F-75P

REFERENCE MANUAL

PDP-7

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

F-75P

Preliminary

PROGRAMMED DATA PROCESSOR - 7

REFERENCE MANUAL

December 1964

DIGITAL EQUIPMENT CORPORATION · MAYNARD, MASSACHUSETTS

The information contained within this manual is preliminary and subject to change without notice. Any comments concerning this manual should be addressed to:

Digital Equipment Corporation
Program Documentation Department
146 Main Street
Maynard, Massachusetts 01754

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

ILLUSTRATIONS

ILLUSTRATIONS (continued)

CHAPTER 1

PDP-7 SYSTEM DESCRIPTION

PROGRAMMED DATA PROCESSOR-7 (PDP-7) is a general purpose, solid state, digital computer designed for high speed data handling in the scientific laboratory, the computing center, or the real time process control system. PDP-7 is a single address, fixed 18-bit word length, binary computer using 1's complement arithmetic and 2's complement arithmetic to facilitate multi-precision arithmetic. A random access magnetic core memory with a complete cycle time of 1.75 microseconds is used to achieve a computation rate of 285,000 additions per second.

The PDP-7 is completely self-contained, requiring no special power sources, air conditioning, or floor bracing. From a single source of 115-volt, 60-cycle, single-phase power, the PDP-7 produces circuit operating dc voltages of -15 volts (±1) and +10 volts (±1) which are varied for marginal checking. Total power consumption is 2000 watts. It is constructed with standard DEC FLIP CHIP™ modules and power supplies. Solid-state components and built-in marginal checking facilities insure reliable machine operation.

Input/output to the PDP-7 is fast parallel transfer and may be connected to a variety of peripheral equipment. In addition to the teleprinter, keyboard and high-speed perforated tape reader and punch supplied with the basic computer, the PDP-7 optional peripheral equipment includes magnetic tape equipment, card equipment and line printers, serial drums, cathode-ray tube displays, a data communication system, and analog-to-digital converters. Special purpose I/O equipment is easily connected using an interface of standard DEC modules.

The basic PDP-7 includes the central processor and control console; 4096 word core memory; input/output control with device selector (up to 64 I/O connections), information collector (seven 18-bit channels), information distributor (six 18-bit channels), program interrupt, data interrupt, I/O trap, I/O skip facility, I/O status check, and real time clock. A high speed paper tape reader (300 cps), high speed paper tape punch (63.3 cps), and KSR 33 teleprinter (10 cps) are standard input/output equipment with the basic PDP-7.

PROCESSOR

The processor performs logical and arithmetic functions, provides access to and from memory and controls the flow of data to and from the computer. It consists of the processor control, the memory, memory control and six other active registers.

ACCUMULATOR (AC) is an 18-bit register which performs arithmetic and logical operations on the data and acts as a transfer register through which data passes to and from the I/O buffer registers.

™FLIP CHIP is a trademark of the Digital Equipment Corporation

LINK (L) is a 1-bit register used to extend the arithmetic facility of the accumulator.

MEMORY ADDRESS REGISTER (MA) is a 13-bit register which holds the address of the core memory cell currently being used.

MEMORY BUFFER REGISTER (MB) is an 18-bit register which acts as a buffer for all information sent to or received from memory.

INSTRUCTION REGISTER (IR) is a 4-bit register which holds the operation code of the program instruction currently being performed.

PROGRAM COUNTER (PC) is a 13-bit register which holds the address of the next memory cell from which an instruction is to be taken.

MEMORY

The high speed random access memory is a 4096 word coincident-current core module with a cycle time of 1.75 microseconds. In one cycle the memory control retrieves an 18-bit word stored in the memory cell specified by the memory address register, writes the word by a parallel transfer into the memory buffer register, and rewrites the word into the same memory cell.

CENTRAL PROCESSOR OPTIONS

CORE MEMORY MODULE TYPE 147

The Core Memory Module extends the memory capacity of the PDP-7 from 4096 words to 8192 words.

CORE MEMORY EXTENSION CONTROL TYPE 148

The Memory Extension Control allows the expansion of the PDP-7 memory from 8192 to 32,768 words in increments of either 4096 or 8192 words, using the Type 149 Memory Modules. The Type 148 includes an Extended Program Counter, an Extended Memory Address Register, and an Extend Mode Control.

EXTENDED ARITHMETIC ELEMENT TYPE 177

The Extended Arithmetic Element is a standard option for the PDP-7 which facilitates high-speed multiplication, division, shifting, and register manipulation. Installation of the EAE adds an 18-bit register, the Multiplier Quotient Register (MQ) to the computer as well as a 6-bit step counter register. The contents of the MQ register are continuously displayed on the operator's console just below the accumulator indicators.

The Type 177 and the basic computer cycle operate asynchronously, permitting computations to be performed in the minimum possible time. Further, since the EAE instructions are microcoded, several operations can be performed by one instruction, thus simplifying associated programming. Average multiplication time is 6.1 μsec, average division time is 9 μsec.

AUTOMATIC PRIORITY INTERRUPT TYPE 172

The Automatic Priority Interrupt increases the capacity of the PDP-7 to handle transfers of information to and from input/output devices. The 172 identifies an interrupting device directly, without the need for flag searching. Multilevel interrupts are permissible where a device of higher priority supersedes an interrupt already in process. These functions increase the speed of the input/output system and simplify the programming. More and faster devices can therefore be serviced efficiently.

The Type 172 contains 16 automatic interrupt channels arranged in a priority chain so that channel 0 has the highest priority and channel 17_8 has the lowest priority. The priority chain guarantees that if two or more in-out devices request an interrupt concurrently, the system grants the interrupt to the device with the highest priority. The other interrupts will be serviced afterwards in priority order.

INPUT/OUTPUT CONTROL

The I/O control links up to 64 input and output stations by lines to the central processor, calls the stations, and collects and distributes the input/output data. It also controls the interleaving of data during a data interrupt, senses the status of I/O devices and skips instructions based on this status, traps IOT (input-output-transfer) instructions initiating a program break, and generates real time signal pulses for use by external peripheral equipment.

No additional interface equipment is required to attach the standard PDP-7 Peripheral Equipment. Word buffers are included within units of standard I/O optional equipment so that the basic PDP-7 can simultaneously control data transfer between several I/O devices. Special-purpose I/O equipment is easily connected to the PDP-7 by assembling an interface using the standard line of FLIP CHIP modules manufactured by DEC.

INPUT/OUTPUT EQUIPMENT

Mass Storage

DECtape DUAL TRANSPORT TYPE 555

A fixed address magnetic tape facility for high speed loading, readout, and on-line program debugging. Read, write, and search speed is 80 inches a second. Density is 375 bits an inch. The two logically independent transports have a storage capacity of 3 million bits each. Features phase recording, rather than amplitude recording; redundant, nonadjacent data tracks; and a prerecorded timing and mark track.

DECtape CONTROL TYPE 550

Controls up to four Type 555 Dual DECtape Transports. Searches in either direction for specified block numbers, then reads or writes data. Units as small as a single word may be addressed.

AUTOMATIC MAGNETIC TAPE CONTROL TYPE 57A

Controls up to eight tape transports automatically. Provides information transfer through computer's data interrupt facility, permitting interlaced program and tape operation. Controls reading or writing of tape at various rates compatible with IBM, BCD, or binary parity modes.

MAGNETIC TAPE TRANSPORT TYPE 570

Tape motion is controlled by pneumatic capstans and brakes, eliminating conventional pinch rollers, clamps, and mechanical arms. Tape speed is either 75 or 112.5 inches per second. Track density, program-selectable, is 200, 556, and 800 bits per inch. Tape width is one-half inch, with six data tracks and one parity track. Format is IBM compatible. Dual heads permit read-checking while writing.

MAGNETIC TAPE TRANSPORT TYPE 50

Reads and writes IBM-compatible magnetic tape at transfer rates of 15,000 or 41,700 cps and 200 or 556 cpi.

BLOCK TRANSFER DRUM SYSTEM TYPE 24

Drum transfers operate through the computer's data interrupt facility permitting interlaced program and drum transfer operation. Storage capacities of 32,768 words, 65,536 words, or 131,072 words are available.

Displays

PRECISION CRT DISPLAY TYPE 30

Plots data point by point on a 16-inch cathode ray tube in a raster 9-3/8 inches square having 1024 points on a side. Separately variable 10-bit X and Y coordinates. Includes program intensity control. Plotting rate is 35 microseconds per point.

PRECISION INCREMENTAL CRT DISPLAY TYPE 340

Plots points, lines, vectors, and characters on a raster identical to the 30. Plotting rate is 1-1/2 microseconds per point in vector, increment, and character modes. Random point plotting is 35 microseconds.

OSCILLOSCOPE DISPLAY TYPE 34

Controls the plotting of data point by point on an X-Y plotting scope such as the Tektronix Model RM 503. Raster size is 1024 x 1024 points.

HIGH SPEED LIGHT PEN TYPE 370

Uses fiber optic light pipe and photomultiplier system for fast detection and modification of information displayed on the precision CRT display.

Analog-To-Digital

ANALOG-TO-DIGITAL CONVERTER TYPE 138B

Transforms an analog voltage to a binary number, selectable from 6 to 11 bits. Conversion time varies, depending on the number of bits and the accuracy required. Twenty-four combinations of switching point accuracy and number of bits can be selected on the front panel.

MULTIPLEXED ANALOG-TO-DIGITAL CONVERTER TYPE 138/139

The Type 139 Multiplexer Control permits up to 64 channels of analog information to be applied singly to the input of the Type 138B Analog-to-Digital Converter. Channels can be selected in sequence or by individual addresses.

HIGH-SPEED ANALOG-TO-DIGITAL CONVERTER TYPE 142

Transforms an analog voltage to a single, 10-bit binary number in 6 microseconds. Conversion accuracy is $\pm 0.15\% \pm 1/2$ least significant bit.

ANALOG-DIGITAL-ANALOG CONVERTER SYSTEM TYPE ADA-1

Performs fast, real-time conversion between digital and analog computers. Maximum sample rate for D/A conversion is 200 kc; for A/D and interlaced conversions, 100 kc. Digital word length is 10 bits. Actual conversion times are 5 microseconds for A/D and 2 microseconds for D/A. Semiautomatic features enable the converter system to perform many of the functions that a computer normally performs for other converter interfaces.

18-BIT OUTPUT RELAY BUFFER TYPE 140

18 spdt relays actuated by computer command for use to directly control or signal external equipment.

INCREMENTAL PLOTTER AND CONTROL TYPE 350

Performs high resolution plotting on paper 12 or 31 inches wide at rates of 12,000 or 18,000 points per minute. Plotting increments are 0.005 and 0.01 inch.

Card Equipment and Printer

CARD READER AND CONTROL TYPE 421A, B

Reads standard 12-row, 80-column punched cards at a rate of 200 or 800 cards per minute in either alphanumeric or binary modes.

CARD PUNCH CONTROL TYPE 40

Controls a card punch such as the IBM Model 523 Summary Punch. Punch Control Buffer holds one 80-bit row.

AUTOMATIC LINE PRINTER AND CONTROL TYPE 64

The automatic line printer prints lines of text 120 columns wide at a maximum rate of 300 lines per minute. Printing is performed by solenoid-actuated hammers. A 64-character set is provided.

Input/Output Systems

DATA COMMUNICATION SYSTEM TYPE 630

Provides a real-time interface for up to 64 remote typewriter stations for on-line inputs and outputs. Used in message switching, data collecting, and data processing in multi-user applications.

DATA INTERRUPT MULTIPLEXER TYPE 173

Provides multiplex control for simultaneous operation of three high-speed devices such as the Type 57A Tape Control or the Type 24 Drum. Maximum combined transfer rate is 570,000 18-bit words per second.

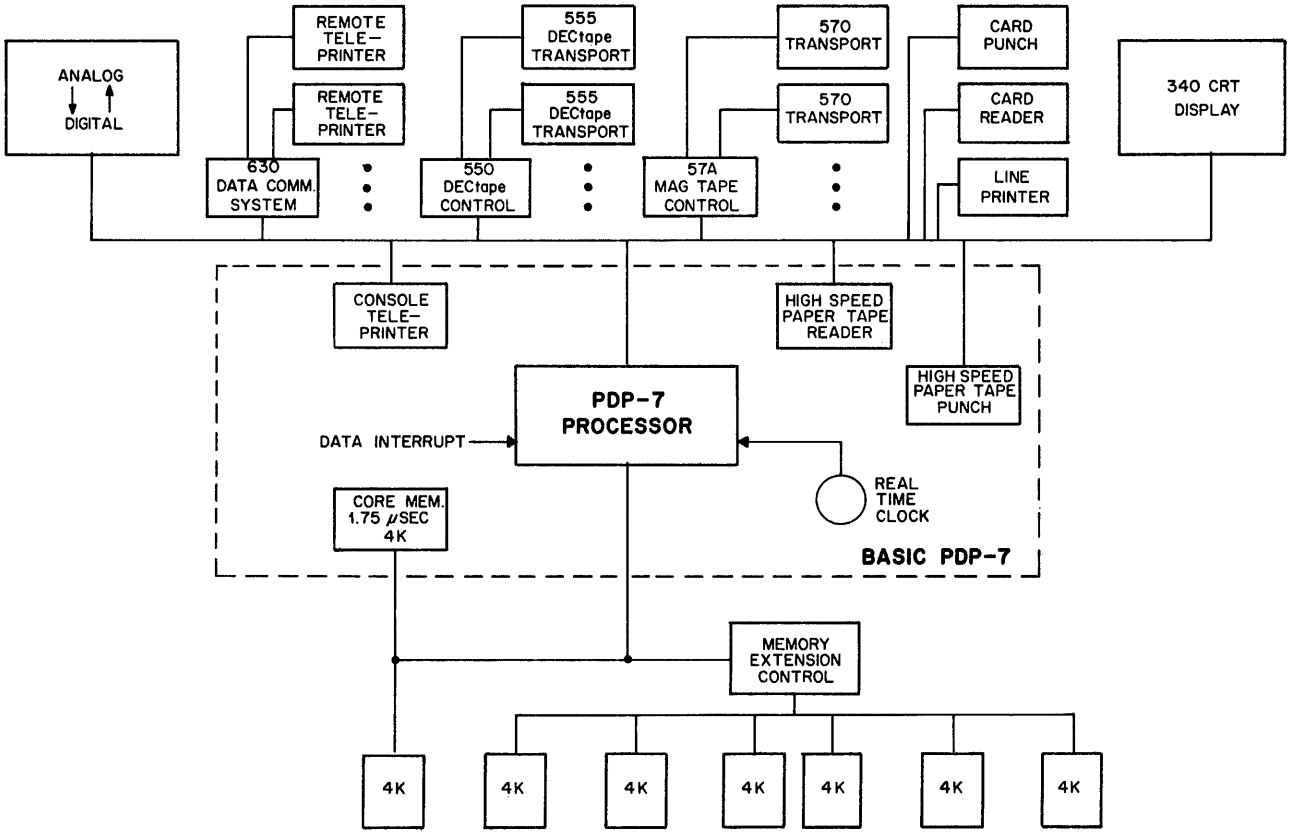


Figure 1-1 Expanded PDP-7 System

PROGRAMMING SYSTEM

The PDP-7 Programming System includes an advanced FORTRAN Compiler, a Symbolic Assembler, Editor, DDT Debugging System, Maintenance routines and a library of arithmetic, utility and programming aids developed on the program-compatible PDP-4. Both the Editor and DDT are designed to allow symbolic debugging and computer-aided editing to replace the tedious manual equivalent. New and updated programs are being developed continuously in the applied programming department.

