a connector pin.
M602	Pulse Generator	The 2 pulse amplifiers in this module provide standard 50 ns or 110 ns pulses for M Series systems.
M617	6-4 Input NOR Buffers	6 4-input positive NOR gates with a fan-in of 1 unit load and a fan-out of 30 unit loads.
M627	Power Amplifier Module	6 4-input high speed positive NAND gates with a fan-in of 2.5 unit loads and a fan-out of 40 unit loads.
M650	Negative Output	The 3 non-inverting level shifters on this module can be used to interface the positive levels or pulses (duration greater than 100 ns) of K and M Series to −3 V logic systems.
M652	Negative Output Converter	These two circuits provide high-speed, non-inverting level shifting for pulses as short as 35 ns or levels from M Series to −3 V systems. The output can drive low impedance terminated cables.
M660	Positive Level Driver	3 circuits provide low impedance, 100-ohm, terminated cable driving capability, using M Series levels or pulses of duration greater than 100 ns. Output drive capability is 50 mA at +3 V or Ground.
M661	Positive Level Driver	3 circuits provide low-impedance unterminated cable driving. Characteristics are similar to M660 with the exception that +3 V drive is 5 mA.
M730	8/I Bus Positive Output Interfacer	General-purpose positive bus output module for use in interfacing many positive level (0 to +20 V) systems to the PDP-8/I or PDP-8/L. Module includes device selector, 12 bit parallel output buffer, and adjustable timing pulses.
M731	8/I Bus Negative Output Interfacer	Identical to M730 except outputs are level shifted for 0 to −20 V systems to the PDP-8/I or PDP-8/L. Module includes device selector, 12 bit parallel input buffer, and adjustable timing pulses.

Type	Function	Description
M732	8/I Bus Positive Input Interfacer	General purpose positive bus input module for use in interfacing many positive level (0 to +20 volt) systems to the PDP-8/I or 8/L. Module includes device selector, 12 bit parallel input buffer, and adjustable timing pulses.
M733	8/I Bus Negative Input Interfacer	Identical to M732 except inputs are level shifted from negative voltage systems.
M901	Flexprint Cable Connector	Double-sided 36 pin shielded mylar cable connector. All pins are available for signals or grounds. Pins A2, B2, U1 and V1 have 10 Ω resistors in series.
M902	Resistor Terminator	Double-sided 36 pin terminator module with 100 Ω terminations on signal leads. Alternate ground are provided as in the M903 and M904.
M903	Connector	Double-sided 36 pin shielded mylar cable connector with alternate grounds for I/O bus cables.
M906	Cable Terminator	18 load resistors clamped to prevent excursions beyond +3 V and Ground. It may be used in conjunction with the M623 to provide cable driving ability.

SECTION 12-3 HARDWARE DETAILS

This portion of the chapter provides the interface designer with information on design procedures, module layout, wiring, and cable selection. Additional help may be obtained from local DEC sales offices.

I/O CABLING

The PDP-8/I, PDP-8/L was designed to provide the user maximum ease and flexibility in implementing special interfaces. External devices and interfaces are constructed and mounted outside of the basic machine, thereby eliminating the necessity for modifications to the basic processor. All signals to and from the computer are carried on either coaxial or shielded mylar cables.

Figure 12-14 illustrates the cable configurations for typical systems. For the negative bus PDP-8/I, six cables are used for programmed I/O and interrupt cable connections in (or out). Five additional cables are used for a total of 11, when Data Break devices are implemented. For positive bus PDP-8/I computers, six cables are also used for programmed I/Os; however, only 4 data break cables are required. Wye-type cables are connected to the first peripheral, thus the bus becomes identical to that of the PDP-8/L after the first peripheral. For the PDP-8/L, three dual cables are used for program interrupt cable connections (in or out). Two additional dual cables are used for a total of five, when Data Break devices are implemented.

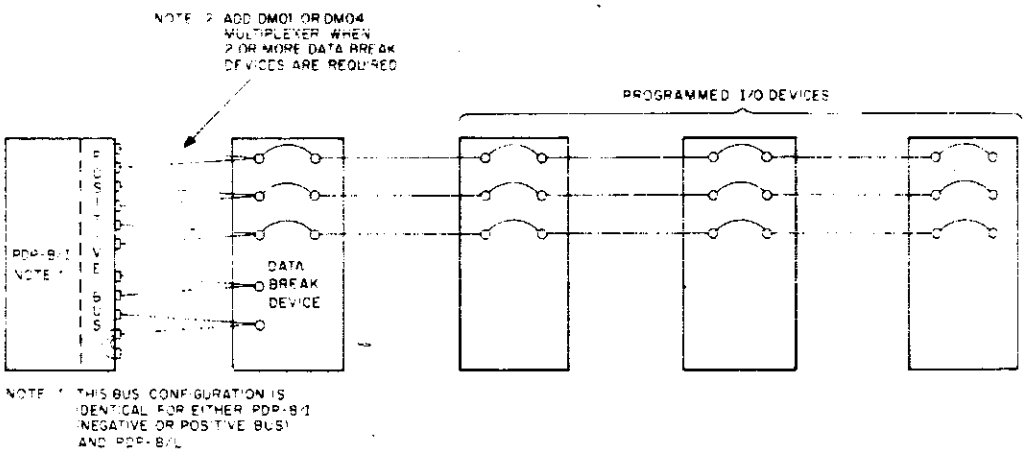


Figure 12-14 Typical System I/O Cabling

MODULE LAYOUT

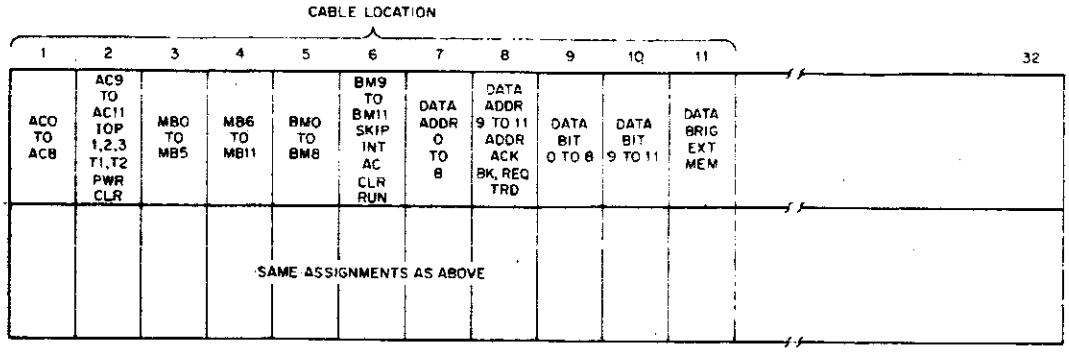
In general, module layout is based on the functional elements within a system and is primarily a matter of common sense. Since wire length could be critical, layout considerations are advisable. Digital Equipment Corporation has, however, layout conventions for I/O cabling to extend devices. The interface designer may wish to use these conventions as a guide. The general rule is DO NOT DEAD-END THE I/O BUS. This means that parallel connections should always be made at each device to handle possible future expansion.

Negative Bus Devices

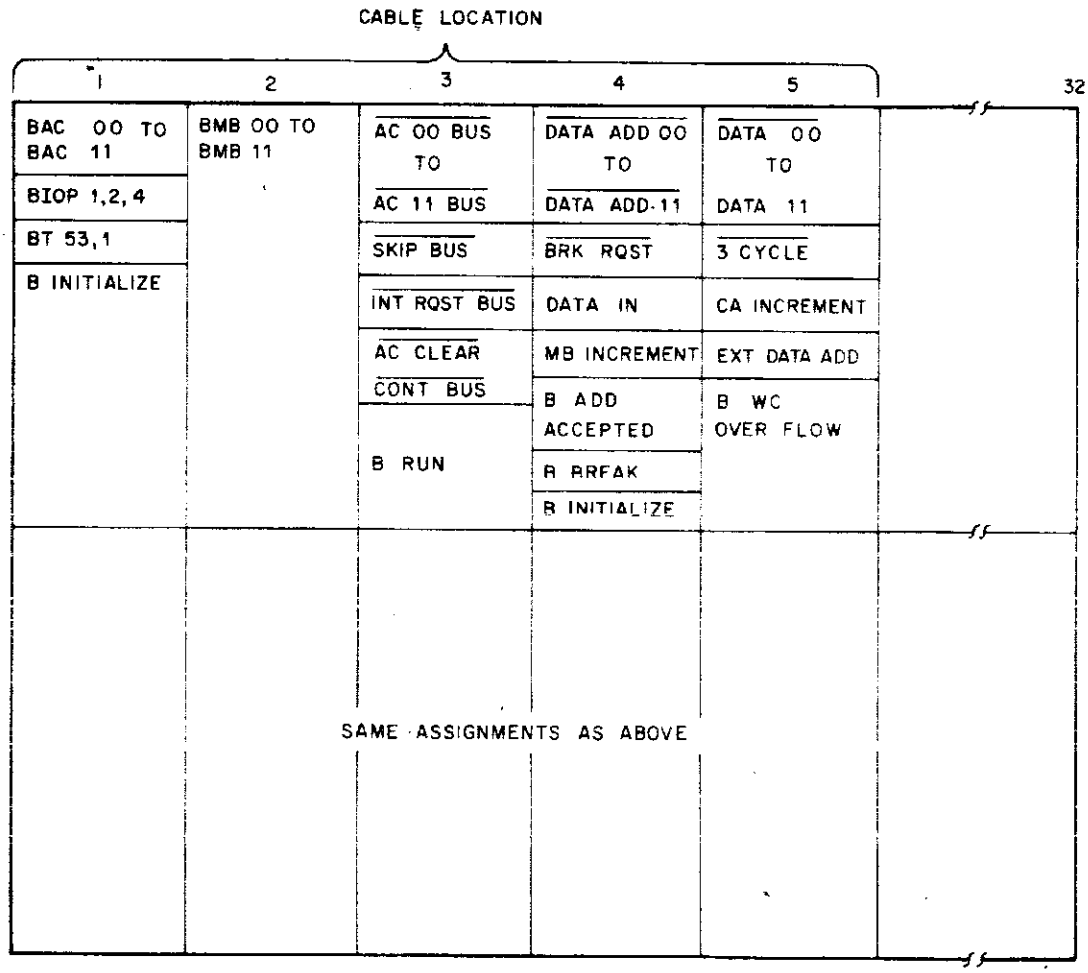
Module locations 1 through 6 (looking at wiring pin side) in an option mounting panel are reserved for program interrupt cable connections in (or out). Data Break information is assigned to locations 7 through 10, with location 11 available for data break use with extended memory. The lower mounting of the optional mounting panel also has slots 1 through 11 reserved, exactly as previously mentioned, for cable in (or out). Cable assignments are shown in Figure 12-15a.

Positive Bus Devices

Module slots 1 through 5 in the bottom half of the option mounting panel are wired in parallel with the top module slot locations 1 through 5. To continue the I/O cabling to the next device, the bottom slots are used and the I/O cable connections are exactly the same as mentioned above. Cable assignments are shown in Figure 12-15b.



(a) For Negative Bus Devices



(b) For Positive Bus Devices

Figure 12-15 I/O Cable Assignments

Cable Selection — Two types of cables are recommended for I/O interface connections.

The first is 9 conductor coaxial cable. This cable protects systems from radiated noise and cross talk between individual lines. Coax cable used and sold by Digital Equipment Corporation has the following nominal specs:

Z = 95 ± 5 Ω
C° = 13.75 pf/foot approx. (unterminated)
L = 124 nH/foot approx.
R = 0.095 Ω/foot nominal
V_P = 79% of velocity of light, approx. (1.5 nsec/ft.)

This cable is available in either flat or round form.

The second cable type is 19 conductor (9 signals and 10 grounds), #30 gauge flat copper shielded mylar.

CONNECTOR SELECTION

Of the many connectors available, several have particular application to I/O connectors. Price and ordering information is available on these and other connectors in the Digital Logic Handbook. Of particular interest are the M903 and M904 types described in the subsequent paragraphs.

a. M903 Connector — Double-sided 36-pin Shielded Mylar cable connector with alternate grounds for I/O bus cables. (Two Shielded Mylar cables are utilized with this connector module.)

b. M904 Connector — Double-sided 36-pin coaxial cable connector with alternate grounds for I/O bus cables. (Two coax cables are utilized with this connector module).

(1) Signals:
B1, D1, E1, H1, J1, L1, M1, P1, S1,
D2, E2, H2, K2, M2, P2, S2, T2, V2

(2) Grounds:
A1, C1, F1, K1, N1, R1, T1,
C2, F2, J2, L2, N2, R2, U2

SIGNAL TERMINATING

Termination is required on programmed I/O cables longer than 20 feet, and may be desirable on shorter cables. For negative bus, use 220-ohm shunt resistors to ground on IOP1, IOP2, IOP4, BTS1, BTS3, and INITIALIZE. No special termination module exists for negative bus. For positive bus, 100 ohms to ground on the same lines should be used. (A G717 module accomplished this and should be inserted at the end of the bus on cable number 1.) If two buses are present in a machine, they are electrically independent, and must be separately terminated.

WIRING HINTS

These suggestions may help reduce mounting panel wiring time. They are not intended to replace any special wiring instructions given on individual module data sheets or in application notes. For fast, neat wiring, the following order is recommended:

a. All power wiring (Pins A2, B2, C2, T1) and any horizontally bussed signal wiring. Use Horizontal Bussing Strips, Type 933. Minus 15V should be supplied to pin B only if that voltage is required by the module in question.

b. Vertical grounding wires interconnecting each chassis ground with Pins C2 & T1 grounds. Run these wires from the uppermost mounting panel to the bottom panel. On the first and last blocks of the mounting panel, connect the grounds to the chassis.

c. All other ground wires. Connect pin C2 of each module to T1 of each module, to C2 of the next module down, etc., making connections to all other pins to be grounded along the way. The usual practice of concern over ground loops should be totally disregarded. At the frequencies involved in digital logic, many parallel paths are of utmost importance. Do not forget that connections should be made to the ground pins on the signal connections.

d. Wire all signal wires in convenient order. Point-to-point wiring produces the shortest wire lengths, goes in fastest, is easiest to trace and change, and generally results in better appearance and performance than cabled wiring. Point-to-point wiring is strongly urged.

The recommended wire size for use with the H803 mounting blocks and H911 mounting panel is #30. Larger or smaller wire may be used depending on the number of connections to be made to each lug. Solid wire and a heat resistant insulation (Kynar) is recommended. The H803 mounting blocks are only available with wire wrap pins which necessitates the use of a wire wrap tool. (Digital can supply #30 gauge wire in 1000-foot rolls.)

ADEQUATE GROUNDING IS ESSENTIAL. IN ADDITION TO THE CONNECTIONS BETWEEN MOUNTING PANELS MENTIONED ABOVE, THERE MUST BE CONTINUITY OF GROUNDS BETWEEN CABINETS AND BETWEEN THE LOGIC ASSEMBLY AND ANY EQUIPMENT WITH WHICH THE LOGIC COMMUNICATES.

When wire wrapping is done on a mounting panel containing modules, the wire wrap tool must be grounded except when all modules are removed from the mounting panel. This procedure must be followed, because even with tools isolated from the ac power line, such as those operated by batteries or compressed air, static charge may build to sufficient amplitudes where damage to semiconductors may result.

COOLING

The low power consumption of M Series modules results in a total of about 15 W dissipation in a typical H911 mounting panel containing 64 modules. Convection cooling is sufficient for a few mounting panels, but forced air cooling should be used when a very large system is built.

IOT ALLOCATIONS FOR PDP-8/I AND PDP-8/L

IOT	OPTION
00	Interrupt
01	High Speed Reader Type PR8
02	High Speed Punch Type PP8
03	Teletype Keyboard/Reader
04	Teletype Teleprinter/Punch
05	Displays, Types VC8 and KV8
06	Displays, Types VC8 and KV8
07	Displays, Type VC8, KV8
10	Memory Parity Option MP8 and Power Fail Option KP8
11	Teletype Systems Type PT08 and DC02
12	Teletype Systems Type PT08 and DC02
13	Real Time Clock Type KW8
14	
15	
16	
17	
20	Memory Extension Control Option Type MC8 and Time Sharing Option KT8
21	Memory Extension Control Option Type MC8
22	Memory Extension Control Option Type MC8
23	Memory Extension Control Option Type MC8
24	Memory Extension Control Option Type MC8
25	Memory Extension Control Option Type MC8 and Time Sharing Option KT8
26	Memory Extension Control Option Type MC8 and Time Sharing Option KT8
27	Memory Extension Control Option Type MC8 and Time Sharing Option KT8
30	Analog-to-Digital Converter/Scope Control Type AX08
31	Analog-to-Digital Converter/Scope Control Type AX08
32	Analog-to-Digital Converter/Scope Control Type AX08
33	Analog-to-Digital Converter/Scope Control Type AX08
34	Analog-to-Digital Converter/Scope Control Type AX08
35	Analog-to-Digital Converter/Scope Control Type AX08
36	Analog-to-Digital Converter/Scope Control Type AX08
37	Analog-to-Digital Converter/Scope Control Type AX08
40	Teletype System Type PT08 and 680I Communications System
41	Teletype System Type PT08 and 680I Communications System
42	Teletype System Type PT08 and 680I Communications System
43	Teletype System Type PT08 and 680I Communications System
44	Teletype System Type PT08 and 680I Communications System
45	Teletype System Type PT08
46	Teletype System Type PT08
47	Teletype System Type PT08
50	Incremental Plotter Type VP8
51	Incremental Plotter Type VP8
52	Incremental Plotter Type VP8
53	General Purpose A/D Converters and Multiplexers, Types AF01A, AF02A, AF03A, AD08, AM08, AM02A, AM03A and AF04A Scanning Digital Voltmeter
54	General Purpose A/D Converters and Multiplexers, Types AF01A, AF02A, AF03A, AD08, AM08, AM02A, AM03A and AF04A Scanning Digital Voltmeter

IOT	OPTION
55	D/A Converter Types AA01A and AA054A
56	D/A Converter Types AA01A and AA054A
57	D/A Converter Types AA01A and AA054A, Sample and Hold Control Type AC01A and AF04A Scanning Digital Voltmeter
60	Random Access Disk File and Control Types DF32 and RF08, and Synchronous Modem Interface Type DP01AA
61	Random Access Disk File and Control Types DF32 and RF08, and Synchronous Modem Interface Type DP01AA
62	Random Access Disk File and Control Types DF32 and RF08, and Synchronous Modem Interface Type DP01AA
63	Card Reader Type CR8 and Maintenance IOTs for DF32 Disk File
64	Synchronous Modem Interface Type DP01AA
65	Synchronous Modem Interface Type DP01AA and Line Printer and Control Type 645
66	Synchronous Modem Interface Type DP01AA and Line Printer and Control Type 645
67	Card Reader Type CR8 and Synchronous Modem Interface Type DP01AA
70	Automatic Mag Tape Type TC58 and Modem Interface Connector Panel Type DC08F
71	Automatic Mag Tape Type TC58 and Modem Interface Connector Panel Type DC08F
72	Automatic Mag Tape Type TC58 and Modem Interface Connector Panel Type DC08F
73	Automatic Mag Tape Type TC58 and Modem Interface Connector Panel Type DC08F
74	Automatic Mag Tape Type TC58 and Modem Interface Connector Panel Type DC08F
75	
76	DECtape Controls TC01 and TC08
77	DECtape Controls TC01 and TC08

*Indicates IOTs used by basic machine.

**Avoid use of IOT 6201 for any purpose other than a CDF instruction. This IOT is used by the Binary Loader program, even if no extended memory is installed on the machine.

CHAPTER 13
INSTALLATION AND PLANNING

SPACE REQUIREMENTS

Adequate space must be provided at the installation site to accommodate the PDP-8/I [PDP-8/L] and related peripheral equipment and to allow access to all doors and panels for maintenance.

All notations enclosed in brackets ([ ]) indicate data for the PDP-8/L Computer only.

The PDP-8/I is available only in a rack mounted configuration. This configuration and related peripherals can be purchased completely installed in DEC cabinets or unmounted for installation in the user's cabinet.

Figure 13-1 illustrates the rack mounted model PDP-8/I. Figure 13-2 illustrates the rack mounted PDP-8/L. The PDP-8/L is also available in a table-top model (see Figure 13-3).

Detailed mounting information is included for PDP-8/I installation in standard BUD and EMCOR racks (see Figures 13-4 and 13-5).

The standard ASR33 Teletype requires floor space approximately 22¼ in. wide by 18½ in. deep. Signal cable length restricts the location of the teletype to within 8 ft. of the side of the computer.

ENVIRONMENTAL REQUIREMENTS

Ambient temperature at the installation site can vary between 50°F and 130°F (between 10°C and 55°C) with no adverse effect on computer operation. To extend the life expectancy of the system, however, it is recommended that ambient temperature be maintained between 70°F and 85°F (between 21°C and 30°C). Humidity is from 10% to 90% without condensation.

During shipping or storing of the system, the ambient temperature may vary between −4°F and 150°F (0°C and 65°C). Although all exposed surfaces of all DEC cabinets and hardware are treated to prevent corrosion, prolonged exposure to extreme humidity should be avoided.

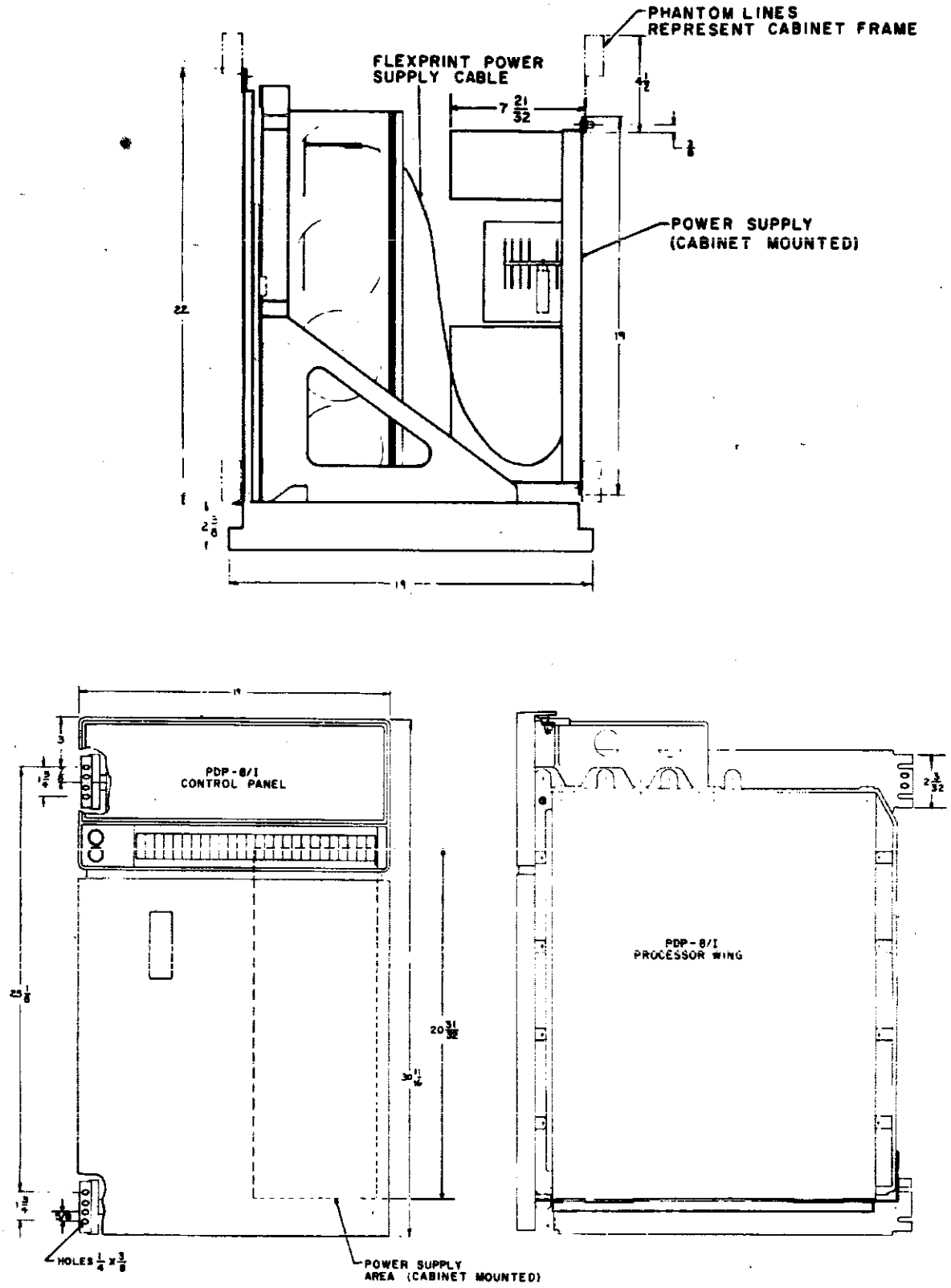


Figure 13-1 Rack mounted Model of PDP-8/I

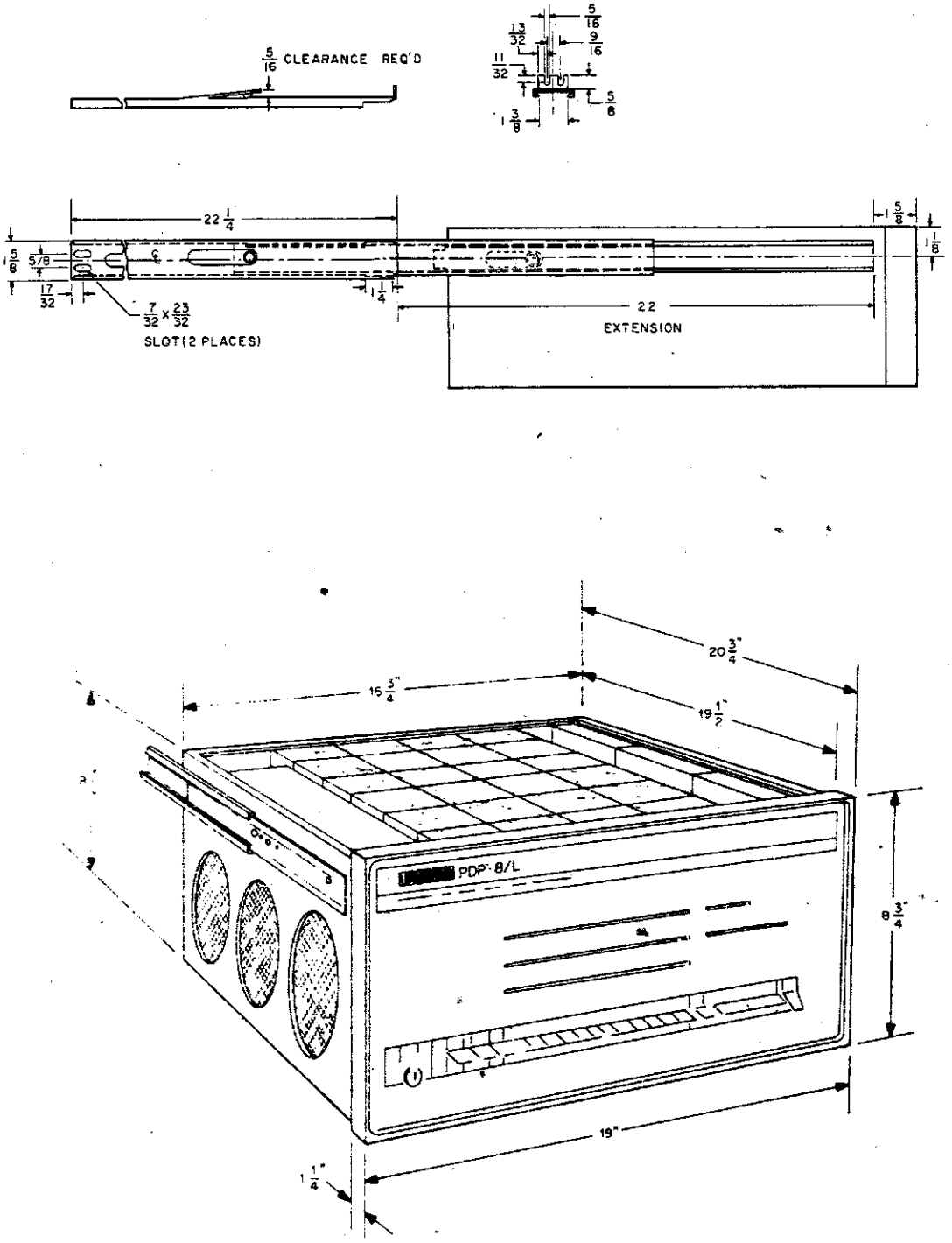


Figure 13-2 Rack mounted Model of PDP-8/L

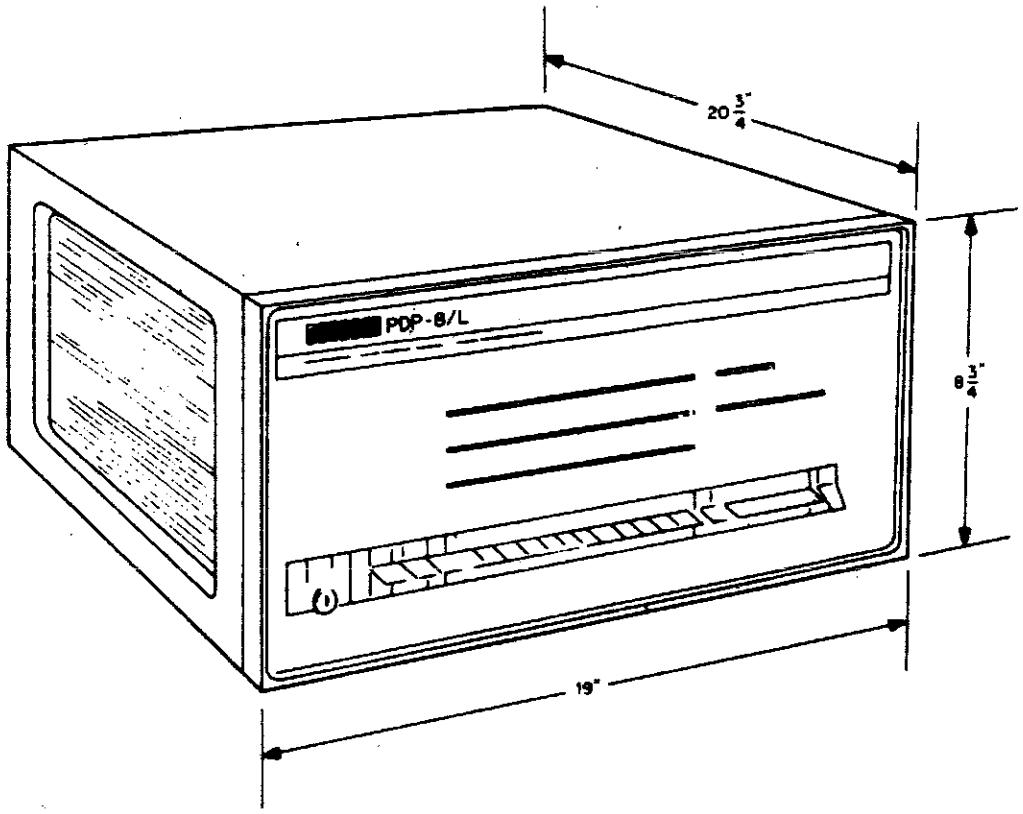


Figure 13-3 PDP-8/L Table-Top Model

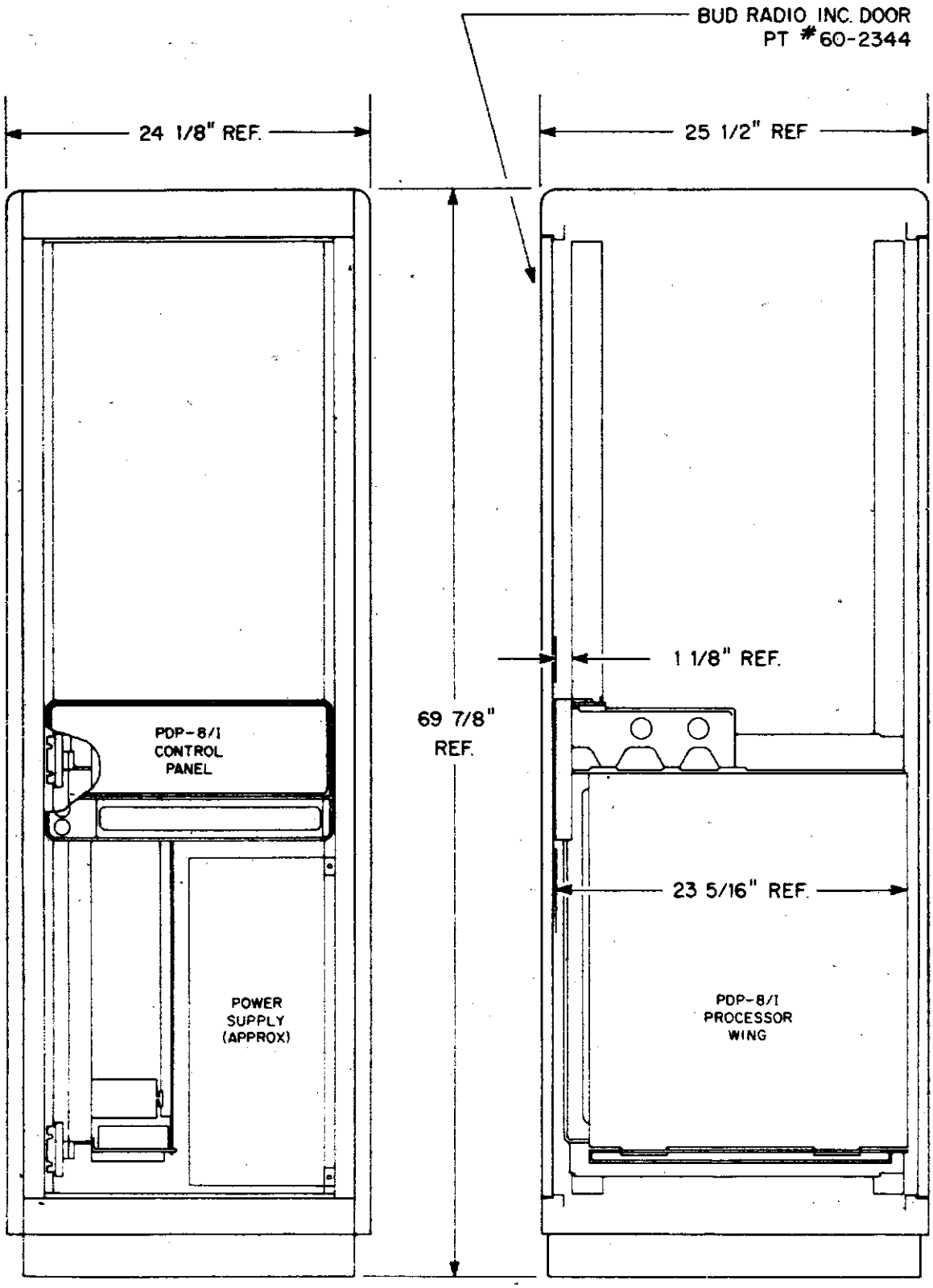


Figure 13-4. Installation BUD Cabinet PDP-8/I

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

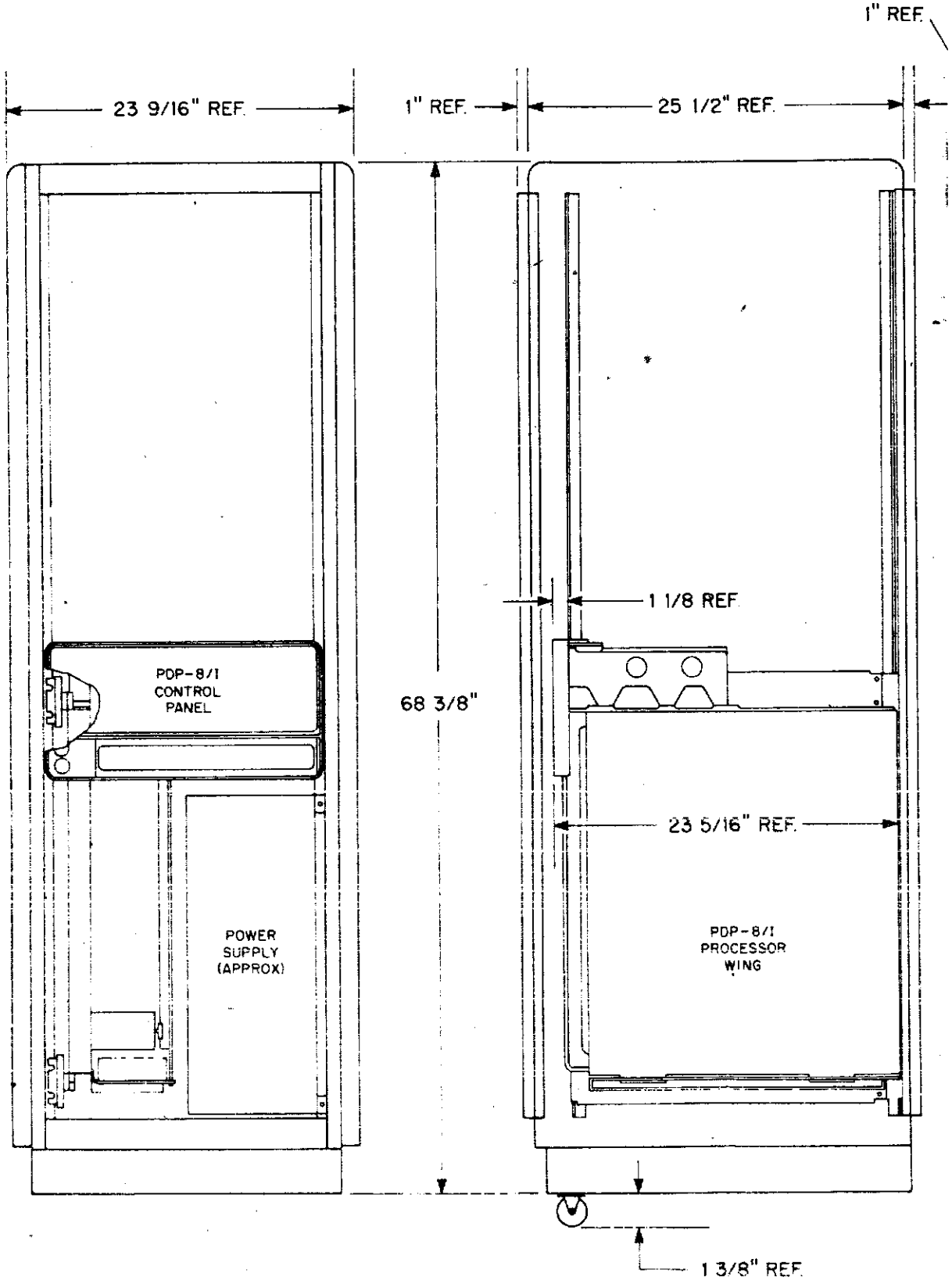


Figure 13-5. Installation EMCOR Cabinet PDP-8/I

POWER REQUIREMENTS

PDP-8/I Requirements — A source of 115 V (±17 V), 60 Hz (±2%), single-phase power, capable of supplying at least 15 A, must be provided to operate a standard PDP-8/I Computer.

PDP-8/L Requirements — The PDP-8/L Computer requires a source of 115 VAC, single-phase, (±15 V, −10 V) AC power at a frequency from 47 to 63 Hz. A 15 A, 3-prong, U-ground power cord on the rear of the chassis is supplied for connection to the power source. An optional step-down transformer can be supplied for 220 V operation. The computer draws 2.5 A and dissipates 250 W.

Common Requirements — Teletype power requirements are common to both the PDP-8/I and PDP-8/L Computers. The teletype requires 115 V AC (±10%), 60 Hz (±0.5 Hz), or 50 Hz (±0.75 Hz) power.

The teletype, which plugs into the rear of the computer and is controlled by the console ON/OFF switch, draws 2.0 A and dissipates 150 W.

General Requirements — To facilitate connection of the computer power cable, the power source should be provided with a Hubbell 3-terminal twistlock plug, 60 Hz systems use a 30 A twistlock plug, 50 Hz systems use a 20 A twistlock plug.

All free-standing cabinets require independent 115 V receptacles. However, these units can be turned on or off, or controlled from the PDP-8/I [PDP-8/L] operator's console.

Cables connect to cabinets through a drop panel in the bottom of the cabinets. Subflooring is not necessary because casters elevate the cabinets to afford sufficient cable clearance.

POWER SPECIFICATIONS FOR COMPUTER INSTALLATION

It is generally advisable to provide a separate load center/breaker panel for the computer system with a breaker for the computer and for each peripheral receptacle. DEC recommends that the wiring include a run of No. 4 gauge wire from the computer frame to a substantial earth ground. A large water pipe or a steel building beam is adequate in most instances, although some systems may require a direct connection to a grounding stake or other high-quality earth ground. Significant operational difficulties are likely in the event of either a poor neutral or poor ground circuit.

Voltage readings should be made at each receptacle in the computer power system to ensure adherence to these power requirements; a checkout procedure for testing the electrical system is provided by DEC on request.

To compensate for voltage variations, transformer terminals are jumpered; the necessary details are included in Figure 13-6.

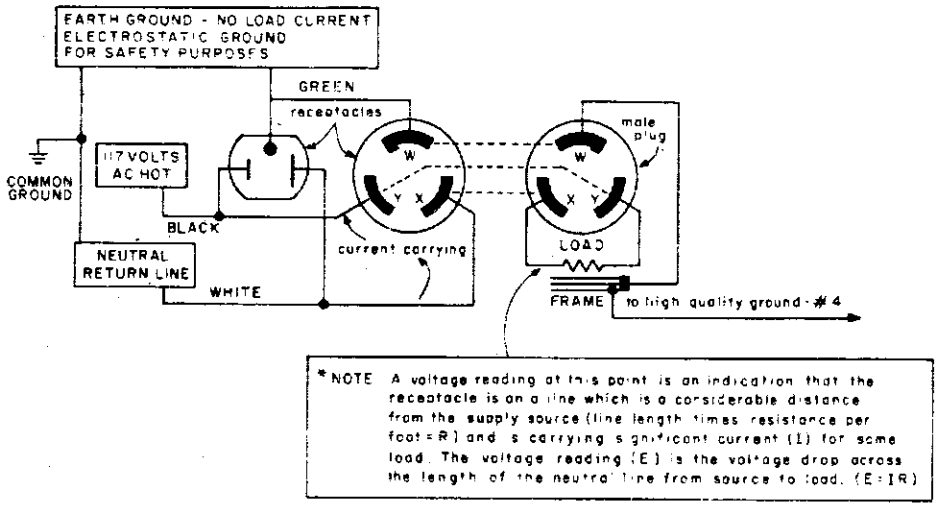


Figure 13-6. Receptacle Jumpering Schematic Diagram

INSTALLATION PROCEDURE

During system checkout, customers are invited to visit DEC's Maynard manufacturing facility to inspect and become familiar with their equipment. Computer customers may also send personnel to instruction courses on computer operation, programming, and maintenance conducted regularly in Maynard, Massachusetts.

DEC engineers are available during installation and testing for assistance or consultation. Further technical assistance in the field is provided by home office design engineers or branch office application engineers.

Table 13-1 provides installation data and space requirements to be considered when installing a PDP-8/I [PDP-8/L].

TABLE 13-1 INSTALLATION DATA

UNIT	NET WEIGHT (lbs.)	DIMENSIONS (in.) Height	Width	Depth	SERVICE CLEARANCE Front	HEAT DISSIPATION Btu/Hr	CURRENT (AMPS) Nom	Surge	POWER DISSIPATION (KW)	MOUNTING PANELS	REMARKS
PDP-8/L Table Top	80	8-3/4	19	20-3/4	—	850	2.5	—	0.25	—	
PDP-8/I PDP-8/L Rack Mounted	190/70	30-11/16 8-3/4	19 19	24-3/4 20-3/4	22 22	2660 850	7.5 2.5	—	0.78 0.25	—	
Standard Cabinet CAB8B (Empty)	225	71-7/16	21-11/16	30	—	—	—	—	—	—	
Teletype ASR-33	70	45	23	19	—	375	2.0	8.0	110	—	Control logic is in basic PDP-8/I or PDP-8/L
Peripheral Expansion Unit BA08	70 (When Full)	8-3/4	19	20-3/4	—	850 (MAX)	2.5	—	0.25 (MAX)	—	Combination of punch and reader designated PCB Weighs 40 lb
Paper Tape Reader PR8	32	10-1/2	19	16	—	510	1.0	—	0.15	2	Control logic mounts within standard PDP-8/I or PDP-8/L package
Paper Tape Punch PR8	32	10-1/2	19	16	—	680	2.0	—	0.20	2	
Card Reader CR8	47	19	14	20	—	900	2.7	—	0.30	(Table space Required)	Control logic mounts within standard PDP-8/I or BA08 unit for PDP-8/L
DECtape Automatic Control TC01	45	15-3/4	19	—	—	680	1.75	—	0.20	3	Controls up to 8 TU55 DECtape transports
DECtape Transport TU55	35	10-1/2	19-1/2	9-3/4	9	410	1	2	0.11	2	
Automatic Mag Tape Control TC58	60	21	19	—	—	1000	2.6	—	0.30	4	Controls up to 8 units (TU20C or TU20D) industry compatible (or IBM compatible).
TU20C & TU20D	400	71-7/16	21-11/16	30	22	4000	6.8	8	0.8		Dimensions are for subsystem mounted in a standard cabinet

Table 13-1 Installation Data (Cont.)

UNIT	NET WEIGHT (lbs.)	DIMENSIONS (in.) Height	Width	Depth	SERVICE CLEARANCE Front	HEAT DISSIPATION Btu/Hr	CURRENT (AMPS) Nom	Surge	POWER DISSIPATION (KW)	MOUNTING PANELS*	REMARKS
Incremental Mag Tape Controller TR02	20	5-1/4	19	--	—	200	0.52	—	0.06	1	Single controller controls one (1) incremental transport. Dual controller controls two (2) incremental transports.
Incremental Tape (8-1/2) Transport TU22 (10-1/2")	60/85	12-1/4/ 24-1/2	19	11-3/4	—	400/600	1.0/1.5	—	0.12/0.18	—	
Incremental Tape (8-1/2) Transport TU25 (10-1/2")	60/85	12-1/4/ 24-1/2	19	11-3/4	—	400/600	1.0/1.5	—	0.12/0.18	—	8-1/2" Reel Transports are 12-1/4" in height. 10-1/2" Reel Transports are 24-1/2" in height
Incremental Tape (8-1/2) Transport TU28 (10-1/2")	60/85	12-1/4/ 24-1/2	19	11-3/4	—	400/600	1.0/1.5	—	0.12/0.18	—	
Random Access Disk File DF32/DS2	75/60	10-1/2	19-1/2	21-1/4	—	1700	3	3	0.50	2	
Disk Control and File RF08/RS08	(Dependent on size of system)	71-7/16	21-11/16	30	22	510/620 (Plus RS08 Motors)	1.3/2.6 Plus 4.0A per RS08 Motor)		0.15/0.30 (Plus RS08 Motors)	—	Dimensions are for subsystem mounted in a standard cabinet.
I/O Conversion Panel DW08	20	5-1/4	19	15	—	375	1.0	—	0.11	1	
Data Channel Multiplexer DM04/DM01	20 30	5-1/4 10 1/2	19 19	— —	— —	170	0.5	—	0.05	1 2	
A/D Converter AF01A	55	8 11/16	19	19-1/2	—	188	0.5	0.5	0.06	2	
A/D Converter AF02A	(Dependent on size of subsystem)	21 11/16	71-7/16	30	22	(Dependent on size of system)				—	Dimensions are for subsystem mounted in a standard cabinet
A/D Converter AF03A	(Dependent on size of subsystem)	71-7/16	21-11/16	30	22	(Dependent on size of subsystem)				—	

Table 13-1 Installation Data (Cont.)

UNIT	NET WEIGHT (lbs.)	DIMENSIONS (in.) Height	Width	Depth	SERVICE CLEARANCE Front	HEAT DISSIPATION Btu/Hr	CURRENT (AMPS) Nom	Surge	POWER DISSIPATION (KW)	MOUNTING PANELS*	REMARKS
IDVM AF04A	300	71 7/16	21 11/16	30	22	2350	6.0	13.2	0.69	—	
A/D Converter AD08	15	5 1/4	19	—	—	170	0.45	—	0.05	1	
D/A Converter AA01A	15	5 1/4	19	—	—	170	0.45	—	0.05	1	
Multiplexer Control AA05/AA07	(Dependent on size of system)	71 7/16	21 11/16	30	22	(Dependent on size of subsystem)				—	Dimensions are for subsystem mounted in a standard cabinet.
Oscilloscope Display VR01-A	28	7	19	17	—	375	1.0	—	0.12	(Table Space Required)	Control logic mounts within standard PDP-8/I or BA08 unit for PDP-8/L.
Storage Tube Display VT01	51	11 7/8	11 5/8	22 3/8	—	850	2.1	—	0.25	(Table Space Required)	Control logic mounts within standard PDP-8/I or BA08 unit for the PDP-8/L.
Asynch. Serial Line Interface PT08	15	5 1/4	19	—	—	170	0.45	—	0.05	1	
Synch Modem Interface DP01AA	30	5 1/4	19	—	—	275	0.70	—	0.08		

Mounting Panels are standard DEC 5 1/4 in. high logic panels.

NOTE: Power supplies for option logic are normally mounted on rear door of DEC cabinets. Customers using their own cabinets should allow additional space for power supplies.

SYSTEM CONFIGURATIONS

PDP-8/I [PDP-8/L] Systems (see Figures 13-7 through 13-9) are mounted in standard DEC cabinets (see Figures 13-10 through 13-13). There are three 10½ in. spaces above the PDP-8/I PDP-8/L Control Panel on the basic cabinet. These spaces are numbered consecutively starting just above the control cabinet. Each space has options assigned in a fixed priority, as illustrated.

When systems require additional cabinet space, an option cabinet must be added. This cabinet is mounted to the left of the basic cabinet (front view), with option priorities assigned as shown in Figures 13-10 and 13-11.

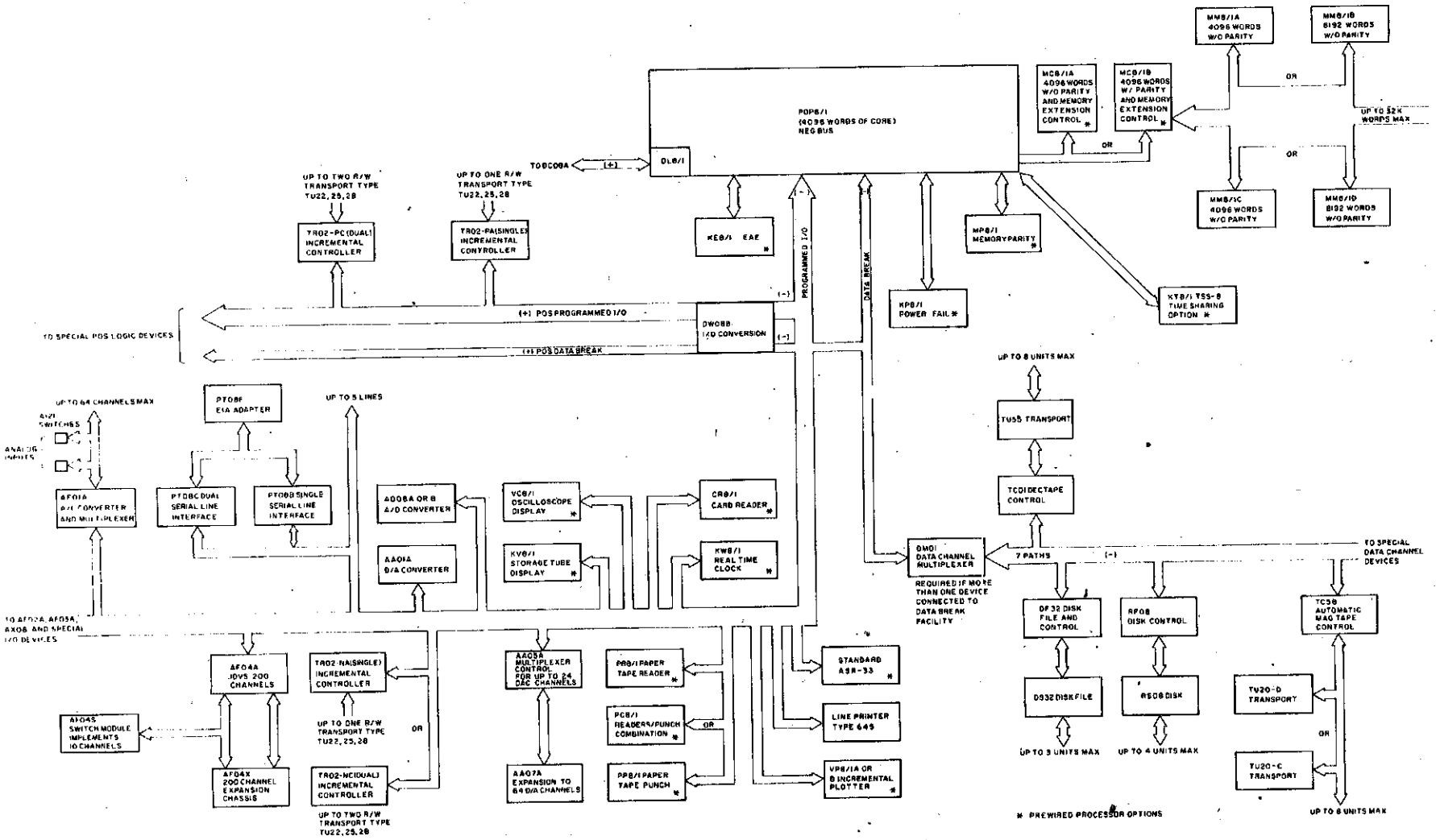


Figure 13-7 Negative Bus PDP-8/I Configuration

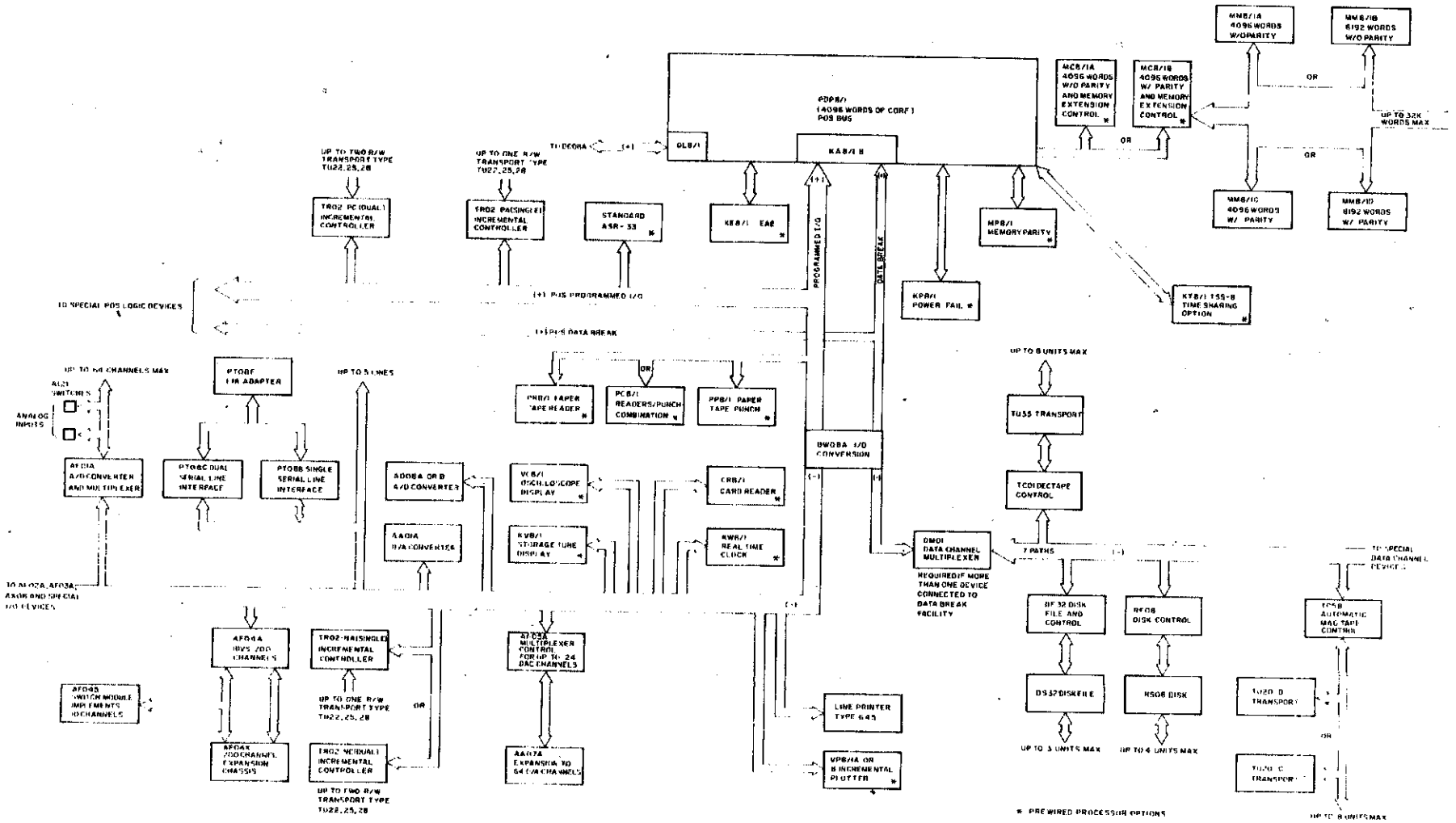


Figure 13-8 Positive Bus PDP-8/I Configuration

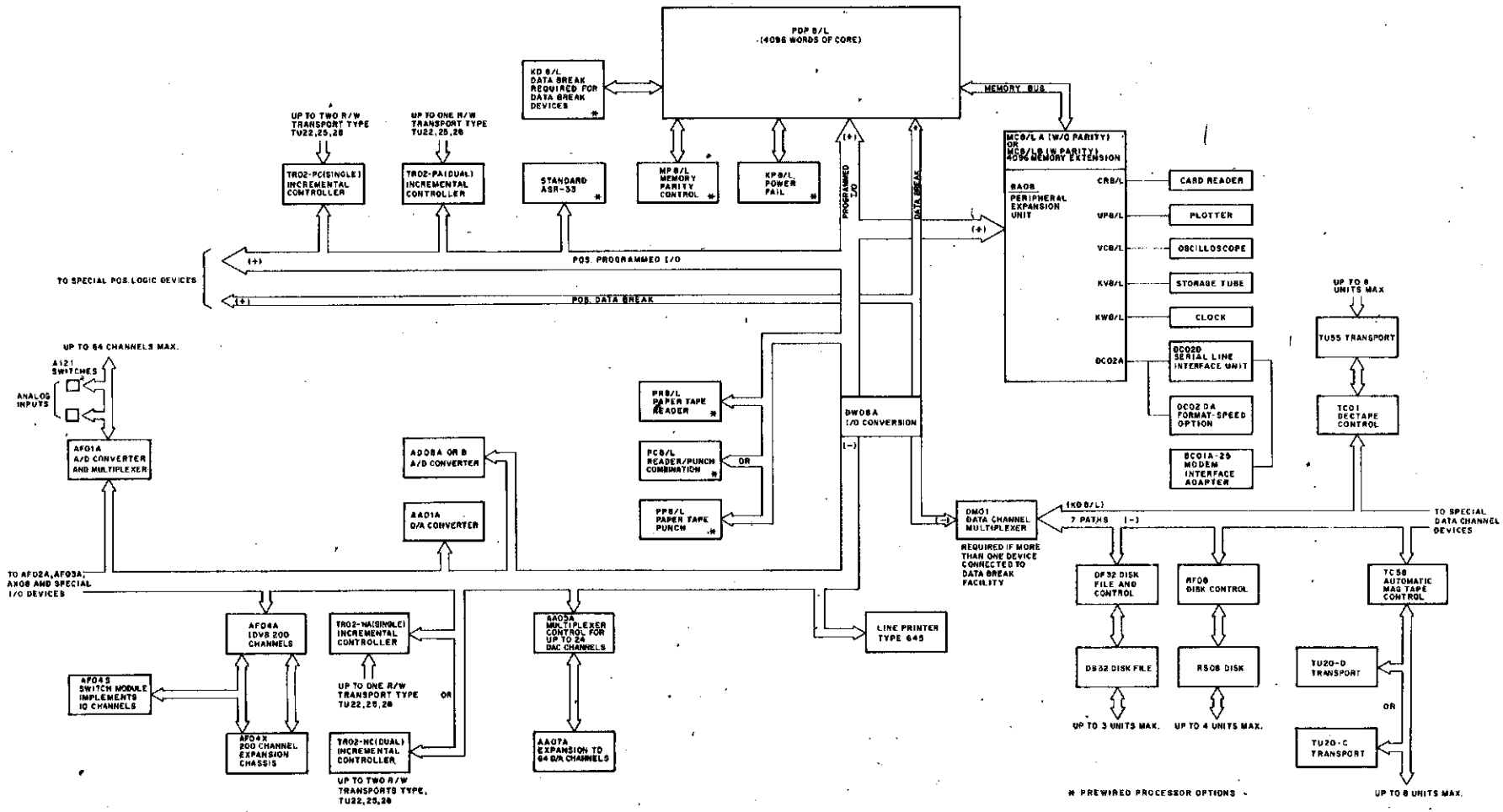


Figure 13-9 PDP-8/L Configuration

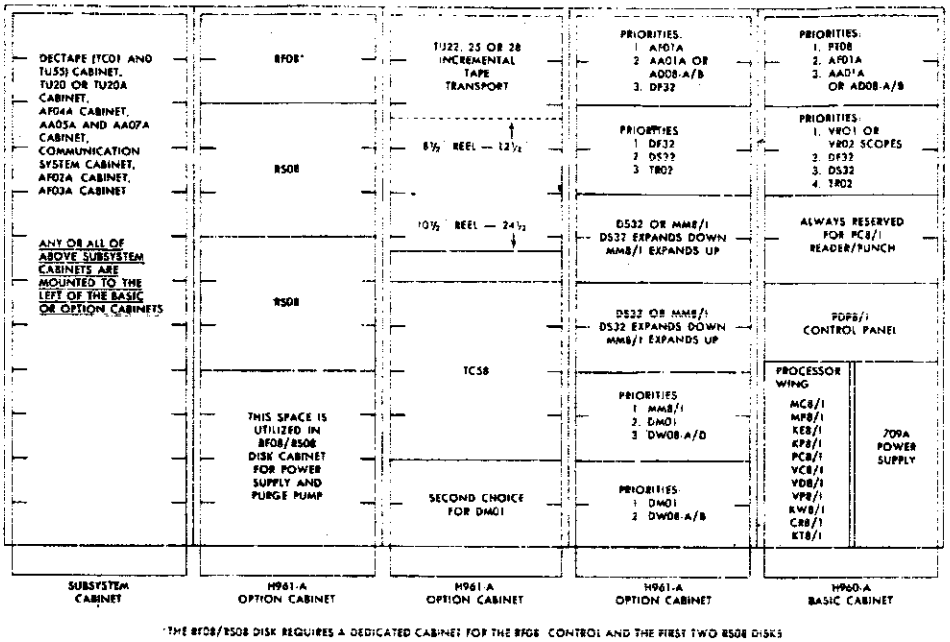


Figure 13-10 Priority Assignment, PDP-8/I

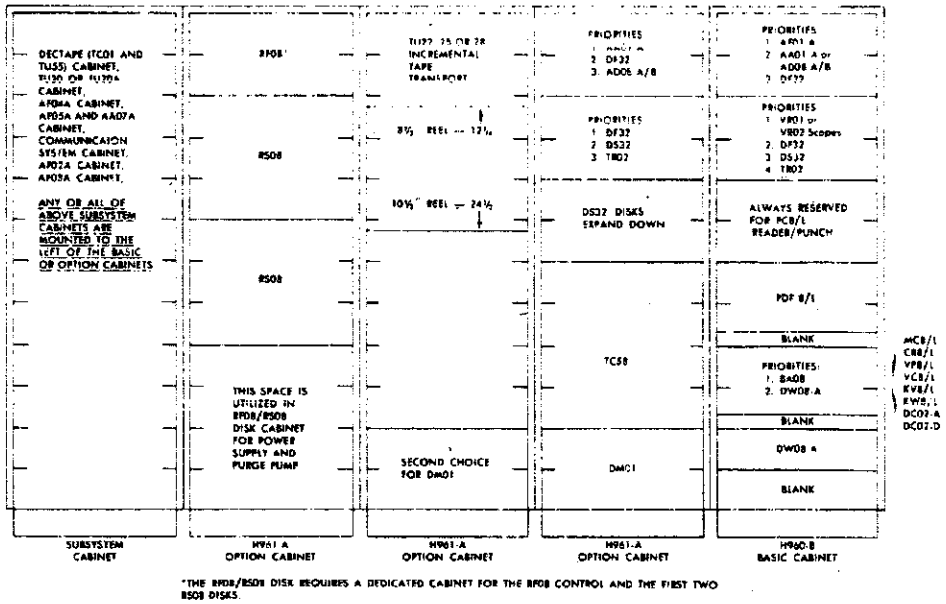


Figure 13-11 Priority Assignment, PDP-8/L

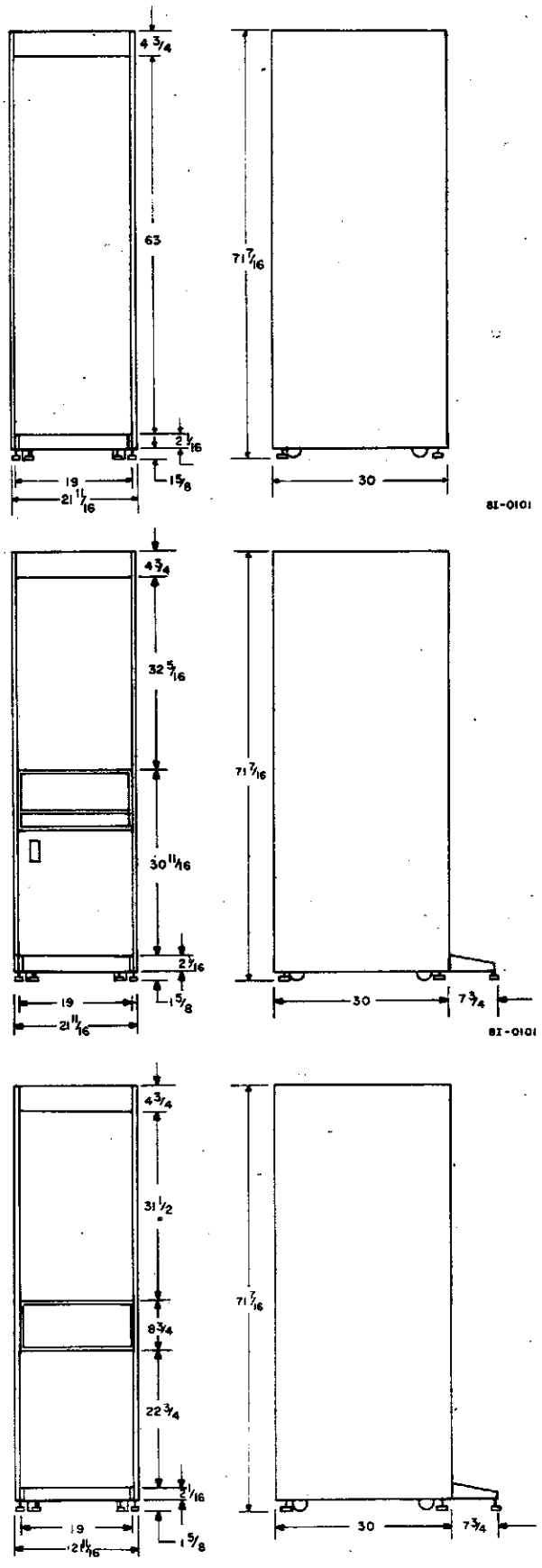


Figure 13-12 Basic PDP-8/I and PDP-8/L Cabinets

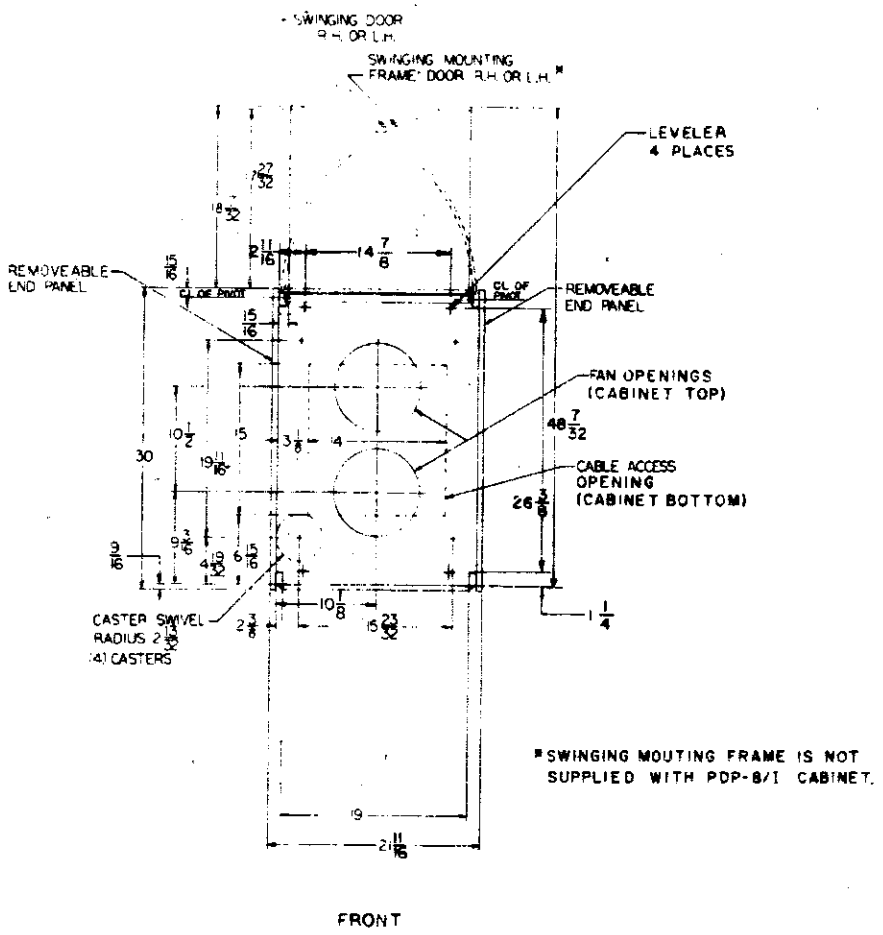


Figure 13 13 Cabinet Layout Dimensions

APPENDIX A
PROGRAM ABSTRACTS

FAMILY-OF-8 PROGRAMS

The PDP-8/L or PDP-8/I is delivered to the user complete with an extensive selection of system programs and routines which make the full data processing capability of the new computer immediately available to each user and eliminate many commonly experienced initial programming delays.

The programs described in these abstracts come from two sources, past programming effort on the PDP-5; 8; 8/S, and present and continuing programming effort on these same machines plus the PDP-8/L and PDP-8/I. Thus, the programming system takes advantage of the many man-years of program development and field testing by Digital computer users. There are over 4000 Family-of-8 systems already installed in the field.

Although many PDP-8/L and PDP-8/I programs originated with previous Family-of-8 computers, all utility and functional program documentation is issued anew.

Programs written by users of Family-of-8 computers and submitted to the DECUS Library (DECUS — Digital Equipment Corporation Users' Society) are immediately available to Family-of-8 users. Consequently, users of all Family-of-8 computers can take advantage of continuing program development.

SYSTEM PROGRAMS

DEC-08-AJAD-D
FOCAL-8

FOCAL-8 (FOrmula CALculator for the PDP-8 Family computers) is an on-line, conversational interpretive language, used as a tool by students, engineers, and scientists in solving a wide variety of numerical problems. The language consists of short imperative English statements and mathematical expressions in standard notation. FOCAL puts the full calculating power and speed of the computer at the user's fingertips without requiring mastery of the intricacies of assembly language programming; in fact, the user need not know anything about computers.

Using FOCAL-8, programs can be entered from the Teletype keyboard and immediately executed, with the interpretive features editing, compiling, and executing the stored program. FOCAL is available with additional segments (overlay tapes) which provide increased calculating accuracy, more core for large programs by utilizing two fields of core, or graphic display capabilities. Worthy of special mention is the system segment that gives each of seven FOCAL users the power and flexibility of FOCAL from one computer. This time-sharing system permits one FOCAL program to serve several users so that each feels he has the whole system to himself.