SYMBOLIC ASSEMBLER

The Symbolic Assembler lets the programmer code instructions in a symbolic language. The assembler used on the PDP-7 allows mnemonic symbols to be used for instruction codes and addresses. Constant and variable storage registers can be automatically assigned. The assembler produces a binary object tape and lists a symbol table with memory allocations and useful diagnostic messages.

DIGITAL DEBUGGING TAPE (DDT)

DDT speeds program debugging by communicating with the user in the address symbols of the source language program. Program debugging time is further shortened when using DDT because program execution and modification are controlled from the teleprinter keyboard. For example, to branch to a new location in the program it is only necessary to type the symbolic location name on the keyboard followed by the character, single quote ('). The same symbol followed by the character, slash (/), causes the contents of that location to be typed. By using DDT to insert break points in a program, the programmer can make corrections or insert patches and try them out immediately. Working corrections can be punched out on the spot in the form of loadable patch tapes, eliminating the necessity of creating new symbolic tapes and reassembling each time an error is found.

SYMBOLIC EDITOR

The Editor permits the editing of source language programs by adding or deleting lines of text. All modification, reading, punching, etc., is controlled by symbols typed at the keyboard. The editor reads parts or all of a symbolic tape into memory where it is available for immediate examination, correction, and relisting.

FORTRAN COMPILER

The FORTRAN used with the PDP-7 is based on the field-proven FORTRAN II used with PDP-4 and is designed for programming flexibility and operating efficiency. An 8K memory is now required for FORTRAN with the PDP-7 to provide a program and data storage capacity commensurate with the power of the PDP-7 Processor. FORTRAN permits the PDP-7 user with little knowledge of the computer's organization and machine language to write effective programs. Programs are written in a language of familiar English words and mathematical symbols. Compilation of the original FORTRAN source program is performed separately from the compilation of associated subroutines. Thus, when errors in FORTRAN coding are detected by the compiler diagnostic, only the erroneous program need be recompiled.

BUS-PAK II

Designed for data processing operations, BUS-PAK is a program assembly system for use by the data processing programmer. Programs written using BUS-PAK enable the PDP-7 to function as a business-oriented computer equipped with a logical instruction set very similar to the instructions used by data processing computers. BUS-PAK operates in a character mode, has a built-in high-speed I/O control, is capable of single and double indexing, multilevel indirect addressing, and makes available 15 accumulators.

CHAPTER 2

OPERATION

This chapter describes console operation of the PDP-7 through use of the console lights, switches, and keys. A second section describes how to load basic system tapes into the computer.

OPERATOR CONSOLE

The operator console contains all of the manual controls necessary to start and stop the PDP-7, to observe the status of all active control processor registers, and to manually address, examine, and change the 18-bit contents of any location (word) in core memory. The functions of the console lights, switches, and keys are described in the following tables.

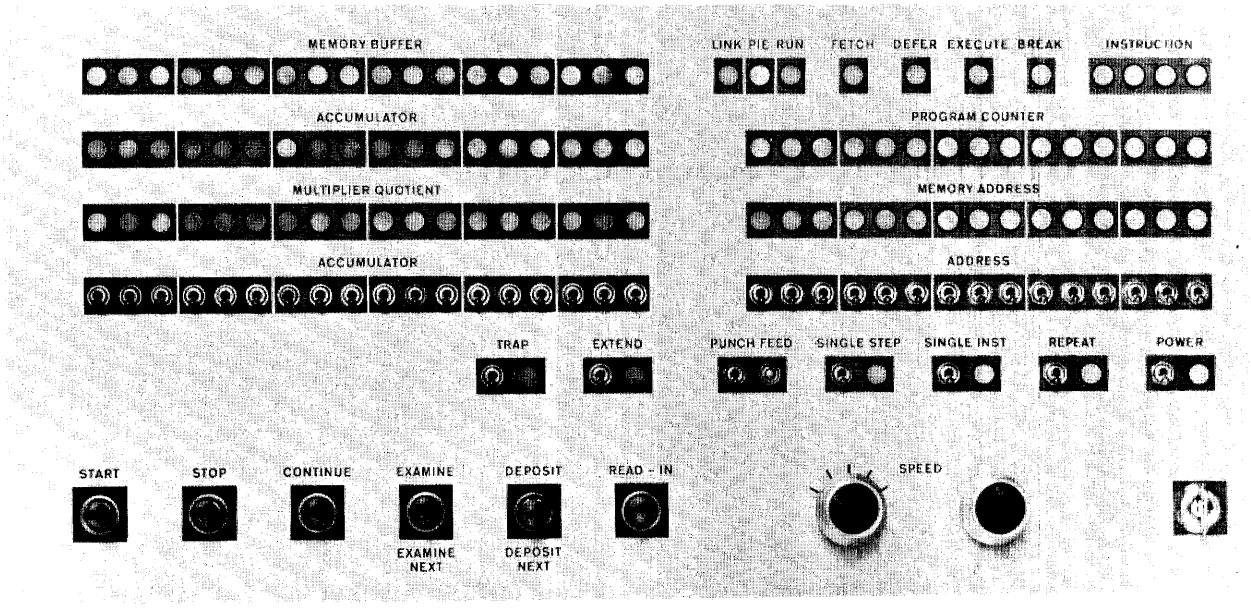


Figure 2-1 Operator Console

SWITCHES AND KEYS

Keys	Function
START	Starts the processor. The first instruction is taken from the memory cell specified by the setting of the ADDRESS switches. The START operation clears the AC and link, and turns off the program interrupt.
STOP	Stops the processor at the completion of the memory cycle in progress at the time of key operation.
CONTINUE	Causes the computer to resume operation from the point at which it was stopped. Besides the normal "off" and momentary "on" positions, this key has a latched "on" position obtained by raising the key instead of depressing it.
EXAMINE	Places the contents of the memory cell specified by the ADDRESS switches into the AC and MB. At the completion of the operation, the contents of the ADDRESS switches appear in the MA, and the PC contains the address of the next cell.
EXAMINE NEXT	Places the contents of the cell specified by the PC into the AC and MB. The C(PC) are incremented by 1, and the MA contains the address of the register examined.
DEPOSIT	Deposits the C(AC switches) into the memory cell specified by the ADDRESS switches. The C(AC switches) remain in the AC and MB. The contents of the ADDRESS switches appear in the MA. The PC contains the address of the next cell.
DEPOSIT NEXT	Deposits the C(AC switches) into the memory cell specified by the PC. The C(PC) are then incremented by 1. At the completion of the operation, the C(AC), C(MA) are the same as for DEPOSIT

READ-IN	Punched paper tape is read in binary mode into a core memory block. The first address of the memory block is given by the C(ADDRESS switches). Control then transfers to the central processor which interprets and executes the last word in the block.

Switches and Speed Controls	Function
ADDRESS	A group of 15 switches used to establish the memory address for the START, EXAMINE, and DEPOSIT operations.
ACCUMULATOR	A group of 18 switches used to set up the word to be placed in memory by the DEPOSIT and DEPOSIT NEXT operations, or the word to be placed in the AC by a program. These 18 switches are used for program sense control.
EXTEND	Enables the Extend Mode of the optional Type 148 Memory Extension Control to be used with all console keys and switches performing memory reference functions.
TRAP	Permits the Trap Mode to be engaged by the program.
PUNCH FEED	Switch - controls perforated tape punch power. When down, punch power is under program control. When up, punch power is on. Button - causes punch to punch tape leader. Punch power remains on for additional 5 seconds as it does under program control.
SINGLE STEP	Causes the computer to stop at the completion of each memory cycle. Repeated operation of CONTINUE while this switch is on steps the program one memory cycle at a time.

SINGLE INSTRUCTION	Causes the computer to stop at the completion of each instruction. Repeated operation of CONTINUE while this switch is on steps the program one instruction at a time. When both switches are on, SINGLE STEP takes precedence over SINGLE INSTRUCTION.
REPEAT	Causes the operations initiated by pressing CONTINUE, EXAMINE NEXT, or DEPOSIT NEXT to be repeated as long as the key is held on. The rate of repetition is controlled by the SPEED knob setting.
SPEED	Two controls that vary the REPEAT interval from approximately 40 microseconds to 8 seconds. The left knob is a 5-position coarse control, the right knob a continuously variable fine control. For both knobs, slowest speed is obtained in extreme left position.
POWER	Controls the primary power to the computer and all external devices attached to it.

CONSOLE LOCK

On the lower right-hand side of the console is a key-operated, 2-position lock. When the key is turned counterclockwise, the console is unlocked and all controls operate normally. When the key is turned clockwise, the console is locked; operation of any of the console keys, the speed controls, or the POWER, SINGLE STEP, SINGLE INSTRUCTION or REPEAT switches has no effect on the running of the computer.

CONSOLE LIGHTS

Lights	Indication
ACCUMULATOR	The contents of the AC
MULTIPLIER-QUOTIENT	The contents of the MQ
MEMORY BUFFER	The contents of the MB
INSTRUCTION	The contents of the IR
MEMORY ADDRESS	The contents of the MA
PROGRAM COUNTER	The contents of the PC
LINK	The contents of the link
PIE	Indicates when the Program Interrupt is Enabled
TRAP	Indicates when the Trap Mode is Enabled
EXTEND	Indicates when the Extend Mode is Enabled
RUN	The computer is executing instructions
FETCH, DEFER, EXECUTE, BREAK	The major control state of the next memory cycle
SINGLE STEP, REPEAT, SINGLE INST, POWER	Indicates the function is active

C (registers) are a binary display.

CHAPTER 3

CENTRAL PROCESSOR

ARITHMETIC AND CONTROL REGISTERS

Six active registers in the central processor are used to perform arithmetic operations, control memory access and to transfer information to and from the computer. Figure 3-1 shows the relation between the central processor registers and other elements of the computer.

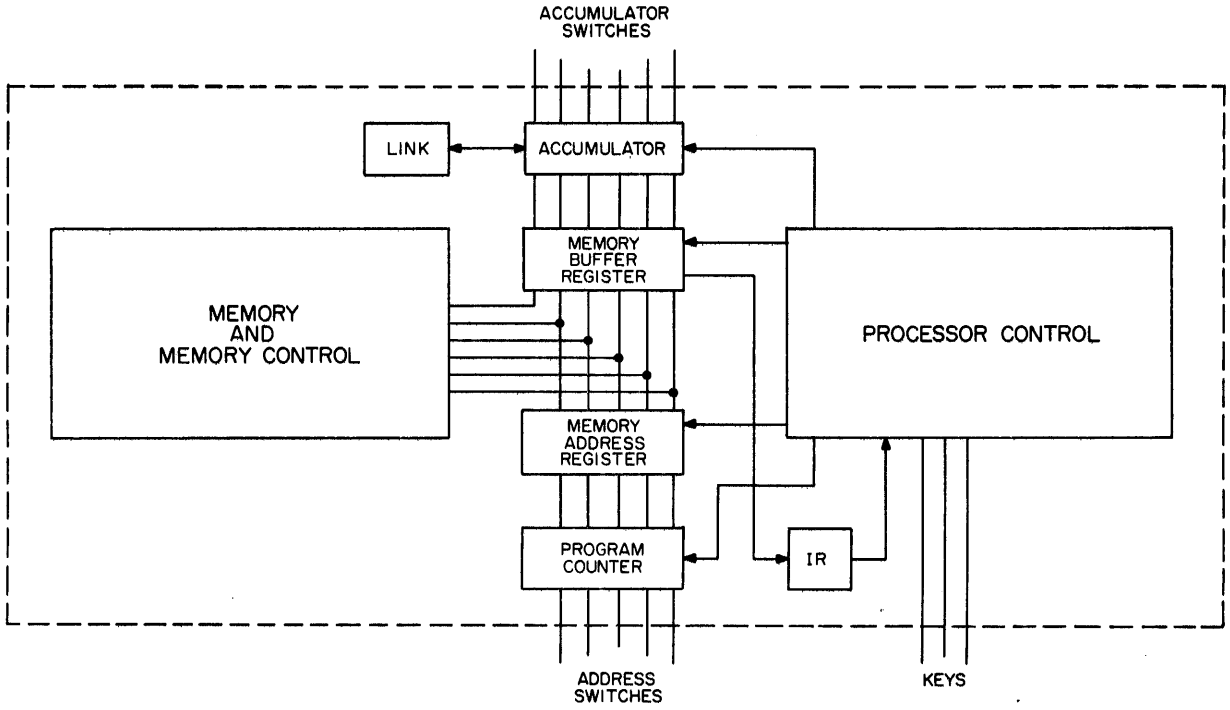


Figure 3-1 PDP-7 Central Processor and Memory

ACCUMULATOR (AC): Arithmetic operations are performed in this 18-bit register. The AC may be cleared and complemented. Its contents may be rotated right or left with the link. The contents of the Memory Buffer may be added to the contents of the AC with the result left in the AC. The contents of both registers may be combined by the logical operations AND and Exclusive OR, the result remaining in the AC. The Inclusive OR may be formed between the AC and the accumulator switches on the operator console (see below), and the result left in the AC. Except in data interrupt transfers, information is transferred between core memory and an external device through the accumulator.

If a 1-cycle instruction is fetched, the operations specified are performed during the last part of the fetch cycle. The next state is fetch. If a two-cycle instruction is fetched, the following control state is either defer or execute.

DEFER: When bit 4 of a memory reference instruction is a 1, the defer state is entered to perform the indirect addressing. The memory location addressed contains the address of the operand, and access to the operand is deferred to the next memory cycle.

EXECUTE: This state is established only when a memory reference instruction is being executed. The contents of the memory cell addressed are brought into the MB, and the operation specified by the contents of the IR is performed.

BREAK: When this state is established, the sequence of instructions is broken for a data interrupt or a program interrupt. In both cases, the break occurs only at the completion of the current instruction. The data interrupt allows information to be transferred between memory and an external device. When this transfer has been completed, the program sequence is resumed from the point of the break. The program interrupt causes the sequences to be altered. The contents of the PC and the contents of the Link are stored in location 0000, and the program continues from location 0001. See Chapter 4, Program Interrupt Control.

INSTRUCTIONS

Instructions are of two types: memory reference and augmented. Memory reference instructions require a memory address, while augmented instructions do not. For clarity, a set of basic logic symbols is used throughout the instruction set.

Instruction Symbol Definitions

SYMBOL	DEFINITION
Y	Designates any register in core memory
Yj	Designates the jth bit of register Y
Yj-k	Designates bits j through k of register Y
C(Y)	The contents of register Y
C(Y) = > C(Z)	The contents of Y replace the contents of Z
I	Designates an instruction, without reference to a register or core location

Instruction Symbol Definitions (continued)

SYMBOL	DEFINITION
I_j	The jth bit of an instruction. For example, $I_4 = 1$ means that bit 4 of an instruction code has a value of 1.
i	In a memory reference instruction, this signifies indirect addressing. It means that in the code for such an instruction $I_4 = 1$.
$\wedge$	And
$\vee$	Inclusive Or
$\not\vee$	Exclusive Or
$\overline{\quad}$ (overbar)	Complement. For example, $C(\overline{Y})$ signifies the complement of the contents of register Y.

Memory Reference Instructions

The bit assignments of the memory reference instruction are shown in Figure 3-2. Bits 0-3 determine the operation to be performed. Bits 5-17 specify the memory address. Bit 4 is used to specify indirect addressing.

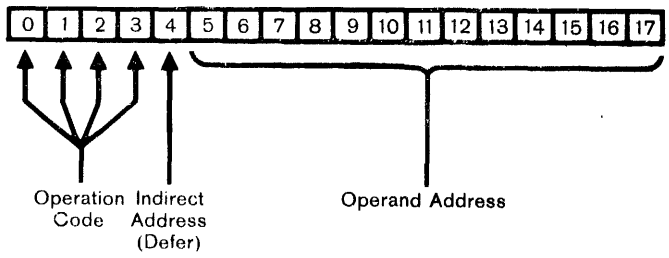


Figure 3-2 Memory Reference Instruction Format

Memory reference instructions require a fetch cycle to interpret the operation and determine the memory address, and an execute cycle to carry out the operation. Information is transferred from the AC to core memory through the Memory Buffer. When an operand is to be combined with the contents of the AC, it is brought from core storage into the MB; the operation is then performed between the AC and the MB. When indirect addressing is specified, an extra (defer) cycle is required to determine the effective address.

The jmp instruction requires an address but not an operand and thus is not a true memory reference instruction; although, it is convenient to list it with them. An execute cycle is not needed, and the instruction is completed in one cycle.

MEMORY REFERENCE INSTRUCTIONS

MNEMONIC SYMBOL	OCTAL CODE (Bits 0-3)	MACHINE CYCLES	OPERATION
lac Y	20	2	Load AC. The C(Y) are loaded into the AC. The previous C(AC) are lost; the C(Y) are unchanged. C(Y) = > C(AC)
dac Y	04	2	Deposit AC. The C(AC) are deposited in the memory cell at location Y. The previous C(Y) are lost; the C(AC) are unchanged. C(AC) = > C(Y)
dzm Y	14	2	Deposit zero in memory. Zero is deposited in memory cell Y. The original C(Y) are lost. The AC is unaffected by this operation. 0 = > C(Y)
add Y	30	2	Add (1's complement). The C(Y) are added to the C(AC) in 1's complement arithmetic. The result is left in the AC and the original C(AC) are lost. The C(Y) are unchanged. The link is set to 1 on overflow. C(Y) + C(AC) = > C(AC)
tad Y	34	2	Twos complement add. The C(Y) are added to the C(AC) in 2's complement arithmetic. The result is left in the AC and the original C(AC) are lost. The C(Y) are unchanged. A carry out of the 0 bit complements the link. C(Y) + C(AC) = > C(AC)
xor Y	24	2	Exclusive OR. The logical operation Exclusive OR is performed between the C(Y) and the C(AC). The result is left in the AC and the

MNEMONIC SYMBOL	OCTAL CODE (Bits 0-3)	MACHINE CYCLES	OPERATION
xor Y (continued)			original C(AC) are lost. The C(Y) are unchanged. Corresponding bits are compared independently. $C(Y_i) \not\vee C(AC_i) => C(AC_i)$. Example C(AC)$_i$ original / C(Y)$_i$ / C(AC)$_i$ final 0 / 0 / 0 0 / 1 / 1 1 / 0 / 1 1 / 1 / 0
and Y	50	2	AND. The logical operation AND is performed between the C(Y) and the C(AC). The result is left in the AC, and the original C(AC) are lost. The C(Y) are unchanged. Corresponding bits are compared independently. $C(Y_i) \wedge C(AC_i) => C(AC_i)$ Example C(AC)$_i$ original / C(Y)$_i$ / C(AC)$_i$ final 0 / 0 / 0 0 / 1 / 0 1 / 0 / 0 1 / 1 / 1
jmp Y	60	1	Jump to Y. The next instruction to be executed is taken from memory cell Y. Y => C(PC)
jms Y	10	2	Jump to subroutine. The C(PC) and the C(L) are deposited in memory cell Y. The next instruction is taken from cell Y + 1. $C(L) = C(Y_0)$. $0 => C(Y_{1-4})$ $C(PC) => C(Y_{5-17})$. $Y + 1 => C(PC)$

MEMORY REFERENCE INSTRUCTIONS (continued)

MNEMONIC SYMBOL	OCTAL CODE (Bits 0-3)	MACHINE CYCLES	OPERATION
cal	00	2	Call subroutine. The address portion of this instruction is ignored. The action is identical to jms 20. The instruction cal i is equivalent to jms i 20.
xct Y	40	1 + time of instruction being executed	Execute. The instruction in memory cell Y is executed. The computer acts as if the instruction located in Y were in the place of the xct, so that the PC sequence is unaltered.
sad Y	54	2	Skip if AC is different from Y. The C(Y) are compared with the C(AC). If the numbers are the same, the computer proceeds to the next instruction. If the numbers are different, the next instruction is skipped. The C(AC) and the C(Y) are unchanged. If C(AC) $\neq$ C(Y) then C(PC) + 1 => C(PC)
isz Y	44	2	Index and skip if zero. The C(Y) are incremented by one in 2's complement arithmetic. If the result is zero, the next instruction is skipped; if not, the computer proceeds to the next instruction. The C(AC) are unaffected. C(Y) + 1 => C(Y) If result = 0, C(PC) + 1 => C(PC)

Execution Time (μsec)

Instruction or program execution times can be computed in microseconds by multiplying the number of machine cycles by 1.75, the microsecond cycle time of the PDP-7.

AUGMENTED INSTRUCTIONS

Augmented instructions do not require a memory reference. As a result, an execute cycle is not needed, and the instructions in this class are completed in one cycle. Bits 4-17 of an augmented instruction are used to specify the operations to be performed. Several operations may be combined by microcoding in a single instruction, as explained below.

Operate Class

The group of augmented instructions with octal code 740000 is used to control bit manipulation and sensing in the operating registers of the PDP-7. Instructions of the 74 octal code belong to the Operate Class. These instructions are designed so that several of them may be combined in a single operate instruction. This is possible with operate instructions whose functions are performed at different event times during the 1.75 μsec instruction cycle. Figure 3-3 shows the Operate Class Instruction Bit Assignment.

A Table of Operate Instructions indicating event times follows. These event times are numbered 1, 2, and 3 and occur in that order. An operation which takes place at event time 2 is completed before event time 3 begins. Certain operations which take place at the same event time may not be combined in the same instruction.

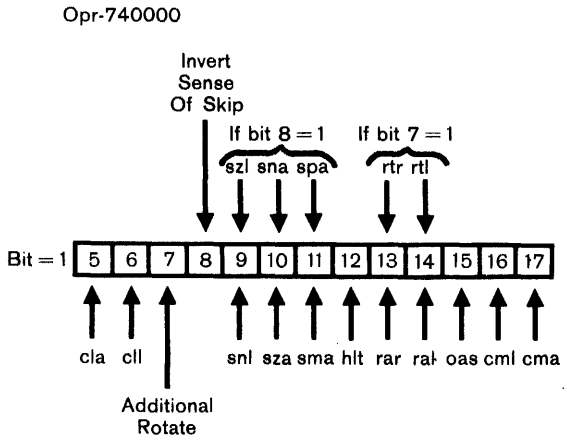


Figure 3-3 Operate Class Instruction - Bit Assignment

OPERATE INSTRUCTIONS

MNEMONIC SYMBOL	OCTAL CODE	EVENT TIME	OPERATION
opr	740000		Operate. Indicates the operate class. When used alone, performs no operation; the computer proceeds to the next instruction after one memory cycle.
cla	750000	2	Clear AC. The AC is cleared to 0. 0 => C(AC)
cma	740001	3	Complement AC. Each bit of the AC is complemented. $\overline{C(AC)}$ => (AC)
cll	744000	2	Clear link. Link is set to 0. 0 => C(L)
cml	740002	3	Complement link. $\overline{C(L)}$ => C(L)
ral	740010	3	Rotate AC left. The C(AC) and the C(L) are rotated left one place. $C(AC_i)$ => $C(AC_{i-1})$ $C(AC_0)$ => C(L). C(L) => $C(AC_{17})$
rtl	742010	2, 3	Rotate two places left. Equivalent to two successive ral's.
rar	740020	3	Rotate AC right. The C(AC) and the C(L) are rotated one place right. $C(AC_i)$ => $C(AC_{i-1})$ C(L) => $C(AC_0)$ $C(AC_{17})$ => C(L)

OPERATE INSTRUCTIONS (continued)

MNEMONIC SYMBOL	OCTAL CODE	EVENT TIME	OPERATION
rtr	742020	2, 3	Rotate two places right. Action taken is equivalent to two successive rar's.
oas	740004	3	OR AC switches. The Inclusive OR of the C(AC) and the C(AC switches) is placed in the AC. If an AC switch is down, it is interpreted as a 0; if up, as a 1. C(AC switches) V C(AC) => C(AC)
sma	740100	1	Skip if minus AC. If the AC is negative, the next instruction is skipped. If AC_0 =1, then C(PC) + 1 => C(PC)
spa	741100	1	Skip if plus AC. If the AC is positive, the next instruction is skipped. If AC_0 = 0, then C(PC) + 1 => C(PC)
sza	740200	1	Skip if zero AC. If C(AC) are 0, the next instruction is skipped. If C(AC) = 0, then C(PC) + 1 => C(PC)
sna	741200	1	Skip if non-zero AC. If C(AC) ≠ 0, then C(PC) + 1 => C(PC)
snl	740400	1	Skip if non-zero link. If C(L) is 1, the next instruction is skipped. If C(L) ≠ 0, then C(PC) + 1 => C(PC)
szl	741400	1	Skip if zero link. If C(L) = 0, then C(PC) +1 => C(PC)
hlt	740040	immediately after the completion of the cycle.	Halt. Stops the computer.

When skip operations are combined in a single instruction, the Inclusive OR of the conditions to be met determines whether or not the skip takes place. For example, if both sza and snl are specified (operation code 740600), the next instruction is skipped if either the C(AC) = 0, the C(L) = 1, or both. When the sense of the skip is inverted (I_8 = 1) in a combined skip, the skip takes place only if both of the conditions are met. For example, both sna and szl are specified (operation code 741600), the next instruction is not skipped if either the C(AC) = 0, the C(L) = 1, or both. The skip occurs only if both C(AC) $\neq$ 0 and C(L) = 0.

The nature of the rotate operations is such that no other operations may take place during the same event time. The following restrictions must therefore be observed:

rar and ral may not be combined with oas, cml, or cma.
rtr and rtl may not be combined with cla, cll, oas, cma, or cml.

INPUT/OUTPUT TRANSFER (IOT) INSTRUCTION

Input/Output Transfer (iot) Instructions are used to control external devices, sense their status and transfer information between them and the central processor. The bit assignments of the iot class are shown in Figure 4-2. Bits 0-3 carry the iot instruction code (70) . Bits 6-11 determine the external device selected; bits 4-5 and 12-13 are used to select a mode of operation or subdevice, and bits 15-17 initiate electrical pulses to the device for direct control of the information transfer or device operation. Descriptions of iot instructions are given along with the I/O equipment description in Chapter 7. The I/O Interface Electrical Characteristics follow the device description.

The Law Instruction

The law instruction (code 760000) is a special case of the operate class instructions. Bits 5-17 are not interpreted; instead, the instruction itself is placed in the AC. In this way an address-size number can be loaded into the AC without using an extra memory location. The law instruction is used to:

load memory addresses for use in indirect addressing,

load characters into the AC for use with I/O equipment,

initialize word count for a magnetic tape transfer,

preset the clock counter.

Only bits 5-17 of the law instruction are regarded by the above addresses, characters, and counts.

Example:

```
law     1234    /the octal number 761234 is entered into
                the AC
dac     15      /C(AC) = >C(location 15)
```

To initialize a memory location with a negative number, where the complete word (bits 0-17) is to be regarded, it is necessary to take the 1's complement of the number and then subtract away the octal code 760000. For example, if the desired count is 755, memory location Y is loaded with -755 as follows. The 1's complement of 000755 is 777022, which can be represented as the sum of 760000 and 1022. Since 760000 is the operation code for law, the resulting program sequence is used:

```
law     1022
dac     Y
```

INDIRECT ADDRESSING

In a memory reference instruction, if bit 4 is a 1, indirect addressing occurs when the instruction is executed. Bits 5-17 of such an instruction are interpreted as the address of the memory location containing not the operand but the address of the operand. Thus, access to the operand is deferred once to another location. The indirect instruction appears as

Example: add i 100 where, C(100) = 001357

where i signifies indirect addressing. The processor interprets the contents of register 100 as the address of the instruction operand and in the next memory cycle adds C(1357) to the AC. Access to an operand can be deferred in this manner only once during the execution of an instruction.

AUTO-INDEXING

Each 8192 word memory field of a PDP-7 computer system contains 8 auto-indexing memory locations in memory locations shown in the table below. When one of these locations is used as an indirect address, the location contents are automatically incremented by one, and the result is taken as the effective address of the instruction. The indexing is done with no added instruction time. Note that indexing of an auto-index location occurs only on an indirect reference; for direct addressing, the auto-index locations are identical to other memory locations.

Machine Size	Memory Fields	Relative Address	Physical locations of Auto-indexing locations
4K	0	10-17	10_8-17_8
8K	0	10-17	10_8-17_8
16K	0, 1	10-17	10_8-17_8, 10010-10017_8
24K	0, 1, 2	10-17	10_8-17_8, 10010-10017_8, 20010-20017_8
32K	0, 1, 2, 3	10-17	10_8-17_8, 10010_8-10017_8, 20010-20017_8, 30010-30017_8

Example

Assume four memory locations initially have the following contents:

Location	Contents
10	100
40	50
100	40
101	41

The following four instructions to load the accumulator illustrate by comparison the use of auto-indexing.

```
        lac         100     Places the number 40 into the AC
        lac     i   100     Places the number 50 into the AC
        lac         10      Places the number 100 into the AC
        lac     i   10      By auto-indexing, C(10) becomes 101;
                            then the number 41 is placed into the AC
```

Auto-indexing is also used to operate on each member of a block of numbers without the need for address arithmetic. The following three examples demonstrate how this is done:

Example 1 Add a column of numbers $Y = \sum_{i=1}^{N} X_i$

```
        Location    Contents
          10/       FIRST -1        /location of first word
        COUNT       -N + 1          /two's complement of number of additions
        cla                         /clear AC
LOOP,   add     i   10              /add into partial sum
        isz         COUNT           /test for completion
        jmp         LOOP            /more in table, go back
        CONTINUE                    /sum in AC
```

Example 2 $C_i = A_i + B_i$ for $i = 1, 2, \ldots N$

Note that three auto-indexing locations are used to simplify the addressing. In the basic machine, eight locations are available for use as auto-indexing registers.

```
            Location    Contents

              10/       L(A) -1          /the location of the A array -1
              11/       L(B) -1          /the location of the B array -1
              12/       L(C) -1          /the location of the C array -1

LOOP,       lac       i    10            /get addend
            add       i    11            /form sum
            dac       i    12            /store sum
            isz            COUNT         /test for completion
            jmp            LOOP          /more in table, go back
            CONTINUE                     /done, continue
```

Example 3 $C_j = C_j + K$ $j = 1, 2, \ldots N$

Modify a list of numbers by adding a constant to each of them. Note that the auto-indexing memory register contains an instruction rather than just an address. This is perfectly acceptable since, when not in extend mode, only the address bits are used in generating the effective address.

```
            Location    Contents

              10/       dac  FIRST -1   /deposit into first location in table -1
            COUNT/          -N +1       /two's complement of number of words in table
            CONST/           K          /the constant

LOOP,       lac       i    10          /pick up initial value from table
            add            CONST       /add the constant
            xct            10          /replace in table
            isz            COUNT       /test for completion
            jmp            LOOP        /more on table, go back
            CONTINUE
```

EXTENDED ARITHMETIC ELEMENT TYPE 177

The Extended Arithmetic Element EAE Type 177 is a standard option for the PDP-7 to facilitate high-speed multiplication, division, shifting and register manipulation. The EAE contains an 18-bit Multiplier Quotient Register (MQ), a 6-bit Step Counter Register, two Sign Registers and the EAE logic. The two panels of EAE logic are normally installed just below the Center Processor in bay one of the PDP-7 Computer. The contents of the MQ register are continually displayed on the operator's console just below the accumulator indicators.