DEC-08-AFCO-D
4K FORTRAN

The 4K FORTRAN Compiler lets the user express problems in a mixture of English words and mathematical statements. It reduces the time needed for program preparation and enables users with little or no knowledge of the computer's organization and operating language to write effective programs.

The 4K FORTRAN language consists of four general types of statements: arithmetic, logic, control, and input/output. FORTRAN functions include addition, subtraction, multiplication, division, sine, cosine, arctangent, square root, natural logarithm, and exponentiation.

DEC-08-KFXB
8K FORTRAN

The 8K FORTRAN system translates a source program into relocatable binary code. The relocatable binary code is output on paper tape and then loaded into the computer for program execution. The 8K FORTRAN system features USA Standard FORTRAN syntax; subroutines; two levels of subscripting; function subprograms; input/output supervisors; relocatable output loaded by the 8K Linking Loader; COMMON statements; I, F, E, A, X, and H format specifications; and arithmetic and trigonometric library subroutines.

The 8K FORTRAN system consists of a one-pass compiler, the 8K SABR Assembler, 8K Linking Loader, and a comprehensive Library of subprograms. The system requires a PDP-8 Family computer with at least 8K words of core memory, an ASR33 Teletype, and a high-speed paper tape reader and punch. 8K FORTRAN utilizes all available core to 32K. 8K FORTRAN is a modified version of USASI Basic FORTRAN.

DEC-08-ARXA-D
8K SABR Assembler

SABR (Symbolic Assembler for Binary Relocatable programs) is an advanced one-pass symbolic assembler for 8K to 32K PDP-8 Family computers. SABR programs are core page independent, allowing programs to be written without regard to the 128-word core page of the computer. SABR automatically generates off-page and off-field references for direct and indirect statements. It also automatically connects instructions on one page to those that overflow onto the next. The list of available pseudo-ops is extensive, including external subroutine calling, argument passing, and conditional assembly. SABR offers an optional second pass to produce a side-by-side octal/symbolic listing of the assembled program.

The relocatable binary tapes produced by SABR are loaded into any field of core memory and executed with the 8K Linking Loader, as is the comprehensive library of subprograms (also used with 8K FORTRAN). These subprograms may be called by any SABR program. SABR also acts as the second pass of the 8K FORTRAN system.

Several other PDP-8 assemblers (some of which can run on larger computers) are available from DECUS.

DEC-08-ASAC-D
PAL III Assembler (4K)

The PAL III Symbolic Assembler is a two-pass assembler which translates symbolic programs written in the PAL III language into binary-coded programs. The assembler offers an optional third pass which produces an octal/symbolic printout and/or punchout of the assembled program. PAL III permits symbolic origins, expressions, and references. Its output is in binary code for use with DEC's standard Binary Loader.

DEC-08-CDDA-D
DDT-8 (Dynamic Debugging Tape)

DDT-8 provides a means for on-line program debugging at the symbolic or mnemonic level. By typing commands on the console teleprinter, memory

locations can be examined and changed, program tapes can be inserted, selected portions of the program can be run, and the updated program can be punched.

DEC-08-CMAA-D
MACRO-8 Assembler

The MACRO-8 symbolic assembler accepts source programs written in symbolic language and translates them into binary form in two passes. MACRO-8 produces an object program tape (binary), a Symbol table (for use with DDT), an octal/symbolic assembly listing, and useful diagnostic messages. MACRO-8 is compatible with PAL III, and has the following additional features: user defined macros; double precision integers, floating point constants; arithmetic and Boolean operators, literals, text facilities and automatic link generation.

DEC-08-COCO-D
ODT-8 (Octal Debugging Technique)

ODT-8 (Octal Debugging Technique) is a debugging aid which facilitates communication with, and alteration of, the program being run. Communication between operator and program occurs via the Teletype, using defined commands and octal numbers. ODT-8 is a subset of DDT-8 and occupies three pages of core storage.

The program may be relocated to occupy any three consecutive pages of core.

DEC-08-ESAB-D
Symbolic Editor

The Symbolic Editor allows the user to prepare and edit symbolic tapes on-line in ASCII code with the Teletype and/or high-speed reader/punch. The tedious task of correcting symbolic program tapes using the Teletype off-line is thereby avoided. Proper use of the Symbolic Editor can substantially ease the labor and reduce the number of passes necessary to correct symbolic program tapes. The Editor reads a page, or section, of symbolic tape into a buffer in core storage, where it is available for examination and correction upon keyboard command. The page buffer occupies all of core not taken up by the Editor itself and has a capacity of approximately 6000_{10} characters. When the Editor has finished reading a page into the buffer, a bell rings to signal the user that he may begin editing. The user may then call for a listing of individual (numbered) lines, in any order, and insert desired changes and corrections. In addition, text may be added to the buffer, or inserted between specified lines. Groups of lines or individual lines may be moved or deleted by a single command, or the entire page may be erased if desired. Searches may be made and parts of lines changed without retyping the entire line. Upon keyboard command, the Editor will then either list or punch out the corrected lines or page on paper tape. The Editor can also be used to generate a new symbolic tape by typing new text directly on the keyboard. Errors in typing may be corrected simply by typing a rubout.

DEC-08-YQYA-D
Floating Point Package

The floating point package permits the PDP-8/I to perform arithmetic operations that could not otherwise be done without the addition of costly optional hardware. With the floating point package, the programmer need not concern himself with writing and repeatedly calling complicated arithmetic subroutines. He merely has to specify in one instruction the desired arithmetic operation (operator) and an argument (operand), and the operation will be performed

automatically. In addition to increasing accuracy, floating point operations relieve the programmer of scaling problems common in fixed-point arithmetic. The floating point subroutines and interpreter permit the programmer to encode arithmetic operations to either 6 or 10 decimals digits of precision as easily as though the machine had floating point hardware.

Digital-8-16-S-D
Master Tape Duplicator

This program will duplicate and verify 8-channel paper tapes using a high-speed reader and high-speed punch. The program uses the program interrupt and allows both the reader and the punch to operate at maximum speed. The program accumulates two types of checksums which reading and while punching: (1) the number of nonzero characters on the tape, and (2) the sum of characters on the tape (both are taken modulo 4096).

When duplicating, the program compares the checksums at the end of the tape with the checksums accumulated by the read routine. If these differ, a reader error has occurred and a message is typed.

Digital-08-USBO-P
Multianalyzer Display and Analysis

The two-dimensional pulse-height analysis reads in and analyzes two-parameter energy and spectra data. The program receives and executes commands from the keyboard. These commands start and stop data taking, control the displays, and control writing and punching of data. The displays available are: isometric, vertical and horizontal slicing, differential and integral contours, and "twinkle box". The program is flexible with respect to the dimensions of the data matrix.

Digital-8-15-S-P
Oceanographic Analysis

This program represents the basic accepted physical oceanography method for the reduction of data concerning depth, temperature, and salinity measurements of the water column.

This program allows the field oceanographer a rapid means of immediately calculating Sigma-T, anomaly of specific volume, and sound velocity following a Nansen cast whereby he may examine results in detail to determine the structure of the environment he has just sampled and to check the validity of his measurements.

The program also contains an interpolation routine as well as a depth integration of the anomaly of specific volume.

DEC-08-G61D-D
Programmed Buffered Display 338 Manual

The Type 338 Programmed Buffered Display is a precision incremental display system, consisting of a small scale, high-speed computer and a display subsystem for control of the CRT. The computer used is Digital Equipment Corporation's PDP-8 (for Programmed Data Processor). It is a single address, fixed word-length (twelve bits) machine. The complete cycle time for its random access magnetic core memory is 1.5 μs. All arithmetic operations are performed in 2's complement notation.

DEC-T8-MRFB-D
TSS/8 Time-Sharing Monitor

TSS/8 (Time-Sharing System for the PDP-8/I and PDP-8 computers) is a general-purpose, stand-alone, time-sharing system. TSS/8 offers each of up

to 24 users a comprehensive library of programs which provide facilities for compiling, assembling, editing, loading, saving, calling, debugging, and running user programs on-line. Any of these library programs can be called into use by typing, in response to Monitor's invitation (the dot), the command R and the assigned name of the program. For example, • R FOCAL brings the FOCAL program into core from the disk and automatically executes FOCAL so that it begins typing out its initial dialogue.

The heart of this time-sharing system is a complex of programs called Monitor. Monitor coordinates the operations of the various units, allocates the time and services of the computer to users, and controls their access to the system. By segregating the central processing operations from the time-consuming interactions with the human users, the computer can in effect work on a number of programs simultaneously. The executions of various programs are interspersed without interfering with one another and without detectable delays in the responses to the individual users.

DEC-08-YISA-D
Variable Stroke Character Generator for KV8

The Variable Stroke Character Generator (VSCG) is a program which generates and displays the 64 principal ASCII characters using the KV8/I Storage Tube Display Controller. By making use of the redundancy in the English character set, all 64 ASCII characters are described by seven vector coordinate masks plus two control words per character. This is quite different from the conventional stroke vector generators, where all ASCII characters are generated from one standard mask, for example, the starburst pattern.

Characters are generated by executing stroke vectors on the storage tube. Each vector is specified by the address of its end-point coordinates. Execution time is minimized by executing only the required stroke vectors for a particular character; thus, this program is in effect a variable stroke character generator (VSCG).

In addition to displaying characters, VSCG provides control functions for carriage return/line feed, "home" to top left of the scope, horizontal tab, and character stabilization.

VSCG is very versatile and easily modified since all control functions and characters are table driven.

ELEMENTARY FUNCTION ROUTINES

DEC-08-FFAC-D
Math Routines Manual
The following routines are described in the Math Routines Manual.

Square Root Subroutine-Single Precision
Forms the square root of a single-precision number. An attempt to take the square root of a negative number will be given 0 for a result.

Signed Multiply Subroutine-Single Precision
Forms a 22-bit signed product from 11-bit signed multiplier and multiplicand.

Signed Divide Subroutine-Single Precision

This routine divides a signed 11-bit divisor into a signed 23-bit dividend giving a signed 11-bit quotient and a remainder of 11 bits with the sign of the dividend.

Double-Precision Multiply Subroutine-Signed

This subroutine multiplies a 23-bit signed multiplicand by a 23-bit signed multiplier and returns with a 46-bit signed product.

Double-Precision Divide Subroutine-Signed

This routine divides a 23-bit signed divisor into a 47-bit signed dividend and returns with a 23-bit signed quotient and a remainder of 23 bits with the sign of the dividend.

Sine Routine-Double Precision

The double-precision sine subroutine evaluates the function Sin (X) for $-4<X<4$ (X is in radians). The argument is a double-precision word, 2 bits representing the integer part and 21 bits representing the fractional part. The result is a 23-bit signed fraction $-1<\text{Sin}(X)<1$.

Cosine Routine-Double Precision

This rubroutine forms the cosine of a double-precision argument (in radians). The input range is $-4<X<4$.

Four-Word Floating-Point Package

This is a basic floating-point package that carries data as three words of mantissa and one word of exponent. Common arithmetic operations are included as well as basic input/output control. No functions are included.

Logical Subroutines

Subroutines for performing the logical operations of inclusive and exclusive OR are presented as a package.

Shift Right, Shift Left Subroutines (Single and Double Precision)

Four basic subroutines, shift right and shift left, each at both single and double precision, are presented as a package.

Logical Shift Routines

Two basic subroutines, shift right at both single and double precision, are presented as a package. The shifts are logical in nature.

LOADERS

DEC-08-LRAA-D
Read-in Mode (RIM Loader)

The RIM Loader is a minimum routine for reading and storing information contained in read-in-mode coded tapes via the ASR33 Perforated Tape Reader.

DEC-08-LBAA-D
Binary Loader

The Binary Loader is a short routine for reading and storing information contained in binary-coded tapes, using the ASR33 Perforated Tape Reader and the Type PR8/L High-Speed Perforated Tape Reader.

The Binary Loader accepts tapes prepared by the use of PAL (Program

Assembly Language) or MACRO-8. Diagnostic messages may be included on tapes produced when using either PAL or MACRO. The Binary Loader will ignore all diagnostic messages.

DEC-08-LHAA-D
"HELP" Loader

The "HELP" Loader loads the standard version of the RIM and BIN Loaders into the PDP-8/L, in less than 90s, replacing manual procedures which require several minutes.

DEC-08-LUAA-D
TC01 Bootstrap Loader

This is a bootstrap for loading the PDP-8 DECtape Library System designed for use with DECtape Control Type TC01 with TU55 Tape Transports.

UTILITY PROGRAMS AND SUBROUTINES

DEC-08-PMPO-D
Read-In Mode (RIM) Punch

The RIM Punch program provides a means of punching out information contained in selected blocks of core memory as RIM-coded tape via the ASR33 Perforated Tape Punch or High-Speed Punch. The punch program may occupy either low or high memory, depending on the version used.

Digital-8-5-U-Sym-D
Binary Punch (ASR33 or PP8/L)

This program provides a means of punching out information contained in selected blocks of core memory as binary-coded tape via the ASR33 Perforated Tape Punch or the High-Speed Punch.

Digital-8-6-U-Sym-D
Octal Memory Dump

This routine will read the console switches twice to obtain the upper and lower limits of an area of memory, then type on the Teletype an absolute address plus the octal contents of the first four words specified and repeat this until the block is exhaused, at which time the user may repeat the operation.

Digital-8-10-U-Sym-D
BCD to Binary Conversion Subroutine

A basic subroutine for converting binary-coded-decimal numbers to their equivalent binary value. Conversion is accomplished by "radix deflation".

Digital-8-11-U-D
Double Precision BCD to Binary Conversion Subroutine

This subroutine converts a 6-digit BCD number to its equivalent binary value in two computer words.

Digital-8-12-U-D
Incremental Plotter Subroutine

This subroutine moves the pen of an incremental plotter to a new position along the best straight line. The pen may be raised or lowered during the motion.

Digital-8-14-U-Sym-D
Binary to BCD Conversion Subroutine

This subroutine provides the basic means of converting binary data to binary-

coded-decimal (BCD) data for typeout, magnetic tape recording, etc.

Digital-8-15-U-Sym-D
Binary to BCD Conversion (Four Digit)

This subroutine extends the method used in Digital-8-14-U-Sym so that binary integers from 0 to 4095 contained in a single computer word may be converted to four binary-coded-decimal characters packed in two computer words.

Digital-8-18-U-Sym-D
Alphanumeric Message Typeout

A basic subroutine to type messages packed in computer words. Two 6-bit characters are packed internally in a single word. All ASR33 codes from 301 to 337 and from 240 to 277 (except 243 and 245) can be typed. The typing of line-feed (code 212) and carriage-return (code 215) are made possible by arbitrarily assigning internal codes of 43 and 45, respectively, to represent these characters, thus preventing the output of ASCII codes 243 (#1) and 245 (%).

Digital-8-19-U-Sym-D
Teletype Output Subroutines

A group of subroutines useful in controlling ASR33 output is presented as a package. Provision is made for the simulation of tabulation stops. The distance "tabbed" may be controlled by the user. Characters whose ASR33 codes are in groups 241 and 277, inclusive, and 300 through 337, inclusive, are legal. Space, carriage return then line feed, and tabulation are provided via subroutines.

Digital-8-20-U-Sym-D
Character String Typeout

A basic subroutine to type messages stored internally as a string of coded characters. All ASR33 characters are legal.

Digital-8-21-U-Sym-D
Symbolic Tape Format Generator

The format generator allows the user to create symbolic tapes with formatting. It may be used to condense tapes with spaces by inserting tabs, or merely to align tabs, instructions, and comments.

Digital-8-22-U-Sym-D
Unsigned Decimal Print

This subroutine permits the typeout of the contents of a computer word as a 4-digit, positive, decimal integer.

Digital-8-23-U-Sym-D
Signed Decimal Print — Single Precision

This subroutine permits the typeout of the contents of a computer word as a signed 2's complement number. If bit 0 of the computer word is a 1, the remaining bits represent a negative integer in 2's complement form; if bit 0 equals 0, the remaining bits represent a positive integer. If the number is negative, a minus sign is printed; if positive, a space.

Digital-8-24-U-Sym-D
Unsigned Decimal Print, Double Precision

This subroutine permits the typeout of a double-precision integer stored in the usual convention for double-precision numbers. The one exception is that all 24 bits are interpreted as magnitude bits (i.e., bit 0 of the high-order

word is not a sign bit). The typeout is in the form of a 7-digit, positive, decimal integer.

Digital-8-25-U-Sym-D
Signed Decimal Print, Double Precision

This subroutine permits the typeout of the contents of two consecutive computer words as one signed, double-precision, 2's complement number. If bit 0 of the high-order word is a 1, the remaining 23 bits represent a negative integer in 2's complement form; if bit 0 equals 0, the remaining bits represent a positive integer. If the number is negative, a minus sign is printed; if positive, a space.

Digital-8-28-U-Sym-D
Single Precision Decimal to Binary Conversion and Typeout ASR333, Signed or Unsigned

This routine accepts a string of up to four decimal digits (single precision for the PDP-8/L) from the Teletype keyboard and converts it to the corresponding 2's complements binary number.

The string may contain as legal characters a sign (+, −, or space) and the digits from 0 through 9. If the first legal character is not a sign, the conversion is unsigned. A back arrow (←) at any point in the string erases the current string and allows the operator to reenter the correct value. Any character after the first, other than another digit or back arrow causes the conversion to terminate and is found in location SISAVE within the subroutine.

Digital-8-29-U-Sym-D
Double Precision Decimal to Binary Conversion and Typeout (ASR33), Signed or Unsigned

This routine accepts a string of up to eight decimal digits (double precision for the PDP-8/L) from the Teletype keyboard and converts it to the corresponding 2's complement binary number.

The string may contain as legal characters a sign (+, −, or space) and the digits 0 through 9. If the first legal character is not a sign, the conversion is unsigned. A back-arrow (←) at any point in the string erases the current string and allows the operator to reenter the value. Any character after the first, other than another digit or "back-arrow", causes the conversion to terminate and is found in location DIDSAV within the subroutine.

DECTAPE SYSTEM SOFTWARE

DEC-08-SUBO-D
DECtape Programming Manual

The DECtape Library System is loaded by a 17_{10} instruction bootstrap routine that starts at 7600_8. This loader calls a larger program into the last memory page, whose function is to preserve on tape the contents of memory from 6000_8-7577_8, and then to load the INDEX program and the directory into those same locations. Since the information in this area of memory has been preserved, it can be restored when operations have been completed. The skeleton system tape contains the following programs:

INDEX — Typing this causes the names of all programs currently on file to be typed out.

UPDATE — Allows the user to add a new program to the files. UPDATE queries the operator about the program's name, its starting address, and its location in core memory.

GETSYS — Generates a skeleton library tape on a specified DECtape unit.

DELETE — Causes a named file to be deleted from the tape.

Starting the skeleton library tape, the user can build up a complete file of his active programs and continuously update it.

DEC-08-FUBO-D
TC01 DECtape Subroutines

These subroutines provide the user with the ability to read, write and search using the TC01 tape system. The read and write subroutines transfer 128_{10} (one memory page) of the specified block (or blocks) although the standard block length is 129_{10} 12-bit words. Successive blocks are read (written) from (into) successive 128 word blocks of core. Provision is made for transfer to and from extended memories.

DEC-08-EUFA-D
TC01/TU55 DECtape Formatter

The purpose of this system is to record the required timing and mark tracks on a DECtape mounted on the TC01-TU55 DECtape unit.

The program, which never stops, obtains the variable information it needs by communication with the operator via the ASR33 Teletype.

Two full passes are required to complete one DECtape. Upon completion of a sequence, another tape may be mounted and formatted, as the last, without renewed communication between the operator and program. Therefore, marked tape may be produced in great numbers with little operator intervention, at a rapid rate; one tape, excluding set-up time, requires two minutes from start to finish (see also Disk Software).

DEC-08-YPTA-D
DECtape Copy Routine

The DECtape copy program (DTC-8) provides a simple, efficient method for copying one DECtape from another on the PDP-8. Features include the capability of handling non-standard block lengths (up to 1550_{10} 12-bit words per block), and the ability to reread and validate the copied data, thus ensuring its correctness. DECtape handling routines are all internal and Monitor independent.

Disk/Dectape Software

DEC-D8-SDAA-D
DISK MONITOR SYSTEM

This system consists of a keyboard-oriented Monitor, which enables the user to efficiently control the flow of programs through his PDP-8/L, and a comprehensive software package, which includes a FORTRAN Compiler, Program Assembly Language (PAL-D), Edit program (EDITOR), Peripheral Interchange Program (PIP) and Dynamic Debugging Technique (DDT) program. Also provided is a program (BUILDER) for generating a customized monitor according to the user's particular machine configuration (amount of core, number of disks or DECtapes, etc.).

The system is modular and open ended, permitting the user to construct the software required in his environment, and allows the user full access to his disk (or DECtape) — referred to as the system device — for storage and retrieval of his programs. By typing appropriate commands to the Monitor, the user can load a program (construct it from one or more units of binary coding previously punched out on paper tape or written on the disk by the Assembler, and assign it core), save it (write it out, with an assigned starting address, on the system device), and later call it (read it back into core from the system device) for execution.

In order to have a complete DISK/DECtape package, the user may order the following in addition to DEC-D8-SDAA-D above:

1.	Disk System Builder	DEC-D8-SBAD-PB
2.	Disk Editor	DEC-D8-ESAB-PB
3.	PIP	DEC-D8-PDAB-PB
4.	Disk DDT	DEC-D8-CDDA-PB
5.	Disk DDT Driver (ASCII)	DEC-D8-CDDA-PA
6.	Disk/DECtape FORTRAN	DEC-D8-AFA(1-6)-PB

DEC-D8-ASAB-D
PAL-D DISK ASSEMBLER

PAL-D is the symbolic assembly program designed primarily for the 4K PDP-8 family of computers with disk or DECtape. The PAL-D Assembler makes machine language programming easier, faster, and more efficient. Basically, the Assembler processes the programmer's source program statements by translating mnemonic operation codes to the binary codes needed in machine instructions, relating symbols to numeric values, assigning absolute core addresses for program instructions and data, and preparing an output listing of the program, which includes notification of any errors detected during the assembly process. PAL-D incorporates virtually all of the features of both PAL III and MACRO-8.

MAINTENANCE PROGRAMS

MAINDEC-08-DOTA-D(D)
Incremental Tape Random Exerciser

This program written for write/read incremental tape units interfaced with either a PDP-8I or PDP-8L, exercises up to two tape units with three separate program routines which are independent of each other. The operator must specify the type of tape unit and the desired test routine via keyboard input. Data errors read from tape are indicated by printouts on the TTY.

MAINDEC-8/I-D01B-D
Instruction Test 1

This is a diagnostic program for testing the AND, TAD, and OPERATE instructions of the PDP-8/L and PDP-8/I.

MAINDEC-8I-D02B-D
Instruction Test 2

This program is an extensive test of autoindexing, indirect addressing, and the DCA instruction for the PDP-8/L, 8/I. It also offers minimal testing for interrupt and the AND, TAD, ISZ, JMP and JMS instructions.

MAINDEC-08-D02B-D
Instruction Test Part 2B

This program is a test of the 2's complement add (TAD) and rotate logic (RAL, RTL, RAR, RTR). Random numbers are used in the Two's Add portion of the test and sequential numbers are used in the Rotate portion. Program control depends on operator manipulation of four switches in the SWITCH REGISTER (bits 0, 1, 2, 3). Error information is normally printed out on the keyboard printer.

MAINDEC-8I-D4CA-D(L)
PDP-8/I, PDP-8/L Memory Parity IOT Test

Memory Parity IOT Test is designed to exercise and detect errors on the memory parity control logic. A routine is also included which writes random numbers in memory field 0, and then checks for data parity errors. Manual intervention after the start of the test is required in order to test the parity IOT's. Printed instructions are given on the TTY printer.

MAINDEC-08-D04B-D
Random JMP Test

This program tests the JMP instruction of the PDP-8/L, 8/I. Most of memory is used as a JUMP field with a random number generator selecting each

"JUMP FROM" AND "JUMP TO" LOCATION.

MAINDEC-08-D05B-D
Random JMP-JMS Test

This is a diagnostic program to test the JMS instruction of the PDP-8/L, 8/I. Random "FROM" and "TO" addresses are selected for each test. The JMP instruction is tested in that each test requires a JMP to reach the JMS.

MAINDEC-08-D07B-D
Random ISZ Test

This program is written to test the ISZ instruction of the PDP-8/L, 8/I. An ISZ instruction is placed in a FROM location, and a TO location contains the OPERAND. Part 1 of the program selects FROM, TO, and OPERAND from a random number generator, with the option of holding any or all constant. Part 2 uses a fixed set of FROM, TO, and OPERAND numbers.

MAINDEC-08-D1AC-D
PDP-8/I, PDP-8/L Memory Power On/Off Test

This program is a memory data validity test to be used after a simulated power failure.

MAINDEC-08-D1B0-D
Memory Address Test

The memory address test checks for proper memory address selection on the PDP-8/L, 8/I.

MAINDEC-08-D1EB-D
Extended Memory Checkerboard

The PDP-8/L, 8/I Extended Memory Checkerboard diagnostic is designed to provide worst case half-select noise conditions in order to determine the operational status of core memory. Four data patterns, and their complements, are written and checked for error. The patterns provided will generate the worst case noise conditions for a PDP-8/L, 8/I equipped with standard or spe-

cially purchased core stacks, and will test systems equipped with 8K words of core memory. Automatic program relocation is provided in order to test all memory stacks from each stack.

Teletype printouts are provided for error identification. Also, the operator is given a degree of control over the program by various SR settings.

MAINDEC-08-D1GB-D
Extended Memory Control
This program tests the extended memory control logic for proper operation. It may be used with a PDP-8/L, 8/I equipped with a minimum of 4K of extended memory. The program exercises and tests the control IOT's; the ability to reference all fields from field 0; program interrupt and interrupt inhibit; auto-indexing in each field, and a special test for the PDP-8/L which tests the presence of a false memory pulse when a nonexistent memory field is referenced. Errors encountered during running will result in a program halt. The halt locations are labeled, and the error may be identified by referencing the program listing or table of error halts.

MAINDEC-08-D1HA-D(D)
Extended Memory Address Test
The PDP-8/L, 8/I Extended Memory Address Test tests all of memory not occupied by the program to make sure that each location can be uniquely addressed. This is performed by a series of four tests. The first two tests write the address and complement address of each memory location into itself, and then check the contents of each location to make sure each is correct. The third test first sets all of memory not occupied by the program to all ones, and then writes a word of all zeroes, except for one bit, into each location and checks for error. The fourth test is similar except that a word of all ones, except for one bit, is written into each location and is checked for error.

MAINDEC-08-D1L0
Basic Memory Checkerboard
The memory checkerboard diagnostic tests memory for core failure on half-selected lines under worst case conditions. Its use is intended for basic 4K memory systems.

MAINDEC-08-D4A0-D
Memory Parity Checkerboard
The PDP-8/L, 8/I Memory Parity Checkerboard diagnostic test the parity bit plane for core failure on half-selected lines under worst case conditions. Its use is intended for basic 4K memory systems.

MAINDEC-08-D4BA-D
Extended Memory Parity Test
The PDP-8/L, 8/I Extended Memory Parity Test is designed to provide worst case half-select noise conditions within the parity bit plane. Four data patterns, and their complements are written, and checked for parity errors after writing each pattern are made. The program will test systems equipped with 8K words of core memory.

Operation of the program is similar to the Extended Memory Checkerboard test except that program relocation is not included, and error halts are provided for error identification.

MAINDEC-08-D2EA-D
High Speed Reader Test

This program tests the performance of the High Speed Perforated Tape Reader and Control by scanning a closed-loop test tape for accuracy of transmission. The reader control is tested for correct operation with the PDP-8/L interrupt system.

An auxiliary program included with the test punches a tape from which the test loop can be made.

MAINDEC-08-D2PD-D
Family-of-8 ASR33/35
Teletype Tests Part 1

The Family-of-8 ASR33/35 Teletype Tests' Part I is the first part of a two part package used to test the ASR33, or ASR35 Teletype when attached to a Family-of-8 system.

Part 1 contains nine selectable programs numbered from 0 to 10 (octal). The programs are selected by means of the switch register (SR).

The programs available are:

PRG0 Basic Input Logic Tests

PRG1 Basic Output Logic Tests

PRG2 Reader Test

RPG3 Test Tape Generator. Punches tape with characters stored in locations 0021 and 0022.

PRG4 Test Tape Generator. Punches Binary Count Pattern test tape.

PRG5 Reader Exerciser. Reads Binary Count pattern tape in random length blocks, and with fixed stalls between characters. The stall is determined at random.

PRG6 Reader Exerciser. Reads Binary Count pattern tape. Fixed stall between characters. Stall count is taken from LOC 0023.

PRG7 Reader Exerciser. Reads tape punched with any two test characters. Random length blocks and fixed stall between characters. The stall is determined at random.

PRG10 Reader Exerciser. Reads tape punched with any two test characters. Fixed stall between characters. Stall count taken from LOC 0023.

MAINDEC-08-D2QD-D
Family-of-8 ASR33/35
Teletype Tests, Part 2

The Family-of-8 ASR33/35 Teletype Tests, Part 2 is the second part of a two part package used to test the ASR33 or ASR35 Teletype when attached to a Family-of-8 system.

Part 2 contains nine selectable programs numbered from 0 to 10 (octal). The programs are selected by means of the switch register (SR).

The available programs are:

PRG0 Printer Test

PRG1 Punch Test

PRG2 Keyboard Test

PRG3 Combined Reader, Printer, Punch Test

PRG4 Printer Exerciser. Prints lines of characters stored in LOC 0021 and 0022. No stalls.

PRG5 Same as PRG4, but stalls between characters.

PRG6 Punch Exerciser. Punches and read checks data blocks of data stored in LOC 0021 and 0022. No stalls.

PRG7 Same as PRG6, but random stalls between characters punched.

PRG10 Punch Exerciser. Punches and read checks blocks of Binary Count pattern. Random stalls between characters punched.

MAINDEC-08-D3BB-D
TC01 Basic Exerciser

The TC01 Basic Exerciser is a series of test programs that may be used to gain a high degree of confidence in the data handling ability of a TC01 DECtape Control and one to eight TU55 DECtape Transports. The Basic Exerciser consists of several basic routines that may be individually selected; each routine will operate on any configuration of one to eight drives. These routines include a Basic Motion Routine, Search Find All Blocks Test, Basic Search Routine, Start/Stop/Turnaround Test, Basic Write/Read Data Test with eight selectable patterns, and a Parity Generation and Checking Test. The operation of the Basic Motion Routine and the Basic Search Routine are controlled by keyboard input. Also, a Write Data Scope Loop, Read Data Scope Loop, and a Search Scope are provided to keep the tape moving from end to end zone.

MAINDEC-08-D3EB-D
TC01 Extended Memory Exerciser

TC01 Extended Memory Exerciser is a test program for the PDP-8/L which test the transfer of data between the TC01 DECtape Control and extended memory fields (more than 4K). It does this by storing a data pattern in an extended memory field, tranferring the data onto DECtape and then reading the data back into the field and checking it for correct transfer.

MAINDEC-08-D3RA-D
DECTREX 1
TC01 Random Exerciser

DECTREX 1 is a DECtape Random Exerciser for the TC01 DECtape control and any configuration of one to eight TU55 DECtape transports. Drive selection, tape direction, number of blocks, sequence of operation and patterns generated are by random selection. The DECtape functions exercised are search, read data and write data in normal and continuous modes, read all in continuous mode, and move.

Also included are a short series of processor tests that are executed while waiting for interrupts and during data breaks while searching, reading, and writing from DECtape.

MAINDEC-8/1-D5BB-D
DF32 Diskless
Logic Test, MiniDisk

Diskless is a test of the DF32 disk logic and its computer interface. This pro-

gram does not test the disk, nor associated analog interface circuits. (The disk is not needed for these routines; if the disk is connected, the disk motor should be turned off. For a complete test of the disk system, use DF32 Disk Data Test.)

MAINDEC-08-D5CC-D
DF32 Disk Data
MiniDisk, Interface, Address, Data Test

The DF32 Disk Data is a complete test of the disk system. Also included is a short processor test that is executed while waiting for interrupts, and during data breaks.

MAINDEC-08-D5DB-D
Exerciser for Master and Slaves Units

"Multi Disk" is a high speed exerciser intended for multi-disk configuration and can control one to four disks.

MAINDEC-08-D6CC-D
Calcomp Plotter Test

Ths program tests the CalComp Plotter and its control. All control and plotting functions are tested.

MAINDEC-08-D6HA-D
AF04A Diagnostic and Demonstration

The Diagnostic and Demonstration program for the AF04A allows the operator to type in up to 1000_8 pseudo instructions and cause analog to digital conversions via the AF04A. The pseudo-instructions which make up the individual pseudo-program will be executed when a "$" (dollar sign) is input from the keyboard. The operator may specify all parameters of the conversion instruction and specify any order of instructions.

MAINDEC-08-D6JD-D
AD08 Diagnostic

For the AD08A this is an I/O Instruction and calibration check. For the AD08B it is also a limited test of multiplexer selection and A/D repeatability.

MAINDEC-81-D6AB-D
AX08 Diagnostic

This unit is tested in three sections (a) an instruction test of the logic, (b) a display test for the scope, (c) and a calibration section for the A/D Converter.

MAINDEC-8/I-D8AD-D
KW8I Real Time Clock

The KW8I Real Time Clock diagnostic program is designed to thoroughly test all IOT and Data transfer instructions used in the M708 Clock Control and M709 Clock Counter. The program consists of two routines; the first routine which starts at address 200, tests all flags, enables, etc., to ascertain if initialize has cleared them, and the second phase of the clock control program tests each of the IOT's to determine if they will set and/or clear each of the controllable flip-flops, it tests for proper skips, program interrupt and for proper operation of any of the three clocks. The error typeout for an error in the clock control test is as follows:

ERROR 0001

NOTE

Any clock system which supplies a clock at a frequency

of less than one clock pulse per 10 seconds will cause a failure Error 0003.

The second routine starts at address 400, and is used only when a M709 Clock Counter module is connected to the clock control. The clock (i.e. crystal) adjustable or line, must be removed from the computer in order for this test to run. The error typeout for an error in the clock control test is as follows:

CLOCK COUNTER FAILED

SENT RXED

0001 000

The SENT refers to a 12-digit number which was loaded into the clock control counter, and the RXED refers to the number which was transferred back to the computer from the counter.

MAINDEC-08-D8FA-D
DP01A Bit-Synchronous Data Communication System
IOT and Data Test

The DP01A test consists of two sequences intended to verify correct operation of all IOT instructions and associated control logic with the DP01A Bit-Synchronous Data Communication System.

MAINDEC-08-D8HB-D
DP01A Bit-Synchronous Data Communication System
IOT and Data Test for IOT's 6301 through 6354

The DP01A test consists of two test sequences intended to verify correct operation of all IOT instructions and associated control logic with the DP01A Bit-Synchronous Data Communication System.

MAINDEC-08-D8KA-D
DP01A IOT and Data Tests (60-67)

The DP01A test consists of two independent test sequences intended to verify correct operation of the IOT instructions and control logic associated with the DP01A Bit-Synchronous Data Communication System. Although the tests are treated separately, they may be in memory at the same time.

MAINDEC-08-D8LA-D
DP01A Bit-Synchronous Data Communication System IOT and Data Test for IOT-6501 through 6564

The DP01A test consists of two test sequences intended to verify correct operation of all IOT instructions and associated control logic with the DP01A Bit-Synchronous Data Communication System.

MAINDEC-08-D9AB-D
TC58 Data Reliability Test (7 Track)

The TC58 Data Reliability Test is primarily designed for the collection of statistical information pertaining to the data reliability of the tape drives that may be associated with the TC58 Magnetic Tape Control. The program is also designed to be usable as an aid to the hardware debugging and maintenance of the TC58 Magnetic Tape Control and its associated magnetic tape drives. This program may also be used as an extended data reliability acceptance test.

MAINDEC-08-D9BA-DL
TC58 Drive Function Timer

The TC58 Drive Function Timer program is designed to be an aid in the hard-

ware debugging and maintenance of the TC58 Magnetic Tape Control and its associated magnetic tape drives. The program will operate on any configuration of 1 to 8, 45 or 75 inch per second, 7 or 9 track drives. Selected operations are initiated, timed and the times are then typed in decimal milliseconds. There is no limit checking on times by the program, the decisions on the validity of times typed must be made external to the program or by the person operating this test.