The Extended Arithmetic Element hardware operates asyncronously to the basic computer cycle, permitting computations to be performed in the minimum possible time. Further, since the EAE instructions are microprogrammed, it is usually possible to simplify programming and shorten computation time by microcoding exactly the arithmetic operation desired.

The EAE instructions are broken up into two parts: The first part permits register manipulation as microprogrammed in the instruction while data is being fetched; the second part is the specified operation itself. Signed and unsigned multiplication would, for example, differ in the microprogrammed first part where the sign manipulation is done. A table showing the bit configuration for the EAE instructions is given in Figure 3-4.
The set-up phase of the instruction is broken up into four event times. Microprogramming for all but the set-up commands uses only the first three event times. The bits corresponding to the 4th event time then specify the step count of commands such as multiply, divide, and the shifts. The unassigned operation code should not be used as it is reserved for future EAE expansion.

Instruction times for operations performed by the EAE depend on the operation, the step count, and the data itself. Each command has a basic operation time to which is added function times depending on the operation.

Operation	Time
Shift/Normalize	1.6 μs plus 0.1μs/step.
Multiply	2.4 μs plus 0.1 μs/step plus 0.25 μs per one-bit in the multiplier.
Divide	2.4 μs plus 0.35 μs/step plus 0.2 μs per one-bit in the quotient.

Since the EAE expects to fine the multiplier or the divisor in the location following the multiply or divide instruction, a short subroutine is usually used to set-up the multiply or divide in the general case. These subroutines in both open and closed form are shown on the following pages. For multiplication or division by a constant, a subroutine is

not required and the maximum speed becomes the true multiplication or division time. Single length numbers (18-bits) are assumed to be of the form: high-order bit is the sign followed by 17 bits in 1's complement notation. Double length numbers (36-bits) use two registers, and are of the form: two high-order bits as signs, followed by 34 bits in 1's complement notation. Both sign bits must be the same. Unsigned numbers may be either 18 or 36 bits in length.

EAE Microprogramming

Arithmetic operations in the EAE assume that the numbers are unsigned 18 or 36-bit words. To properly manipulate sign numbers, the EAE instructions are microprogrammed to take complements and arrange the signs. In multiplication, the 18-bit number in the MQ register is multiplied by the number in the memory location following the instruction. The multiplier in the MQ register at the beginning of the operation can be either positive or negative. If it is negative, its sign must also appear in the EAE AC sign register. If this register is a one, the MQ register is complemented prior to the multiplication. Microprogramming makes it possible to set up the EAE AC sign register and to move the AC to the MQ while the data is being fetched.

When the multiplicand is taken from the memory location following the instruction, it must be a positive number with the original sign in the Link. The exclusive OR of the Link and the EAE sign register (the two registers containing the original signs of the numbers) form the sign of the product. If the sign of the product is a one (negative) the AC and MQ are complemented at the end of the operation. For the signed multiplication, the two most significant bits of the AC contain the sign of the product.

To produce a full 36-bit product or quotient, the step count of the multiply instruction should be 18 and for the divide instruction 19. However, for calculation not requiring 36-bit accuracy before rounding, the step count may be set lower to reduce the time required for the arithmetic operation.

For unsigned operations the Link must be zero.

A list of microprogrammed EAE Register Manipulation instructions is given on the following pages. Microprograms other than those common enough to warrant mnemonics are possible. An example is an instruction to place the contents of the AC into the MQ. The operation code for this instruction would be formed by using the EAE Setup op-code, code bit 5, to clear the MQ at event time 1 and bit 7 to OR the AC into the MQ at event time 2. An instruction of this type, however, is usually not necessary since the contents of the AC are automatically transferred to the MQ prior to multiplication by the microprogrammed mul or muls instruction.

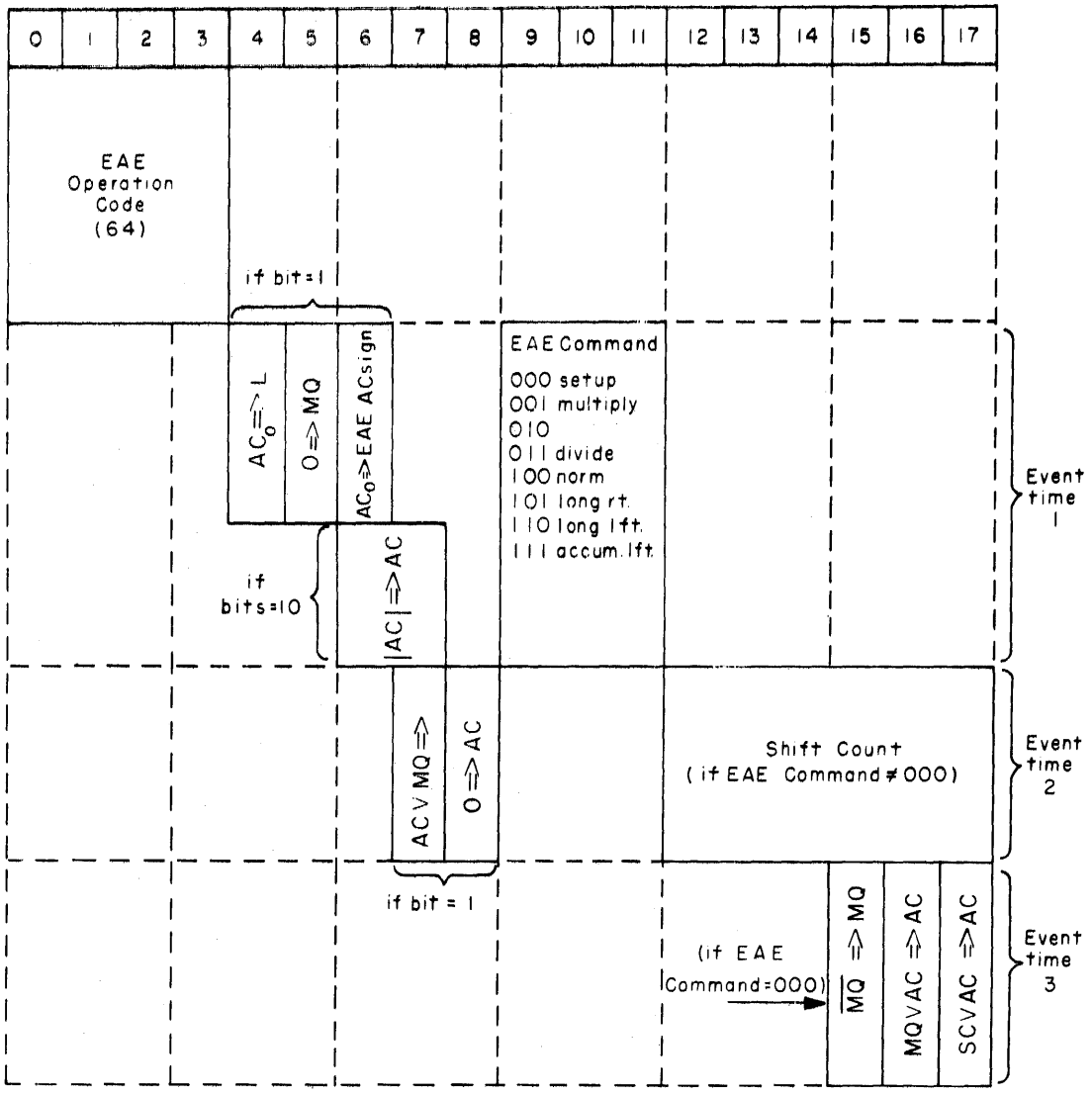


Figure 3-4 EAE Instruction Bit Assignment

EAE BIT ASSIGNMENTS AND OPERATIONS

(Refer to Figure 3-4)

BIT POSITIONS	BITS	FUNCTION
0, 1, 2, 3	1101	EAE operation code.
4	1	Place the AC sign in the Link. Used for signed operations.
5	1	Clear the MQ.

EAE BIT ASSIGNMENTS AND OPERATIONS (continued)

BIT POSITIONS	BITS	FUNCTION
6	1	Read the AC sign into the EAE AC Sign Register prior to carrying out a stepped operation. Used for the signed operations multiply and divide.
6, 7	10	Take the absolute value of the AC. Takes place after the AC sign is read into the EAE AC sign.
7	1	Inclusive OR the AC with the MQ and read into MQ. (If bit 5 is a 1, this reads the AC into the MQ).
8	1	Clear the AC.
9, 10, 11	000	Setup: Specifies no stepped EAE operation, and enables the use of bits 15, 16, and 17. It is used as a preliminary to multiplying, dividing, and shifting signed numbers. Execution time is one cycle.
9, 10, 11	001	Multiply: causes the number in the MQ to be multiplied by the number in the memory location following this instruction. If the EAE AC Sign Register is 1, the MQ will be complemented prior to multiplication. The exclusive OR of the EAE AC sign and the Link will be placed in the EAE Sign Register (the sign of product and quotient). The product is left in the AC and MQ, with the lowest order bit in MQ bit 17. The program continues at the location of this instruction plus two. At the completion of this instruction the Link is cleared and if the EAE sign was 1, the AC and MQ are complemented. The step count of this instruction should be 22 (octal) for a 36-bit multiplication, but can be varied to speed up the operation. The execution time is 4.2 to 8.7 μs, depending on number of 1 bits in the MQ.
9, 10, 11	010	(This is an unused operation code reserved for possible future expansion).
9, 10, 11	011	Divide: causes the 36-bit number in the AC and MQ to be divided by the 18-bit number in the register following the instruction. If the EAE AC sign is 1, the MQ is

BIT POSITIONS	BITS	FUNCTION
		complemented prior to starting the division. The magnitude of the AC is taken by microprogramming the instruction. The exclusive OR of the AC sign and the Link are placed in the EAE sign. The part of the dividend in the AC must be less than the divisor or overflow occurs. In that case the Link is set at the end of the divide;otherwise, the Link is cleared. At the completion of this instruction, if the EAE sign was 1, the MQ is complemented; and if the EAE AC sign was 1, the AC is complemented. Thus the remainder has the same sign as the dividend. The step count of this instruction is normally 23 (octal) but can be decreased for certain operations. The execution time is 3.5 μs in the case of divide overflow or from 9.0 - 12.6 μs otherwise.
9, 10, 11	101	Long right shift: causes the AC and MQ to be shifted right together as a 36-bit register the number of times specified in the step count of the instruction. On each step the Link fills AC bit-0, AC bit-17 fills MQ bit-0, and MQ bit-17 is lost. The link remains unchanged. The time is 0.1 n + 1.6 μs, where n is the step count.
9, 10, 11	110	Long left shift: causes the AC and MQ to be shifted left together the number of times specified in the step count of the instruction. On each step, MQ bit 17 is filled by the Link; the Link remains unchanged. MQ bit 0 fills AC bit 17 and AC bit 0 is lost. The time is 0.1 n + 1.6 μsec, where n is the shift count.
9, 10, 11	100	Normalize: causes the AC and MQ to be shifted left together until either the step count is exceeded or AC bit 0 $\neq$ AC bit 1. MQ bit 17 is filled by the Link, but the Link is not changed. The step count of this instruction would normally be 44 (octal). When the step counter is read into the AC, it contains the number of shifts minus the initial shift count as a 2's complement 6-bit number. The time is 0.1 n + 1.6 μs, where n is the number of steps in the shift counter or the number required to effect normalization, whichever is less.

EAE BIT ASSIGNMENTS AND OPERATIONS (continued)

BIT POSITIONS	BITS	FUNCTION
9, 10, 11	111	Accumulator left shift: causes the AC to be shifted left the number of times specified in the shift count. AC bit 17 is filled by the Link, but the Link is unchanged. The time is 0.1 n + 1.6 μs, where n is the step count.
12-17		Specify the step count except in the case of the setup command, which does not change the step counter.
15	1	On the setup command only, causes the MQ to be complemented.
16	1	On the setup command only, causes the MQ to be inclusive ORed with the AC and the result placed in AC. (If the AC has been cleared, this will place the MQ into the AC).
17	1	On the setup command only, causes the AC to be inclusive ORed with the SC and the results placed in AC bits 12-17. (If the AC has been cleared, this will place the SC into the AC).

Microprogramming High Speed Arithmetic

The following example uses the microprogramming capabilities of the Extended Arithmetic Element to maximize speed in arithmetic operations. Consider the case of normalizing a 12-bit word located in the AC (possibly converted from analog data) by multiplying it by a 5-bit constant. The speed increase is achieved by reducing the step count of the multiplication to the minimum necessary for this particular case. Note also that the multiply instruction will, through microprogramming, move the contents of the AC to the MQ prior to the multiplication operation. It is assumed that the desired result, a 17-bit number, will appear in the AC subsequent to the operation. It is also assumed that the Link is 0 prior to the instructions and that the 12 data bits represent an unsigned (positive) number.

The multiply instruction is formed with the following bit configuration. Bits 0-3 are EAE instruction operation code; bit 5 is set to clear the MQ prior to the multiply instructions; bit 7 is set to place the AC into the MQ prior to the multiply; bit 8 is set to clear the AC at the same time; bits 9 and 10 are 0; bit 11 indicates a multiply EAE instruction; bits 12 through 17 are used to set the step count of the instruction. The desired step count is 11 decimal (13 octal). The octal code for this special multiply operation is 653113. After

multiplying, it is necessary to long left shift the AC and MQ to restore the product to the AC only. The step count must be 11 for this instruction; hence, the octal code is 640613.

When programmed, the above multiplication routine occupies three sequential memory locations, the multiply instruction 653113, followed by the 5-bit constant, followed by long shift instruction 640613. The total execution time of this open subroutine is (average) 5.0 microseconds for the multiply, followed by 2.7 microseconds for the shift requiring a total of 7.7 microseconds to normalize the 12-bit analog data. The final result appears as a 17-bit number in the AC where it is then available for further operations.

```
norm,    muls - 7        /multiply with step count of 11 (13 octal)
         -               /storage of constant
         llss + 13       /signed left shift with step count of 11
                          (13 octal)
```

EAE INSTRUCTION LIST

MNEMONIC	OCTAL	OPERATION
eae	640000	Basic EAE command - No operation.
lrs	640500	Long right shift.
lrss	660500	Long right shift, signed (AC sign => Link).
lls	640600	Long left shift.
llss	660600	Long left shift, signed (AC sign => L).
als	640700	Accumulator left shift.
alss	660700	Accumulator left shift, signed (AC sign => L).
norm	640444	Normalize: max. shift is 44.
norms	660444	Normalize, signed (AC sign => L).
mul	653122	Multiply C(AC) x C(C(PC)) as 18-bit unsigned numbers, leave result in AC V MQ. The Link must be 0.
muls	657122	Multiply signed, C(AC) x C(C(PC)). The C(C(PC)) must be positive and its original sign must be in the Link. The signed result appears in AC V MQ right adjusted.