MAINDEC-08-D9CB-D
TC58 Random Exerciser

The TC58 Random Exerciser Test is a test program designed to stimulate tape system usage. Any configuration of 1 through 8 TU20 (or similar) 7- and/or 9-track drives may be concurrently tested.

MAINDEC-08-D9DC-D
TC58 Instruction Test — Part 1

The TC58 Instruction Test is a series of incremental subtests designed to aid in the checkout and maintenance of the TC58 magnetic tape system.

MAINDEC-08-D9EC
TC58 Instruction Test — Part 2

The TC58 Instruction Test is a series of incremental subtests designed to aid in the checkout and maintenance of the TC58 magnetic tape system.

MAINDEC-08-D1KA-D
KP81/KR01 Power Fail Test

This diagnostic is a complete test of the PDP-8/L and PDP-8/I Power Fail option with the intervention of the operator.

MAINDEC-08-D2GE-D
Family-of-8 High-Speed Punch and Reader Tests

The Family-of-8 High-Speed Reader and Punch Tests are a test package used to test the Type PC02 and PC03 High-Speed Reader Punch when attached to any of the Family-of-8 systems. The tests perform basic input and output control logic tests, reader and punch tests, reader and punch speed printouts, and provide maintenance loops useful in adjusting the reader and punch.

MAINDEC-08-D4CA-D
PDP-8, 8/I, 8/L Incremental
Tape Compatibility Test

The PDP-8, 8/I, 8/L PEC Incremental Tape Compatibility Test verifies that the TU22, 25, 28 Incremental Write-Synchronous Read Transports generate IBM compatible format. This is accomplished by generating a test tape using pattern #1 of the TC59 Data Reliability Test. The TC59 is the control unit for TU20 transport which is IBM compatible. The test tape is then read back on the TU20 transport using the appropriate TC59 Data Reliability Test program. For operating procedures of TC59 Data Reliability Test (7 or 9 track) refer to MAINDEC-9A-D4DC-D and MAINDEC-9A-D4EB-D.

MAINDEC-08-D4EA-D (D)
TR22, 25, 28 Incremental
Tape Instruction Test

The Incremental Tape Instruction Test is designed to test all IOT instructions associated with the TR02A. The program may be used on R/W transports, and 7 and 9 track transports at any of the three standard bit densities. The program exercises each IOT, but does not test validity of the LRCC or CRC. These

characters are checked in the Data Reliability diagnostic.

The program initially tests as many functions as possible without a supply reel mounted. A supply reel is then mounted, and all remaining tape movement IOTs are checked. Instructions which the operator must follow are printed on the TTY.

MAINDEC-08-D6GC-D
A/D Calibration Check

The A/D Calibration Check for the converters is to be used to ascertain the accuracy of converter adjustments. This tape is to be used with an adjustable voltage source. The converted value will be displayed in the AC, and the switch register will be used to select multiplexer channels. (Passing of these checks do not guarantee 100% monotonicity and steady state accuracy, since all of the 4096 possible conditions are not checked.)

MAINDEC-08-D6KB-D
Display Test 34D/VC8-I

This diagnostic is designed to check out all configurations of the 34D VC8/I display, including all IOTs, display circuitry, intensity logic and light pen, if included. Under normal conditions, without a light pen, the picture which is displayed is controlled by SR0, SR1. One of four pictures will be displayed. The intensity of the display is controlled by SR3, SR4. If the light pen is connected and SR5 is set, the picture will not change unless a light pen hit is made (i.e., the light pen iris is open, the gain is properly set and the light pen is pointed toward any source of light, to which light pens are normally sensitive).

MAINDEC-08-D6MA-D-(D)
VS38 Display Diagnostic

This program was written, using MAINDEC-08-D6IA (Little Pictures for an 8) as a foundation, to test the VS38 Slave Option control logic and the slave CRTs and light pens. The program is composed of 12 display routines which exercise the CRTs to show their proper operation, and 2 diagnostic routines to diagnose VS38 control logic errors. The 12 display routines cause the images to be shown on all slave displays simultaneously, 1 diagnostic routine checks control gating with no images shown, and 1 diagnostic routine exercises the hardware of the selected slave.

MAINDEC-08-D6TA-D
AA05/AA07 Calibration Tape

These routines are to be used for the calibration of the AA05 Digital to Analog Converter and the AA07 Expander unit.

MAINDEC-08-D8EB-D
DP01A IOT And Data Tests

The DP01A test consists of two independent test sequence intended to verify correct operation of the IOT instructions and control logic associated with the DP01A Bit-Synchronous Data Communication System. Although the tests are treated separately, they may be in memory at the same time.

MAINDEC-08-D8MB
VA-38 Character Generator Test

This program tests the VA38 Character Generator option of the 338 Visual Buffered Display. The program is subdivided into four parts, each of which tests a different phase of the VA38 operation.

MAINDEC-08-D8PA
PT08 Test Program for use with Dataphone Options

This test provides a means of checking character transmission and reception using PT08 controls. Testing is performed by connecting the control output to the control input and comparing transmitted and received characters. In no way does this test supercede any existing Teletype tests; in fact, it is not necessary for checking Teletype control operation. However, when an F option is added to a PT08B or a PT08C control, this test provides sufficient testing to guarantee an acceptable production unit.

MAINDEC-08-D8SB-D
DM01 Exerciser

The DM01 Exerciser is a program written to exercise the DM01 Data Break Multiplexer to assure that it can properly interlace data breaks from several peripheral devices to the PDP-8 computer. It does this by exercising several data break devices simultaneously.

MAINDEC-08-D8TA
SIM360 — IBM S360 Channel Simulator

Program SIM360, when executed in a PDP-8 equipped with the S360 Interface Tester, simulates and monitors the control signals, command codes and data transfer sequence required for on-line IBM S360 Multiplexor or Selector Channel operation. (Read IBM Reference Manual A22-6843-3 for a complete description of selector channel and multiplexor channel characteristics and operation.)

MAINDEC-08-D8WA-D (L)
DC02 Teletype Flag Test

This program consists of three parts and will test all flags and data handling capability of the DC02 and from 1 to 12 associated teletypes.

MAINDEC-08-D8XA-D(L)
XOR Buffer Option Diagnostic for use with DP01A

The XOR Buffer Diagnostic consists of 12 tests which test power clear COB (6661), ROB (6662) and both the inclusive and exclusive OR functions.

MAINDEC-08-D9FA-D
TC58 Data Reliability Test (9 Track)

The TC58 Data Reliability Test is designed primarily for the collection of statistical information pertaining to the data reliability of the tape drives that may be associated with the TC58 Magnetic Tape Control. The program is also designed to be usable as an aid to the hardware debugging and maintenance of the TC58 Magnetic Tape Control and its associated magnetic tape drives. This program may also be used as an extended data reliability acceptance test.

MAINDEC-8I-D8AB
DC08T1: DC08 Off-Line IOT and Data Test

Program DC08T1 is designed to completely test the IOTs and data control logic associated with from 1 to 128 asynchronous data lines in an off-line configuration.

MAINDEC-8I-D8BA
DC08T2-DC08 On-Line Data Exercise

Program DC08T2 provides a DC08 data transfer exercise capable of communicating with local 5- or 8-level code data terminals such as Models 28, 33 or 35 Teletype.

The program offers data echo and data transmit operation on a prespecified mix of lines associated with a 5-level code or 8-level code data clock.

MAINDEC-8I-D8CA-D
689AGT1 — 689 AG Control and Data Test with Optional ACU Test

Program 689AGT1, used in conjunction with the 689 Modem Interface Cable and 689 ACU Simulator, provides a complete functional test of the 689AG without the need for on-line modem connections.

The program consists of three independent tests designed to check the 689 Modem Interface Control and IOTs, DC08A/689 Data Logic, and the optional ACU feature control and IOTs.

MAINDEC-8I-D8DA-D
689 AGT2-689AG On-Line Diagnostic Exercise

Program 689AGT2 provides a diagnostic exercise capable of operating with an on-line DC08 system equipped with the 689AG Modem Interface option and standard customer-supplied serial data equipment, (i.e., 103A or 103F Modems).

The program is designed to accept a number of system parameters which, in addition to specifying line numbers and modem type, define a mode of operation appropriate for the particular system configuration.

MAINDEC-T8-D8AA-D(L)
Time Sharing-8 Option Test

This is a test of the time sharing option, and is based upon switching between the executive and user modes testing all "trapable" instructions.

MAINDEC-T8-D8B8-D
TIME SHARE-8 Hardware Exerciser

TIME SHARE-8 Hardware Exerciser is a program written to assure that the hardware can function properly both in and out of time-share mode with data break devices running. It does this by exercising several data break devices simultaneously (if available) and running a program in time-share mode.

DECUS LIBRARY PROGRAM ABSTRACTS*

DECUS No. 6/8-12
PDP-8 Assembler for PDP-6

Assembles PDP-8 program written in PAL on a PDP-6 using any I/O devices.

DECUS No. 5/8-1.1
BPAK — A Binary Input/Output Package

A revision of the binary package originally written by A. D. Hause of Bell Telephone Laboratories. With BPAK the user can read in binary tapes via the photoreader and punch them out via the Teletype punch. It may be used with any in-out device, but is presently written for the photoreader and Teletype punch. A simple modification converts BPAK so that it reads from the Teletype reader if the photoreader is disabled. In its present form it occupies locations 7600 - 7777.

*Note: Programs listed as DECUS No. 5-(number) indicate that they work on the PDP-5, and compatibility to the PDP-8 is uncertain. Programs marked DECUS No. 5/8-(number) indicate they can be used on both PDP-5 and 8 computers. Those marked DECUS No. 8-(number) work on the PDP-8, and compatibility to the PDP-5 is uncertain.

DECUS No. 5-2.1
OPAK — An On-Line Debugging Program

A utility program which enables the user to load, examine, and modify computer programs by means of the Teletype. This program is a revision of the program written by A. D. Hause, Bell Telephone Laboratories. Extensive use of the program has suggested many refinements and revisions of the original program, the most significant additions being the word search and the breakpoint. The standard version of OPAK is stored in 6200 to 7577 and also 0006. An abbreviated version is available (7000 to 7577, 0006), which is identical to the other except that it has no provision for symbolic dump. Both programs are easily relocated. Control is via Teletype, with mnemonic codes, (e.g., "B" for inserting breakpoint, "P" for proceed, etc.).

DECUS No. 5-3
BRL — A Binary Relocatable Loader with Transfer Vector Options for the PDP-5

A binary loader program occupying locations 4640_8 to 6177_8, also 160 to 177. It has two main functions:

1. It allows a PDP-5 operator to read a suitably prepared binary program into any page location in memory except those occupied by BRL.
2. It greatly simplifies the calling of programmed subroutines by allowing the programmer to use an arbitrary subroutine calling sequence when writing his program, instead of having to remember the location of the subroutines.

DECUS No. 5-4
Octal Typeout of Memory Area with Format Option

(Write-up and Listing Only)

DECUS No. 5-5
Expanded Adding Machine

Expanded Adding Machine is a minimum-space version of Expensive Adding Machine (DEC-5-43-D) using a table look-up method including an error space facility.

This is a basic version to which additional control functions can easily be added. Optional vertical or horizontal format, optional storage of intermediate result without reentrys, fixed-point ouput of results within reason, and other features that can be had in little additional space under switch register control. (Write-up and Listing Only)

DECUS No. 5-6
BCD to Binary Conversion of 3-Digit Numbers

This program is based on DEC-5-4 and is intended to illustrate the use of alternative models in program construction.

While not the fastest possible, this program has one or two interesting features. It converts any 3-digit BCD-coded decimal number, $D_1D_2D_3$ into binary in the invariant time of 372 microseconds. Efficient use is made of BCD positional logic to work the conversion formula $(10D_1 + D_2)\ 10 + D_3$ by right shifts in the accumulator. In special situations, it could be profitable to insert and initial text/exit on zero, adding 12 microseconds to the time for nonzero numbers. (Write-up and Listing Only)

DECUS No. 5/8-7
Decimal to Binary Conversion by Radix Deflation on PDP-8

(Write-up and Listing Only)

DECUS No. 5/8-8
Obsolete

DECUS No. 5/8-9
Analysis of Variance PDP-5/8
An analysis of variance program for the standard PDP-5/8 configuration. The output consists of:
For each sample:

1. sample number
2. sample size
3. sample mean
4. sample variance
5. sample standard deviation

The grand mean

Analysis of Variance Table:

1. the grand mean
2. the weighted sum of squares of class means about the grand means
3. the degrees of freedom between samples
4. the variance between samples
5. the pooled sum of squares of individual values about the means of their respective classes
6. the degrees of freedom within samples
7. the variance within samples
8. the total sum of squares of deviations from the grand mean
9. the degrees of freedom
10. the total variance
11. the ratio of the variance between samples to the variance with samples.

This is the standard analysis of variance table that can be used with the F test to determine the significance, if any, of the difference between sample means. The output is also useful as a first description of the data.

Other Programs Needed: Floating Point Interpretive Package (DEC-8-5-S)

DECUS No. 5-10
Paper Tape Reader Test
A test tape can be produced and will be continuously read as an endless tape. Five kinds of errors will be detected and printed out. The Read routine is in 6033-6040.

Storage Requirements: Registers in locations 10, 11, 4-67 (save 63, 64), and 6000-7777.

DECUS No. 5-11
PDP-5 Debug System
Purpose of this program is to provide a system capable of:

1. Octal dump 1 word per line.
2. Octal dump 10_8 words per line.
3. Modifying memory using the typewriter keyboard.

4. Clearing to zero parts of memory.
5. Setting to HALT codes part of memory.
6. Entering breakpoints into a program.
7. Initiating jumps to any part of memory.
8. Punching leader on tape.
9. Punching memory on tape in RIM format.
10. Punching memory on tape in PARITY format.
11. Load memory from tape in PARITY format.

DECUS No. 5-12
Pack-Punch Processor and Reader for the PDP-5

The processor converts a standard binary-format tape into a more compressed format, with two 12-bit words contained on every three lines of tape. Checksums are punched at frequent intervals, with each origin setting, or at least every 200 words.

The reader, which occupies locations 7421 to 7577 in the memory, will load a program which is punched in the compressed format. A test for checksum error is made for each group of 200 words or less and the program will halt on error detection. Only the most recent group of words need be reloaded. Read-in time is about 10% less than for conventional binary format, but the principal advantage is that little time is lost when a checksum error is detected, no matter how long the tape.

DECUS No. 5-13
PDP-5 Assembler (for use on IBM 7044/7094)

This program accepts IBM 7044/7094 symbolic programs punched on cards and assembles them for the PDP-5. An assembly listing is produced, and a magnetic tape can be converted to paper tape and then read into the PDP-5 or it can be read directly into a PDP-5 with an IBM compatible tape unit. Cards are available.

DECUS No. 5/8-14
Dice Game for the PDP-5/8

Enables a user to play the game DICE on either the PDP-5 or PDP-8.

DECUS No. 5/8-15
ATEPO: Auto Test in Elementary Programming and Operation of a PDP-5/8 Computer.

The program will type questions or instructions to be performed by the operator of a 4K PDP-5/8. The program will check to see if the operator has answered the questions correctly. If this is the case, it will type the next question or instruction.

DECUS No. 5-16
Paper Tape Duplicator for the PDP-5

The Paper Tape Duplicator for the PDP-5 is a single buffered read and punch program utilizing the program interrupt. It computes a character count and checksum for each tape and compares with checks at the end of the tape. Checks are also computed and compared during punching.

DECUS No. 5/8-17
Type 250 Drum Transfer Routine for use on PDP-5/8

Transfers data from drum to core (Read) or core to drum (Write) via ASR33 Keyboard Control.

DECUS No. 5/8-18A
Binary Tape Disassembly Program

Disassembles a PDP-5 or 8 program which is on tape in BIN format. It prints the margin setting, address, octal contents, mnemonic interpretation (PAL) of the octal contents. A normal program or a program which uses Floating Point may be disassembled.

DECUS No. 5/8-18C
Disassembler with Symbols

This disassembler accepts a binary tape of standard format and produces a listing of the tape in PAL III mnemonics, and a cross-reference table of all addresses referenced by any memory-reference instruction. A symbol table may be entered to produce a listing similar to a PAL III Pass 3 listing. A patch to produce only a cross-reference table is included.

Minimum Hardware: PDP-8 with 4K, ASR33, High-Speed Reader, EAE

Storage Requirement: $20\text{-}1773_8$ for program, $1773\text{-}7577_8$ for scratch.

DECUS No. 8-19a
DDT-UP: Octal-Symbolic Debugging Program

DDT-UP is an octal-symbolic debugging program for a 4K PDP-8 which occupies locations 5600-7667. The mnemonics for the eight basic instructions are defined internal to this area. Other symbols are stored, four locations per symbol, from 5577 down towards 0000. The mnemonics for the standard OPR and IOT group instructions are initially defined in this area. Thus, the highest location initially avalable to the user is 5363. Beginning at this location the user may define symbols one at a time using the comma (,) operator.

From the Teletype, the user can symbolically examine and modify the contents of any memory location. DDT-UP allows the user to punch a corrected program in CBL format.

DDT-UP has a breakpoint facility to help the user run sections of his program. When this facility is used the debugger also uses location 0005.

DECUS No. 5/8-20
Remote Operator FORTRAN System

Program modification and instructions to make the FORTRAN OTS version dated 2/12/65 operate from remote stations.

DECUS No. 5/8-21
Triple Precision Arithmetic Package for the PDP-5/8

An arithmetic package to operate on 36-bit signed integers. The operations are add, subtract, multiply, divide, input conversion, and output conversion. The largest integer which may be represented is $2^{35} - 1$ or 10 decimal digits. The routines simulated a 36-bit (3 word) accumulator in core locations 40, 41 and 42 and a 36-bit multiplier quotient register in core locations 43, 44 and 45. Aside from the few locations in page 0, the routines use less core storage space than the equivalent double precision routines.

DECUS No. 5-22
DECtape Duplicate (552)

A DECtape routine for the PDP-5 to transfer all of one reel (transport 1) to another (transport 2). Occupies one page of memory beginning at 7400. The last page of memory is not used during the operation of the program; however, the memory from 1 to 7436 is used to set the DECtape reels in the proper

starting attitude and is then destroyed during duplication. Duplication will commence after which both reels will rewind. Parity error will cause the program to halt with 0040 in the accumulator.

DECUS No. 5/8-23A
PDP-5/8 Oscilloscope Symbol Generator (4 x 6 Matrix)

The subroutine may be called to write a string of characters, a pair of characters, or a single character on an oscilloscope. Seventy (octal) symbols in ASCII Trimmed Code and four special 'format" commands are acceptable to this routine. The program is operated in a fashion similar to the DEC Teletype Output Package.

DECUS No. 5/8-23B
PDP-5/8 Oscilloscope Symbol Generator (5 x 7 Matrix)

This subroutine may be called to write a string of characters, a pair of characters, or a single character on a 34D Oscilloscope. Twenty-six alphabetic characters and 0 - 9 numeric characters are acceptable. However, there is space available to include any symbol the user desires. The program is operated in a fashion similar to the DEC Teletype Output Package (Digital 8-19-U).

Source Language: MACRO-8

Storage Requirement: 200_8-777_8 registers

DECUS No. 5-25
A Pseudo Random Number Generator

The random number generator subroutine, when called repeatedly, will return a sequence of 12-bit numbers which, though deterministic, appears to be drawn from a random sequence uniform over the interval 0000_8 to 7777_8. Successive numbers will be found statistically uncorrelated. The sequence will not repeat itself until it has been called over 4 billion times.

DECUS No. 8-26A
Compressed Binary Loader (CBL)

The CBL (Compressed Binary Loader) format in contrast to BIN format utilizes all eight information channels of the tape, thus achieving nearly 25% in time savings.

As BIN tapes include only one checksum at the end of the tape, CBL tapes are divided into many independent blocks, each of which includes its own checksum. Each block has an initial loading address for the block and a word count of the number of words to be loaded.

Storage Requirement: 7700-7777

DECUS No. 8-26B.1
CBL2BN and BN2CBL

CBL2BN is a short utility program which converts paper tape in CBL format to BIN and BN2CBL converts paper tape from BIN to CBL format. It offers high or low speed I/O and proper punching of field characters.

Program Language: PDPMAP — similar to PAL

Storage Requirement: 300_8 and 200_8 Buffer;
400_8 and 200_8 Buffer

DECUS No. 8-26C
XCBL: Extended Memory CBL Loader

XCBL is used to load binary tapes punched in CBL format into a PDP-8 with more than standard 4K memory. This loader occupies locations 7670 through 7777 of any memory field.

DECUS No. 8-26D
XCBL Punch Program

This program permits a user to prepare an XCBL tape of protions of a PDP-8 extended memory through the control the keyboard of the on-line Teletype.

The program is loaded by the BIN Loader.

There are two versions of the program so that any section of memory may be punched:

LOW XCBL occupies 00000-00377 and its starting address is 00000.

HIGH XCBL occupies 17200-17577 and its starting address is 17200.

If bit 0 is a ONE, the operation of XCBL PUNCH is similar to that of DDT-UP (DECUS No. 8-19a).

DECUS No. 5/8-27 and 5/8-27a
Bootstrap Loader and Absolute Memory Clear

Bootstrap Loader inserts a bootstrap loading program in page 0 from a minimum of toggled instructions.

Absolute Memory Clear leaves the machine in an absolutely clear state and, therefore, cycling around memory obeying an AND instruction with location zero. Should not be used unless one plans to reinsert the loader program.

DECUS No. 5/8-28a
PAL III Modifications — Phoenix Assembler

This modification of the PAL III Assembler speeds up assembly on the ASR33/35 and operates only with this I/O device. Operation is essentially the same as PAL III, except that an additional pass has been added, Pass 0. This pass, started in the usual manner, but with the switches set to zero, reads the symbolic tape into a core buffer area. Subsequent passes then read the tape image from storage instead of from the Teletype.

DECUS No. 5/8-29
BCD to Binary Conversion Subroutines

These two subroutines improve upon the DEC-supplied conversion routine. Comparison cannot be made to the DECUS-supplied fixed-time conversions, DECUS No. 5-6, because it is specified only for the PDP-5. One routine is designed for minimal storage, the other for minimal time. Both are fixed-time conversions; time specified is for a 1.5μs machine.

Execution Time: Minimal Time routine — 73.6μs, 32 locations

Minimal Storage routine — 85μs, 29 locations

DEC routine — 64 — 237μs, 37 locations

DECUS No. 5-30
GENPLOT: General Plotting Subroutines for the PDP-5

This self-contained subroutine is for the PDP-5 with a 4K memory and a CalComp incremental plotter. The subroutine can move (with the pen in the

up position) to locations (x, y), make an "x" at this location, draw a line from this present position to location (x, y) and initialize the program location counters.

DECUS No. 5-31
FORPLOT: FORTRAN Plotting Program for PDP-5

FORPLOT is a general-purpose plotting program for the PDP-5 computer in conjunction with the CalComp 560 Plotter. It is self-contained and occupies memory locations 0000_8 to 4177_8. FORPLOT accepts decimal data input on paper tape in either fixed or floating point formats. Formats can be mixed at will. PDP-5 FORTRAN output tapes are acceptable directly and any comments on these are filtered out.

Storage Requirement: $0000\text{-}4177_8$

DECUS No. 5/8-32a
Program to Relocate and Pack Programs in Binary Format

This program provides a means to shuffle machine language programs around in memory to make the most efficient use of computer storage.

DECUS No. 5/8-33
Tape to Memory Comparator

Tape to Memory Comparator is a debugging program which allows comparison of the computer memory with a binary tape. It is particularly useful for detecting reader problems, or during stages of debugging a new program. Presently uses high-speed reader, but may be modified for TTY reader.

DECUS No. 5-34
Memory Halt — A PDP-5 Program to Store Halt in Most of Memory

With Memory Halt and OPAK in memory, it is possible to store Halt (7402) in the following memory locations: 0001 to 0005, 0007 to 6177, and 7402 to 7403.

Other Programs Needed: OPAK (DECUS No. 5-2.1)

DECUS No. 8/8-35
BCD to Binary Conversion Subroutine and Binary to BCD Subroutine (Double Precision)

This program consists of a pair of relatively simple and straightforward double precision conversions.

DECUS No. 5-36
Octal Memory Dump Revised

The Octal Memory Dump on Teletype is a DEC routine (DEC-5-8-U) which dumps memory by reading the switch register twice; once for a lower unit and again for an upper limit. It then types an address, the contents of the program and the next three locations, issues a CR/LF, then repeats the process for the next four locations. This leaves the right two-thirds of the Teletype page unused. The 78_{10} instructions occupy two pages.

This revised routine uses the complete width of the Teletype page and occupies only one memory page, using less paper and two less instructions. Now an address and the contents of 15 locations are typed out before a carriage return.

DECUS No. 5-37
Transfer II

For users who have more than one memory bank attached, the PDP-5/8,

Transfer II may prove valuable in moving information from one field to another. When debugging, Transfer II enables a programmer to make a few changes in a new program and test it without reading in the original program again. Transfer II enables more extensive use of memory banks.

DECUS No. 5/8-38
FTYPE: Fractional Signed Decimal Type-In

Enables a user to type fractions of the form: .582, -.73, etc., which will be interpreted as sign plus 11 bits (e.g., 0.5 - 2000_8). Subroutine reads into 300-3177 and is easily relocated, as it will work on any page without modifications.

DECUS No. 5/8-39
DSDPRINT, DDTYPE: Double-Precision Signed Decimal Input-Output Package

DSDPRINT, when given a signed 24-bit integer, types a space or minus sign, and then a 7-digit decimal number in the range −8388608 to +8388607. DDTYPE enables a user to type in a signed decimal number in either single or double precision. These routines are already separately available, but the present subroutine package occupies only one memory page and allows for more efficient memory allocation. Location in 3000-3177, but will work on any page.

DECUS No. 5-40
ICS DECtape Routines (One-Page 552 Control)

The routines will read or write from the specified DECtape unit and delay the program until all I/O is completed. The last block read will overflow the specified region and destroy one core location. Only standard 129 word DECtape blocks will be read or written. The routines will halt if an error occurs with the status bits in the AC.

DECUS No. 5-41
Breakpoint

This debugging routine has been reduced to a minimum operation. It is a mobile routine which can operate around any program that leaves an extra 30 cells of memory space.

Its function is to insert breakpoints in any given location of the program being debugged, and to hold the contents of AC and Link. The programmer may examine any locations desired and then continue to the next breakpoint. It is presently located in 140_8 - 170_8, but may be easily relocated.

Storage Requirement: 140_8 - 170_8

DECUS No. 5-42
Alphanumeric Input

With the Alphanumeric Input Package, any character may be read into the PDP-5 through either the Teletype or the high-speed reader. The characters are packed two per cell and stored in the address indicated in the switch register.

DECUS No. 5/8-43
Unsigned Octal — Decimal Fraction Conversion

This routine accepts a four-digit octal fraction in the accumulator and prints it out as an N-digit decimal fraction where N=12 unless otherwise specified. After N digits, the fraction is truncated. Programs are included for use on the PDP-5 with Type 153 Automatic Multiply-Divide and the PDP-8 with Type

182 Extended Arithmetic Element.

Storage Requirement: 55_8 locations for the PDP-5

47_8 locations for the PDP-8

DECUS No. 8-44
Modifications to the Fixed Point Output in the PDP-8 Floating Point Package (Digital 8-5-S)

The Floating Point Package (Digital 8-5-S) includes an Output Controller which allows output in fixed point as well as floating point format. This Output Controller takes the form of a certain number of patches to the "Floating Output E Format" routine, plus an additional page of coding.

Using the Calculator program (Digital 8-10-S), which includes the Floating Point Package, certain deficiences were noted in the fixed-point output format, particularly the lack of any automatic rounding off.

This new version of the Output Controller is also in the form of patches to the Floating Output with an additional page of coding, thereby not increasing the size of the Floating Point Package.

The following summarizes this new version:

1. The number output is automatically rounded off to the last digit printed. or the sixth significant digit, whichever is reached first. Floating point output is rounded off to six significant figures since the seventh is usually meaningless.
2. A number less than one is printed with a zero preceding the decimal point (e.g., +0.5 instead of +.5).
3. A zero result, after rounding off, is printed as 0 instead of +.
4. The basic Floating Point Package includes the facility to specify a carriage return/line feed after the number, using location 55 as a flag for this purpose. The patches for the Output Controller caused this facility to be lost. This version restores the facility.

DECUS No. 5/8-45
PDP-5/8 Remote and Time-Shared System

A time-shared programming system which allows remote stations immediate access to the computer and a wide selection of programs.

DECUS No. 5/8-46a
PDP-5/8 Utility Programs

Consists of seven programs (listed below) each of which may be selected via the teletypewriter. When the program is started, either by a self-starting binary loader or by manually starting the computer in address 200_8, it is in its executive mode. In this mode, it will respond only to eight keys and perform the following functions:

B—go to BIN to QK Converter Program

E—go to Editor Program

F—FORTRAN Tape formatter

L—type a section of leader and stay in executive

N—go to Editor Program without typing leader

P—go to Page Format Program

T—Assembly language tape formatter

Q—go to QK to BIN Converter Program

DECUS No. 8-47
ALBIN — A PDP-8 Loader for Relocatable Binary Programs

ALBIN is a simple method for constructing relocatable binary formatted programs, using the PAL III Assembler. Allocation of these programs can be varied in units of one memory page (128_{10} registers). When loading an ALBIN program, the actual absolute addresses of indicated program elements (e.g., the keypoint of subroutines) are noted down in fixed program-specified location on page zero. In order to make a DEC symbolic program suitable for translation into its relocatable binary equivalent, minor changes are required which, however, do not influence the length of the program. Due to its similarity to the standard DEC BIN loader, the ALBIN loader is also able to read-in normal DEC binary tapes. ALBIN requires 122_{10} locations, RIM loader included. Piling-up in core memory of ALBIN programs stored on conventional or DECtape can be achieved using the same method with some modifications.

DECUS No. 5/8-48
Modified Binary Loader MKIV

The Mark IV Loader was developed to accomplish four objectives:

1. Incorporate the self-starting format described in DECUS 5/8-27, ERC Boot.
2. Select the reader in use, automatically, without switch register settings.
3. Enable a newly-prepared binary tape to be checked prior to loading by calculating the checksum.
4. Reduce the storage requirements for the loader so that a special program would fit on the last page of memory with it.

DECUS No. 8-49
Relativistic Dynamics

Prints tables for the relativistic particle collisions and decay in the same format as the Oxford Kinematic Tables. It can be used in two ways:

1. Two-particle Collisions — Given the masses of incident, target, and emitted particles, the incident energy and centre-of-mass angles, the program calculates angles and energies of the emitted particles in the Lab frame. If the process is forbidden energetically, program outputs "E" allowing the threshold energy to be found.
2. Single Particle Decays by specifying M2=0 (target), the problem will be treated as a decay, and tables similar to the above will be printed.

DECUS No. 5/8-50
Additions to Symbolic Tape Format Generator (DEC-8-21-4)

These routines allow the user to perform further useful functions by the addition of a few octal patches. By making the appropriate octal patches via the toggles, the Format Generator can also format FORTRAN tapes, shorten tape by converting space to tabs, and convert the type of tape.

DECUS No. 5/8-51
Character Packing and Unpacking Routines

ASCII characters may be packed two to a word and recovered. Control

characters are also packable but are preceded by a 37 before being packed into the buffer.

Storage requirement: 63_{10} words

DECUS No. 8-52
Tiny Tape Editor

This Tiny Tape Character Editor fits in core at the same time as the PAL III or MACRO-8 Assemblers. A tape may be duplicated at three speeds and stopped at any character for insertion or deletion. The toggle switches control the speed and the functions desired.

Storage Requirement: 72_{10} registers

DECUS No. 5/8-53
COPCAT (DECtape Copy 552)

COPCAT is a tape-to-tape copy routine for the PDP-5 and PDP-8 DECtape (552 Control).

DECUS No. 5/8-54
Tic-Tac-Toe Learning Program

This program plays Tic-Tac-Toe basing its moves on stored descriptions of previously lost games. The main program is written in FORTRAN. There is a short subroutine written in PAL II used to print out the Tic-Tac-Toe board. The program comes already educated with about 32 lost games scored.

Other Programs Needed: FORTRAN Object Time System

DECUS No. 5/8-55
PALEX — An On-Line Debugging Program for the PDP-5/8

One problem with programs written in Program Assembly Language (PAL) for operation on a PDP-5/8 computer is the danger of an untested program being self-destructive, running wild, destroying other programs residing in memory such as loading programs. PALEX prevents any of the above unwanted operations from occurring while it gives the operator-programmer valuable debugging information and enables him to make changes in his program and try out the modified program. Once running, PALEX cannot be destroyed by any program or instruction in memory, the operator need not touch any manual console controls, and all required information is printed in easy-to-read format on the Teletype console.

DECUS No. 8-56
Fixed Point Trace No. 1

A minimum size monitor program which executes the users' program one instruction at a time and reports the contents of the program counter, the octal instruction, the contents of the accumulator and link and the contents of the effective address by means of the ASR33 Teletype.

Storage Requirement: two pages

DECUS No. 8-57
Fixed Point Trace No. 2

Similar to Fixed Point No. 1 (DECUS No. 8-56) except that the symbolic tape provided has a single origin setting instruction of 6000. Any four consecutive memory pages can be used, with the exception of page zero, by changing this one instruction.

DECUS No. 8-58
One-Page DECtape Routine (552 Control)

A general-purpose program for reading, writing, and searching of magnetic tape. This program was written for the Type 552 Control. It has many advantages over both the standard DEC routines and also over the DECUS No. 5-46. The routines are one page long and can be operated with the interrupt on or off. The DEC program delays the calling program while waiting for the unit and movement delays to time-out. This routine returns control to the calling program. This saves ¼ second every time the tape searches forward and half that time when it reverses. In addition, it will read and write block 0. This program is an advantage over the previous one-page routines in that it allows interrupt operations, does not overflow by one location, interrupts the end zone correctly and not as an error, and provides a calling sequence identical to the DEC program.

DECUS No. 8-59
PALDT — PAL Modifications for DECtape (552 Control)

When assembling programs, PALDT requires that the symbolic tape be read in only once. The program writes on the library tape itself after finding the next available block from the directory. During pass 0 the tape is read-in using the entire user's symbol table. During passes, 1, 2, 3, as much of the symbol table is used as possible. This means the fewest tape passes possible. As an added advantage, pass 0 ignores blank tapes, leader-trailer, line feeds, form feeds, and rubouts; thereby saving space. The whole program decreases the user's symbol table by only three pages: one for the DECtape program above, one for pass 0, and one for the minimal length read in buffer.

DECUS No. 8-60
Square Root Function by Subtraction Reduction

A single-precision square root routine using EAE. This routine is usually faster than the DEC routine and can easily be modified for double precision calculation at only twice the computation time.

DECUS No. 8-61
Improvement to Digital 8-9-F Square Root

An improved version of the DEC Single-Precision Square Root Routine (without EAE). Saves a few words of storage; execution is speeded up 12 percent.

DECUS No. 5-63
SBUG-4

SBUG-4 allows the PDP-5 to execute one instruction of any given program at a time, returning to SBUG-4 following each instruction and printing out the contents of various registers. This permits following the path of a program which has gone astray or examining some defective operation.

DECUS No. 5/8-64
DECtape Programming System (552 and TC01 Controls)

This program provides rapid access to DEC software and utilizes routines through the use of DECtape. Programs may be stored, edited, assembled, listed, or executed without reliance upon paper tape.

May be used with both TC01 and 552 DECtape Controls.

DECUS No. 8-65
Editor Modified for DECtape (552)

This program consists of modifications to the Digital 8-1-S Symbolic Editor to enable reading and writing on DECtape. This results in considerable time savings in assembling PAL programs since PAL has also been modified to accept the symbolic program directly from DECtape. The DECtape compatibility is also useful for storing text for later use and for regaining Editor memory space lost due to delete and change commands.

DECUS No. 8-66
Editor Modified for DECtape (552)

This program consists of modifications to the Digital 8-1-S Symbolic Editor to enable reading and writing on DECtape. This results in considerable time savings in assembling PAL programs since PAL has also been modified to accept the symbolic program directly from DECtape. The DECtape compatibility is also useful for storing text for later use and for regaining Editor memory space lost due to delete and change commands.

In addition, the overflow detection routine is now foolproof and results in a HALT.

Storage Requirement: Editor <0, 1461>

Modifications: <1462, 1502>
<6376, 7177>

DECtape Routines: <7200, 7577>

Minimum Hardware: PDP-8 with EAE, ASR33, DECtape

DECUS No. 8-67
PAL Modified for DECtape Input (552)

This program is a modification to the Digital 8-3L-S PAL Assembly Program enabling PAL to obtain the symbolic program from DECtape (in addition to paper tape), and output the assembled program in the usual manner. (The symbolic program is written onto DECtape by use of the "Editor Modified for DECtape" Program.) This modification also makes it possible to assemble sections or commands from the keyboard with those from DECtape. The resulting assembly is limited in speed mainly by the punching of the assembled program during Pass 2; and Pass 1 is speeded considerably. Also included is a tabulator interpreter, providing Pass 3 listings in tabulated format.

Storage Requirement: PAL III — <0, 3561> plus symbol table

Modifications — <6555, 7177>

DECtape Routines — <6555, 7177>

Minimum Hardware: PDP-8 with EAE, ASR33, DECtape

DECUS No. 8-68a
LABEL for PDP-8

The LABEL Program punches labels for paper tapes on the Teletype punch. When a key on the Teletype keyboard is depressed, no echo is performed, but a few characters of tape are punched which form the outline of the character associated with the key. Outlines are punched for all characters whose code is between 240 and 337.

Store Requirement: Locations 200-677 of any memory field. 400-677 of Readable Punch.

DECUS No. 5/8-69
LESQ29 and LESQ11

The purpose of the program is to fit the best sequence of parabolas to a given 400 point data curve in order to remove extraneous noise; rather than rely on a single 400 point parabola least squares fit to approximate a given data curve. Approximately 400 individual parabolas are computed as follows:

LESQ29

Data values 1 through 29 are subjected to a second order least squares fit. The median point of the resulting parabola (point #15) is then substituted for the original data value #15.

A second parabola is then computed using data values 2 through 30. The median point of this parabola (point #16) is then substituted for point #16 of the original data curve.

This procedure is repeated until all data values have been replaced (except for the first and last 14 points which are excluded by the mechanics of the operation).

LESQ11

Process identical to LESQ29 except that an 11 rather than a 29 point smooth interval is used. First point replaced is point #6, and only the first and last 5 points are excluded from smoothing.

LESQ11 will preserve higher frequency data than LESQ29 for a given data curve with constant time between data points.

Minimum Hardware:	4K PDP-5 or PDP-8 Teletypewriter (plotter, DECtape optional)
Other Programs Needed:	Floating Point Interpretive Package (Digital 8-5-S) and appropriate data handling routines.
Storage Requirement:	LESQ11: 400-564; 700-716 LESQ29: 400-564; 700-751
Execution Time:	(PDP-5) LESQ11: 1 minute LESQ29: 2.5 minutes
Restrictions:	Positive integer data $<3777_8$; time between data points constant.

DECUS No. 8-70
EAE Routines for FORTRAN Operating System (DEC-08-CFA3)

These are two binary patches to the FORTRAN Operating System which utilizes the Type 182 EAE hardware for single precision multiplication and normalization, replacing the software routines in FOSSIL (the operating system). The binary tape is loaded by the BIN Loader after FOSSIL has been loaded. Execution time of a Guass-Jordan matrix inversion is reduced by approximately 30%.

Minimum Hardware: PDP-8 with Type 182 EAE

Other Programs Needed: FORTRAN Operating System DEC-08-CFA3-PB Dated March 2, 1967

DECUS No. 8-71
Perpetual Calendar

The program is designed as a computer demonstration. When a valid date is fed into the computer, the corresponding day of the week is typed out. The program is based on the Gregorian Calendar and is limited to years between 1500 and 4095.

Minimum Hardware: PDP-8 with an ASR33 Teletype

Storage Requirement: 20-1333

DECUS No. 8-72

Matrix Inversion, Real Numbers

The program inverts a matrix, up to size 12 x 12, of real numbers. The algorithm used is the Gauss-Jordan method. A unit vector of appropriate size is generated internally at each stage. Following the Gauss sweep-out, the matrix is shifted in storage, another unit vector is generated and the calculation proceeds.

Minimum Hardware: PDP-8

Other Programs Needed: FORTRAN Compiler and FORTRAN Operating System

Storage Requirement: Uses all of core not used by the FORTRAN Operating System.

Execution Time: Actual computation takes less than 10 seconds. Data read-in and read-out may take up to five minutes.

DECUS No. 8-73
Matrix Inversion, Complex Numbers

The program inverts a matrix, up to size 6 x 6, of complex numbers. The algorithm used is the Gauss-Jordan method, programmed to carry out complex number calculations. A unit vector of appropriate size is generated internally. Following the Gauss sweep-out, the matrix is shifted, another unit vector is generated, and the calculation proceeds.

Other Programs Needed: FORTRAN Compiler and FORTRAN Operating System

Storage Requirement: This program uses essentially all core not used by the FORTRAN Operating System

Execution Time: Actual computation takes less than 10 seconds. Data read-in may take up to five minutes.

DECUS No. 8-74

Solution of System of Linear Equations AX-B, by Matrix Inversion and Vector Manipulation

This program solves the set of linear algebraic equations AX-B by inverting matrix A using a Gauss-Jordan method. When the inverse matrix has been calculated, it is printed out. At that point, the program requests the B-vector entries. After read-in of the B-vector, the product is computed and printed out. The program then loops back to request another B-vector, allowing the system to solve many sets of B-vectors without the need to invert matrix A again. Maximum size is 8 x 8.

Other Programs Needed: FORTRAN Compiler and FORTRAN Operating System

Storage Requirement: This program uses essentially all of core not used by the FORTRAN Operating System.

DECUS No. 8-75

Matrix Multiplication — Including Conforming Rectangular Matrices

This program multiplies two matrices, not necessarily square but which conform for multiplication.

Other Programs Needed: FORTRAN Operating System and FORTRAN Compiler

DECUS No. 8-76

PDP NAVIG 2/2

This program utilizes the output of the U. S. Navy's AN/SRN-9 satellite navigation receiver to obtain fixes on a PDP-8 or PDP-8/S. This program, except for some details of input and output, follows very closely NAVIG2 written for the IBM 1620, which in turn is derived from the TRIDON program written at the Applied Physics Laboratory of Johns Hopkins University for the IBM 7090.

PDP-NAVIG2/2 is written in PAL III for 4K machine with ASR33. Floating point numbers using two 12-bit words as mantissa and one 12-bit word as exponent are employed.

Restriction: The accuracy is slightly less than that using 7 decimal digits per word.

Minimum Hardware: PDP-8 with an ASR33

DECUS No. 8-77

The purpose of this system is to expedite the programming of multi-processing problems on the PDP-8 and PDP-8/S. It maximizes both the input speed and the portion of real time actually used for calculations by allowing the program to run during the intervals between issuing I/O commands and the raising of the device flag to signal completion of the command. The technique also allows queuing of input data or commands so that the user need not wait while his last line is being processed, and so that each line of input may be processed as fast as possible regardless of its length. The system uses the interrupt facilities and has less than 3% overhead on the PDP-8/S (about 0.1% on the PDP-8).

This method is especially useful for a slower machine where the problem may

easily be calculation limited but would, without such a system, become I/O bound.

The program may also be easily extended to handle input from an A/D converter. Here, the input would be buffered by groups of readings terminated either arbitrarily in groups of N or by zero crossings.

This program can increase the I/O to computation efficiency of some programs by 100%. It can do this even for a single Teletype. Each user will probably want to tailor the program to his individual needs.

Storage Requirement: 600_8 registers for two TTYs plus buffer space. (Several device configurations are possible.)

DECUS No. 8-78
Diagnose: A Versatile Trace Routine for the PDP-8 with EAE

This trace routine will track down logical errors in a program (the "sick" program). Starting at any convenient location in the "sick" program, instructions are executed, one at a time, and a record of all operations is printed out via the Teletype. To avoid tracing proven subroutines, an option is provided to omit subroutine tracing. The present routine is significantly more versatile than two other trace routines in the DECUS Library for the PDP-8 (DECUS Nos. 8-56 and 8-57) in that it is able to trace "sick" programs containing floating point, extended arithmetic and a variety of input/output instructions. Diagnose is, however, at a disadvantage compared with DECUS No. 8-56 in requiring more memory space (five pages as opposed to two); and compared with DECUS No. 8-57 in not possessing the trace-supppression features of the latter. The mode of operation of Diagnose is quite different from the other trace routines.

Minimum Hardware: PDP-8 with EAE

Other Programs Needed: Floating Point Package needed for floating point tracing. (DEC-8-5-S)

Miscellaneous: Program is relocatable

DECUS No. 8-79
TIC-TAC-TOE (Trinity College Version)

This TIC-TAC-TOE game is programmed, using internal logic, so that the computer will either win or stalemate, but not lose a game. At the termination of a game, the program restarts for the next game.

DECUS No. 8-80
Determination of Real Eigenvalues of a Real Matrix

This is a two-part program for determining the real eigenvalues of a real-valued matrix. The matrix does not have to be symmetric. Part I uses the power method of iterating on an eigenvector to determine the largest eigenvalue of the matrix. Part II then deflates the matrix using the results of Part I so as to produce a matrix of order one less than that solved for in Part I. Part I can then be reloaded, and the next eigenvalue in line may be calculated. In this, all the real eigenvalues may be computed in order.

DECUS No. 8-81
A BIN or RIM Format Data or Program Tape Generator

This program enables a PDP-8 operator to generate tapes under Teletype control in RIM or PAL BIN format without formal assembly, assuming the operator knows the octal codes corresponding to each instruction. This is

particularly useful when one is dealing with small programs for testing interface equipment or when making small modifications to larger programs saving reassembling time. Tapes generated using this program can be appended to existing BIN or RIM tapes and can then be loaded with the original tape into core with the appropriate leader. Another use of this program is in the preparation of data tapes in RIM or BIN format so that data can be loaded directly into PDP-8 core via the usual loaders. The program also generates leader/trailer code and a checksum under program control.

Storage Requirement: Locations 6000-6077.

DECUS No. 8-82
Library System for 580 Magnetic Tape (Preliminary Version)

The system provides for storing program files (or other files) on the 580 Magnetic Tape with PDP-8, and recalling them at will without altering the state of the rest of the computer. In general principle, it is similar to the DECtape Library System, and the only effective storage requirement is the last page of memory.

At present, the system consists of three programs known as BOOTSTRAP 1, BOOTSTRAP 2, and the LIBRARY Routines.

BOOTSTRAP 1 is a minimal loader program which resides in the last page of memory. Its function is to rewind the tape and load BOOTSTRAP 2 into the last page, automatically transferring control to it. Bootstrap saves the area of core to be used by the system as a record on the magnetic tape, loads the LIBRARY Routines into core, and transfers control to them.

The LIBRARY Routines comprise a Directory of the files on tape, an Input-Output package enabling communication with the Teletype, and four system programs:

LIst	Types out the names of files in the Directory
CAll	Transfers a file into core and exits
DUmp	Writes a file on tape, rewrites the Directory, and exits
EXit	Restores the computer to its original state, with BOOTSTRAP 1 and BIN on the last page.

The magnetic tape subroutine and some control functions are included in BOOTSTRAP 2. Each entry in the directory consists of three words: the name of the file, its first location in core, and the number of words it occupies. The capacity of the directory is 22_{10} entries.

DECUS No. 5/8-83A & B
Octal Debugging Package (with and without Floating Point)

This program is an on-line debugger which will communicate with the operator through the ASR33 Teletype. It allows register examination and modification, octal dumping, binary punching, multiple and simultaneous breakpoints, starting a program, and running at a particular location with preset AC and link. ODP is completely relocatable at the beginning of all pages except page zero, and is compatible with the PDP-5, the PDP-8, and the PDP-8/S.

Storage Requirement:	The high version of ODP requires locations 7000-7577. The low version requires locations 0200-0777. All versions will require three pages. Also, location 0002 is used for a breakpoint pointer to ODP.
Minimum Hardware:	The standard PDP-5, 8, or 8/S, with ASR33 Teletype is required. A high-speed punch is optional.

DECUS No. 8-84
One-Pass PAL III

This is a modification to Digital 8-3L-S. For use on an 8K PDP-8 with ASR33. The principle of the modification is to store the incoming characters during Pass 1 into the memory extension and take them from there during Pass 2 and 3. Source programs must be limited to 4095 characters. This modification can save about 40% of assembly time.

Operation of the program is the same as for PAL III except that the reading of the source program for Pass 2 and 3 need not be repeated. For these passes, one simply presses CONTINUE after setting the correct switches.

Restriction: The program does not work with high-speed reader and punch.

DECUS No. 5/8-85
Set Memory Equal to Anything

This program will preset all locations to any desired settings; thus combining a memory clear, set memory equal to HALT, etc. into a single program. The program is loaded via the switch registers into core.

DECUS No. 8-87
XMAP

This program types out the contents of the DECtape directory on TTY keyboard. The list includes the name of the program, its initial block number, the amount of blocks used, the starting address and the locations of the program in core. The above restriction is only a format restriction due to the line length on the TTY unit. At present, this program is operational only with the TC01 control; however, the symbolic version may be modified for use with the 552 control.

Storage Requirement: 0000-1232, 6000-6577 (directory)

Restrictions: Each program on tape is assumed to occupy no more than three successive sequences of memory pages.

DECUS No. 8-88
DECtape Symbolic Format Generator

These are DECtape versions of the Symbolic Tape Format Generator, Digital 8-21-U, that operate under the DECtape Programming System, DECUS 5/8-64. They provide neat format for symbolic files generated with XEDIT, and a means to get symbolic programs out on paper. They compact a program containing extra spaces and give the number of blocks actually used in the output file. The library tape is executable on TC01 equipment only, but instructions are given for altering it for 552 equipment.

Minimum Hardware: PDP-8 with TC01 Control

Other Programs Needed: XRDCT, XWDCT, XBUFF, (DECUS No. 5/8-64)

DECUS No. 8-89
XOD: Extended Octal Debugging Program

XOD is an octal debugging program for a PDP-8 with extended memory which preserves the status of the program interrupt system at breakpoints. The program occupies locations 6430 through 7577 of any memory field. From the on-line Teletype, the user can examine and modify the contents of any memory location. Positive and negative block searches with a mask may also be performed.
XOD includes an elaborate breakpoint facility to help the user run sections

of his program. When this facility is used, the debugger also uses locations 0005, 0006, and 0007 of every memory field.
Restrictions: The ability to punch binary tapes is not included in XOD.

DECUS No. 8-90
Histogram on Teletype

This routine plots histograms on the Teletype when there is no CRT display available or no means of making a permanent copy of a CRT display. Input to the routine consists of a vertical scaling factor, the size of the table to be plotted (limited only by the size of the Teletype print line), the starting address of two core areas: one containing the data to be plotted, and one for use as temporary storage by the machine.
Storage Requirement: 128_{10} words plus tables

DECUS No. 8-91
MICRO-8: An On-Line Assembler

MICRO-8 is a short assembler program for the PDP-8 that translates typed mnemonic instructions into the appropriate binary code and places them in specified memory locations immediately ready to function. It processes the typed instructions by a table look-up procedure.

It is especially useful for programs of less than one page which are to be run immediately. Only octal (not symbolic) addresses may be specified, but the user has control of the zero page and indirect addressing bits. An octal typeout routine permits examination of any memory location.

Storage Requirement: 3200-4200
Restrictions: MICRO-8 is quite capable of modifying itself.

DECUS No. 8-92
Analysis of Pulse-Height Analyzer Test Data with a Small Computer

This PDP-8 computer program is used in the evaluation of test data for multichannel pulse-height analyzers. The program determines integral and differential nonlinearities and examines smooth spectra of radioactive decay.

DECUS No. 8-93
CHEW — Convert Any BCD to Binary — Double Precision

This subroutine converts a double precision (6-digit) unsigned-integral binary-coded decimal (BCD) number with bit values of 4, 2, 2, and 1 to its integral-positive-binary equivalent in two computer words. It is possible to change the bit values to any desired values and thereby convert any BCD number to binary.

Storage Requirement: 0109_{10}

DECUS No. 8-94A
BLACKJACK

This program enables a person to play BLACKJACK with the computer. The computer acts as dealer and keeps track of bets, cards played, etc.

Storage Requirement: $0\text{-}3777_8$

Minimum Hardware: PDP-8 with EAE

DECUS No. 8-94B
Patch for BLACKJACK

This patch contains two overlays for BLACKJACK (DECUS No. 8-94A). The first eliminates the need for the EAE hardware, the second allows one to

"double down" on any two cards with the instruction "D" (0 response to "HIT?" is made invalid).

Minimum Hardware: PDP-8, 8/S, or 8/I

Other Programs Needed: DECUS No. 8-94A

DECUS No. 8-95
TRACE for EAE

TRACE interpretively executes a PDP-8 program. At the same time, a printout is provided of the contents of the program counter, the instruction, the link, accumulator, and multiplier-quotient registers, and where applicable the effective address and the contents of the effective address. This printout may be for all or a selected type of instruction within selected memory bounds. The program is capable of handling any PDP-8 instruction including IOT, two-word EAE, and interrupt instructions. TRACE cannot be destroyed by the program being traced while TRACE is in control.

Minimum Hardware: PDP-8 with Type 182 EAE, ASR33 Teletype

Storage Requirement: 400_8 or 500_8 locations.

DECUS No. 8-96
J Bessel Function (FORTRAN)

This program computes the J Bessel Function for a given argument and order. It is complete PDP-8 FORTRAN program that operates in a conversational mode.

Other Programs Needed: FORTRAN Compiler/Operating System

DECUS No. 8-97
GOOF

A one page program which allows insertion of instruction (xxxx) in location (nnnn) by means of the TTY keyboard. A feature of automatically incrementing the current address permits rapid insertion of blocks of data or instructions. Typing RUB-OUT reinitializes the program.

Storage Requirement: 175_8 locations (1 page)

DECUS No. 8-98
3D DRAW for the 338

This program is a demonstration of the capabilities of the 338 system. The program allows the user to sketch three dimensional objects on the scope and rotate them in real time.
Minimum Hardware: PDP-8 with 338 Display

DECUS No. 8-99A
Kaleidoscope

The program creates pictures on the PDP-8 or PDP-8/S with 34D Display. They are varied by manipulating the sense switches (within the range 0000 — 0007). The program was submitted without comments by an anonymous donor.

DECUS No. 8-99B
Kaleidoscope — 338

The program creates varied pictures by manipulating the buttons of the 338 Display pushbutton bank (within the range 0000-0007).
Storage Requirement: 200_8 - 274_8

DECUS No. 8-100
Double-Precision Binary Coded Decimal Arithmetic Package

Consists of the following routines:

BCDADD — The single-precision BCD addition routine is the basic component of the BCD arithmetic package. This routine functions simply by masking out and adding together corresponding BCD digits (i.e., four bits) and checking for carry (i.e., when the sum of two four-bit numbers is greater than 9 (1001)).

MPYBCD — This routine multiplies a single precision (three digit) number times a double precision one to produce another double precision number. Overflow is indicated in the link; the arguments are not affected.

SUBBCD — One double precision BCD number is subtracted from a second by this routine. It uses a 9's complement routine and the double precision add routine.

DOLOUT — Special formats: ("$XXXX•YY "); ("XXXXXX "); (3 nonprinting data codes); ("XXX ").

DECUS No. 8-101
Symbolic Editor with View

This program is an extended version of the standard PDP-8 Symbolic Editor (high-speed I/O) program. One extra command has been added, "V", which takes the lines specified by the arguments and displays them on the CRT (338). The program, otherwise, operates in the same way as the Editor.

The following pushbutton options are provided:

0: Count Up Scale

1: Count Down Scale

2: Count Up Intensity

3: Count Down Intensity

Minimum Hardware: 8K PDP-8, 338 CRT, and VC38 Character Generator

DECUS No. 8-102a
A LISP Interpreter for the PDP-8

LISP is a programming language for list manipulation. The system is particularly suitable for conversational use and teaching. There are very few restrictions to the language apart from the total storage space. More than half of the storage is used as list space.

Minimum Hardware: 4K PDP-8 and ASR33

DECUS No. 8-103A
Four Word Floating Point Function Package

This program package, written for use with Digital's Four Word Floating Point Package (DEC-08-FMHA-PB), includes subroutines to evaluate square, square root, sine, cosine, arctangent, natural logarithm, and exponential functions.

DECUS No. 8-103B
Four Word Floating Point Rudimentary Calculator

This is a minimum space program to perform calculations with the 10.5 decimal place precision of Digital's Four Word Floating Point Package (DEC-08-FMHA-PB), and uses the Four Word Floating Point Function Package

(DECUS 8-103A). Operations are performed in the sequence in which they are entered. One storage register is provided. Up to five user-defined operation routines may be called.

DECUS No. 8-103C
Four Word Floating Point Output Controller with Rounding

This subprogram is almost identical to the output controller for the Three Word Floating Point Package (Digital 8-5-S) with the rounding addition (DECUS No. 8-44) except that the Four Word Floating Point Package (DEC-08-FMHA-PB) is used.

DECUS No. 8-103D
Additional Instructions for use with Four Word Floating Point Package

These subroutines allow the Four Word Floating Point Interpreter to perform the operations: read a floating point number, skip positive floating point accumulator, skip zero floating point accumulator, no operation, unconditional jump, negate floating point accumulator, and halt. The two skip instructions and the jump instruction allow forward or backward jumping up to 15 locations of instructions.

DECUS No. 8-104
Card Reader Subroutine for the PDP-8 FORTRAN Compiler

Modifications and additions which allow the PDP-8 FORTRAN Compiler to read source programs from cards. The standard FORTRAN card format is used with only minor modifications.

Minimum Hardware: 8K PDP-8 and a Type CR01C Card Reader

DECUS No. 8-105
D-BUG

D-BUG is an aid used in debugging PDP-8 programs by facilitating communication with the program being run. Communication between operator and program is via the ASR33 Teletype. D-BUG is similar to DEC's ODT II (DEC-08-COA1-PB); however, it uses the DEC Floating Point Interpreter (Digital 8-5-S).

Two modes of operation are possible, fixed and floating point. D-BUG features include register examination and modification, control transfer, octal dumping, and instruction trap-outs to D-BUG control. Registers containing floating point numbers may also be examined, and break-traps can be inserted in floating point programs.

DECUS No. 8-106
Readable Punch

This program enables the user to type a character on the keyboard and produce the character in readable form on paper tape. The program uses the high-speed punch. The readable characters on tape are produced by means of a table which contains the format of a 6 x 5 matrix using three words of storage per character to be punched. In addition, channel 8 is punched throughout. The program is terminated by typing a carriage return which generates 6 inches of tape. (Reference DECUS No. 8-68A).

DECUS No. 8-107
CHESSBOARD Display on the 338

This program displays a chessboard on the screen of a DEC 338 Display with all thirty-two chessmen set up on their initial board positions. There is no

provision to move them about the board; it is just a demonstration picture.

Storage Requirement: $03000 - 04320_8$

Minimum Hardware: PDP-8 with 338 Display

DECUS No. 8-108
INCMOD: Increment Mode Compiler (338)

The INCMOD program for the DEC 338 Display allows the user to build a display subroutine composed of increments only. The user inputs information by pointing with the light pen. The program displays the figure he is constructing in each of the four available scale settings. The program is of value as a demonstration and may be of help for maintenance purposes. It occupies locations 0000- 1231 and builds the increment mode display file beginning at location 1232.

Storage Requirement: $0000\text{-}1231_8$

Minimum Hardware: PDP-8 with 338 Display

DECUS No. 8-109
SEETXT Subroutine (338)

SEETXT is a subroutine for the DEC 338 Display which can be called instead of the normal typeout subroutine. In addition to typing, it displays all printed characters on the screen corresponding to the last twenty lines which have been typed out.

The program includes the option of suppressing the typing so that output can occur at a much higher rate than ten characters per second. The user has the option of controlling the length of a decay loop in the subroutine so that output rate may range from nearly immediate to Teletype rate.

The maximum number of lines displayed, the scale, and intensity may be altered at any time. There is also the option of clearing the screen of displaying a blinking marker at the current typing position.

Minimum Hardware: PDP-8 with 338 Display

DECUS No. 8-110
DIREC: Directory Print

DIREC is a system program to be used with the PDP-8 Disk Monitor System. The program lists an index of the file directly for the disk on the on-line Teletype. The user has the option of seeing the index to system files or user files, or both.

DIREC can also be used in conjunction with the SEETXT Subroutines for the 338 Display to obtain a listing of the directory on the display screen.

Other Programs Needed: Disk Monitor System

Minimum Hardware: PDP-8 with Disk

DECUS No. 8-111
DISKLOOK

DISKLOOK is a small utility program for a PDP-8 with a 32K DF32 Disk. Using the on-line Teletype, the user may examine and alter any location (in octal) on the disk. Masked searches are also available.

Minimum Hardware: PDP-8 with 32K DF32 Disk

Storage Requirement: $200 - 777_8$

DECUS No. 8-112
Sentence Generator

This program generates random English language sentences, using a dictionary (provided by the user) of ten basic word groups (A-J). The dictionary is used in conjunction with a random number generator and a syntactical algorithm to provide an output of randomly constructed English language sentences.

The program is an excellent vehicle for computer demonstration purposes. It may also be used in English teaching programs to aid students in perceiving sentence structure and errors in the use of words.

DECUS No. 8-113
Conversion of Friden (EIA) to ASCII

This program will translate tapes prepared on a Friden Flexowriter (EIA) into ASCII for direct assembly, further editing, or feeding into the FORTRAN program. Alphabetic characters may be in either upper or lower case. The program uses a table lookup and comparison with the negative complement ASCII character.

Storage Requirement: 213_8 including 2 autoindex registers

DECUS No. 8-114
Rounded Decimal Output Modification for PDP-8 FORTRAN

The program loads over the PDP-8 FORTRAN Operating System (DEC-08-AFA3-PB) and provides output in conventional decimal form: rounded, aligned, and with plus sign, leading zeros (other than one, in the case of fractional numbers), and trailing decimal point replaced by spaces. The FORTRAN trigonometrical routines are over-written. The source program must begin with two statements assigning integer variables representing, respectively, the numbers of digits required to the right of the decimal point, and the total number of digits (these can be reassigned, by program or manually). Output is called in the normal way, i.e. by TYPE statements referring to FORMAT statements containing the symbol E. If output of a number is not possible in the format requested, the decimal point is shifted to the right in the field; if formatted output is still impossible, or if zero or negative total digits were requested, output reverts to "E" format.

Restrictions: FORTRAN source language programs must begin with two special statements defining format required.

DECUS No. 8-115a
Double Precision Integer Interpretive Package

This program is similar in operation to the Floating Point Package (Digital 8-5-S). It consists of addition, subtraction, multiplication, division, load, store, jump and branch subroutines coupled to an interpreter. It allows direct and indirect addressing in the normal assembly language manner. The operation is faster and more compact than the collected individual double precision subroutines.

Minimum Hardware: PDP-5, 8, 8/S, or 8/I.

Storage Requirement: 14 words in page 0 and an additional 2 page of memory

DECUS No. 8-116
PDP-8 Automatic Magnetic Tape Control (Type 57A) Library System

The PDP-8 Automatic Magnetic Tape Control (Type 57A) Library System is a series of bootstrap programs which load library programs into memory from an IBM-compatible magnetic tape read using a Type 57A Automatic Magnetic Tape Control. A program is selected by entering the appropriate code number into the switch register on the computer console.
A copy of the IBM-compatible library tape may be obtained by sending a ½ inch magnetic tape to the author.

DECUS No. 8-117
A PDP-8 Interface for a Charged-Particle Nuclear Physics Experiment

Documentation (only) describing an interface constructed to use a PDP-8 computer with a charged-particle detector system employing three solid-state detectors and flight-time analysis. Up to 48 bits from each randomly-occurring event are transferred through the data (break) channel to a hardware-selected buffer region in the core of a PDP-8 computer. Designed for use as a magnetic tape analyzer for the most complex cases, the system assumes that the 48 bits originate in flag bits set by fast logic and in (presently four) amplitude digitizers, all of which are assumed to contain information for the same event. The system includes some limited capability for controlling the course of the experiment, and provides for read-out through the computer of a series of external fast counters. The report summarizes the design concepts, shows schematic flow diagrams, defines the computer instructions associated with the interface system, and gives simple model programs to illustrate methods of applications.

DECUS No. 8-118
General Linear Regression

The major section of this program is the "Main Arithmetic IX" which consists of four initializing statements, an input section; a weighting section; a section which cumulates means, sums of squares, etc.; a section which cumulates the relevant regression coefficients, etc.; and a section which calculates confidence limits as variances.

The section which calculates the relevant regression coefficients allows for both cases of linear regression, and in the computation of standard error of the intercept, uses (N-2) degrees of freedom to provide a better estimate for small values of N while providing negligible differences from conventional calculation when N is larger.

The section which calculates confidence limits as variances provides a calculation of the variance of the error of the estimate of the dependent variable again using (N-2) degrees of freedom for the general case. This calculation is fully corrected for both random variance within the tested population of data and for the difference between the independent variable and the mean of the independent variable for the population of data.

DECUS No. 8-119
Off-Line TIC-TAC-TOE (PAL)

TIC-TAC-TOE is a self-learning program which will improve its game as it plays. Whenever its human opponent wins, the program changes its strategy such that it can never be beaten again in the same way. Thus, the program gains "experience" every time it loses. The program will punch its experience on paper tape in binary format on request. This experience tape can be reread by the program at any time and will reset the program to the level of experi-

ence it had when the tape was punched. The program will notify the operator if any error is made in reading the experience tape and gets very upset if the player tries to cheat.

Storage Requirement: Locations 10-4000 (approximately)
Will operate with low or high speed tape input/output equipment.

Minimum Hardware: PDP-8, ASR33, or high-speed reader and punch

DECUS No. 8-120
Disk/DECtape FAILSAFE

This program will punch the contents of the disk (or DECtape) onto paper tape which can be loaded back onto the disk using the same program. The paper tape is punched in 200_8 word blocks in binary format, with a checksum for each block. FAILSAFE simplifies and speeds the process of rebuilding the Disk System Monitor after running disk tests.

Minimum Hardware: PDP-8, 8/S, 8/I, with 32K Disk or DECtape

DECUS No. 8-121
DECtape Handler (552 DECtape)

This program allows quick, controlled data-block transfers between the PDP-8 and DECtape. It reads, writes and searches in minimum time (interrupt mode), requires minimum space (overlay with lastpage BIN, RIM, DECSYS Loaders) and occupies only two blocks on tape (block 0 = System, block 1 = Return-System). It is protected against destruction and gives, after the transfer, the status levels for testing purposes. It is usable as a Switch Register controlled program or as a subroutine with or without interrupt, giving the possibility of quick data storage, program shuffling and overlay technique with PDP-8 and DECtape.

Minimum Hardware: PDP-8, DECtape 552 Control

DECUS No. 8-122A
SNAP: Simplified Numerical Analysis Program with EAE

SNAP is a computer language for real-time interactive computation which can be learned in less than one hour. It is particularly useful in teaching programming to beginners.

A unique feature of SNAP is its ability to interact on-line with other laboratory instruments. SNAP can accept electrical inputs directly and can read inputs from a real-time clock. Both of these functions are incorporated in a single SNAP instruction.

Another feature particularly useful for biological problems is Table Instructions. A list of 100 numbers may be entered from the keyboard or from punched paper tape.

DECUS No. 8-122B
SNAP: Simplified Numerical Analysis Program Without EAE

See 8-122A for Abstract

DECUS No. 8-123
UNIDEC Assembler

The UNIDEC Assembler runs on the Univac 1108 and passes assembled PDP-8 code over the electronic link between the 1108 and PDP-8. The source statements are punched on cards for input into the 1108 in a format nearly

identical to that of MACRO-8. A printed listing and object code are produced as fast as the cards can be read.
Note: Source deck and documentation only available.

DECUS No. 8-124
PDP-8 Assembler for IBM 360/50 and above

The 360/PDP-8 Assembler is a collection of programs written mostly in FORTRAN IV (G) which operates on the IBM 360/50 and above. It assembles programs for PDP-5 and PDP-8 computers. Once a program has been assembled, it may be punched on cards, saved in a file, or transmitted through the Data Concentrator over data lines. It is also possible to obtain binary paper tapes by use of the Data Concentrator.
The Assembler follows the PAL III operation code and addressing convention. The input format and program listing conventions are slightly different from those of PAL III, because it is organized around a line format, while PAL III is organized around a paper tape format.
Note: Source deck and documentation only available.

DECUS No. 8-125
PDP-8 Relocatable Assembler for IBM 360/50 and above

The documentation available describes a method for segmenting PDP-8 programs for the purpose of facilitating program maintenance and residence in MTS (Michigan Terminal System) files. The method provides for program storage on a page-relocatable basis with relocation information continuous to but not necessarily integral with text information. Linkages between separately assembled program segments are provided in a form very similar to those used in IBM System/360 systems.

Currently available utilities within MTS provide assembly and link-editing facilities, using programs stored either as punched card decks or in MTS files. Utilities are also included for the purpose of paper tape transcription either in PAL-compatible format or in a special format useful for dynamic loading via a data link to a remote machine. In addition to these MTS utilities, two relocating PDP-8 loaders are available which operate using the special dynamic-loading format. Each of these programs occupies one dedicated page of PDP-8 memory and operates in a multicore-bank environment. One of these programs is designed to operate as a stand-alone utility, while the other is designed to operate within the RAMP system.

DECUS No. 5/8-126
Cumulative Gaussian Distribution Curve Fitting

This is a curve fitting program that will take a set of any number of points with any spacing describing a cumulative Gaussian distribution and determine the mean and standard deviation by an iterative least squares differential-correction technique. The mean square error of the final curve is also computer.

Other Programs Needed: FORTRAN Compiler and Operating System

Source Language: PDP-5/8 FORTRAN

DECUS No. 8-127
XDDT: Extended Octal-Symbolic Debugging Program

XDDT is an octal-symbolic debugging program for extended memory which preserves the status of the program interrupt system at breakponts. It is the result of merging the features of the DDT-UP (DECUS No. 8-19a) and XOD

(DECUS No. 8-89) debugging programs, and it includes many improvements over its predecessors.

From the Teletype, the user can symbolically examine and modify the contents of any memory location in a variety of formats. Positive and negative block searches with a mask may also be performed.

XDDT includes an elaborate single-breakpoint facility to help the user run sections of his program.

The ability to punch binary tapes is not included in XDDT.

Minimum Hardware:	Basic PDP-8, but recommended for 8K or more
Source Language:	PDPMAP
Storage Requirement:	With initial symbol table, 4200-7577

DECUS No. 8-128
PDP-8 Oscilloscope Display of Mathematical Function

This is a general-purpose FORTRAN program for oscilloscope display of single-valued functions, y = f(x). The FORTRAN statement of the function can be changed by the user so as to display specific functions of interest to the user. The user must specify a range for the independent variable. Scaling of the function for an appropriate display is carried out automatically by the program. The user may then interrupt the display to respecify the range of either independent or dependent variable. The display will be flicker-free on a conventional (nonstore) oscilloscope.

Minimum Hardware:	4K PDP-8, Type 34D Display Unit
Other Programs Needed:	FORTRAN compiler and operating system, PAL Assembler
Source Language:	FORTRAN (main program), PAL (subroutine)

DECUS No. 8-129
PDP-8/57A Magnetic Tape Program Library System

Programs may be written on and called off IBM-compatible tape by name from the Teletype. BIN and RIM loaders may also be called in from the Teletype. Only the last page of core is used. Library programs may be corrected, modified, or added to at any time. When called in, programs may be relocated in core. It is possible to subdivide programs as they are written on tape and then individually relocate each portion as it is loaded in.

Minimum Hardware:	Standard PDP-8, 57A Tape Control
Source Language:	MACRO-8
Storage Requirement:	7600-7777

DECUS No. 8-130A
REBIL8: Relocating Binary Tape Loader for the PDP-8/S

Sections of the DEC-08-LBAA-LA Binary Loader have been rewritten to extend its duties to loading of suitably prepared relocatable binary program tapes as well as address and data modifications. Requirements are the same as the standard DEC loader, and REBIL8 will load standard DEC binary tapes.

Minimum Hardware:	PDP-8/S and ASR33
Source Language:	MACRO-8

DECUS No. 8-130B
RELCON: Binary to Relocatable Binary Tape Converter

RELCON is used to tag data, used by memory reference instructions for indirect addressing, with the Data Modification Mark (376 Code). It may also be used to adjust addresses so that the relocatable version begins loading memory at page 0 if no address modification is specified. This does not mean that the program will operate in this area of memory but serves to simplify address specification at load time.