EAE INSTRUCTION LIST (continued)

MNEMONIC	OCTAL	OPERATION
div	640323	Divide C(AC and MQ) as a 36-bit unsigned number by C(C(PC)). Leave quotient in MQ and remainder in AC. The Link must be 0.
divs	644323	Divide C(AC and MQ) as a 1's complement signed number by the C(C(PC)). The C(C(PC)) must be positive and its original sign must be in Link. The signed quotient is in the MQ and the remainder, having the same sign as the dividend, will be in the AC.
idiv	653323	Integer divide. Divide C(AC) as an 18-bit unsigned integer by C(C(PC)). The MQ is ignored. The quotient will be in the MQ and the remainder in the AC. Link must be 0.
idivs	657323	Integer divide, signed. Same as idiv but C(AC) is 17-bit signed and the usual convention on C(C(PC)) and Link apply.
frdiv	650323	Fraction divide. Divide the 18-bit fraction in the AC by the 18-bit fraction in the C(C(PC)). The Link must be 0; the MQ is ignored. The quotient replaces the MQ and the remainder replaces the AC.
frdivs	654323	Fraction divide, signed. Same as frdiv, but C(AC) is 17-bit signed and the usual conventions of C(C(PC)) and Link apply.
lacq	641002	Replace the C(AC) with the C(MQ).
lacs	641001	Replace the C(AC) with the C(SC).
clq	650000	Clear MQ.
abs	644000	Place absolute value of AC in the AC.
gsm	664000	Get sign and magnitude, thus setting up divisor or multiplicand. Places AC sign in the Link and takes the absolute value of AC.
osc	640001	Inclusive OR the SC into the AC.
omq	640002	Inclusive OR AC with MQ and place results in AC.
cmq	640004	Complement the MQ.

Signed Multiply and Divide Closed Subroutines

```
/ signed divide subroutine                    37-42 μs
/ calling sequence
/       dividend in AC + MQ
/       jms divide
/       pickup other factor
divide,     0                                 entry to subroutine
            dac tem
            xct i divide
            gsm
            dac divl
            lac tem
            divs
divl,       0                                 / location of divisor
            isz divide
            jmp i divide


/ signed multiply subroutine                  25-31 μs
/ calling sequence:
/       one factor in AC
/       jms mpy
/       pickup other factor                   / lac xxx or lac  i xxx
mpy,        0                                 / entry to subroutine
            gsm                               / fix multiplicand magnitude
            dac .+3
            xct i mpy                         / get multiplier
            muls
            0                                 / location of multiplicand
            isz mpy                           / index return
            jmp i mpy
```

CHAPTER 4

INPUT/OUTPUT CONTROL AND INTERFACE

Functions

Information is transferred between the PDP-7 and peripheral equipment by the input/output control. This interface sets up the information path between computer and device, controls the transfer, and monitors the state of availability of each device. It also includes facilities for data, clock, and program interrupts. Figure 4-1 shows in schematic form the section of the input/output control. The input/output control is itself controlled by the programmed input/output transfer (iot) instructions. An iot instruction causes the input/output control to produce pulses. These pulses select an I/O device and initiate a data transfer. The single iot instruction is microprogrammed to control all input/output devices.

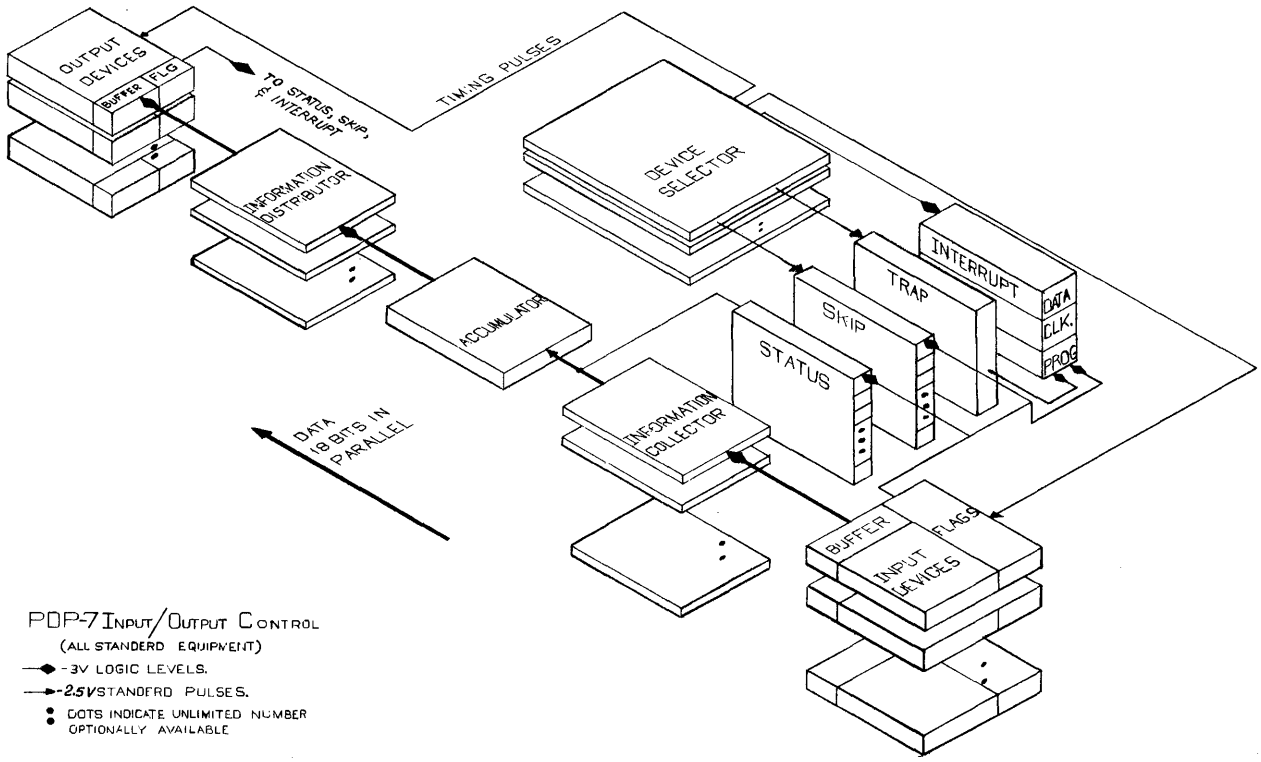


Figure 4-1 I/O Control Schematic

IOT INSTRUCTION

The input/output transfer (iot) instruction causes the input/output control to produce pulses which select I/O devices and transfer information. All iot instructions are octal code 70 with a bit assignment as shown in Figure 4-2.

Mnemonic	Instruction Code	Operation
iot	700000	input/output transfer

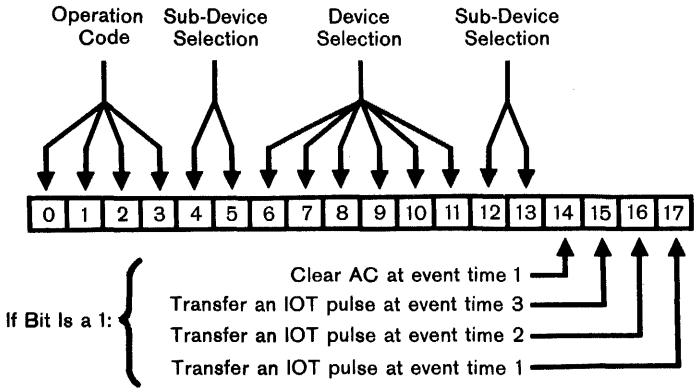


Figure 4-2 Bit Assignment for Input-Output Transfer Instruction (iot)

Bits 0-3 signify the iot instruction. Bits 4-13 specify the external device and its mode. When bit 14 is a 1, the accumulator will be cleared prior to the data transfer. Bits 15-17 select the pulses sent to the device during event times 1, 2, and 3. For ease of recognition, the iot pulses are coded according to bits 17, 16, and 15 as I/OP1, I/OP2, and I/OP4 respectively. I/OP1 is used to check the status of a device. I/OP2 and I/OP4 are initiated by the device selector to cause a transfer of information to and from the information collector and the information distributor.

PROGRAM FLAGS

The status of each I/O device is indicated to central processor by flag bits. A program reads the flag bits of a device and initiates appropriate action. In this way, input/output transfers and program operation are easily coordinated. Flags are connected to the program interrupt control, status bits, and the input/output skip facility.

A flag may indicate one of several things depending upon the location of its connection.

1. Connected to the program interrupt, a flag indicates that:

    a. An output transfer has been completed and the device buffer is now available for refilling.

b. An input buffer contains information for transfer into the computer.

c. A device operating asynchronously has information for input or requires information for output.

2. Connected to the input/output skip facility, a flag may indicate:

a. Skip the next instruction if the device buffer if full.

b. Skip the next instruction if an output operation has been completed.

3. Connected to the status register, a flag may indicate the:

a. Occurrence of an error

b. Direction of data transfer

c. Direction device is operating, forward, reverse

d. Mode of operation in a device

e. Subdevice connected to a central device

f. Busy or idle condition of a device.

DEVICE SELECTOR (DS)

The device selector selects an input/output device or subdevice according to the address code of the device in memory buffer bits 4-13 of the iot instruction. It then generates an I/OP pulse at event 1 if memory buffer bit 17 is a 1, event time 2 if memory buffer bit 16 is a 1, and at event time 3 if memory buffer bit 15 is a 1. The I/O event times differ from those of the microporgrammed operate group event times. A complete table of the I/OP pulses and corresponding times is given below and in Figure 4-4.

Event Time	Computer Cycle Time	Instruction Bit	I/OP Number
1	5	17	1
2	7	16	2
3	1 (next cycle)	15	4

Upon receipt of an iot instruction, the device selector determines which device has been selected, and then performs one or all of the following functions:

1. I/OP 1 is used to sense the state of the flag or flags associated with a device.

2. I/OP 2 is used to clear the flag or flags associated with a device and to read the contents of the device buffer into the IC.

3. I/OP 4 is used to transfer data from the accumulator through the information distributor into the buffer of an output device or to initiate operations within a peripheral device (ex. a line of perforated tape is read into the tape buffer or a card is moved to a reading or punching station).

The specific function or functions an I/OP performs are selectable and depend on the device and its timing requirements. A device may use any number of combinations of the three pulses. Devices requiring more than three pulses may use multiple device codes. For extremely expanded mode selection, a device may sense the state of the accumulator bits loaded prior to the iot instruction.

The 6-bit device selection numbers, memory buffer bits 6-11, are decoded by a diode decoder module B171. (See Figure 4-3.) The 6-bit code, therefore, produces an assertion level for the selected device. This level, in turn, controls I/O pulses through the device selector gates. The device selector amplifiers transmit pulses to the selected device according to bits 15, 16, and 17 of the iot instruction. These pulses can be of various types, depending on the type of the pulse amplifier used. Two different pulse amplifiers are available and produce the following range of (ground reference) pulses:

1. 2.5 volts, positive or negative pulses at 70-nanosecond intervals

2. 2.5 volts, positive or negative pulses at 400-nanosecond intervals

The standard device selector contains selector modules for the standard devices and has provisions for up to 6 additional decoders, gates, and amplifiers. When additional peripheral I/O devices are added to the PDP-7, a device code is easily established in the device selector by clipping out the diode of the unasserted level in the B171 module. Figure 4-3 shows the B171 with the clipping point marked with a ⊗.

Slow Cycle

For input/output equipment requiring a timing pulse chain slower than the normal I/O pulse cycle, a second timing chain may be requested by a signal from the device selector (see Figure 4-4). The slow cycle permits equipment with slow logic to be easily interfaced to a PDP-7 system.

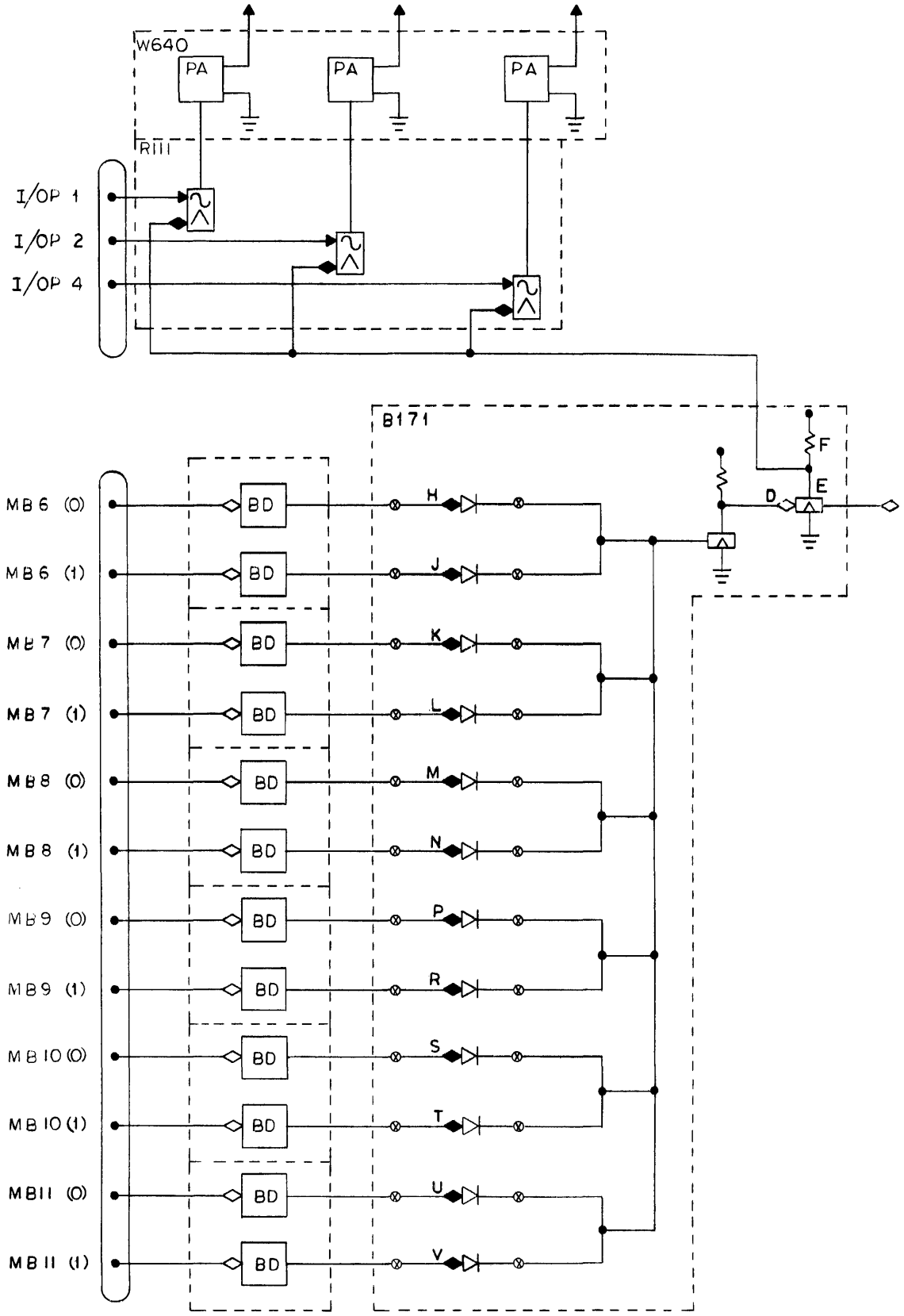


Figure 4-3 Device Selector Decoder

The slow cycle timing chain is preset by DEC to give optimal operation for the slowest piece of I/O equipment in the computer system. Only one slow cycle time can be requested by the DS. The PDP-7 enters the slow cycle each time an iot instruction is given to transfer data to or from a slow device.

INFORMATION COLLECTOR (IC)

The information collector is a seven-channel gated mixer which controls the transfer of 18-bit words from external devices to the accumulator. Pulses from the DS control the IC gates according to the device specified by the iot instruction. Because the accumulator must be cleared before a word is transferred through the IC to the AC, the iot instructions are usually microcoded to clear the accumulator (bit 14 is a 1) at the same time the external device is activated.