DECUS No. 8-131
SRCD: Software Rapid Character Display

SRCD (Software Rapid Character Display) is not a program but a method for quick display of the maximum number of text characters. A listing of increment-mode command words is supplied for the sixty-four characters on the Teletype keyboard. Each character is drawn with a 5 x 7 dot matrix followed by two blank points to provide spacing. It is useful for displaying buffers of text, such as for editing programs or in utilization of the display as a satellite processor in time-sharing systems. In these applications, the PDP-8 is frequently sitting in a loop, "listening" for keyboard characters, or simply doing nothing. With SRCD, the main frame is constantly engaged in background work, helping to display characters, and I/O handling for interrupt servicing routines.

DECUS No. 8-132
STRIP: A Data Display and Analysis Program for the PDP-8, 8/I

This program accepts paper tape data listings and displays the result on the display unit. Some elementary computations are made on the data and are also displayed. The program is deliberately designed to be open ended, and most users will want to add features peculiar to their own problem. Almost all functions are carried out in subroutine form; these subroutines can be called either from the keyboard or within another subroutine.

DECUS No. 8-133
First Order Kinetics

First order kinetic processes are common in chemistry and in other areas. This program accepts up to 42 data points, calculates the rate constant and intercept by the method of least squares, and give the rms deviation, the correlation coefficient, and an estimate of the error in slope. It permits graphical (CRT) examination of deviations from the least squares line and iteration to a "best" infinity value. It also provides options for plotting the deviation between observed and calculated quantities on a CRT and may be used in other cases in which one wishes to correlate the natural logarithm of one quantity with another, as in linear free-energy relationships.

Source Language: MACRO-8

Storage Requirement: The program occupies essentially all of core.

DECUS No. 8-134
LSQ: Least Squares Subroutine

The subroutine calculates the slope and intercept for the equation $y_i + mx_i = b$ by the method of least squares. It also returns the rms deviation of y, the correlation coefficient and an estimate of the error in the slope. The calculated values of Y and the differences between the given and calculated values are also available on return from the subroutine.

Other Programs Needed:	FLOAT, floating point interpreter "C" — (Digital 08-YQYA)
Source Language:	MACRO-8
Storage Requirement:	1.5 pages plus page 0 locations

DECUS No. 8-135
DNHELP: Directory Assistor Program

DNHELP is a four-page disk utility program that may reside in core with DIREC (DECUS No. 8-110) and DISLOOK (DECUS No. 8-111). It is designed to assist programmers in investigating the contents of the DN and SAM blocks on the disk under the DEC Disk Monitor System.

Minimum Hardware:	PDP-8 with DF32 Disk or TC01 DECtape.
Other Programs Needed:	System Monitor Head — DEC Disk (Tape) Monitor System and SYSIO.
Storage Requirement:	5000-5777, Buffer from 7400-7577

DECUS No. 8-136
Fourier Transform Program in FORTRAN II

The program, written in PDP-8 FORTRAN II, performs the discrete Fourier Transform of a function defined over N($N \geq 220$) evenly spaced points. I/O is via the ASR33. The program requests the number of function points, then that number of function values, and then prints out the values of the sine and cosine components at the function of each defined harmonic. A conventional (not Cooley-Tukey) algorithm is used since I/O time relative to computing time is significant.

DECUS No. 8-137
Programs for Storage, Manipulation and Calculation for Data Using DECtape

These programs use DECtape for the storage of data files. Once data has been stored on DECtape, the statistical or calculation programs will operate on particular parts of it selected by the user. All programs are conversant. They ask questions regarding execution and accept answers via the Teletype.

DATRIT is a program to write data on DECtape directly from the ASR33. Numerical data is stored on DECtape in floating point format.

EDATA is a program to edit data files created on DECtape by DATRIT.

SDT is a program to calculate mean and standard deviation from data files stored on DECtape.

FORT calculates an analysis of variance table similar to DECUS No. 5/8-9 using data files stored on DECtape.

COVAR calculates the necessary values for an analysis of covariance from data files stored on DECtape. The paired input consists of matching files of x and y data.

LCOVAR is a semilogarithmic version of COVAR. y values are converted to log y before calculation so that each "Y" in the output format means log y. This program is useful for semilogarithmic regression analysis.

TPAIR performs a paired T test on data files stored on DECtape. The input consists of paired files of x and y data.

BCALC enables the user to perform calculations using data files on DECtape as variables in the calculation. Results of the calculation are stored on DECtape. BCALC is a master program for handling the data files. The user must supply a floating point program, which is called by BCALC as a subroutine, for each specific calculation.

LCALC enables the user to perform calculations from data stored on DECtape using specific lines of a file as variables in the calculations. The result of the calculation may be stored on one line of the same file or on a different file. LCALC is similar to BCALC.

SUBS is a package of four subroutines used by most of these programs. SUBS contains six pointers on page zero and subroutines in the area from 4000 to 7577.

These subroutines are:

MESSAGE	Type packed text
UNFLOAT	Unfloat floating point numbers
RWTAPE	Read and write DECtape
FPOINT	Floating point output controller

FLEX is an extended version of Floating Point which lacks the Output Controller. It is used to overlay the FPOINT section of SUBS in the programs which use extended Floating Point.

DECUS No. 8-138
PAL III.5

PAL III.5 features several modes making pass 3 output more legible. The line number feature makes subsequent editing significantly easier. With the exception of SR bit 2, normal operating instructions apply. The tapes are complete (not an overlay to PAL III).

DECUS No. 8-139
Editor

This editor is a program which adheres fairly closely to DEC traditions in the area of text editors with major exceptions, such as line members, line specification, buffer capacity, and availability.

Storage Requirement: 0_8-2507_8

DECUS No. 8-140
Binary Tape Consolidator

The Binary Tape Consolidator is an extremely useful system-generation tool. It will make a single tape from any number of smaller ones and may be used for duplicating tapes.

DECUS No. 8-141
SYSLUK

SYSLUK is a four-page utility program for examining and modifying blocks on the system I/O device; i.e., DF32 Disk or TC01 DECtape. Its operation is inde-

pendent of which monitor head is resident, provided either is there. The user has the facility to examine and modify locations and to perform masked searches.

Minimum Hardware:	PDP-8 with DF32 Disk or TC01 DECtape
Other Programs Needed:	SYSIO — "system device" routine for DEC Disk (Tape) Monitor System
Source Language:	MACRO-8
Storage Requirement:	200-1177 (buffer from 7377-7577)

DECUS No. 8-142
Binary Punch — Extended Memory

This program is a revision of Digital 8-5-U Binary Punch which allows for extended memory. Tapes produced may be loaded by Digital 8-2-U Binary Loader.

Source Language:	MACRO-8
Storage Requirement:	7600-7730

DECUS No. 8-143
FFTS-R: Fast Fourier Transform Subroutine for Real Valued Functions

This subroutine computes the Fast Fourier Transform (FFT) or its inverse of a data sequence which has been stored in core. It will accommodate up to 2048 time samples and will transform that number in under 5 seconds.

Minimum Hardware:	PDP-8 or 8/I with EAE
Source Language:	PAL
Storage Requirement:	3-7, 20-107, 400-6401

DECUS No. 8-144
FFTS-C: Fast Fourier Transform Subroutine for Complex Data

FFTS-C enables computation of the discrete Fourier transformation in a minimum amount of time. By using the Cooley-Tukey algorithm, up to 1024 points may be transformed in only 4.5 seconds, introducing a reduction of 99 percent in computation time.

Minimum Hardware:	PDP-8 or PDP-8/I
Source Language:	PAL III
Storage Requirement:	3-7, 20-55, 400-5777

DECUS No. 8-145
A Time-of-Flight Analyzer Based on a Small On-Line Computer

This program enables the computer to interact with the TOF-converter and to generate spectrum displays on an oscilloscope. The TOF Converter provides the computer with digital information about the time a neutron takes to travel from the scattering sample to a detector (up to 12 detectors can be accommodated) and which detector was involved.

The TOF analyzer for which this program was written is in use with a double chopper facility installed at the ISPRA-I reactor. It consists of a PDP-8 on-line

computer with 4K memory, the automatic restart option, and a display unit; a TOF Converter; and conventional counting electronics.

DECUS No. 8-146
High Speed Interrupt Executive

These routines are designed to handle the priority scheduling of up to 12 interruptable devices. Each I/O device is assigned a priority level, and upon receipt of an interrupt from that device, execution of its routine in initialized. If the prority of an I/O device "x" is less than that of an I/O device "y" which is currently being serviced, device "x" will be queued until "y" has been serviced. These routines allow a user to prohibit interrupts on any (or all) levels.

Minimum Hardware: PDP-8 with EAE

Source Language: MACRO-8

Storage Requirement: Three memory pages

DECUS No. 8-147
Incremental Plotter Printout Subroutines

A group of subroutines providing character-output facilities for the incremental plotter is presented as a package. Virtually all the ASCII characters may be printed in any of 8 formats and 63 sizes. One routine sets a control code to determine the size and orientation of the characters and the direction the line is to run, another prints out a string of characters according to this code, a third prints just one character held in AC 6-11, and a fourth routine prints the signed decimal equivalent of the contents of the accumulator.

Minimum Hardware: PDP-8, Type 350 Plotter and Control

Source Language: Digital 8-12-U "Incremental Plotter Subroutine

Storage Requirement: Five memory pages (1177 locations)

DECUS No. 8-148
Plotter System

This is a generalized plotting system for the CalComp Plotter allowing "plot time" modification of the data. The main program tape accepts all plotting commands and data from the Teletype. Patch tape #1 modifies the system to a high-speed reader. Patch tape #2 modifies all input through the high-speed reader.

Minimum Hardware: PDP-8, CalComp Plotter, and High-Speed Reader

Source Language: MACRO-8

Storage Requirement: Five memory pages (1177 locations)

DECUS No. 8-149
Core Window

The 34D Scope displays the octal contents of any 64 consecutive core locations, beginning at the address set in the Switch Register and Data Field switches (if Extended Memory is used). There are 16 lines, each with an address plus four memory words. A special character generator program refreshes the display 11 times per second.

Minimum Hardware: PDP-8, 34D Scope

Source Language: MACRO-8

Storage Requirement: 15; 7240_8-7573_8

DECUS No. 8-150
PTOD8 High and PTOD8 Low

PTOD8 (PTT Trace and Octal On-Line Debugging Program for the PDP-8, is a means to debug a running users program. It features register examination and modification, multiple breakpoints (traps), memory protection of a chosen block, word search (masked or not masked), tracing a running users program (gives a full printout of consequently executed instructions), interrupt proof, and also features binary tape punching (automatic leader-trailer code and checksum).

Storage Requirement: PTOD8 requires 1343_8 registers
PTOD8 HIGH: 6200_8-7543_8,
PTOD8 LOW: 200_8-1543_8

DECUS No. 8-151
On-Line TIC-TAC-TOE

This program plays the game of TIC-TAC-TOE with the user. By means of a previously stored algorithm, it selects the best move for any given situation. Conversation and ultimate defeat is via the Teletype.

Source Language: PAL

DECUS No. 8-152
PDP-8 Music Program

The coding program allows the user to type a song on the Teletype and produce a coded binary tape of that song. It accepts musical information in a form more compatible with ordinary sheet music and converts it to a coding scheme. The playing program plays the song "Penny Lane" via the coding program with the use of a power amplifier and speaker.

Minimum Hardware: PDP-8 with D/A Converter, Power Amplifier and Speaker

Source Language: PAL III

Restrictions: 6577_8 notes

DECUS No. 8-153
Tape/Disk Transfer Programs

This series of programs was written to create and recall disk images on magnetic tape. They were written initially to facilitate rebuilding the disk system in the event of an accidential or deliberate wipe out. The usefulness of the DF32 was significantly enhanced by the ability to store and easily recall a number of different disk images. A single reel of DECtape can hold up to five complete images, each of which occupies 400_8 blocks.

Minimum Hardware: PDP-8 with DF32 Disk and TC01 DECtape

Source Language: PAL III

DECUS No. 8-154
SWAP

Using self-contained I/O, SWAP may be employed to load the disk from or dump the disk onto DECtape. It is faster and more versatile than the Disk to DECtape FAILSAFE.

Storage Requirement: PDP-8 with DF32 and TC01 DECtape

Minimum Hardware: "PDPMAP" (similar to PAL)

Source Language: 600_8 to 4200_8 buffer

DECUS No. 8-155
HEP

HEP is a program which gives calculating machine type operation and stored program operation. It is based on Floating Point Package (DEC-8-5-S-D) and Floating Point Controller (DECUS No. 8-44). Calculations have an accuracy of just over six decimal digits and printout is routined to six decimal digits. It includes facilities for format control, program control and tests, subroutines, and an array of variables. Although it was designed mainly for quick results from small calculations, it also has facilities and space for quite large and eliborate programs. Note, HEPTRACE, DECUS No. 8-156.

Minimum Hardware: PDP-8 with Teleprinter

Storage Requirement: 0003_8-7577_8

DECUS No. 8-156
HEPTRACE

This program is used in conjunction with HEP (DECUS No. 8-155) to give trace and one shot facility during the execution of HEP programs.

DECUS No. 8-157
Square Root Patch for DEC-8-5-S

This program is a patch to the standard SQRT routine (DEC-8-5-S). It is a shorter and faster way of giving exact roots of exact squares.

Storage Requirement: 6656_8-6747_8

DECUS No. 8-158
AX08 Symbol Generator

This subroutine may be called to display single characters of a string of characters on the oscilloscope of an AX08 (LAB-8) system. Sixty different symbols, in addition to four special "format" codes, are provided by the routine. Software control of character scaling and "margins" on the display is provided.

Minimum Hardware: LAB-8 with oscilloscope

Source Language: PAL

Storage Requirement: 223_{10} locations

DECUS No. 8-159
CINET-BASIC

This interpretive compiler was patterned after Dartmouth's BASIC. It was built by modifying DEC's FOCAL, and uses many of the same subroutines and/or methods. Error messages are given in terms of an error number and line number.

Storage Requirement: Main program locations 0000-3252 and 4600-7600 and user's code from 3252 on.

Minimum Hardware: A PDP-8 with 4K and ASR-33

DECUS No. 8-160
FASTLOAD

FASTLOAD is a minimal bootstrap loader for the PDP-8 requiring only eight instructions to load in the upper page of memory.

DECUS No. 8-161
EXPO: A Flexible PDP-8 Data Acquisition Program

EXPO is a PDP-8 program which reads various kinds of data from experimental apparatus, optionally logs data on magnetic tape, and accumulates one- or two-dimensional histograms of selected variables. These histograms may be displayed on the Teletype or scope, simultaneous with data acquisition. From the keyboard the user defines what variables are to be histogrammed and under what conditions: variable names are symbolic and numerical parameters are decimal. Also from the keyboard, the user may call for Teletype or scope output with some control of format. Because of its flexible user-oriented input-output, EXPO has proven to be very useful in debugging and utilizing complex apparatus in a high-energy physics experiment; it is likely to be useful in similar experimental situations in science or engineering. The write-up includes a useful general discussion of interrupt handling on the PDP-8.

Minimum Hardware:	4K PDP-8 with EAE, Magtape, Scope Display, and Plotter optional
Source Language:	LRL Assembler (DECUS No. 5-13) on cards
Storage Requirement:	0-7177 if all options used

DECUS No. 8-162
Demonstration Programs for the PDP-8

1. Fran the Barmaid — Focal
2. World War 1 — Snoopy
3. PDP-8 Music
4. Night Watchman's Clock
5. Matching Pennies
6. Hangman Game — Focal

DECUS No. 8-163
FOCAL Routines

Most FOCAL programs listed here are distributed in write-up form. Some are available with paper tape. Please refer to numerical index for specifics.

a. Mathematics
   - Square Matrix Multiply
   - Least Common Multiply
   - Base to Base Integer Conversion
   - Prime Number Generator
   - Repeating Decimal Program
   - Nth Degree Polynomial Data Point Fitting Routine
   - Nth Degree Polynomial Data Point Fitting Routine with RMS Error
   - Simultaneous Equations
   - Abbreviated Simultaneous Equations
   - Curve Fittings

Root Finder Program
Determinot Program
Prime Plots
Rectangular to Polar Conversion
Polar to Rectangular Conversion

b. Plot
   One Line Routines
   X^3 and Circle
   Superposition
   Circle

c. Education
   Sine
   Factors
   Figure Eight
   Right Triangle Solution

d. Engineering
   Column Width
   Traverse
   Least Square "Linear" Fit
   Nozzle Weight Flow
   Filter Design
   OHM's Law
   Second Order Differential Equations
   — Y Transforms for Complex Numbers
   Y — Transforms for Complex Numbers
   Series Resonant
   Circuit Analysis

e. Accounting
   Payroll Calculations
   Grades: A Grade Averaging and Display Program

f. Statistics
   One-Sample Statistics: Two-Sample: Welch Procedure
   One-way Analysis of Variance
   Sheffe's Contrast Between Means
   Strip FOCAL: A Data Display and Storage Routine

g. Physics
   Monte Carlo Solution to Neutron Penetration Problem
   Multipulse
   Multipulse-2
   Least Squares Fit to an Exponential

h. Oceanography
   Seismic Refraction Sloping Layer Program

DECUS No. 8-164
Prime Number Determination

This program, types out by successive division all previously determined primes which are less than or equal to square roots of the numbers being evaluated. Short additional programs cause additional typeout of all nonprime numbers as a product of the integer.

Other Programs Needed: Floating Point Interpretive Package (DEC-8-5-S)

Storage Requirement: 4K

DECUS No. 8-165
The PDP-8 Simulator System for Philco 212

A program has been developed which provides the facility to simulate the operation of a PDP-8 computer within a Philco 212 computer. The system includes the ability to assemble PDP-8 programs in assembly language, produce paper tapes suitable for running on a PDP-8 and to simulate the execution of a PDP-8 program completely within the Philco 212. The simulation facility should be of use to people who are anticipating the delivery of a PDP-8 computer, and who wish to do program development before its arrival. The simulator portion is written in Philco 212 assembly language (it is not 211 compatible), and the remainder of the system is written in the MAD language.

Write-up only available.

DECUS No. 8-166
Interim Technical Report — PDPMAP Assembler for IBM 7040

This report describes the PDPMAP Assembly System which is used to assemble symbolic programs written for a PDP-8 or DEC-338 with up to 16K memory locations. The system is implemented at the University of Pennsylvania on an IBM 7040 and DEC PDP-8 connected by a high-speed data channel (IBM 7904 and DEC DM03). The PDPMAP System uses the powerful assembler of a large computer (IBM 7040 MAP Assembler) to quickly assemble programs for a small computer.

Report only available.

DECUS No. 8-167
CIRCUITS

CIRCUITS is a program which enables electronics circuits to be drawn using the DEC-338 Display system. The compute circuit can be stored on paper tape and read in for future modifications.

Minimum Hardware: A PDP-8 with 338 Display, Teletype, High Speed Reader and Punch, and Character Generator

DECUS No. 8-168
CalComp Plotting Package

This package is a series of subroutines designed to be used with the CalComp and PDP-8. The subroutines are:

PLOTX — a modified 8-12-U, general move routine

ALPHA — alphanumeric packed string plotting

DLTR — an 8-bit ASCII letter drawer

AXIS — an axis drawing routine

NUMBER — a signed 11-bit binary number output routine

DSYM — centered symbol drawing routine

LINE — vector plotting routine

This package is issued only on DECtape.

Minimum Hardware: A PDP-8 computer with DECtape 350B Interface and CalComp Model 565 Digital Plotter with a step size of 0.01 inches.

DECUS No. 8-169
Physical Oceanography Data Reduction Program for the PDP-8 (II)

This system gives the capability of automating and improving handling and processing of oceanographic data at sea.

It presents an oceanographic data processing system for use at sea on a small computer with a basic configuration of 4K memory, ASR33 Teletype, high-speed paper tape reader/punch and a 31 inch CalComp Plotter.

(1) to increase the speed, ease and accuracy of the data reduction at sea.
(2) to give quality control on the original data. This allows malfunctioning instruments to be quickly detected so that appropriate action may be taken.
(3) to calculate various parameters which are used by the oceanographer. The manual calculation of these parameters is not practical because of their complexity.
(4) to digitize the data suitable for transfer to a larger and more powerful short-based system.

This PDP-8 oceanographic data processing system is capable of accepting pressure, temperature, salinity, oxygen and silicate as measured parameters. It is usual to have the information necessary to calculate the pressure and temperature shortly after the completion of an oceanographic station. However, it is not possible to complete the determination of the chemistry (salinity, oxygen and silicate) until several hours later. Hence, the data input to the system is divided into two parts. The first basic input contains the information from the 'deck sheet' which is used to calculate the corrected temperatures and observed pressures. A least square polynominal is fitted to the observed pressures to give a final pressure at each sampling depth. At this point a plot of temperature vs. pressure may be obtained. To continue processing, it is necessary to have the chemistry data which constitutes the second basic input to the system. For each station the chemistry data are combined with the pressures and corrected temperatures. From these data the depth, specific volume anomaly, potential temperature and density anomalies are calculated.

Certain approximations were made in the system because of the limited core memory and to simplify the programming. However, in all cases sufficient accuracy has been retained to ensure meaningful results.

The following routines are included:

Temperature Formatting
Additions to Floating Point Package
Pack Thermometer Calibration
Pressure Curve Fit
Final Pass
DHUN
PLOPRM
Distance and Bearing
Formatting of Chemistry
Department PLOTCO

DECUS No. 8-170
FORTRAN Source Conversion Program

This program will allow the user to convert FORTRAN source programs written for DEC-08-AFC1 (FORTRAN Compiler, Old Version) to the new format, FORTRAN (DEC-08-AFC1-PB).

DECUS No. 8-171
Real-Time System for Behavioral Science Experiments

This program controls the operations of ten behavioral chambers using four classical experimental designs; Punishment Discrimination (PD), Non-discriminated Avoidance (NDA), Fixed Ratio (FR), and Differential Rate of Low Response (DRL). Besides controlling the experiments, certain statistics are accumulated during the experiments for printout at the end of each test run.

Write-up only available.

Minimum Hardware:	PDP-8 with an ASR33; requires a special interface between computer and behavioral equipment.
Storage Requirement:	4K

DECUS No. 8-172
Octal Systems Edit

Octal System Edit allows advanced users to perform direct octal editing of the information on their systems device. It makes block format compatible with system blocks. All editing is via the Teletype; commands allow reading, writing, and moving blocks; listing, changing, and punching individual words in block.

Minimum Hardware:	PDP-8, 8/I, 8/S with DF32 or TC01/TU55
Other Programs Needed:	Disk/DECtape monitor (DEC-D8-SBAC-PB)
Source Language:	PAL-D
Storage Requirement:	200_8-1177_8 (may be reassembled into any 4 pages)
Restrictions:	Requires that Monitor Head be present in 7600_8-7779_8

DECUS No. 8-173
TIC 5/8 (Scope Version)

TIC 5/8 plays a fast game of TIC-TAC-TOE on the display scope. The program can be reset to a learning configuration by hitting two keys on the Teletype. and will begin to learn winning strategies from each game it loses until it has become a master player again. The program makes use of the program interrupt facility and makes necessary changes for a PDP-5 or PDP-8.

Minimum Hardware:	PDP-5/8 family and 34D scope
Source Language:	LRL PDP Assembly Language
Storage Requirement:	1-3 and 41-3000
Restrictions:	Should not be copied after use. Execution time excludes use on PDP-8/S. All program interrupt flags must be cleared for use (room is provided)

DECUS No. 8-174
Medium

Medium is a demonstration program for use on the PDP-5 or PDP-8 family. Messages typed on the Teletype are displayed on the scope, advancing across the screen from right to left similar to the Times Square News Sign.

Minimum Hardware:	PDP-5 or PDP-8 family with 34D or VC8/I Scope.

Source Language: LRL PDP-5 Assembly Language

Storage Requirement: 41_8-1500_8

DECUS No. 8-175
Post Stimulus Interval Histogram for AX08

This program, using the Schmitt triggers, generates a post stimulus interval histogram for one channel.

Minimum Hardware: LAB-8

Other Programs Needed: LAB-8 compiler

Restrictions: Maximum count per interval is 4095_{10}
Maximum number of epochs is 4095_{10}
Maximum numberof intervals is 3456_{10}

DECUS No. 8-176
PAL CHOP

PAL CHOP produces minimum-length copies of PAL source tapes by removing all comments, tabs, multiple spaces, and multiple carriage-return line-feeds. It is especially useful in facilitating the handling and storage of sections of extremely large programs which have been debugged.

Minimum Hardware: PDP-5, PDP-8, 8/I, 8/L and ASR33. High-speed reader and punch optional.

Source Language: PAL-D

Storage Requirement: Program occupies 10_8-366_8; uses 400_8-1177_8 as buffer

Execution Time: I/O limited

DECUS No. 8-177
COPY

COPY is an extension of PIP. Its purpose is to copy disk files onto paper tape and vice versa. COPY's major advantage is that it saves time in putting files on and off the disk. This can be very useful for those with one disk and limited space.

Minimum Hardware: PDP-8, 8/I or 8/S with disk and Teletype

Other Programs Needed: Disk Monitor I/O routine in core and command decoder stored on disk starting in block 15.

Source Language: PAL-D

Storage Requirement: 0-2777_8: only 0-1474_8 for program — rest buffers

DECUS No. 8-178
Reverse Assembler

The Reverse Assembler accepts a paper tape in binary format and produces either a printed listing or a paper tape that is acceptable to the PAL Assembler as a symbolic tape. It produces the mnemonics for almost all input/output devices as well as PAL III and Floating Point instructions.

Minimum Hardware: PDP-8 with ASR33

Source Language:	PAL III
Storage Requirement:	$0\text{-}5400_8$
Execution Time:	Input/output limited two-five seconds for ASR33 per line.

DECUS No. 8-179
EAE Modifications for Binary Disassembler with Symbols

This patch permits use of the Binary Disassembler with Symbols, (DECUS No. 5/8-18C) by users without EAE. The patch shortens the space for the cross reference table by approximately one page, and changes all EAE instructions to JMSs to routine which take their place. The patch also changes the octal type routine to make space for links on page zero.

Minimum Hardware:	PDP-8 with 4K, ASR33, High-speed reader
Other Programs Needed:	Binary Disassembler with Symbols (DECUS No. 5/8-18C)

DECUS No. 8-180
Editor and Assembler for 57A Magnetic Tape System

The Symbolic Editor and MACRO-8 Assembler have been modified to replace paper tape with IBM-compatible magnetic tape for more rapid and convenient program development. The system requires an 8K PDP-8 and a 57A Tape Control with one transport.

The Editor reads and writes ASCII text in a file on magnetic tape. Text is stored in pages which may be individually accessed by Teletype commands. All the original operations are retained, including paper tape I/O.

MACRO-8 assembles the text file, completing all three passes before halting. Binary output is on high or low speed paper tape. The symbol table and Pass 3 listing are on a line printer.

A third program moves pages of text from one area of tape to another whenever rediting and reassembly are necessary.

Minimum Hardware:	PDP-8, 8K memory, ASR33, 57A Magnetic Tape Control with one transport.
Other Programs Needed:	Symbolic Editor (DEC-08-ESAB) High-speed MACRO-8 (DEC-8-8-S)
Storage Requirement:	Fields 0 and 1; locations 0-7577
Source Language:	MACRO-8
Restrictions:	The 57A needs modification for Extended Memory operations

DECUS No. 8-181
Automatic Binary Loader and Duplicator-Coder for Auto Bin

Automatic Binary Loader will automatically start tapes it has loaded into core in any memory field.

The Duplicator-Coder for Auto Bin computes checksums and notifies the operator of an error. It will select the correct input/output devices to be used. It can also be used to format the tapes for the Automatic Binary Loader.

Minimum Hardware:	Basic PDP-8
Source Language:	PAL
Storage Requirement:	Automatic Binary Loader (7600_8-7754_8) Duplicator-Coder for Auto Bin (0010_8-0431_8)
Restrictions:	These programs will not load tapes formatted for automatic memory extensions control; (i.e., channel 8 punched). Both programs will indicate a checksum error.

DECUS No. 8-182
Memory Compare

Memory Compare resides in page 36_8 of either field. It compares contents of similar addresses in pages 0-35_8 of both fields and outputs any differences detected.

Minimum Hardware:	PDP-8 with extended memory
Source Language:	PAL-D
Storage Requirement:	1 page

DECUS No. 8-183
The WANG Loader

The WANG Loader will load any program that ends at location 7777. The program consists of 8 instructions that are loaded via the toggle switches, and a tape that will boot-in the BIN and RIM loaders.

Minimum Hardware:	PDP-8 with ASR33

DECUS No. 8-184
Page Routine

This program will arrange listings in page lengths and sequentially number the pages.

Minimum Hardware:	PDP-8 with ASR33
Source Language:	PAL III
Storage Requirement:	Approximately 200_8 words
Restrictions:	Maximum of 99 pages per listing

DECUS No. 8-185
Modifications to Symbolic Editor and Symbolic Tape Format Generator

The modifications to Symbolic Editor (DEC-08-ESAB) are: 200_8 code becomes a valid character which can be stored or generated; T and F output 200_8 code; all three punching commands, T, F and P are followed by halts to enable the punch to be turned on; T also halts after punching trailer. These changes simplify editing of tapes which contains sections of text or data separated by lengths of leader/trailer.

The modified Format Generator produces a symbolic format which saves tape, editor buffer space and Teletype time.

Minimum Hardware:	PDP-8

Other Programs Needed: Symbolic Editor (DEC-08-ESAB) and Symbolic Tape Format Generator (Digital 8-21)

Source Language: PAL III

DECUS No. 8-186
EAE FORTRAN Patch for the PDP-8

This patch to the PDP-8 FORTRAN Operating System utilizes the extended arithmetic unit option (Type 182 EAE). Four arithmetic routines were rewritten: alignment, normalize, multiply and divide. The reducton in excution time is rather significant.

Another improvement besides the faster execution time was gained with EAE FORTRAN. Since the multiply routine calculates a full 48-bit product and rounds instead of truncates to 24 bits, an increase in significance of the product was noted.

These modifications work with the FORTRAN Operating System of March 2. 1967, They have not been tested with any other version, but would probably work. No changes must be made in operating procedure or any other portion of the program, as this modification loads over the regular arithmetic sub-routines.

Minimum Hardware: PDP-8 with Type 182 EAE

Other Programs Needed: FORTRAN Operating System (DEC-08-AFCO)

Source Language: FORTRAN

DECUS No. 8-187
Keyboard Controlled Binary Punch

This program makes binary tape copies of selected areas of core. It is entirely keyboard controlled, and has provisions for punching leader, data, checksums and field marks for extended memory programming.

Minimum Hardware: PDP-8, High-speed Punch and Extended Memory (optional)

Source Language: PAL-D

Storage Requirement: 1 page (versions included occupy 1, 36 and 37).

DECUS No. 8-188
Patch to Utilize Four Word Floating Point Package (DEC-08-FMHA-8B)

This patch will allow the DEC Floating Point Package to be entered from any memory bank if the arguments and operands processed by the Floating Point Routine all reside in the same memory bank from which the package is called. The patch uses only free locations within the package.

Other Programs Needed: Floating Point Package (DEC-08-FMHA-8B)

DECUS No. 8-189
LKDN: Look into the Directory Name Block

LKDN: will find the appropriate directory name entry when given the file name. It will decode and type out the contents of the entry. The output gives the disk location of the directory entry (in xxx. yyy form, see DISKLOOK

gives the disk location of the directory entry (in xxx.yyy form, see DISKLOOK DECUS No. 8-111) and also the option of the disk block locations for each core page stored.

Minimum Hardware: PDP-8 with DF32 Disk

Other Programs Needed: Disk Operating System (DEC-08-SDAA)

Source Language: PAL-D

Storage Requirement: Program — locations 12_8 and 20_8-1377_8
Buffer — locations 1400_8 - 1777_8
If stored on disk, the program requires 6 blocks; it can be saved with the command "SAVE LKDN! 0-1377; 200."

DECUS No. 8-190
PATCH Utility Program

This program, a utility routine, allows duplicating and updating of DECtape files of any PDP-8 TC01 format. It is derived from a combination of ODT (DEC-08-COBO-D) and XDUP (DECUS No. 5/8-64). The user should be familiar with the operation of both of these programs.

Minimum Hardware: PDP-8 with TC01 Control

DECUS No. 8-191
FIELDS

FIELDS, a demonstration program, calculates and displays the surface potential of a given boundary conditional plane. Each output facility is called by a 338 Display pushbutton giving a numerical and/or pictorial result.

Minimum Hardware: PDP-8 with High-Speed punch, ASR33 Teletype and 338 Display

Storage Requirement: 8K

DECUS No. 8-192
T.A.L.C. (Taylor's Algebraic Linear Calculator)

T.A.L.C. is a general-purpose calculator designed to evaluate a general algebraic equation, given all quantities involved in the equation. In effect, T.A.L.C. turns any of the family-of-eight computers into a powerful desk calculator capable of evaluating complex algebraic, trigonometric, and logarithmic functions. In addition, T.A.L.C. utilizes the concept of "idiot-proofing" to virtually eliminate the possibility of an operator error invalidating the equation. Thus, T.A.L.C. is easy to use and presents unlimited possibilities in any field where fast and accurate calculations are required.

Minimum Hardware: 4K PDP-8, High-Speed Reader, DF32 Disk File and ASR33/35

Other Programs Needed: Floating Point Package (Digital-8-5-S)

Storage Requirement: 4K

Source Language: PAL III

DECUS No. 8-193
DISP

DISP provides a simple means of using the 34D Display with FORTRAN-D.

It allows the operator to display varying numbers of points with movable X and Y axes.

Minimum Hardware:	4K PDP-8 with DF32 Disk
Other Programs Needed:	FORTRAN-D Compiler (DEC-08-AFC0)
Source Language:	PAL-D
Storage Requirement:	$600\text{-}777_8$, $7400\text{-}7577_8$
Restrictions:	Destroys FORTRAN-D disk read/write option (e.g., Read 3, 10)

DECUS No. 8-194
NMR Simulator

NMR Simulator is designed to calculate the theoretical spectrum of compounds containing hydrogen, florine, carbon-13 and other nuclei of spin 112. The calculated theoretical spectrum is displayed on an oscilloscope.

Options for punched and typewritten output, change in X-axis offset (sweep offset) and spectrum resolution are available. Chemical shifts and coupling constant parameters may be varied successively until the displayed spectrum matches that obtained experimentally. Redisplay of a "library" of theoretical spectra is possible by retaining punched output tapes.

Minimum Hardware:	8K PDP-8, Oscilloscope, and High-Speed Reader Punch
Source Language:	PAL III
Storage Requirement:	8K
Execution Time:	1 second to 15 minutes

DECUS No. 8-195
POLY BASIC

POLY BASIC is a compiler and operating stand-along system designed for the PDP-8 family. It has a total user program storage of 32K characters in which the disk is utilized. Some of the features of the compiler are:

a. It has all BASIC system comands
b. It has all BASIC operations
c. It contains all built-in functions except TAN
d. Its accuracy is 1 part in 2^{23} rather than 1 part in 2^{35}, because of word size difference
e. Maximum program size is 6144 characters as in regular (Dartmouth) BASIC
f. Maximum usable statement number is 4095 rather than 9999
g. Maximum array space is 3600 characters, and maximum number of statements is 330; however, these can be traded off against one another at the rate of 25 array elements per statement
h. There are no matrix operations
i. The argument "EDIT" renumbers the user file from line number 100 in steps of 10
j. There is a set of error messages to signal compilation errors and a set for execution errors.

Minimum Hardware: PDP-8 with ASR33 Teletype and DF32 Disk

DECUS No. 8-196
DET: Detect Key Words

DET will detect a key word or words from any sentence that is typed via the Teletype. This program contains "spell", a routine that will check the spelling of the names of all the United States.

DECUS No. 8-197
Overlay for Standard Editor and PAL III Assembler

This overlay enables the user of the Editor (DEC-08-ESAB) and the PAL III Assembler (DEC-08-ASAC) to save approximately half the time required when using the ASR33/35. This patch has proven to be a time saver when debugging is necessary.

Minimum Hardware: PDP-8 with 8K

Other Programs Needed: Editor (DEC-08-ESAB) and PAL III (DEC-08-ASAC)

DECUS No. 8-198
SYSHLP: Monitor System Utility Program

SYSHLP is a combined version of DNHELP (DECUS No. 8-135) and SYSLUK (DECUS No. 8-141). Besides more convenient alteration between the two programs, SYSHLP features improved search coding in the SYSLUK portion.

Minimum Hardware: PDP-8 with DF32 or TC01

Other Programs Needed: System Monitor Head (DEC-D8-SDAA)

Source Language: PDPMAP (DECUS No. 8-166)

Storage Requirement: 200_8-2177_8

DECUS No. 8-199
Accessing Data Arrays and Teletype Text Input/Output

Thes two subroutines provide the user with a powerful yet concise programming methodology when used with the Floating Point Package (DEC-08-FMHA).

The array accessing subroutine permits the user to access both fixed and floating point data located anywhere in the first 2K words of core storage regardless of page overlap. Both data storage and retrieval can be performed on terms analogous to single variable, subscripted FORTRAN array terms such as "ARRAY" (a*J±b).

The second subroutine, TTY Text I/O, provides a concise facility for text output (63 characters), character input, line spacing and page tabulation.

Other Programs Needed: Floating Point Package (DEC-08-FMHA)

Source Language: PAL III

Storage Requirement: Array Accessing — 119_{10} words; Teletype Text I/O — 56_{10} words

DECUS No. 8-200
BOSS

BOSS allows a series of system programs to be brought into core and executed in either one or any number of runs without keyboard input, other than the initial listing of programs and a single decision input at the end of each run.

Minimum Hardware:	PDP-8 with DF32 or DECtape
Other Programs Needed:	Disk Monitor System (DEC-08-SDAA)
Source Language:	PAL III
Storage Requirement:	2 blocks of device (3 DECtapes)
Restriction:	File name must begin with a letter

DECUS No. 8-201
DECSW

DECSW is a subroutine which accepts the contents of decimal switches at a remote location and converts the number into the following forms:

a. As an insert into a BCD string which may be typed out or displayed on CRT screen.
b. As a floating point number in the floating point accumulator.
c. As the binary equivalent in the accumulator, if the number was an integer.

Minimum Hardware: PDP-8 with Digital Switches

DECUS No. 8-205
MTSAFE

MTSAFE is a TC58 version of DECUS No. 8-120, Disk/DECtape FAILSAFE. It is self-explanatory and incorporates additional messages for the operator to service the magtape.

Minimum Hardware:	PDP-8 with TC58 Magtape
Other Programs Needed:	Disk Monitor System (DEC-D8-SDAA)
Storage Requirement:	SA-0200; occupies $100\text{-}1377_8$ and uses $1400\text{-}2177_8$ as buffer

DECUS No. 8-206
DUMP

DUMP types out the octal contents of any 128-word block on tape or disk. The link (129th) word will be printed and identified. The typeout may be halted in the middle to proceed to the next block on the same device, to switch to a different block and/or device, or to return to the monitor.

Minimum Hardware:	4K PDP-8 with DF32 Disk or TC01 DECtape
Other Programs Needed:	DECtape Operating System (DEC-08-S B) or Disk Operating System (DEC-08-SDAA)
Source Language:	PAL-D
Storage Requirement:	Location 0-1177; starting address is 1000; 2000-2200 is used as buffer.

DECUS No. 8-207
Cube Root Subroutine

The Cube Root Subroutine is called with an effective "JMS CUBE" with the argument in the accumulator. The subroutine returns to the memory location following "JMS CUBE" with the result in the accumulator and the remainder

in MAGIC. The algorithm makes use of the fact that the third order difference of any list of consecutive cubes is always equal to 6.

Source Language: PAL

Storage Requirement: 27_{10} locations

DECUS No. 8-208
Evaluating Determinant

Evaluates determinants with the order in the range from 2 through 17.

Other Programs Needed: FORTRAN Compiler and FORTRAN Operating System

Source Language: FORTRAN

Storage Requirement: 4K

Execution Time: Less than 30 seconds

DECUS No. 8-209
Editor-with-View

Editor-With-View is the same as the library distributed version of the disk editor (DEC-D8-ESAB) with the exception of the V(View) comand addition. This command is exactly like the L command for the TTY, except that results are displayed on a storage scope via the software character generator, and the VD8/I or 34D scope controller. The ALTMODE key will display the next line of the text buffer.

Minimum Hardware: 8K PDP-8 with VD8/I or 34D Scope

APPENDIX B
TABLES OF INSTRUCTIONS
PDP-8/I [PDP-8/L] MEMORY REFERENCE INSTRUCTIONS

Mnemonic Symbol	Operation Code	Direct Addr. States Entered	Direct Addr. Execution Time (μs)	Indirect Addr. States Entered	Indirect Addr. Execution Time (μs)	Operation
AND Y	0	F, E	3.0 [3.2]	F, D, E	4.5 [4.8]	Logical AND. 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. ACj $\wedge$ Yj $\rightarrow$ ACj
TAD Y	1	F, E	3.0 [3.2]	F, D, E	4.5 [4.8]	Two's complement add. 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. AC + Y $\rightarrow$ AC
ISZ Y	2	F, E	3.0 [3.2]	F, D, E	4.5 [4.8]	Increment and skip if zero. 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 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.

PDP-8/I [PDP-8/L] MEMORY REFERENCE INSTRUCTIONS (cont.)

Mnemonic Symbol	Operation Code	Direct Addr. States Entered	Direct Addr. Execution Time (μs)	Indirect Addr. States Entered	Indirect Addr. Execution Time (μs)	Operation
						If resultant Y = 0, PC + 1 → PC.
DCA Y	3	3.0 [3.2]	F, E	F, D, E	4.5 [4.8]	Deposit and clear AC. 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. AC → Y; 0 → AC
JMS Y	4	3.0 [3.2]	F, E	F, D, E	4.5 [4.8]	Jump to subroutine. The content of the PC is deposited in core memory location Y and the next instruction is taken from core memory location Y + 1. PC + 1 → Y; Y + 1 → PC
JMP Y	5	1.5 [1.6]	F	F, D	3.0 [3.2]	Jump to Y. 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. Y → PC

PDP-8/I & PDP-8/L GROUP 1 OPERATE MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Sequence	Operation
NOP	7000	—	No operation. Causes a 1.5 μs program delay.
IAC	7001	3	Increment AC. The content of the AC is incremented by one in two's complement arithmetic.
RAL	7004	4	Rotate AC and L left. The content of the AC and the L are rotated left one place.
RTL	7006	4	Rotate two places to the left. Equivalent to two successive RAL operations.
RAR	7010	4	Rotate AC and L right. The content of the AC and L are rotated right one place.
RTR	7012	4	Rotate two places to the right. Equivalent to two successive RAR operations.
CML	7020	2	Complement L.
CMA	7040	2	Complement AC. The content of the AC is set to the one's complement of its current content.
CIA	7041	2, 3	Complement and increment accumulator. Used to form two's complement.
CLL	7100	1	Clear L.
CLL RAL	7104	1, 4	Shift positive number one left.
CLL RTL	7106	1, 4	Clear link, rotate two left.
CLL RAR	7110	1, 4	Shift positive number one right.
CLL RTR	7112	1, 4	Clear link, rotate two right.
STL	7120	1, 2	Set link. The L is set to contain a binary 1.
CLA	7200	1	Clear AC. To be used alone or in OPR 1 combinations.
CLA IAC	7201	1, 3	Set AC = 1.
GLK	7204	1, 4	Get link. Transfer L into AC11.
CLA CLL	7300	1	Clear AC and L.
STA	7240	2	Set AC = −1. Each bit of the AC is set to contain a 1.

PDP-8/I & PDP-8/L GROUP 2 OPERATE MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Sequence	Operation
HLT	7402	3	Halt. Stops the program after completion of the cycle in process. If this instruction is combined with others in the OPR 2 group the other operations are completed before the end of the cycle.
OSR	7404	3	OR with switch register. The OR function is performed between the content of the SR and the content of the AC, with the result left in the AC.
SKP	7410	1	Skip, unconditional. The next instruction is skipped.
SNL	7420	1	Skip if L $\neq$ 0.
SZL	7430	1	Skip if L = 0.
SZA	7440	1	Skip if AC = 0.
SNA	7450	1	Skip if AC $\neq$ 0.
SZA SNL	7460	1	Skip if AC = 0, or L = 1, or both.
SNA SZL	7470	1	Skip if AC $\neq$ 0 and L = 0.
SMA	7500	1	Skip on minus AC. If the content of the AC is a negative number, the next instruction is skipped.
SPA	7510	1	Skip on positive AC. If the content of the AC is a positive number, the next instruction is skipped.
SMA SNL	7520	1	Skip if AC $<$ 0, or L = 1, or both.
SPA SZL	7530	1	Skip if AC $>$ 0 and if L = 0.
SMA SZA	7540	1	Skip if AC $<$ 0.
SPA SNA	7550	1	Skip if AC $>$ 0.
CLA	7600	2	Clear AC. To be used alone or in OPR 2 combinations.
LAS	7604	1, 3	Load AC with SR.
SZA CLA	7640	1, 2	Skip if AC = 0, then clear AC.
SNA CLA	7650	1, 2	Skip if AC $\neq$ 0, then clear AC.
SMA CLA	7700	1, 2	Skip if AC $<$ 0, then clear AC.
SPA CLA	7710	1, 2	Skip if AC $>$ 0, then clear AC.

PDP-8/I EXTENDED ARITHMETIC ELEMENT MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Sequence	Operation
MUY	7405	3	Multiply. The number held in the MQ is multiplied by the number held in core memory location PC + 1 (or the next successive core memory location after the MUY Command). At the conclusion of this command the most significant 12 bits of the product are contained in the AC and the least significant 12 bits of the product are contained in the MQ. Y X MQ → AC, MQ.
DVI	7407	3	Divide. The 24-bit dividend held in the AC (most significant 12 bits) and the MQ (least significant 12 bits) is divided by the number held in core memory location PC + 1 (or the next successive core memory location following the DVI command). At the conclusion of this command the quotient is held in the MQ, the remainder is in the AC, and the L contains a 0. If the L contains a 1, divide overflow occurred and the operation was concluded after the first cycle of the division. AC, MQ ÷ Y → MQ.
NMI	7411	3	Normalize. This instruction is used as part of the conversion of a binary number to a fraction and an exponent for use in floating point arithmetic. The combined content of the AC and the MQ is shifted left by this one command, until the content of AC0 is not equal to the content of AC1, to form the fraction. Zeros are shifted into vacated MQ11 positions for each shift. At the conclusion of this operation, the step counter contains a number equal to the number of shifts performed. The content of L is lost. ACj → ACj — 1 AC0 → L MQ0 → AC11 MQj → MQj — 1 0 → MQ11 until AC0 ≠ AC1
SHL	7413	3	Shift left. This instruction shifts the combined content of the AC and MQ to the left one position more than the number of positions indicated by the content of core memory at address PC + 1 (or the next successive core memory location following the SHL command). During the shifting, zeros are shifted into vacated MQ11 positions.

PDP-8/I EXTENDED ARITHMETIC ELEMENT MICROINSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Sequence	Operation
			Shift Y + 1 positions as follows: ACj → ACj — 1 AC0 → L MQ0 → AC11 MQj → MQj — 1 0 → MQ11
ASR	7415	3	Arithmetic shift right. The combined content of the AC and the MQ is shifted right one position more than the number contained in memory location PC + 1 (or the next successive core memory location following the ASR command). The sign bit, contained in AC0, enters vacated positions, the sign bit is preserved, information shifted out of MQ11 is lost, and the L is undisturbed during this operation. Shift Y + 1 positions as follows: AC0 → AC0 ACj → ACj + 1 AC11 → MQ0 MQj → MQj + 1
LSR	7417	3	Logical shift right. The combined content of the AC and MQ is shifted right one position more than the number contained in memory location PC + 1 (or the next successive core memory location following the LSR command). This command is similar to the ASR command except that zeros enter instead of the sign bit vacated positions. Information shifted out of MQ11 is lost and the L is undisturbed during this operation. Shift Y + 1 positions as follows: 0 → AC0 ACj → ACj + 1 AC11 → MQ0 MQj → MQj + 1
MQL	7421	2	Load multiplier quotient. This command clears the MQ, loads the content of the AC into the MQ, then clears the AC. 0 → MQ AC → MQ 0 → AC
SCA	7441	2	Step counter load into accumulator. The content of the step counter is transferred into the AC. The AC should be cleared prior to issuing this command or the CLA command can be

PDP-8/I EXTENDED ARITHMETIC ELEMENT MICROINSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Sequence	Operation
			combined with the SCA to clear the AC, then effect the transfer. SC V AC → AC
SCL	7403	3	Step counter load from memory. Loads complemented bits 7 through 11 of the word in memory following the instruction into the step counter. MB7-11 → SC PC + 2 → PC
MQA	7501	2	Multiplier quotient load into accumulator. The content of the MQ is transferred into the AC. This command is given to load the 12 least significant bits of the product into the AC following a multiplication or to load the quotient into the AC following a division. The AC should be cleared prior to issuing this command or the CLA command can be combined with the MQA to clear the AC then effect the transfer. MQ V AC → AC
CLA	7601	1	Clear accumulator. The AC is cleared during sequence 1, allowing this command to be combined with the outer EAE commands that load the AC during sequence 2 (such as SCA and MQA). 0 → AC
CAM	7621	1, 2	Clear accumulator and multiplier quotient. CAM = CLA LMQ.

BASIC IOT MICROINSTRUCTIONS

Mnemonic	Octal	Operation
Program Interrupt		
ION	6001	Turn interrupt on and enable the computer to respond to an interrupt request. 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.
IOF	6002	Turn interrupt off; i.e., disable the interrupt.
High Speed Perforated Tape Reader and Control		
RSF	6011	Skip if reader flag is a 1.
RRB	6012	Read the content of the reader buffer and clear the reader flag. (This instruction does not clear the AC.) RB V AC4-11 → AC4-11
RFC	6014	Clear reader flag and reader buffer, fetch one character from tape and load it into the reader buffer, and set the reader flag when done.
High Speed Perforated Tape Punch and Control		
PSF	6021	Skip if punch flag is a 1.
PCF	6022	Clear punch flag and punch buffer.
PPC	6024	Load the punch buffer from bits 4 through 11 of the AC and punch the character. (This instruction does not clear the punch flag or punch buffer.) AC4-11 V PB → PB
PLS	6026	Clear the punch flag, clear the punch buffer, load the punch buffer from the content of bits 4 through 11 of the accumulator, punch the character, and set the punch flag to 1 when done.
Teletype Keyboard/Reader		
KSF	6031	Skip if keyboard flag is a 1.
KCC	6032	Clear AC and clear keyboard flag.
KRS	6034	Read keyboard buffer static. (This is a static command in that neither the AC nor the keyboard flag is cleared.) TTI V AC4-11 → AC4-11
KRB	6036	Clear AC, clear keyboard flag, and read the content of the keyboard buffer into the content of AC4-11.
Teletype Teleprinter/Punch		
TSF	6041	Skip if teleprinter flag is a 1.
TCF	6042	Clear teleprinter flag.
TPC	6044	Load the TTO from the content of AC4-11 and print and/or punch the character.
TLS	6046	Load the TTO from the content of AC4-11, clear the teleprinter flag, and print and/or punch the character.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
Oscilloscope Display Type VC8/I [VC8/L]		
DCX	6051	Clear X coordinate buffer.
DXL	6053	Clear and load X coordinate buffer. AC2-11 → YB
DIX	6054	Intensify the point defined by the content of the X and Y coordinate buffers.
DXS	6057	Executes the combined functions of DXL followed by. DIX.
DCY	6061	Clear Y coordinate buffer.
DYL	6063	Clear and load Y coordinate buffer. AC2-11 → YB
DIY	6064	Intensify the point defined by the content of the X and Y coordinate buffers.
DYS	6067	Executes the combined functions of DYL followed by DIY.
DSB	6075	Set minimum brightness.
DSB	6076	Set medium brightness.
DSB	6077	Set maximum brightness.
DSB	6074	Zero brightness.
Light Pen Type 370		
DSF	6071	Skip if display flag is a 1.
DCF	6072	Clear the display flag.
Memory Parity Type MP8/I [MP8/L]		
SMP	6101	Skip if memory parity error flag = 0.
CMP	6104	Clear memory parity error flag.
Automatic Restart Type KP8/I [KP8/L]		
SPL	6102	Skip if power is low.
Memory Extension Control Type MC8/I		
CDF	62N1	Change to data field N. The data field register is loaded with the selected field number (0 to 7). All subsequent memory requests for operands are automatically switched to that data field until the data field number is changed by a new CDF command.
CIF	62N2	Prepare to change to instruction field N. The instruction buffer register is loaded with the selected field number (0 to 7). The next JMP or JMS instruction causes the new field to be entered.
RDF	6214	Read data field into AC6-8. Bits 0-5 and 9-11 of the AC are not affected.
RIF	6224	Same as RDF except reads the instruction field.
RIB	6234	Read interrupt buffer. The instruction field and data field stored during an interrupt are read into AC6-8 and AC9-11, respectively.
RMF	6244	Restore memory field. Used to exit from a program interrupt.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
Memory Extension Control Type MC8/L		
CDF	62N1	Change to data field N. The data field register is loaded with the selected field number (0 or 1). All subsequent memory requests for operands are automatically switched to that data field until the data field number is changed by a new CDF command.
CIF	62N2	Prepare to change to instruction field N. The instruction buffer register is loaded with the selected field number (0 or 1). The next JMP or JMS instruction causes the new field to be entered.
RDF	6214	Read data field into AC 8. Bits 0-5 and 9-11 of the AC are not affected. Bits 6 & 7 are cleared.
RIF	6224	Same as RDF except reads the instruction field
RIB	6234	Read interrupt buffer. The instruction field and data field stored during an interrupt are read into AC 8 and 11 respectively.
RMF	6244	Restore memory field. Used to exit from a program interrupt.
Incremental Plotter and Control Type VP8/I		
PLSF	6501	Skip if plotter flag is a 1.
PLCF	6502	Clear plotter flag.
PLPU	6504	Plotter pen up. Raise pen off of paper.
PLPR	6511	Plotter pen right.
PLDU	6512	Plotter drum (paper) upward.
PLDD	6514	Plotter drum (paper) downward.
PLPL	6521	Plotter pen left.
PLUD	6522	Plotter drum (paper) upward. (Same as 6512.)
PLPD	6524	Plotter pen down. Lower pen on to paper.
Random Access Disc File (Type DF32)		
DCMA	6601	Clears memory address register, parity error and completion flags. This instruction clears the disk memory request flag and interrupt flags.
DMAR	6603	The contents of the AC are loaded into the disk memory address register and the AC is cleared. Begin to read information from the disk into the specified core location. Clears parity error and completion flags. Clears interrupt flags.
DMAW	6605	The contents of the AC are loaded into the disk memory address register and the AC is cleared. Begin to write information into the disk from the specified core location. Clears parity error and completion flags.
DCEA	6611	Clears the disk extended address and memory address extension register.
DSAC	6612	Skips next instruction if address confirmed flag is a 1. (AC is cleared.)
DEAL	6615	The disk extended-address extension registers are cleared and loaded with the track data held in the AC.

Mnemonic	Octal	Operation
Random Access Disc File (Type DF32) (continued)		
DEAC	6616	Clear the AC then loads the contents of the disk extended-address register into the AC to allow program evaluation. Skip next instruction if address confirmed flag is a 1.
DFSE	6621	Skips next instruction if parity error, data request late, or write lock switch flag is a zero. Indicates no errors.
DFSC	6622	Skip next instruction if the completion flag is a 1. Indicates data transfer is complete.
DMAC	6626	Clear the AC then loads contents of disk memory address register into the AC to allow program evaluation.
Automatic Line Printer and Control Type 645		
LSE	6651	Skip if line printer error flag is a 1.
LCB	6652	Clear both sections of the printing buffer.
LLB	6654	Load printing buffer from the content of AC6-11 and clear the AC.
LSD	6661	Skip if the printer done flag is a 1.
LCF	6662	Clear line printer done and error flags.
LPR	6664	Clear the format register, load the format register from the content of AC9-11, print the line contained in the section of the printer buffer loaded last, clear the AC, and advance the paper in accordance with the selected channel of the format tape if the content of AC8 = 1. If the content of AC8 = 0, the line is printed and paper advance is inhibited.
DECtape Transport Type TU55 and DECtape Control Type TC01		
DTRA	6761	The content of status register A is read into AC0-9 by an OR transfer. The bit assignments are: AC0-2 = Transport unit select number AC3-4 = Motion AC5 = Mode AC6-8 = Function AC9 = Enable/disable DECtape control flag
DCTA	6762	Clear status register A. All flags undisturbed.
DTXA	6764	Status register A is loaded by an exclusive OR transfer from the content of the AC, and AC10 and AC11 are sampled. If AC10 = 0, the error flags are cleared. If AC11 = 0, the DECtape control flag is cleared.
DTSF	6771	Skip if error flag is a 1 or if DECtape control flag is a 1.

Mnemonic	Octal	Operation
DECtape Transport Type TU55 and DECtape Control Type TC01 (continued)		
DTRB	6772	The content of status register B is read into the AC by an OR transfer. The bit assignments are: AC0 = Error flag AC1 = Mark track error AC2 = End of tape AC3 = Select error AC4 = Parity error AC5 = Timing error AC6-8 = Memory field AC9-10 = Unused AC11 = DECtape flag
DTLB	6774	The memory field portion of status register B is loaded from the content of AC6-8.
Card Reader and Control Type CR8/I [CR8/L]		
RCSF	6631	Generates an IOP pulse (IOP 1) to test the data-ready flag output. If the data ready flag is 1, the next sequential program instruction is skipped.
RCRA	6632	Generates an IOP pulse (IOP 2) to read the alpha-numeric data at the control-logic buffer register and clear the data ready flag.
RCRB	6634	Generates an IOP pulse (IOP 4) to read the BCD data at the control logic buffer register and clear the data ready flag.
RCSD	6671	Generates an IOP pulse (IOP 1) to test the card-done flag output. If the card done flag is 1, the next sequential program instruction is skipped.
RCSE	6672	Generates an IOP pulse (IOP 2) to advance the card, clear the card done flag, and produce a skip flag if reader is ready. If skip flag is generated, the next sequential program instruction is skipped.
RCRD	6674	Generates an IOP pulse (IOP 4) to clear the card done flag.
Automatic Magnetic Tape Control Type TC58		
MTSF	6701	Skip on error flag or magnetic tape flag. The status of the error flag (EF) and the magnetic tape flag (MTF) are sampled. If either or both are set to 1, the content of the PC is incremented by one to skip the next sequential instruction.
MTCR	6711	Skip on tape control ready (TCR). If the tape control is ready to receive a command, the PC is incremented by one to skip the next sequential instruction.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
Automatic Magnetic Tape Control Type TC58 (continued)		
MTTR	6721	Skip on tape transport ready (TTR). The next sequential instruction is skipped if the tape transport is ready.
MTAF	6712	Clear the status and command registers, and the EF and MTF if tape control ready. If tape control not ready, clears MTF and EF flags only.
MTRC	6724	Inclusively OR the contents of the command register into AC0-11.
MTCM	6714	Inclusively OR the contents of AC0-5, AC9-11 into the command register; JAM transfer bits 6, 7, 8 (command function).
MTLC	6716	Load the contents of AC0-11 into the command register.
— —	6704	Inclusively OR the contents of the status register into AC0-11.
MTRS	6706	Read the contents of the status register into AC0-11.
MTGO	6722	Set "go" bit to execute command in the command register if command is legal.
— —	6702	Clear the accumulator.
General Purpose Converter and Multiplexer Control Type AF01A		
ADSE	6531	Skip if A/D converter flag is a 1.
ADCV	6532	Clear A/D converter flag and convert input voltage to a digital number, flag will set to 1 at end of conversion. Number of bits in converted number determined by switch setting, 11 bits maximum. clear flag.
ADRB	6534	Read A/D converter buffer into AC, left justified, and
ADCC	6541	Clear multiplexer channel address register.
ADSC	6542	Set up multiplexer channel as per AC6-11. Maximum of 64 single ended or 32 differential input channels.
ADIC	6544	Index multiplexer channel address (present address + 1). Upon reaching address limit, increment will cause channel 00 to be selected.
Guarded Scanning Digital Voltmeter Type AF04A		
VSEL	6542	The contents of the accumulator are transferred to the AF04A control register.
VCNV	6541	The contents of the accumulator are transferred to the AF04A channel address register. Analog signal on selected channel is automatically digitized.
VINX	6544	The last channel address is incremented by one and the analog signal on the selected channel is automatically digitized.
VSDR	6531	Skip if data ready flag is a 1.
VRD	6532	Selected byte of voltmeter is transferred to the accumulator and the data ready flag is cleared.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
VBA	6534	BYTE ADVANCE command requests next twelve bits, data ready flag is set.
VSCC	6571	SAMPLE CURRENT CHANNEL when required to digitize analog signal on current channel repeatedly.
Real Time Clock, Type KW8/I [KW8/L]		
CCFF	6132	The flag, flag buffer, clock enable, and interrupt enable flip-flops are cleared. This disables the real-time clock.
CCEC	6136	All clock control flip-flops are first cleared, then the clock enable flip-flop is set. For the variable frequency clock, the frequency source is enabled synchronously with program operation. With all clocks, the data input to the flag is enabled after IOP2 time. This represents an 800-ns mask, after the clock is enabled.
CECI	6137	All clock control flip-flops are cleared, then the clock enable, and interrupt enable flip-flops are set. The clock enable flip-flop is described with the CCEC instruction. The interrupt enable flip-flop allows an IO BUS IN INT signal when the flag is set.
CSCF	6133	When the flag flip-flop has been set by a clock pulse, the flag buffer flip-flop is set to a 1. Upon execution of this instruction, an IO BUS IN SKIP is generated if the flag is set. The content of the PC is incremented by 1, so that the next sequential instruction is skipped. The flag flip-flop is then cleared. If the flag flip-flop has not been set, no skip is generated nor is the flag flip-flop cleared.
CCFF	6132	The flag, flag buffer, clock enable, and interrupt enable flip-flops are cleared. This disables the real-time clock. In addition, the OVERFLOW gating is disabled.
CECL	6136	The operations are the same as that of the CCEC instruction for the basic clock, except that the data input to the flag is not enabled until **both** CLOCK ENABLE and OVERFLOW are set. All counter bits are set at IOP2, and then cleared according to the accumulator prior to IOP4. At IOP4 the contents of the counter (the one's complement of the accumulator) are transferred to the output buffer. At the end of IOP4, the counter is incremented by one to provide the two's complement of the accumulator.
CEIL	6137	Operations are the same as those described in the CECI instruction for the basic clock, except that the counter is loaded according to the CECL instruction.

Mnemonic	Octal	Operation
CRCA	6134	The output buffer is gated to the I/O BUS during IOP4, and a CLK AC CLR signal generated. This register contains the last count in the count register. The transfer from the count register is synchronized with this instruction so that a transfer that would occur during this instruction is not made.
CSCF	6133	When the flag flip-flop has been set by a clock pulse, the flag buffer flip-flop is set to a one. Upon execution of this instruction an IO BUS IN SKIP is generated if the flag is set. The content of the PC is incremented by one so the next sequential instruction is skipped. The flag flip-flop is then cleared. If the flag flip-flop has not been set, no skip is generated nor is the flag flip-flop cleared.

Asynchronous Serial Line Interface, Type PT08

TSFXXX	6441	Skip if teleprinter/punch 3 flag is a 1.
TCFXXX	6442	Clear teleprinter/punch 3 flag.
TPCXXX	6444	Load teleprinter 3 buffer (TTOX) with AC4-11 and print/punch the character.
TLSXXX	6446	Load TTOX with AC4-11, 3 flag, print/punch the character and clear teleprinter/punch.
KSFXXX	6451	Skip if keyboard/reader 3 flag is a 1.
KCCXXX	6452	Clear AC and keyboard/reader 3 flag.
KRSXXX	6454	Read keyboard/reader 3 buffer (TTI3) static. (TTI3 is loaded into AC4-11 by an OR transfer.)
KRBXXX	6456	Clear the AC, read TTI3 into AC4-11, and clear keyboard 3 flag.

Time Sharing Hardware Modification, Type KT8/I

CINT	6204	Clear user interrupt. Resets the user interrupt (UINT) flip-flop to the 0 state.
SINT	6254	Skip on user interrupt. When the user interrupt (UINT) flip-flop is in the 1 state, sets the user skip flip-flop (USF) to the 1 state and causes the program to skip the next instruction.
CUF	6264	Clears the user flag. Clears the user buffer (UB) flip-flop.
SUF	6274	Sets the user flag. Sets user buffer (UB) and inhibits processor interrupts until the next JMP or JMS instruction. Generation of IB → IF during the next JMP or JMS instruction transfers the state of UB to the user field (UF) flip-flop.

Mnemonic	Octal	Operation
Storage Tube Display Control, Type KV8/I [KV8/L]		
SNC	6051	Senses the condition of the cursor interrupt flag. The flag produces an interrupt request when set by operation of the interrupt pushbutton on the joystick. The flag is initially cleared when the computer is started. As with all flag-sent instructions, SNC can be used under interrupt conditions to detect the source of the interrupt, or it can be used under interrupt on (ION) when the interrupt request has been caused by the operation of the cursor interrupt button. In a program running with the interrupt off, SNC can be used to ignore the cursor successive approximation subroutine in the program if a request for service has not been made from the joystick controller.
CCF	6052	This instruction is used to clear the cursor flag after a request for service has been acknowledged by the program.
SAC	6062	The analog comparator is set to compare the analog content of any one of six analog sources with the content of the digital-to-analog converter. The analog sources are chosen according to a 3-bit binary code. This code establishes the parameter for choosing the wanted register according to the content of AC2, AC3, and AC6.
LDF	6063	This instruction is used to establish the mode in which a wanted graphic is to be produced according to a 2-bit binary code. This code determines whether the wanted vector will be linear absolute relative, whether the point plot mode will be used, or whether the cursor will be displayed. This code establishes the parameters for these formats according to the content of AC2 and AC3. The LDF instruction must precede the LDX and LDY instructions.
LDX	6064	The X-axis sample and hold register is loaded with the binary equivalent of the X-axis coordinate according to the contents of AC2-11. This data appears at the output of the digital-to-analog converter as the analog equivalent of the X-axis value of the binary word stored in the AC. The LDX instruction clears an existing ready flag and sets the ready flag after 100 ± 20 μs.
LDY	6065	The Y-axis sample and hold register is loaded with the binary equivalent of the Y-axis coordinates according to the contents of AC2-11. This data appears at the output of the digital-to-analog converter as the analog equivalent of the binary word in the AC. The LDY instruction clears an existing ready flag and sets the ready flag after 100 ± 20 μs.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
Storage Tube Display Control, Type KV8/I [KV8/L] (continued)		
EXC	6066	Used to execute the wanted vector according to the contents of AC2-4 and AC6-11. The parameter word establishes long or short formats, circular vectors, display erasure, reset of the integrators, and intensification of the vector. The EXC instruction clears an existing ready flag and sets the ready flag as follows: a) after 20 ± 5 μs for a point or vector continue b) after 250 μs for short vectors c) after 4.05 ms for long vectors d) after 500 ms for an erase.
SRF	6071	Used to determine when the controller is ready to perform the next execute instruction. The ready flag produces an interrupt condition when set. The flag can be set by pressing the erase pushbutton on the VT01 unit. Normally, however, the state of this flag is determined by the controller. This flag is initially cleared when the computer is started and prior to an LDX, LDY, or EXC instruction.
CRF	6072	This instruction clears the ready flag after a skip instruction has been acknowledged.
SDA	6073	Used in the successive approximation subroutine to determine the digital equivalent of the selected analog holding register. This instruction is used with the SAC (6062_8) instruction.
LDA	6074	This instruction is used to load the content of AC2-11. This instruction is used with SDA (6073_8) in the successive approximation subroutine to determine the digital value of the content of the selected analog holding register. Does not change flag states.
Incremental Magnetic Tape Controller, Type TR02		
IRS/IRSA	6701/6711	When data is ready to be strobed into the AC from the read buffer (RB), the PC is incremented by one to skip the next sequential instruction. The read done flag is cleared only if the skip occurs.
ISR/ISRA	6702/6712	The content of the status register (STR) is read into AC0-8. The AC should be cleared before it is read by this instruction.
IWS/IWSA	6703/6713	If the write done flag is set, the next instruction is skipped and the write done flag is cleared.
IMC/IMCA	6704/6714	The move command decoded from AC0-2 is generated. This instruction also clears the read done, write done, and gap detect flags. The indicated flag is set when the command has been executed.
IGS/IGSA	6705/6715	If the gap detect flag is set, the next instruction is skipped and the gap detect flag is cleared.
IWR/IWRA	6706/6716	The contents of the AC are loaded into the tape input data buffer (WB) and a write step command is generated. The write done flag is set when writing is completed.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
Incremental Magnetic Tape Controller, Type TR02 (continued)		
IRD/IRDA	6707/6717	The AC is cleared and the content of the read buffer (RB) is loaded into the AC. Data bits are transferred into AC6-11 (7-track) or AC4-11 (9-track). Parity error is transferred into AC0 which is 0 if there is no parity error.
Disk File and Control, Type RF08/Expander Disk File, Type RS08		
DCIM	6611	Clear the disk interrupt enable and core memory address extension registers.
DIML	6615	Clear the interrupt enable and memory address extension registers, then load the interrupt enable and memory address extension registers with data held in the AC. Then clear the AC. NOTE: Transfers cannot occur across memory fields. Attempts to do so will cause the transfer to "wrap around" within the specified memory field.
DIMA	6616	Clear the AC. Then load the contents of the status register (STR) into the AC to allow program evaluation.
DFSE	6621	Skip next instruction if there is parity error, data request late, write lock status, or nonexistent disk flag set.
DISK	6623	If either the error or data completion flag (or both) is set, the next instruction is skipped.
DCXA	6641	Clear the high order 8-bit disk address register (DAR).
DXAL	6643	Clear the high order 8 bits of the DAR. Then load the DAR from data stored in the AC. Then clear the AC.
DXAC	6645	Clear the AC; then load the contents of the high order 8-bit DAR into the AC.
DMMT	6646	For maintenance purposes only with the appropriate maintenance cable connections and the disk disconnected from the RS08 logic, the (given) standard signals may be generated by IOT 6646 and associated AC bits. The AC is cleared and the maintenance register is initiated by issuing an IOT 6601 command. (For standard signals, see Chapter 7, Section 7-7.)
Sample and Hold Control Option, Type AC01A		
HRAN	6571	The contents of AC3-5 are transferred to the channel address register (CHAR). The 3-bit code is decoded to address any of the 8 channels.
HSIM	6572	Simultaneously places all 8 channels into the hold mode.
Digital-to-Analog Converter, Type AA01A		
SAMP	6574	Places all 8 channels into the sample (or track) mode.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
DAL1	6551	The character in the accumulator is loaded into the channel 1 buffer. The DAC then converts the buffered value to the analog equivalent. (NOTE: Similar instructions for DAL2 and DAL3 load respective DACs.)
Digital-to-Analog Converter, Type AA05/AA07		
CLDA	6551	The address register in the AA05/AA07 is cleared.
LDAD	6552	The address register in the AA05/AA07 is loaded with the contents of AC0-5.
LDAR	6562	The buffer (input buffer, if the channel is double-buffered) of the DAC is loaded from AC0-9.
UPDT	6564	The contents of the input buffers of all double-buffered channels are transferred to their respective output buffers. The input buffer is not affected by this instruction.
Synchronous Modem Interface, Type DP01AA		
STF	6611	Causes the program to skip the next instruction if the transmit flag is in the 0 state. When the transmit flag is in the 1 state, the transmit buffer register (TB) is ready to accept another character.
CTF	6602	Resets the transmit flag. If transmit active flag is set, CTF also causes the program to skip the next instruction.
TAC	6601	Causes the contents of the AC (6, 7, 8, or 9 bits right-justified) to be transferred into the TB.
CIM	6604	Resets the transmit logic idle mode (IM) flip-flop.
SIM	6614	Sets the transmit idle mode (IM) flip-flop.
SRF	6651	Causes the program to skip the next instruction if the receive flag is 0. The flag is set when a received character is ready for transfer to the AC and the flag is cleared when an RRB instruction is issued.
RRB	6612	Transfers the contents of the receiver buffer (RB) (6, 7, 8, or 9 bits, right-justified) to the computer AC. RRB also resets the receive flag.
SEF	6621	Causes the program to skip the next instruction if the receive end flag is 0. (The receive end flag flip-flop is set when the receive logic has stopped receiving serial data from the communications equipment due to termination of the SERIAL CLOCK RECEIVE pulse train.)
CEF	6622	Clears the receive end flag.
SRE	6624	Sets the ring enable (RE) flip-flop to a 1, which permits the ring flag to request a program interrupt.
CRE	6644	Clears the ring enable (RE) flip-flop.
SRI	6631	Causes the program to skip the next instruction if the ring flag is 0. The ring flag is set when a ring input is received.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
Synchronous Modem Interface, Type DP01AA (continued)		
CRF	6632	Clears the ring flag.
STR	6634	Sets the terminal ready (TR) flip-flop to the 1 state. This causes the terminal ready lead to the modem to be set on the ON state. The state changes to OFF for CTR.
CTR	6642	Clears the terminal ready (TR) flip-flop (see STR).
SSR	6641	Causes the program to skip the next instruction if the data-set-ready lead from the modem is in the ON state.
CRA	6652	Clears the receive active (RA) flip-flop, taking the receive logic out of the active state. This inhibits any more receive flags until a new sync character is received.
COB	6661	Clears the XOR buffer.
IOB	6664	Transfers 1s from the AC to the buffer register (BR).
ROB	6662	Transfers the buffer register (BR) content to the AC.
XOB	6654	Causes an exclusive OR of the AC with the buffer register (BR).
Data Communications System 680/I		
TTI	6402	Causes a JMS to be executed (N+3) if the R register does not equal 0 and either the line hold bit of the selected line (specified by bits 2-8 of the LSW) is in the 1 state, or as a result of jamming the line state into and shifting the CAW; bit 11 of the CAW is a 1.
TTO	6404	Clears the link and shifts the link and accumulator one bit position to the right. Bit 11 of the accumulator is shifted into the line unit specified by the line register. The previous contents (1 bit) of the selected line unit is lost.
TTCL	6411	This command sets the contents of the line register to 0.
TTSL	6412	The contents of AC5-11 are ORed into the line register.
TTLL	6413	The contents of AC5-11 are transferred into the line register. This is a microprogram of TTCL and TTSL.
TTRL	6414	The contents of the line register are ORed into AC5-11. The AC must be 0 for a true transfer.
TTINCR	6401	This instruction causes the contents of the line register to be incremented by 1. This command, when microprogrammed with a TTO command is executed.
TTRINC	6461	This command causes the contents of the R register to be incremented by 1. Because it is loaded with a 2's complement number, the result is a subtract. This instruction can be microprogrammed with TTRR.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
Data Communications System 680/I (continued)		
TTRR	6464	This command reads the contents of the R register into AC7-11. The contents of the AC must be 0s before issuing this instruction. This instruction, when microprogrammed with TTINCR, causes the incremented results to be read into the AC.
TTCR	6471	This command causes the R register to be set to 0.
TTLR	6472	This command causes the contents of AC7-11 to be ORed into the R register.
TTLDR	6473	This is a microprogram of TTCR and TTLR and causes the contents of the AC7-11 to be transferred into the R register.
T1on	6424	Clock Control Instruction: Enables clock 1 to set its flag at the predetermined clock rate. The flag in the 1 state causes a program interrupt when the interrupt is enabled. This instruction also sets the flag to the 0 state.
T1off	6422	Clock Control Instruction: Inhibits clock 1 from setting its flag. This instruction also sets the flag to the 0 state.
T1skip	6421	Clock Control Instruction: Causes the program to skip the next instruction if clock flag 1 is in the 1 state. To clear the flag, either T1on or T1off can be used.
T2on	6434	Clock Control Instruction: Enables clock 2 to set its flag at the predetermined clock rate. The flag in the 1 state causes a program interrupt when the interrupt is enabled. This instruction also sets the flag to the 0 state.
T2off	6432	Clock Control Instruction: Inhibits clock 2 from setting its flag. This instruction also sets the flag to the 0 state.
T2skip	6431	Clock Control Instruction: Causes the program to skip the next instruction if clock flag 2 is in the 1 state. To clear the flag, either a T2on or T2off can be used.
T3on	6444	Same as T2on, except enables clock 3.
T3off	6442	Same as T2off, except inhibits clock 3.
T3skip	6441	Same as T2skip, except flag 3 is in the 1 state.
T4on	6454	Same as T2on, except enables clock 4.
T4off	6452	Same as T2off, except inhibits clock 4.
T4skip	6421	Same as T2skip, except flag 4 is in the 1 state.
Multiple Asynchronous Line Unit, Type DC02A		
MTSF	6121	Skip the next instruction if the teleprinter flag is set.
MTCF	6122	Clear the teleprinter flag.
MTPC	6124	Load AC4-11 into the shift register (begin print/punch).
MTLS	6126	Clear the teleprinter flag and load AC4-11 into the shift register (MTCF and MTPC combined).