In the standard PDP-7, seven channels of IC are used. The paper tape reader and I/O status bits each occupy one 18-bit IC channel. The teleprinter occupies eight bits of a third channel. The remaining four and one-half channels are available for connection to any peripheral and optional input equipment. Each PDP-7 input option connects directly into one channel of the IC (ex. Extended Arithmetic Element Type 177, A-D Converter Type 138, DECtape Control Unit Type 550).

For operation of more than seven input devices, the IC is easily expandable in blocks of seven channels to accommodate any number of channels.

The modules used in the IC are the seven-channel R141 gates. The R141 accepts standard levels of 0 and -3 volts or standard 70-nanosecond or wider pulses. The input load is 0.5 ma per grounded input.

Bits transferred to the AC correspond to the incoming polarities:

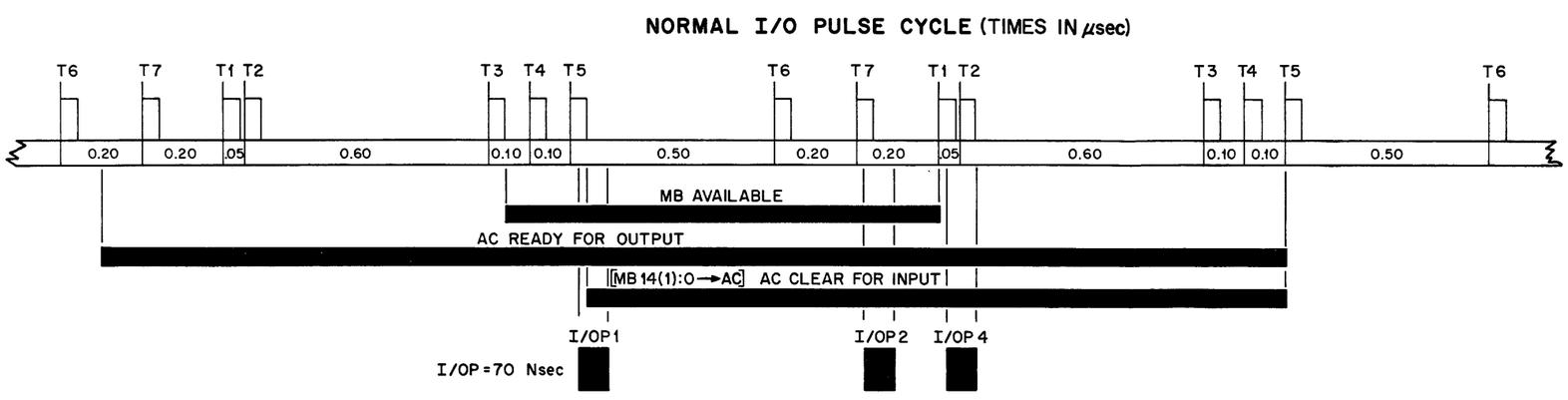
0 volts	0 transmitted to AC
-3 volts	1 transmitted to AC

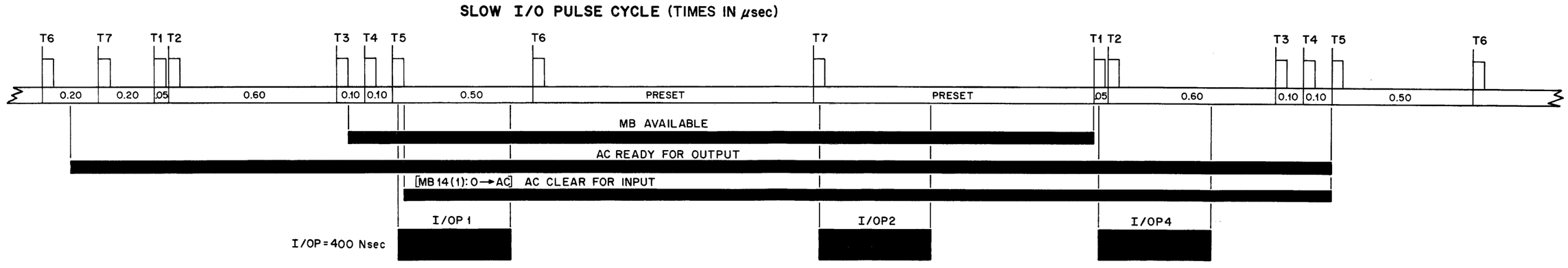
INFORMATION DISTRIBUTOR (ID)

The information distributor is an output bus system through which information is transferred from the accumulator to external devices. Eighteen line drivers buffer and drive the accumulator output through the external device connection cables. Other drivers and cable slots are used to transfer memory buffer and device control bits. Six 18-bit ID channels are standard on the PDP-7. The paper tape punch and teleprinter use two of the six channels. A third channel is used for the expanded ID connection.

Other external devices are easily connected to the information distributor. Each device receives pulses from the device selector to gate in bits from the bus.

The ID can be expanded to any number of output channels.





INPUT/OUTPUT PULSE CYCLES

NORMAL
- TIMING PULSE — STD. DEC PULSE, 40 Ns
- I/O PULSE — STD. DEC PULSE, 70 Ns
- CYCLE — 1.75 μsec

SLOW
- TIMING PULSE — STD. DEC PULSE, 40 Ns
- I/O PULSE — STD. DEC PULSE, 400 Ns
- CYCLE — PRESET BY DEC

Figure 4-4 I/O Pulse Cycle Diagram

The signal polarities presented to the output device by the ID are:

-3 volts	AC bit contains a 0
0 volts	AC bit contains a 1

INPUT/OUTPUT STATUS

The status of each I/O device, as indicated by its flags, may be read into assigned bits of the AC. Figure 4-5 shows the standard assignment for the commonly used devices. An X indicates that the flag is connected to the program interrupt control. The presence of a flag is reflected by a 1 in the corresponding AC bit.

The status of 18 flags may be read into the AC at one time using the following iot instructions.

iors	700314	Input/output read status. The contents of given flags replace the contents of the assigned AC bit.

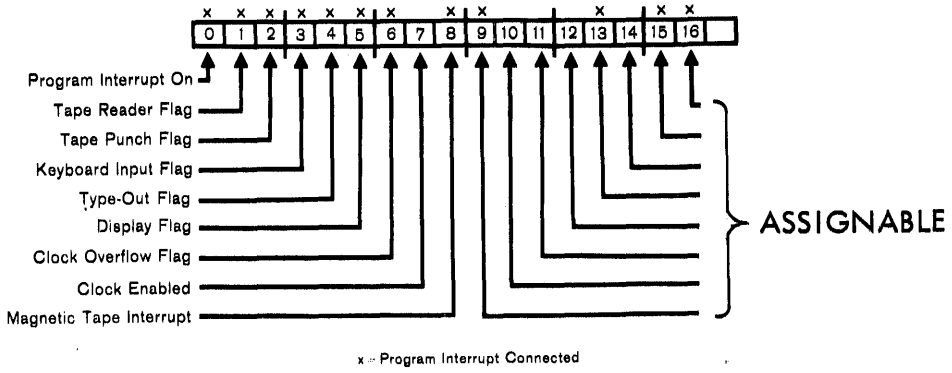


Figure 4-5 Input-Output Status Instruction - Bit Assignment

INPUT/OUTPUT SKIP FACILITY (IOS)

The input/output skip facility enables the program to branch according to the status of an external device. The IOS has fourteen flag inputs and is expandable to any number, five of which are used by the basic computer equipment. When an input/output skip instruction is executed, the DS sends iot pulses to the selected device input. If the flag connected to that input is set to 0, the next instruction in the program sequence is executed. If the flag status is 1, the next instruction is skipped. An I/O pulse for a skip must occur at event time 1.

The I/O skip facility is expandable through the addition of R141 modules, each of which contains seven additional skip inputs. A -3 volts indicates the presence of a flag.

Commonly used skip instructions are:

clsf	700001	Skip if clock has overflowed.
rsf	700101	Skip if paper tape reader buffer has a character.
psf	700201	Skip if paper tape punch is ready.
ksf	700301	Skip if teleprinter keyboard buffer has a character.
tsf	700401	Skip if teleprinter is ready to output.
dsf	700501	Skip on display flag (light pen).
cpsf	706401	Skip if card punch is ready.
lpsf	706501	Skip if line printer is ready.
lssf	706601	Skip if line printer spacing flag is a 1.
crsf	706701	Skip if card reader buffer has a character.

INPUT/OUTPUT TRAP

The PDP-7 I/O trap is designed to simplify programming of sophisticated input/output routines and to provide the basic hardware necessary for a time-shared or multi-user system. The effect of the trap is to insert a program break in place of the iot instruction. Two other conditions are also trapped, an xct instruction whose subject instruction is also an xct and the hlt portion of an operate class instruction.

The trap provides the PDP-7 with the basic hardware necessary to use the PDP-7 in a time-shared mode. With the use of the extend and trap modes, multi-user installations with full memory bank protection are possible. A program operating on one or more independent 8K (or smaller) memory banks can be protected from accidental disturbance by a program operating in other memory banks. All I/O operations can be monitored to check for use of restricted I/O devices or restricted memory locations. In this way, the PDP-7 can be used for real-time process control and simultaneously be available to share time with other programs in other memory banks without the threat of program interference.

The trap mode is enabled by the iton instruction (700062) with the console trap switch on. The trap mode is disabled by any program break. The iton (700062) also turns on the program interrupt through a microcoding of the ion instruction (700042). Since the I/O trap may not be disabled by a program without causing a program break, control over input/output rests entirely with the I/O interrupt routines. Other uses of the program interrupt and extend mode are controlled by the trap, for the extend status may not be changed and the interrupt mode may not be disabled by a program running in the trap mode.

The trap initiates a sequence of events depending on the trapped instruction.

iot	An iot instruction is trapped.
xct	An xct of an xct instruction is trapped.
hlt	A microprogrammed hlt of an operate class (740040) instruction is trapped.

A program break in place of the trapped instruction increments the program counter and stores its contents in location 0, bits 3 to 17, stores the link in bit 0, stores the extend status in bit 1, and stores the status of the trap mode in bit 2 (in this case 1). Control then transfers to location 2. The extend mode is enabled and the program interruptis turned off. The next instructions are taken from the appropriate I/O routine.

PROGRAM INTERRUPT CONTROL (PIC)

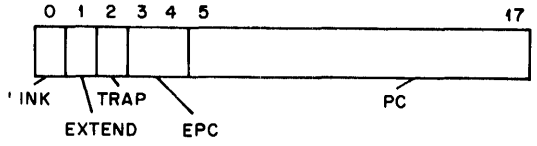
The program interrupt control increases the efficiency of input/output operations by freeing a program from the necessity of constantly monitoring program flags. When the PIC is enabled and a peripheral device becomes available, the PIC automatically interrupts the program sequence and causes a program break to occur. A subprogram beginning at the break location may then sense the program flags to determine which of the devices caused the interrupt. The device is then serviced and control returns to the main program. Fourteen device flags connect to the basic PIC, and more flag connections can be easily added.

The PIC may be enabled or disabled by the program. When it is disabled, program interrupts do not occur, although device flags may be set. Interrupts for these devices occur when the PIC is re-enabled. When the computer is operating with interrupt-producing devices, the PIC is normally enabled.

The following iot instructions control the PIC:

iof	700002	Interrupt off. Disable the PIC
ion	700042	Interrupt on. Enable the PIC

Each of the input/output devices has associated with it a program flag which is set whenever the device has completed a transfer and is ready for another. When the interrupt is enabled and the device is ready, the setting of the device flag (connected to the PIC) causes a program interrupt. The main instruction sequence is halted, the program counter, link, extend mode, and trap mode status are stored in location 0 and control transfers to location 1. Thus, a jms 0 has effectively been executed. The interrupt is then disabled and the extend mode is turned off. The word stored in location 0 has the following format:



Example

When the program interrupt is used to free the central processor between data transfers on a slow I/O device, the PDP-7 can do arithmetic or other I/O transfers while the slow device is in operation. The following sequence gives the limiting usable rate at which the PDP-7 could acknowledge repetitive program interrupts from the same device. Each data transfer is 18 bits.

Cycles	Location			
1	0	-		/contents of PC and Link
1	1	jmp	SERVICE	
2	SERVICE,	dac	TEMP	/save AC
1		iot		/transfer data from device buffer /to AC
3		dac i	10	/store data in memory list
2		isz	COUNT	
1		jmp	.+2	
-		jmp	END	
2		lac	TEMP	/reload AC
1		ion		/turn on interrupt
2		jmp i	0	/return to program
16				

The routine takes 16 machine cycles, or 28.0 microseconds per loop. When operating with a slow I/O device, the PDP-7 can perform other computations or other input/output operations in between program interrupts.

If the paper tape reader (300 cps), paper tape punch (63 cps) and teleprinter (10 cps) were all operating at full speed simultaneously through the PIC, the per cent computer time taken for I/O servicing is roughly

$$\%\text{I/O time} = \text{device rates (cps)} \times \text{service time } (\mu\text{s/interrupt}) \times \frac{100}{10^6}$$

In this case,

$$\%\ \text{I/O time} = (300+63+10) \times (28) \times \frac{100}{10^6}$$

or the time required to service the paper tape reader, punch, and teleprinter operating simultaneously in roughly less than 1.5% of the computer time.

The routine beginning in location 1 is responsible for finding and servicing the device that caused the interrupt. When a program interrupt occurs, the PIC is automatically disabled since only single-level interrupting is provided. The interrupt routine can re-enable the interrupt mode at any time.

The status of the PIC is displayed on the operator console by the indicator marked PIE, program interrupt enabled.

AUTOMATIC PRIORITY INTERRUPT TYPE 172

The (optional) Automatic Priority Interrupt Type 172 increases the capability of the PDP-7 to handle transfers of information to and from input/output devices. The 172 identifies an interrupting device directly without the need for flag searching. Multi-level interrupts are permissible where a device of higher priority supersedes an interrupt already in process. These functions increase the speed of the input/output system and simplify the programming. In this way more and higher-speed devices can be serviced efficiently.

The Type 172 contains 16 automatic interrupt channels arranged in a priority chain so that channel 0 has the highest priority and channel 17_8 has the lowest priority. Each channel is assigned a unique, fixed, memory location in the range of 40_8 through 57_8 starting with channel 0. When establishing priority, each I/O device is assigned a unique interrupt channel. The priority chain guarantees that if two or more I/O devices request an interrupt concurrently, the system grants the interrupt to the device with the highest priority. The other interrupt requests will be serviced afterward in priority order.

The automatic priority interrupt is assigned a priority just below that of the data interrupt, a position held by the real time clock. The 172 replaces the real time clock. The priority interrupt system may operate in either of two modes, the multi-instruction subroutine mode or the single instruction subroutine mode. The mode is determined by the instruction in the memory location assigned to the channel.

The Multi-Instruction Subroutine Mode

This mode is generally used to service an I/O device that requires control information from the PDP-7. Such devices are alarms, slow electromechanical devices, teleprinters, punches, etc. Each device requires a servicing subroutine that includes instructions to manipulate data and give further instructions, such as continue, halt, etc., to the interrupting device.

An interrupt request from a device is granted if the following conditions are met:

The 172 is in the enabled condition (by program control).
There is no data interrupt request present.
The requesting channel is in the enabled condition (by program control).
There is no interrupt in progress on a channel of higher priority.
There is no interrupt in progress on the requesting channel.

When an interrupt is granted, the contents of the channel memory location are transferred to the MB and executed. If the instruction executed is jms Y, the system operates in the multi-instruction subroutine mode. The contents of the program counter and the condition

of the link are stored in location Y, and the device-servicing subroutine starts in Y + 1. (Note that it is often useful to store the contents of the AC before servicing the device and to restore the AC prior to exiting from the servicing routine.)

The interrupt flag is normally lowered by the I72, but can be lowered by an iot instruction if desired. Program control now rests with the servicing routine.

A return to the main program is accomplished by a restore the AC and link, a debreak int and a jump indirect to location Y, where the contents of the PC prior to interrupt are stored. The debreaking iot requires no channel designator, since the interrupt priority chain automatically releases the correct channel and returns it to the receptive state. This iot normally inhibits all other interrupts for one memory cycle to insure that the jump indirect Y is executed immediately.

The following program example illustrates the action that takes place during the multi-instruction subroutine mode. Assume an interrupt on channel 3.

Memory Location	Instruction	Function
1000	add 2650	Instruction being executed when interrupt request occurs.
0043	jms 3000	Instruction executed as a result of interrupt on channel 3. The jms determines multi-instruction mode.
3000	-	The link, condition of the extend mode, and the PC are stored in location 3000.
3001	dac 3050	First instruction of servicing routines stores AC.
3002 3003 3004	-	Instructions servicing the interrupting in-out device.
3005 3006		
3007	lac 3050	Restores AC for main program.
3010	dbr	Debreaking iot releases channel.
3011	jmp i 3000	Return to main program sequence.
1001	-	Next instruction executed from here unless another priority interrupt is waiting.

The Single Instruction Subroutine Mode

In some instances, it is desirable for the PDP-7 to receive information from an external device but not to send control information to the device. Such an application would be the counting of real time clock pulses to determine elapsed time. The single instruction subroutine mode simplifies programming a counter.

An interrupt request is subject to the same conditions as in the multi-instruction mode, and the appropriate memory location is addressed as before. Then the single instruction subroutine mode is entered if the channel memory location does not contain a jms instruction. Normally the instruction is isz. In any case, since the single-instruction constitutes the entire subroutine, the interrupt system automatically lowers the interrupt flag, debreaks the interrupting channel, and returns the channel to the receptive condition.

If the isz instruction is used, the 172 acknowledges only the indexing operation and neglects the skip to avoid changing the contents of the program counter. If an overflow results from the indexing a flag is set. This flag can be entered in another channel of the interrupt system to cause a further program interrupt.

The following program coding illustrates operation in the single instruction subroutine mode. Assume an interrupt on channel 6.

Memory Location	Instruction	Operation
1200	dac 1600	Operation being executed when interrupt occurs.
0046	isz 3200	Instruction executed as a result of break on Channel 6. If overflow, flag is set, PC not changed.
1201	lac 1620	Next instruction in sequence of main program.

Priority Interrupt Instructions

The following instructions are added to the PDP-7 with the installation of the 172. Some instructions, for example cac and asc, can be microprogrammed.

Octal Code	Mnemonic	Operation
cac	705501	Clear all channels. Turn off all channels.
asc	705502	Enable selected channel(s). AC bits 2-17 are used to select the channel(s).

dsc	705604	Disable selected channel(s). AC bits 2-17 are used to select the channel(s).
epi	700004	Enable automatic priority interrupt system. Same as real time clock clon.
dpi	700044	Disable automatic priority interrupt system. Same as real time clock clof.
isc	705504	Initiate break on selected channel (for maintenance purposes). AC bits 2-17 are used to select the channel.
dbr	705601	Debreak. Returns highest priority channel to receptive state. Used to exit from multi-instruction subroutine mode.

AC bits 0 and 1 are available for expansion of the basic automatic priority interrupt system to 4 groups of 16 channels.

REAL TIME CLOCK

The clock produces a pulse every 1/60 second (16.7 milliseconds). When the clock is enabled, every clock pulse causes a clock interrupt. The clock interrupt is similar to a data interrupt in that the contents of an active register are not changed. This interrupt has priority over a program interrupt but is of lower priority than a data interrupt. During the interrupt the contents of memory location 7 are incremented by 1. If the contents of location 7 overflow, the clock flag is set to 1. The clock flag is connected to the program interrupt system and may cause a program interrupt.

Three iot instructions are associated with the clock:

clsf	700001	Skip the next instruction if the clock flag is set to 1.
clof	700004	Clear the clock flag and disable the clock.
clon	700044	Clear the clock flag and enable the clock.

Clock frequencies other than 60 cps can be (optionally) selected for use with the clock interrupt. Depressing the START key on the operator console clears the clock flag and disables the clock.

Since the clock register is in core memory location 7, its contents may be loaded or deposited by a program. A standard technique for using the clock is to preset the contents of location 7 with the complement of the desired count and then to enable the program interrupt and the clock. An interrupt will occur at the end of the desired time. To cause an interrupt at the end of 1 second, the following routine can be used:

```
0/
1/          jmp end-of-time
clock       lam* -60        /load -60 into accumulator (same as
                            law 17720).
            dac 7           /preset clock to -60.
            clon            /turn on clock.
            ion             /turn on interrupt.
                            /continue with 1 second worth of program.
```

DATA INTERRUPT CHANNEL

The data interrupt channel allows a high-speed input/output device such as a magnetic tape unit or drum, to operate independently once the information transfer has been initiated. The data address (15 bits) is transmitted directly to the memory address register. The data itself is read directly into the MB, bypassing the AC entirely. Since the data interrupt has priority over all other interrupts, a request will be granted at the completion of the current instruction. When a data interrupt occurs, the program is delayed for one cycle while the transfer is made; the program then resumes. A transfer rate of 570,000 18-bit words/second (1,710,000 6-bit characters/second) is possible.

The external device must supply 15 address lines, 18 data lines, a request line, and a transfer in (out) line. All lines are -3 volts for assertion, ground for 0. To accommodate slow I/Odevices, the external device may request the computer to slow its cycle for the duration of the transfer (see DS, Slow Cycle).

The optional Type 173 Data Interrupt Multiplexer increases the data interrupt facility to 4 channels arranged in a priority chain. Thus, several high-speed devices such as a Type 57A Tape Control, a type 24 Drum, etc., may operate simultaneously at a maximum combined transfer rate of 570 KC words/second.

The optional Type 174 Data Control controls and buffers high speed transfer between the computer and external devices which do not have the necessary control facilities. The Type 57A Tape Control and Type 24 Drum do not require this data control. Maximum transfer rate is 570 KC words/second.

*lam is a pseudo-instruction to the assembler which generates the equivalent machine instruction using a law instruction.

DATA INTERRUPT MULTIPLEXER CONTROL TYPE 173

The Data Interrupt Multiplexer Type 173 permits four high-speed input/output devices to operate into the standard PDP-7 data interrupt channel. The 173 operates at a combined transfer rate of 570,000 18-bit words per second and is designed for use with high-speed equipment such as magnetic tape systems, drum systems, and multiple high-speed analog-to-digital converters.

The 173 multiplexer operates through the standard data interrupt facilities of the PDP-7 computer. A signal to the data interrupt control causes the operating program to halt or pause for one cycle while the information is either deposited or removed from core memory. During this pause, there is no change in the status of the arithmetic registers. The operating program automatically resumes after the multiplexer access.

When an external device is addressed or addresses core memory through the Type 173 Multiplexer and the data interrupt, the following events occur:

1. The multiplexer switches to the device.

2. A -3-volt level from the 173 control to the device indicates that the central processor is ready for the data break.

3. The 1.75-microsecond data break cycle begins.

   Time 1 Computer samples the 15 address lines.

   Time 2 Data is transferred in or out of core memory through the 173 multiplexer.

   Time 3 Data is transferred in or out of core memory through the 173 multiplexer.

4. End of data break is indicated by -3-volt level on the multiplexer select line.

Line connections to the multiplexer control must supply the data, an address, the direction of transfer, and the data request signal:

Data	18 data lines, -3 volts asserts a 1 bit.
Address	15 address lines, -3 volts asserts a 1 bit.
Direction	One line, -3 volts indicates data transfer into the multiplexer; 0 volts indicates data transfer out of the multiplexer.
Data Request	One line, -3 volts for data request.

Address Lines	(15)	−3 volt level for assertion
Request for Transfer	(1)	−3 volt level for assertion
Direction of Transfer	(1)	−3 volt level for IN (from device to computer) GND for OUT (from computer to device)
Data Bits — IN	(18)	−3 volt level for assertion

The following lines are sent from the data multiplexer control to the device:

MPXB Sel	1 per device	−3 volt level when the device is selected
DATA∧ B	(1)	−3 volt level when a data request has been granted and the computer is in a break
T1		−3 volt pulse occurs at computer cycle time one, address accept time. (70 NS pulse)
T3		−3 volt pulse occurs at computer cycle time three, data accept or transfer out time. (70 NS pulse)

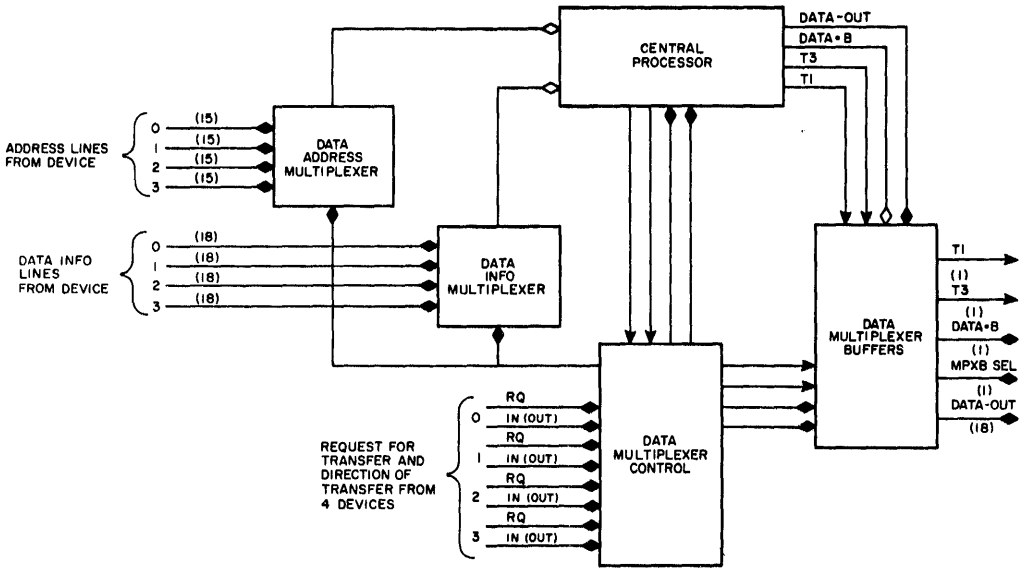


Figure 4-6 Data Interrupt Multiplexer Signal Diagram

DATA CONTROL TYPE 174

The Data Control Type 174 controls and buffers the transfer of data blocks between the PDP-7 and up to three external devices. Block transfers are made from consecutive memory locations to one device at a time. The data control counts the number of data words transferred, buffers either incoming or outgoing information until the transfer is complete and signals the completion of a transfer. Maximum data transfer rate is 1.75 microseconds per 18-bit word, or 570,000 18-bit words per second.

Data is transferred between the two 18-bit buffers of the data control and the PDP-7 through the memory buffer register. The data control includes four hardware registers: two data buffer registers, one word count register, and one initial location register. The word counter contains the complement of the number of words to be transferred in a block and is indexed on each transfer. The location register contains the address of the next data word to be transferred and is indexed on each transfer.

The PDP-7 IOT instructions initiate a transfer of control information to the Type 174 Data Control registers. The initial memory address specifications, direction of transfer and word count are all sent from the accumulator through the information distributor to the 174 control. Once it is initiated, a transfer of block data is interleaved with the running program and requires no additional instructions.

Completion of a block transfer can be indicated through the program interrupt channel or through the automatic priority interrupt channel. The data control may be operated directly through the data interrupt channel on the PDP-7 or indirectly through the Data Interrupt Multiplexer Type 173. Up to four 174 Data controls can draw information through the data interrupt multiplexer.

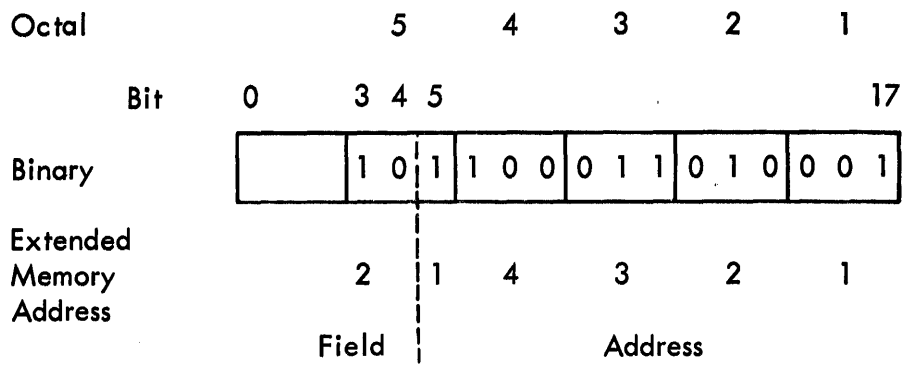
CHAPTER 5

MEMORY EXTENSION CONTROL TYPE 148

The Type 148 Memory Extension Control allows the expansion of the PDP-7 memory from 8,192 to 32,768 words in increments of either 4,096 or 8,192 words, using the Type 149 Memory Modules. The Type 148 includes an Extended Program Counter, an Extended Memory Address Register, and an Extend Mode Control. Locations outside the current 8,192 word field are accessed by indirect addressing while in the Extend Mode. In this mode, bits 3-17 in the effective address of an indirectly addressed instruction contain the Memory Field Number (bits 3 and 4) and the Memory Address (bits 5-17). If not in the Extend Mode, bits 3 and 4 of the effective address are ignored and the field number is taken from the Extended Program Counter. Thus, when not in the Extend Mode, the instruction and data must be in the same 8,192 word field. In the following example, the program starts at location 66666 (Memory Field 3):

```
66666/      lac i 2345
62345/      54321
```

The effective address, 54321, is interpreted as follows:



If not in the Extend Mode, bits 3 and 4 are ignored and the memory address is interpreted as 14321 in the current memory field 3. The physical address is 74321.

The current memory field, from which instructions are executed, is stored in the Extended Program Counter (EPC). The EPC is changed by jumping (jmp i or jms i) while in the Extend Mode. The Program Counter (PC) will not increment across memory field boundaries, it counts from 00000_8 to 17777_8 and back to 00000_8 of the current memory field. A load accumulator instruction, or other memory reference instructions with indirect addressing

(lac i), can access data from any memory field, however,it does not change the EPC. In the Extend Mode, cal goes to location 00020 of field 0. When not in the Extend Mode, it goes to location 20 in the current field. Program interrupts always reference field 0, location 0. Extend Mode is automatically cleared and the condition is stored in the bit position one (1) of location 0. The Extend Mode condition may be re-established at the end of an interrupt routine by the instruction emir. The following diagram illustrates the configuration of bits which are stored in location 0 on a program interrupt.