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic	Octal	Operation
Multiple Asynchronous Serial Line Interface Unit, Type DC02D		
MKSF	6111	Skip the next instruction if the keyboard flag is set.
MKCC	6112	Clear the keyboard and reader flags; clear the AC.
MKRS	6114	Transfer the shift register contents to AC 4-11.
MKRB	6116	Clear the keyboard and reader flags, clear the AC; transfer the shift register contents to AC 4-11 (MKCC and MKRS combined).
MTKF	6123	Transfer status of keyboard flags to AC 0-3.
MINS	6125	Skip if the interrupt request is active (if interrupt is on and any flag is raised).
MTRS	6127	Transfer the status of the selection register to AC 0-3.
Multiple Asynchronous Serial Line Interface Unit, Type DC02A		
MTPF	6113	Transfer status of teleprinter flags to AC 0-3.
MINT	6115	Interrupt on if AC 11 is set (interrupt request, if any flags).
MTON	6117	Transfer AC0-3 to selection register (SELR) (select stations when bit is set).

APPENDIX C
TABLES OF CODES

MODEL 33 ASR/KSR TELETYPE CODE (ASCII) IN OCTAL FORM

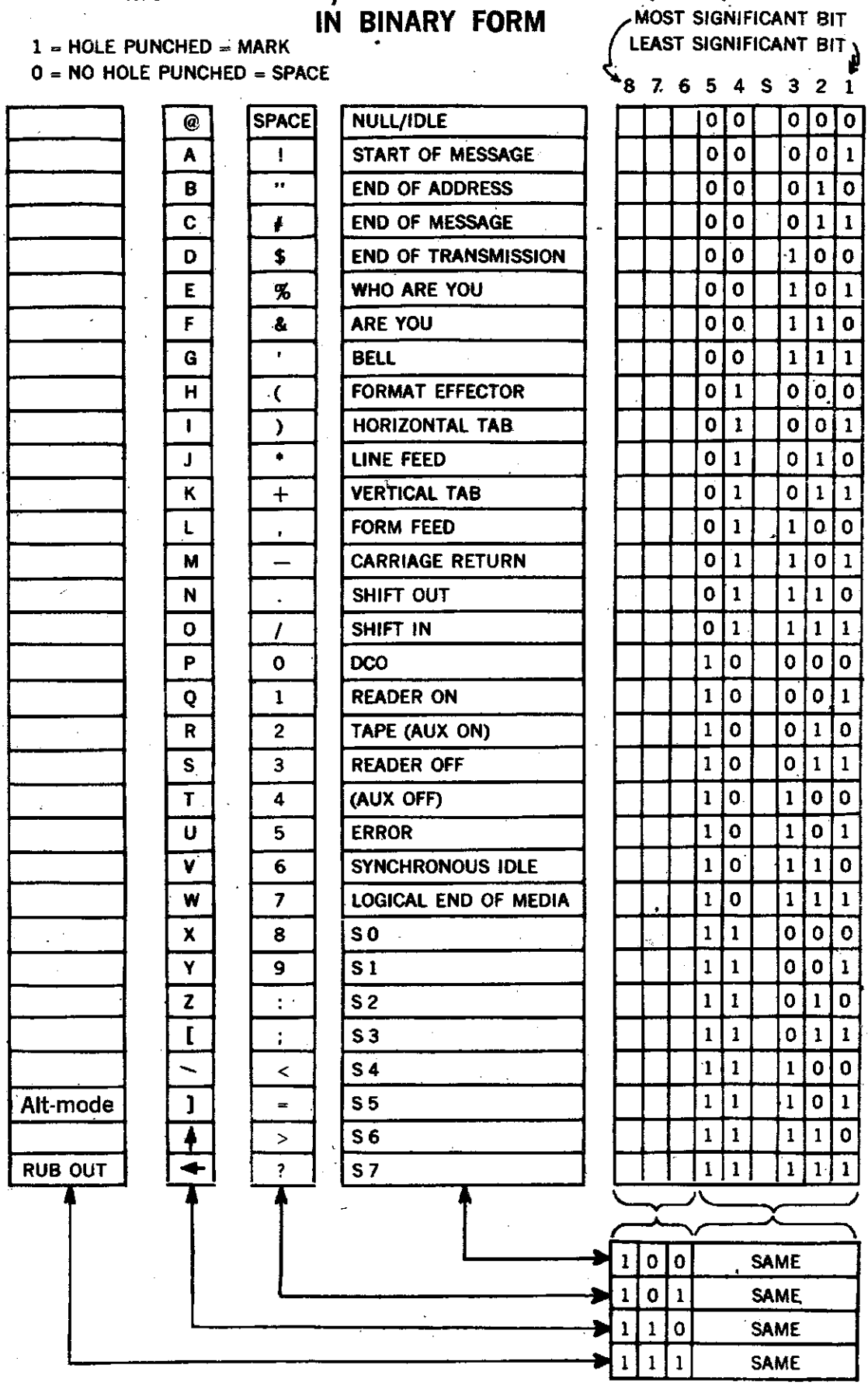
Character	8-Bit Code (in octal)	Character	8-Bit Code (in octal)	Character	8-Bit Code (in octal)
A	301	1	261	=	275
B	302	2	262	>	276
C	303	3	263	?	277
D	304	4	264	@	300
E	305	5	265	[	333
F	306	6	266	\	334
G	307	7	267	]	335
H	310	8	270	↑	336
I	311	9	271	←	337
J	312				
K	313	!	241	Leader/Trailer	200
L	314	"	242	Line-Feed	212
M	315	#	243	Carriage-Return	215
N	316	$	244	Space	240
O	317	%	245	Rub-out	377
P	320	&	246	Blank	000
Q	321	'	247	Alt-mode	375
R	322	(	250		
S	323	)	251	WRU	205
T	324	*	252	TAPE	222
U	325	+	253		224
V	326	,	254	TAB	211
W	327	—	255	X OFF	223
X	330	.	256	EOT	204
Y	331	/	257	RU	206
Z	332	:	272	BELL	207
		;	273	VT	213
0	260	<	274	FORM	214

Note 1: Shift key adds hole 5.

Note 2: Control key deletes hole 2.

MODEL 33 ASR/KSR TELETYPE CODE (ASCII) IN BINARY FORM

1 = HOLE PUNCHED = MARK
0 = NO HOLE PUNCHED = SPACE



CARD READER CODE

INTERNAL CODE	CARD ZONE	CODE NUM.	IBM 26 CHARACTER	IBM 29 CHARACTER
00	NONE		SPACE	SPACE
01	—	1	1	1
02	—	2	2	2
03	—	3	3	3
04	—	4	4	4
05	—	5	5	5
06	—	6	6	6
07	—	7	7	7
10	—	8	8	8
11	—	9	9	9
12	—	8-2	ASSIGNABLE	
13	—	8-3	#	=
14	—	8-4	@	'
15	—	8-5	ASSIGNABLE	
16	—	8-6	ASSIGNABLE	
17	—	8-7	ASSIGNABLE	
20	0	—	0	0
21	0	1	/	/
22	0	2	S	S
23	0	3	T	T
24	0	4	U	U
25	0	5	V	V
26	0	6	W	W
27	0	7	X	X
30	0	8	Y	Y
31	0	9	Z	Z
32	0	8-2	ASSIGNABLE	
33	0	8-3	,	,
34	0	8-4	%	(
35	0	8-5	ASSIGNABLE	
36	0	8-6	ASSIGNABLE	
37	0	8-7	ASSIGNABLE	
40	11	—	—	—
41	11	1	J	J
42	11	2	K	K
43	11	3	L	L
44	11	4	M	M

INTERNAL CODE	CARD ZONE	CODE NUM.	IBM 26 CHARACTER	IBM 29 CHARACTER
45	11	5	N	N
46	11	6	O	O
47	11	7	P	P
50	11	8	Q	Q
51	11	9	R	R
52	11	8-2	ASSIGNABLE	
53	11	8-3	$	$
54	11	8-4	*	*
55	11	8-5	ASSIGNABLE	
56	11	8-6	ASSIGNABLE	
57	11	8-7	ASSIGNABLE	
60	12	—	&	+
61	12	1	A	A
62	12	2	B	B
63	12	3	C	C
64	12	4	D	D
65	12	5	E	E
66	12	6	F	F
67	12	7	G	G
70	12	8	H	H
71	12	9	I	I
72	12	8-2	ASSIGNABLE	
73	12	8-3	.	.
74	12	8-4	<	)
75	12	8-5	ASSIGNABLE	
76	12	8-6	ASSIGNABLE	
77	12	8-7	ASSIGNABLE	

WILL NOT DETECT INVALID PUNCH COMBINATIONS

APPENDIX D
PERFORATED-TAPE LOADER SEQUENCES

READIN MODE LOADER

The readin mode (RIM) loader is a minimum length, basic perforated-tape reader program for the ASR33, it is initially stored in memory by manual use of the operator console keys and switches. The loader is permanently stored in 18 locations of page 37.

A perforated tape to be read by the RIM loader must be in RIM format:

Tape Channel 8 7 6 5 4 S 3 2 1	Format
1 0 0 0 0 . 0 0 0	Leader-trailer code
0 1 A1 . A2 0 0 A3 . A4	Absolute address to contain next 4 digits
0 0 X1 . X2 0 0 X3 . X4	Content of previous 4-digit address
0 1 A1 . A2 0 0 A3 . A4	Address
0 0 X1 . X2 0 0 X3 . X4	Content
(Etc.)	(Etc.)
1 0 0 0 0 . 0 0 0	Leader-trailer code

The RIM loader can only be used in conjunction with the ASR33 reader (not the high-speed perforated-tape reader). Because a tape in RIM format is, in effect, twice as long as it need be, it is suggested that the RIM loader be used only to read the binary loader when using the ASR33. (Note that PDP-8 diagnostic program tapes are in RIM format.)

The complete PDP-8/I RIM loader (SA = 7756) is as follows:

Absolute Address	Octal Content	Tag	Instruction I Z	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 CONTENT
7774,	3376		DCA TEMP	/STORE ADDRESS
7775,	5356		JMP BEG	/NEXT WORD
7776,	0	TEMP,	0	/TEMP STORAGE
7777,	5XXX		JMP X	/JMP START OF BIN LOADER

Placing the RIM loader in core memory by way of the operator console keys and switches is accomplished as follows:

1. Set the starting address 7756 in the switch register (SR).
2. Press LOAD ADDRESS key.
3. Set the first instruction (6032) in the SR.
4. Press the DEPOSIT key.
5. Set the next instruction (6031) in the SR.
6. Press DEPOSIT key.
7. Repeat steps 5 and 6 until all 16 instructions have been deposited.

To load a tape in RIM format, place the tape in the reader, set the SR to the starting address 7756 of the RIM loader (not of the program being read), press the LOAD ADDRESS key, press the START key, and start the Teletype reader.

Refer to Digital Program Library document DEC-08-LRAA-D for additional information on the Readin Mode Loader program.

BINARY LOADER

The binary loader (BIN) is used to read machine language tapes (in binary format) produced by the program assembly language (PAL). A tape in binary format is about one-half the length of the comparable RIM format tape. It can, therefore, be read about twice as fast as a RIM tape and is, for this reason, the more desirable format to use with the 10 cps ASR33 reader or the Type PR8/I High-Speed Perforated-Tape Reader.

The format of a binary tape is as follows:

LEADER: about 2 feet of leader-trailer codes.

BODY: characters representing the absolute, machine language program in easy-to-read binary (or octal) form. The section of tape may contain characters representing instructions (channel 8 and 7 not punched) or origin resettings (channel 8 not punched, channel 7 punched) and is concluded by 2 characters (channel 8 and 7 not punched) that represent a check sum for the entire section.

TRAILER: same as leader.

Example of the format of a binary tape:

Tape Channel 8 7 6 5 4 S 3 2 1	Memory Location	Content	Comments
1 0 0 0 0 . 0 0 0			leader-trailer code
0 1 0 0 0 . 0 1 0 0 0 0 0 0 . 0 0 0		0200	
0 0 1 1 1 . 0 1 0 0 0 0 0 0 . 0 0 0	0200	CLA	origin-setting
0 0 0 0 1 . 0 1 0 0 0 1 1 1 . 1 1 1	0201	TAD 277	
0 0 0 1 1 . 0 1 0 0 0 1 1 1 . 1 1 0	0202	DCA 276	
0 0 1 1 1 . 1 0 0 0 0 0 0 0 . 0 1 0	0203	HLT	
0 1 0 0 0 . 0 1 0 0 0 1 1 1 . 1 1 1		0277	origin-setting
0 0 0 0 0 . 0 0 0 0 0 1 0 1 . 0 1 1	0277	0053	
0 0 0 0 1 . 0 0 0 0 0 0 0 0 . 1 1 1		1007	sum check
1 0 0 0 0 . 0 0 0			leader-trailer code

After a BIN tape has been read in, one of the two following conditions exists:

a. No checksum error: halt with AC = 0
b. Checksum error: halt with AC = (completed checksum) — (tape checksum)

Operation of the BIN loader in no way depends upon or uses the RIM loader. To load a tape in BIN format place the tape in the reader, set the SR to 7777 (the starting address of the BIN loader), press the LOAD ADDRESS key, set SR switch 0 up for loading via the Teletype unit or down for loading via the high speed reader, then press the START key, and start the tape reader.

Refer to Digital Program Library document Digital-8-2-U [DEC-08-LBAA-D] for additional information on the Binary Loader program.

APPENDIX E

SCALES OF NOTATION

2^x IN DECIMAL

x	2^x	x	2^x	x	2^x
0.001	1.00069 33874 62581	0.01	1.00695 55500 56719	0.1	1.07177 34625 36293
0.002	1.00138 72557 11335	0.02	1.01395 94797 90029	0.2	1.14869 83549 97035
0.003	1.00208 16050 79633	0.03	1.02101 21257 07193	0.3	1.23114 44133 44916
0.004	1.00277 64359 01078	0.04	1.02811 38266 56067	0.4	1.31950 79107 72894
0.005	1.00347 17485 09503	0.05	1.03526 49238 41377	0.5	1.41421 35623 73095
0.006	1.00416 75432 38973	0.06	1.04246 57608 41121	0.6	1.51571 65665 10398
0.007	1.00486 38204 23785	0.07	1.04971 66836 23067	0.7	1.62450 47927 12471
0.008	1.00556 05803 98468	0.08	1.05701 80405 61380	0.8	1.74110 11265 92248
0.009	1.00625 78234 97782	0.09	1.06437 01824 53360	0.9	1.86606 59830 73615

$10^{\pm n}$ IN OCTAL

10^n	n	10^{-n}	10^n	n	10^{-n}
1	0	1.000 000 000 000 000 000 00	112 402 762 000	10	0.000 000 000 006 676 337 66
12	1	0.063 146 314 631 463 146 31	1 351 035 564 000	11	0.000 000 000 000 537 657 77
144	2	0.005 075 341 217 270 243 66	16 432 451 210 000	12	0.000 000 000 000 043 136 32
1 750	3	0.000 406 111 564 570 651 77	221 411 634 520 000	13	0.000 000 000 000 003 411 35
23 420	4	0.000 032 155 613 530 704 15	2 657 142 036 440 000	14	0.000 000 000 000 000 264 11
303 240	5	0.000 002 476 132 610 706 64	34 327 724 461 500 000	15	0.000 000 000 000 000 022 01
3 641 100	6	0.000 000 206 157 364 055 37	434 157 115 760 200 000	16	0.000 000 000 000 000 001 63
46 113 200	7	0.000 000 015 327 745 152 75	5 432 127 413 542 400 000	17	0.000 000 000 000 000 000 14
575 360 400	8	0.000 000 001 257 143 561 06	67 405 553 164 731 000 000	18	0.000 000 000 000 000 000 01
7 346 545 000	9	0.000 000 000 104 560 276 41			

$n \log_{10} 2$, $n \log_2 10$ IN DECIMAL

n	$n \log_{10} 2$	$n \log_2 10$	n	$n \log_{10} 2$	$n \log_2 10$
1	0.30102 99957	3.32192 80949	6	1.80617 99740	19.93156 85693
2	0.60205 99913	6.64385 61898	7	2.10720 99696	23.25349 66642
3	0.90308 99870	9.96578 42847	8	2.40823 99653	26.57542 47591
4	1.20411 99827	13.28771 23795	9	2.70926 99610	29.89735 28540
5	1.50514 99783	16.60964 04744	10	3.01029 99566	33.21928 09489

ADDITION AND MULTIPLICATION TABLES

Addition | Multiplication

Binary Scale

$$0+0=0 \quad 0+1=1+0=1 \quad 1+1=10$$

$$0\times0=0 \quad 0\times1=1\times0=0 \quad 1\times1=1$$

Octal Scale

	01	02	03	04	05	06	07
1	02	03	04	05	06	07	10
2	03	04	05	06	07	10	11
3	04	05	06	07	10	11	12
4	05	06	07	10	11	12	13
5	06	07	10	11	12	13	14
6	07	10	11	12	13	14	15
7	10	11	12	13	14	15	16

	02	03	04	05	06	07
2	04	06	10	12	14	16
3	06	11	14	17	22	25
4	10	14	20	24	30	34
5	12	17	24	31	36	43
6	14	22	30	36	44	52
7	16	25	34	43	52	61

MATHEMATICAL CONSTANTS IN OCTAL SCALE

$\pi = 3.11037\ 552421_8$	$e = 2.55760\ 521305_8$	$\gamma = 0.44742\ 147707_8$
$\pi^{-1} = 0.24276\ 301556_8$	$e^{-1} = 0.27426\ 530661_8$	$\ln \gamma = -0.43127\ 233602_8$
$\sqrt{\pi} = 1.61337\ 611067_8$	$\sqrt{e} = 1.51411\ 230704_8$	$\log_2 \gamma = -0.62573\ 030645_8$
$\ln \pi = 1.11206\ 404435_8$	$\log_{10} e = 0.33626\ 754251_8$	$\sqrt{2} = 1.32404\ 746320_8$
$\log_2 \pi = 1.51544\ 163223_8$	$\log_2 e = 1.34252\ 166245_8$	$\ln 2 = 0.54271\ 027760_8$
$\sqrt{10} = 3.12305\ 407267_8$	$\log_2 10 = 3.24464\ 741136_8$	$\ln 10 = 2.23273\ 067355_8$

APPENDIX F

POWERS OF TWO

2^n	n	n^{-2}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 808 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 848	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 081 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
562 949 953 421 312	49	0.000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
1 125 899 906 842 634	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
2 251 799 813 985 248	51	0.000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
4 503 599 627 370 496	52	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 668 164 062 5
9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 834 582 031 25
18 014 398 509 481 984	54	0.000 000 000 000 000 055 511 151 231 257 827 021 171 513 417 041 015 625
36 028 797 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 708 520 507 812 5
72 057 594 037 927 936	56	0.000 000 000 000 000 013 877 787 807 814 456 755 215 395 854 260 253 906 25
144 115 188 075 855 872	57	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 927 130 126 953 125
288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 963 565 063 476 562 5
576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 782 531 738 281 25
1 152 921 504 606 846 976	60	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 891 265 869 140 625

APPENDIX G

OCTAL-DECIMAL CONVERSION

OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal) | 0000 to 0511 (Decimal)

Octal	Decimal
10000	4096
20000	8192
30000	12288
40000	16384
50000	20480
60000	24576
70000	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal) | 0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

2000 to 2777 (Octal)	1024 to 1535 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

3000 to 3777 (Octal)	1536 to 2047 (Decimal)

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

4000 to 4777 (Octal)	2048 to 2559 (Decimal)

Octal	Decimal
10000	4096
20000	8192
30000	12288
40000	16384
50000	20480
60000	24576
70000	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 to 5777 (Octal)	2560 to 3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

6000 to 6777 (Octal) | 3072 to 3583 (Decimal)

Octal	Decimal
10000	4096
20000	8192
30000	12288
40000	16384
50000	20480
60000	24576
70000	28672

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

7000 to 7777 (Octal) | 3584 to 4095 (Decimal)

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	1022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

OCTAL-DECIMAL FRACTION CONVERSION TABLE

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

OCTAL-DECIMAL FRACTION CONVERSION TABLE (continued)

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

OCTAL-DECIMAL FRACTION CONVERSION TABLE (continued)

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001872
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

DEC PRODUCT SUMMARY

DIGITAL MODULES

DIGITAL is one of the world's largest suppliers of digital circuit modules. DIGITAL modules have been used in computers, interfaces and special-purpose systems since 1958.

The two major module lines are the M Series TTL integrated circuit modules and the K Series low-speed noise-immune logic. The M Series line has now been expanded to over 60 modules. The K Series product offering has also been expanded, and now comprises nearly 70 modules, all compatible with M Series. Complementing its basic logic, expanded K series offers specialized functional modules such as sensor converters, communication interfaces, 120 V ac interfaces, drivers for solenoids and motors, and logic level converters for tying either high-speed or noise-sensitive logic to other devices.

As the cost of the logic itself decreases, it becomes increasingly important that efficient, reliable and inexpensive hardware be available to keep total system costs down. DIGITAL provides this hardware. Now, from a few to thousands of modules can be connected, wired, mounted, powered and enclosed efficiently with the lowest-cost-per-function in the industry.

DIGITAL's complete line of power and hardware accessories provides everything needed to put your designs into action, from connector blocks to mounting cabinets. Seven power supplies, six connector block variations; ten mounting panels, twelve blank module configurations and five types of connector cards are among the well over 100 different hardware and accessory items offered by DIGITAL.

PDP-10

PDP-10 is an expandable, 36-bit computer system designed to perform conversational time-sharing, real time applications, and batch processing simultaneously. It is currently serving in such wide spread applications as physics and bio-medical research, process control, university and industrial computation, time-sharing utilities, chemical research, and hybrid simulation.

The PDP-10 includes an extremely powerful processor with 16 general purpose registers for use as accumulators or index registers, from 16,384 to 262,144 words of 36-bit core memory, and a seven level priority interrupt subsystem. The PDP-10 features an I/O bus which provides 200K word/sec transfer rate and interfaces up to 128 devices or device controllers with the processor. It has 366 easy-to-learn and logically complete instructions.

All PDP-10 systems have in common two basic hardware elements: the central processor and core memory. The same central processor is used in every PDP-10 configuration, but core memory can be composed of any mix of several modules, which vary in size, speed, and cost.

The remainder of the system depends upon the functions it will perform. Software choices include three levels of monitors: single user, multiprogramming, and the multiprogramming/swapping monitor. Most software is re-entrant so that a single compiler can serve many users simultaneously, allowing more users to reside in core at the same time and providing better system response.

Other software includes FORTRAN IV, BASIC, AID, COBOL, a macro assembler, a context editor, a symbolic debugging program, a peripheral interchange program, a desk calculator and FORTRAN library programs. All software systems assure upward compatibility from the standard 16,384 words of memory through the multiprogramming and swapping systems at both the symbolic and relocatable binary level.

Hardware choices include three types of disk systems, magnetic tape units, DECtape systems, line printers, card readers and card punches, communications equipment, CRT displays, plotters, and real-time interface equipment.

PDP-10 features a 1-microsecond cycle time, a 2.6 microsecond add time and a modular, proven software package that expands to make full use of all hardware configurations. Memory can be expanded in 8,192, 16,384, or 32,768 word increments to the maximum directly addressable 262,144 words.

PDP-12

Digital's PDP-12 is a general-purpose computer system. It is designed as a simple to operate, yet uniquely flexible tool for a wide variety of research and real-time data handling applications.

Performance characteristics of the PDP-12 have been optimized around a complete hardware/software system, rather than an expandable minimum configuration. The PDP-12 systems concept works to the advantage of users at all levels of programming sophistication. By simplifying programming tasks, the PDP-12 frees users from the mere mechanics of program preparation to concentrate on the more creative aspects of their work.

The following is a brief list of some of the PDP-12's outstanding features:

- All-new, unified display-based programming system
- Automatic program loading from magnetic tape
- Built-in program debugging hardware
- 7" x 9" CRT display with graphic and alphanumeric capabilities
- Large existing library of applications programs
- TTL integrated-circuit modules throughout
- LINCtape-addressable, bi-directional program and data storage
- Free-standing cabinet and console table

Specifications:

- 4,096 12-bit words of core memory, expandable to 32,768 words
- 1.6 microsecond cycle time
- 43 basic instructions including 29 memory reference instructions
- 15 auto-index registers
- 6 programmable SPDT relays
- 6 sense switches
- 12 external sense lines
- Peripherals including 16-channel A/D converter, DECtape units, and new 7" x 9" display, all fully buffered
- Software: new unified display-based system; FORTRAN, FOCAL, BASIC, mathematical, maintenance, and utility routines

PDP-15

PDP-15 systems offer comprehensive solutions to real-time data problems. They combine new design concepts with a wide array of traditional features that spring from Digital's years of leadership in the medium-scale scientific computer field. Both elements share the common purpose of simplifying the user's tasks in a demanding real-time environment.

Since certain types of data-handling tasks require specific hardware and software configurations, Digital has developed four standard PDP-15 systems, ranging in power from the modestly priced basic PDP-15/10 to the real-time disk monitor environment of the PDP-15/40. At every level, the capabilities of the hardware are under the control of a monitor designed specifically for them, so that for every step of hardware growth there is a straightforward step of software control to match.

- PDP-15/10: 4,096 18-bit words of 800-nanosecond core memory, Teletype Model ASR-33 console teleprinter.
  COMPACT Software System including assembler, editor, debugging aids, and mathematical and utility routines.

- PDP-15/20 Advanced Monitor System: 8,192 words of core memory, KSR-35 Teletype for extra reliability, two DECtape transports and control unit, high-speed paper tape reader/punch, Extended Arithmetic Element for high-speed arithmetic operations and register manipulation.
  Advanced Monitor System with FORTRAN IV, FOCAL-15, MACRO-15 macro assembler, linking loader, batch processor, system generator, scientific library, and comprehensive debugging and utility routines.

- PDP-15/30 Background/Foreground System: 16,384 words of core memory, KSR-35 Teletype, Extended Arithmetic Element, Automatic Priority Interrupt system, Memory Protect system, high-speed paper tape reader/punch, three DECtape transports and control unit real-time clock, and a second on-line Teletype for background use.

  Background/Foreground Monitor System combining all Advanced Monitor functions with concurrent execution of real-time foreground tasks and program development or other low-priority background computation.

- PDP-15/40 Disk-Oriented Background/Foreground System: 24,576 words of core memory, KSR-35 Teletype, Automatic Priority Interrupt system, Relocation/Protect system, high-speed paper tape reader/punch, two DECtape transports and control unit, RF15 DECdisk control and two RS09 random-access disk files, real-time clock, and a second on-line Teletype for background use.
  Disk-oriented version of the Background/Foreground Monitor system, allowing concurrent execution of real-time tasks and background computation, in addition to all standard Advanced Monitor functions.

COMPUTER LAB

The COMPUTER LAB is a new high performance low-cost digital logic trainer. The COMPUTER LAB uses the same monolithic integrated transistor-transistor logic circuitry used in DIGITAL's latest PDP computers.

The digital logic fundamentals presented by the COMPUTER LAB constitute the basic knowledge required to pursue a career in computer technology as computer technician, engineer, programmer or operator. The COMPUTER LAB will also help the math-oriented student to understand "New Math" concepts because computer logic operates with binary numbers according to Boolean algebraic laws.

Wiring is easy because of the standard logic symbology used on the front panel and the color coded Patchcords which are easily inserted and removed. An improper circuit will not damage the COMPUTER LAB. The faulty circuit merely "waits" for correction.

Features:

- Transistor—Transistor logic circuitry as used in DIGITAL's PDP computers
- Teaches modern computer logic
- Easy to use: MIL-STD 806 logic symbology on front panel
- Portable: Dimensions of 12½" x 17" x 3¼", weighing only 11 lbs.
- Comprehensive Workbook provides:
  - —Ten detailed chapters
  - —More than 30 experiments
  - —Over 200 hours of laboratory study
  - —Dozens of tables and diagrams
  - —An extensive appendix of supplementary information
- Teacher's Guide with answers, additional text, extra problems, course plans, at only $5.00
- Low cost: COMPUTER LAB, Workbook and Patchcord set, ready to use $445.00

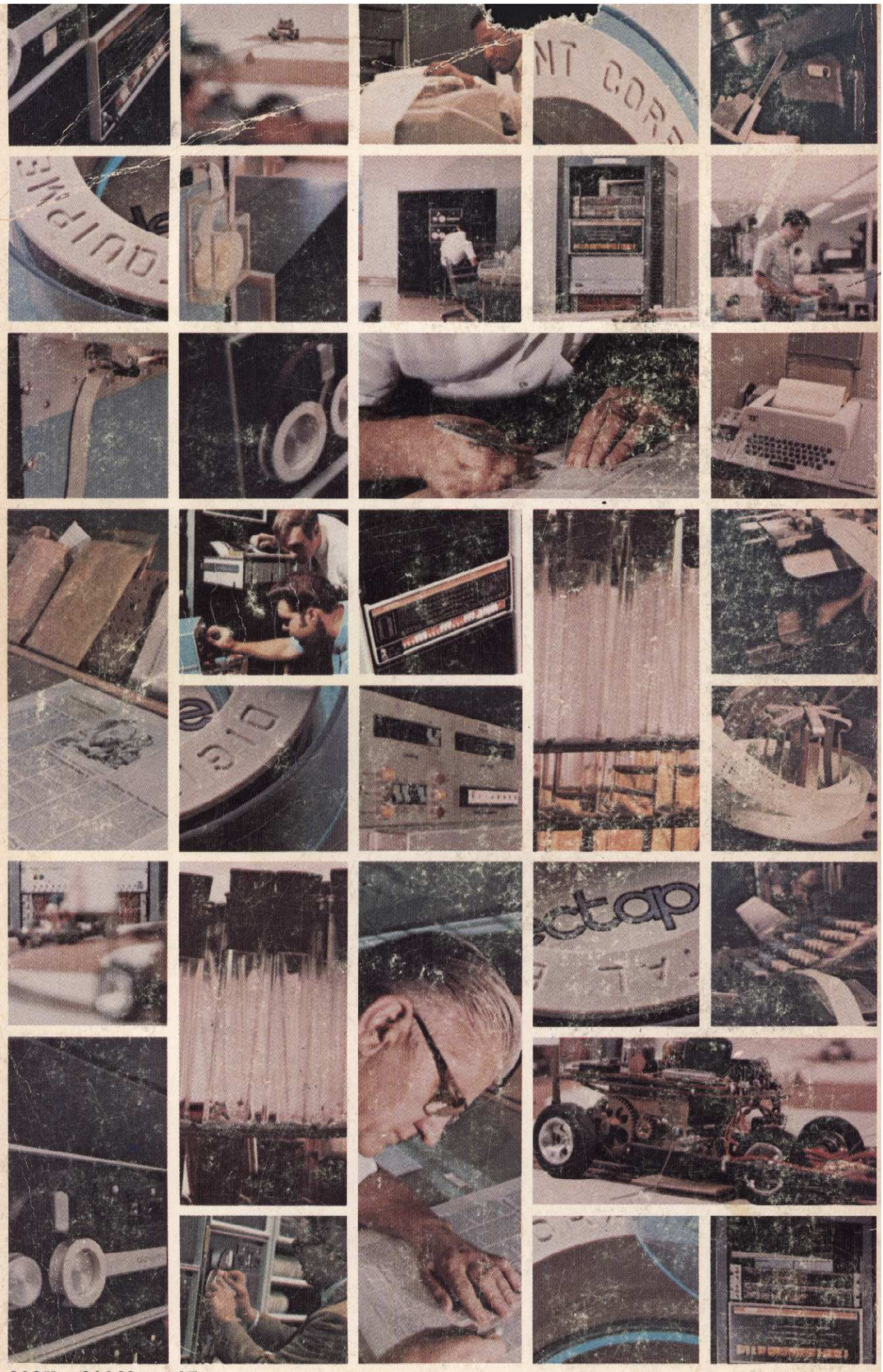
ALPHABETICAL INDEX

Alphabetical Index (Cont.)

Alphabetical Index (Cont.)

Alphabetical Index (Cont.)

Alphabetical Index (Cont.)



082X 01169 AJF
F 09 100