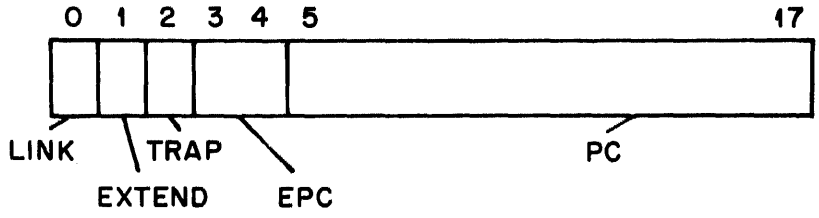


Figure 5-1 PIC Word Format

Each memory field which is added contains eight auto-index registers as does the basic memory. The locations for auto-index registers with 32K of memory are:

00010	00017
20010	20017
40010	40017
60010	60017

Four instructions are added with the Type 148 Memory Extension Control:

sem	707701	Skip if in Extend Mode
eem	707702	Enter Extend Mode
lem	707704	Leave Extend Mode
emir	707742	Extend Mode Interrupt Restore

The following sequence will re-establish the condition of the Extend Mode upon completion of an interrupt servicing routine:

```
emir        /Extend Mode Interrupt Restore
ion         /turn program on
jmp i 0     /return
```

The actual effect of the emir instruction is to turn the Extend Mode on then off again if the effective address of the jmp i 0 instruction has bit 1 equal to 0, (Extend Mode was off when the interrupt routine was entered). The emir instruction can be given at any time prior to leaving the interrupt routine and indirect addressing may be used without effect on the Extend Mode. Only jmp i will restore the Extend Mode Condition.

Existing programs lacking Extend Mode instructions can operate within any memory field providing they do not use program interrupt. If interrupt is used, the following routine must be in field 0.

```
0/
1/      jmp     SIM
SIM,    dac     AC      /save AC
        lac     0       /pick up return
        and     MASK    /select field bits
        dac     ADDR    /set up new location
        lac     0       /pick up return
        emir            /Extend Mode interrupt restore
        dac i   ADDR    /store in new location
        isz     ADDR    /set up jump
        lac     AC      /restore AC
        jmp i   ADDR    /return
ADDR,
MASK,           260000
AC,
```

Data interrupts must supply a 15-bit address. The condition of the Extend Mode is not changed.

CHAPTER 6

PROGRAMMING

The purpose of this chapter is to introduce the basic aids of programming and program preparation. The complete description of each program is given in its accompanying Program Library Document. Questions regarding programs and programming techniques should be addressed to the Manager of Applied Programming, Digital Equipment Corporation, Maynard, Mass.

Programming aids introduced:

Assembler
DDT (Digital Debugging Tape)
Editor
FORTRAN
BUS-PAK II

ASSEMBLER

The PDP-7 Assembler is a one-pass system which translates a symbolic source program into a form suitable for execution. The source program permits the user to express the operations he wishes the computer to perform in a form more legible to the programmer than the binary code in which the PDP-7 must receive its instructions. Instructions for the central processor and input/output standard options are included in the assembler.

By using this assembler, the programmer may employ mnemonic codes for the instructions and assign symbolic addresses in the program. For example, if the programmer uses the characters "lac," the assembler will transform this to the value 200000_8 as stored in memory. The assembly process consists of substituting the value of each symbol for the symbol itself and punching it out on the binary output tape.

During assembly, the assembler keeps a current address indicator which indicates the address of the register into which the next instruction or data word will be stored. For each word assembled, this address is increased by one. The initial address may be preset to allow assembly at any location. Normal assembly starts at location 22.

The assembler performs its action in one pass (i.e., the source language tape is processed only once to produce the binary object language tape). Certain functions which cannot be handled at assembly time must be handled by the loader when the program is loaded into memory.

A summary of the more important parts of the source language is listed below. For a complete list, refer to the PDP-7 Assembler Manual.

Source Language

1. Character Set

All of the lower-case characters of the alphabet are used along with numerals and certain punctuation characters. These punctuation characters and their meaning to the assembler are listed below.

Symbol		Meaning
	space	add syllables
+	plus	add syllables
-	minus	subtract syllables
&	logical and	combine syllables
!	logical or	combine syllables
↵	carriage ret.	terminate words
→|	tabulation	terminate words
,	comma	terminate words
=	equals	define a parameter
/	slash	comment, or address assignment
(	left paren.	initiate constant
)	right paren.	terminate constant (optional)
.	period	current address indicator

2. Syllables

(a) Number - any sequence of digits delimited by punctuation characters.

eg. 1
12
4374

(b) Symbols - any sequence of characters delimited by punctuation characters with the initial character alphabetic (a-z).

eg. a
a121b
larrys

(c) Current Address Indicator - the character "." (period) has the value of the current address.

(d) Constant - a number or syllable consisting of one of the following forms:

(alpha)
(alpha ↙
(alpha→|

Constants may consist of several syllables connected by syllabic operations as long as no more than one syllable or symbol is undefined.

3. Expressions

The value of an expression is computed by combining the component parts in the manner indicated by the connecting punctuation.

```
eg.          a
           a+3
        lac a-5
        szaVsnl
```

Note: The instructions "szl", "sna", and "spa" may be combined to form an expression; the instructions "snl", "sza" and "sma" may also be combined. However, instructions from one set may not be combined with instructions from the other, due to the use of bit 8.

4. Storage Words

Storage words are expressions delimited by tabs or carriage returns. They occupy one register in the program.

```
eg.     lac a
        jmp .+5
        lac (4)
        add 520
        lac (jmp b-6
```

5. Symbol definitions

(a) Parameter - may be assigned with the use of the equals sign (=).

```
eg.     a = 6
        exit = jmp i 20
```

(b) Address assignment

The use of a / (slash) if immediately preceded by an expression sets the current address equal to the value of that expression.

```
eg.     300/            lac (56
        begin -240+a/   lac (56
```

The expression must be defined at the time of assignment.

(c) Comma

If the expression to the left of a comma consists of a single, undefined symbol and that symbol is not from the permanent symbol list, the assembler will set the value of the symbol to the current address, thus defining that symbol.

```
eg.     begin,          lac load
          .               .
          .               .
          .               jmp begin
```

6 Variables

Any storage register which is reserved for data which may change during the program is referred to as a variable. To indicate a variable, it is only necessary to include the character $ anywhere within the first six characters of the variable name the first time it is specified.

7. Pseudo Instructions

Pseudo instructions command the assembler to take certain action during processing of the source language tape. They are transparent to the part of the assembler which processes syllables for output and are disregarded after performing their control function. The more important ones are described below.

(a) Radix Control

The programmer can indicate the radix which the assembler should use when interpreting digits.

Decimal - All numbers are interpreted as decimal numbers until the next occurrence of the pseudo instruction "octal".

Octal - All numbers are interpreted as octal numbers until the next occurrence of the pseudo instruction "decimal".

When the assembler is initially read into core, the mode is octal.

(b) Start

This pseudo instruction indicates the end of the symbolic source tape. It must be followed by a carriage return. After the binary tape is read, the AC lights indicate the last address used by the program. If Start is followed by a space and symbolic expression (inserted before the carriage return), the loader will jump to the address equivalent of the symbolic expression when the program has been read (load and go).

(c) Pause

Performs the same function as "start" except that the program halts on read in. If Pause is accompanied by a symbolic expression, the program may be started at the address indicated by that expression by depressing CONTINUE.

(d) Variables

All variables which have appeared in the program up to this point but have not had locations assigned to them will be stored sequentially starting at the address indicated by the current address counter. Then processing of the program continues

Source Language Tapes

A source language tape can be produced off line using any 8-bit ASCII code equipment. On-line source tapes can be prepared under program control with a greater flexibility for error correction and modification using the Editor program.

For a complete description of the Assembler and its operation, refer to the PDP-7 Assembler Manual, Digital 7-3-S.

DDT (DIGITAL DEBUGGING TAPE)

DDT is a debugging program for the PDP-7 computer. In both 4K and 8K computers, DDT occupies the highest 2000_8 registers of memory. Program modification and execution is from the Teletype keyboard and output is on the teleprinter or punched tape, as desired by the programmer. For example, to branch to a new location in the program it is only necessary to type the symbolic location name on the keyboard followed by the character single quote('). The same symbol followed by the character slash (/) causes the contents of that location to be typed. Working corrections can be punched out on the spot in the form of loadable patch tapes, eliminating the necessity of creating new symbolic tapes and reassembling each time an error is found.

Breakpoints - One of the DDT's most useful features is the breakpoint. A simplified way of thinking of a breakpoint is to think of hlt's being inserted in a program at critical points.

The breakpoint control characters are as follows:

" (double quote)	DDT inserts a breakpoint at the address specified before the ". DDT will remove the instruction at the break location and save it for future restoration. The instruction at the break location is only executed after the proceed is given. To proceed, execute (!).

! (exclamation) — After a break occurs, this character causes DDT to proceed with the user's program. This proceed will cause the instruction which was at the break location to be executed and control to return to the user's program. It is possible to test a loop and break before the last time around (ex. Nth time), by supplying a number before the (!). The break will then occur during the Nth cycle.

' (single quote) — Go to the location specified before the '. This character starts the program running and will run until it encounters the register which was specified as a breakpoint.

As an example of breakpoint use, consider the program section

```
                .
                .
                .
begin,          lac a
                add b
                dac c
                .
                .
                .
```

Suppose this program is giving a wrong answer and you want to find where the error is in the program. Break at "begin + 1"; start the program running at "begin". Suppose a contains 15 and b contains 20:

You type: BEGIN + 1"

You type: BEGIN '

DDT types: BEGIN + 1) 15

DDT types out the break location followed by a right paren, some spaces, and the current contents of the AC. At this point, the programmer is free to change registers or just examine them, change the contents of the AC, or use any of DDT's other features.

For a complete description of DDT, refer to the DDT Program Description, Digital-7-4-S.

EDITOR

The Editor program reads sections of the symbolic source tape into memory where it is available for examination and correction. Corrections are entered directly from the teleprinter keyboard. The corrected text can then be punched out on a new tape. Text may also be entered and punched for original tape preparation. Tape input and output may be either FIO-DEC or ASCII codes, and the Editor will convert from one code to the other.

The information to be edited is stored in a text buffer, which occupies all of memory not taken up by the Editor itself, and has a capacity for about 4,000 characters in a PDP-7 with 4096 words of memory, or about 16,000 characters in one with 8192 words.

Operating Modes

In order to distinguish between commands to itself and text to be entered into the buffer, the Editor operates in one of two modes. In command mode, typed input is interpreted as directions to the Editor to perform some operation. In text mode, all typed input is taken as text to be inserted in or appended to the contents of the text buffer. To help the user keep track of the mode, a visual indication is provided by the LINK light on the PDP-7 console. In command mode, this light is off; in text mode, it is on.

Listed below are five of the special functions which are part of the Editor.

Carriage Return (↵)

In both command and text modes, this is the signal for the Editor to process the information just typed. In command mode, the operation specified is to be performed. In text mode, it means that the preceding line of text is to be placed in the buffer.

Continuation ($ ↵)

In text mode this facilitates adding comments to successive lines or for end-of-line corrections. If a line of text is terminated by this pair instead of a single carriage return, the line will be entered as usual; then the line immediately following it will be printed up to but not including its carriage return. Thus, the new line is left open for additions for corrections.

Line Feed (↓)

This character has two meanings, depending on when it is used. If it is struck after some information has been typed, it causes that information to be deleted. Used thus in either mode, it has the effect of erasing mistakes. When it has

processed the line feed, the Editor responds with a carriage return. If, in command mode, line feed is the first character typed on a line, the next line of text (line .+1) will be printed.

Rub Out (ro)

This key has three distinct functions. Typing ro in command mode will cause the next line of text to be printed. The use of ro for this purpose is preferred to that of line feed, since it provides a neater printout.

Pressing ro in text mode will cause the last character of an incomplete line of text to be deleted from the input buffer. Continued striking of this key will cause successive characters. to be deleted one by one, working from the end of the line back to the beginning. In this way, a mistake can be corrected without having to retype the whole line.

Example: Instead of DAC PTEM, the following line was typed:

```
DAC CTE
```

To correct the line, ro is struck three times, erasing the last three letters in succession, E, T, and C. The correct text is then typed, and the resulting line appears on the Teleprinter as:

```
DAC CTEPTEM
```

It is stored in the text buffer, however, in correct form, as:

```
DAC PTEM
```

In text mode, the ro key has another function. Typed immediately after a carriage return, it signals the Editor to return to command mode. If one deletes all the characters in an incomplete line and then strikes ro one more time, the Editor will also return to command mode. No keyboard response is provided by the Editor; but when it enters command mode, the LINK light, which has been on while in text mode, goes out.

Colon (:)

When this symbol is typed in command mode, the Editor will print the decimal value of the argument that precedes it followed by a carriage return. It is frequently used for determining the number of lines of text in the buffer.

Example:

```
/: 57
```

or in determining the number of the current line:

```
.: 32
```

For a complete description of the Editor and its operation, refer to the PDP-7 Editor Manual, Digital 7-1-S.

FORTRAN

Based on the field proven FORTRAN used with the PDP-4, PDP-7 FORTRAN is written for two different hardware configurations. One is for perforated tape systems and the other is for a configuration which includes at least two logical tape units (either two magnetic tape units or one dual magnetic DECtape unit). Both FORTRAN systems require an 8K memory. Approximately 4000 (decimal) registers are available for stored program and data. The principal subsections of the FORTRAN system are:

Compiler
Fortran Assembler
Object Time System
Library

The compiler accepts input in the FORTRAN language and produces an output in an intermediate language acceptable to the assembler. The assembler accepts the compiler output and produces a binary relocatable version of the program and a binary version of the linking loader.

When the user is ready to execute a program, he loads the main program and any subprogram, followed by any built-in functions called from the library. Once the total program is in memory, he loads the object time system and executes the program. The object time system contains an interpreter for floating point arithmetic, an interpreter for format statements, routines such as fixed floating number conversions, and the I/O routines. The object time system must be in memory when a FORTRAN program is executed.

FORTRAN has the following characteristics:

FIXED POINT CONSTANTS 1-6 decimal digits absolute value $\leq 131,071$.

FLOATING POINT CONSTANTS 10 decimal digits precision. Exponent range from plus $2^{17} - 1$ to minus $2^{17} - 1$.

SUBSCRIPTS Any arithmetic expression representing an integer quantity: Variables in a subscript may themselves be subscripted to any depth. N dimensional arrays are permitted.

STATEMENTS Mixed expressions containing both fixed and floating point variables are permitted. A maximum of 300 characters are allowed (statement numbers not counted).

STATEMENT NUMBERS 1 - 99999.

FUNCTIONS AND SUBROUTINES Subroutines not contained in the FORTRAN library may be compiled by the use of Function and Subroutine statements. Functions and subroutines may have fixed or floating point values as defined by the programmer. Users are required to insure consistent references.

INPUT AND OUTPUT DECtape (Digital's Microtape system), magnetic tape, paper tape, Teletype. Format may be specified by use of a FORMAT statement.

STATEMENTS AVAILABLE Arithmetic statements, I/O statements with FORMAT, DO, Dimension, Common, IF, GOTO, Assign, Continue, Call, Subroutine, Function, Return.

TYPE DECLARATIONS Variables may be declared as real, integer, and FORTRAN. Variable names are 1-6 alphanumeric characters.

MIXED CODES Symbolic instructions can be intermixed with FORTRAN statements.

VARIABLE PRECISION ARITHMETIC Variable precision floating point arithmetic is used with a choice of mantissa (25 or 36 bits) and exponent (8 or up to 99 bits).

ARRAYS Arrays of up to 4 dimensions, either fixed or floating may be defined.

For a complete description of FORTRAN and its operation, refer to the PDP-7 FORTRAN Manual, Digital 7-2-S.

BUS-PAK II

Bus-Pak II is a program assembly system designed for data processing operations. By operating on a character-by-character basis, its instructions are powerful, yet easy to learn and understand. Bus-Pak II offers programming features such as Editing, two modes of indexing and complete input/output control. The Bus-Pak II programming system was developed so that many of the manual record keeping and updating operations could easily be converted to make use of a PDP-4 or PDP-7 computing system. Bus-Pak II users do not have to understand the computer operation. Through the use of the pseudo-language, the PDP-7 is operated as a business-oriented computer, performing all functions including the handling of peripheral input/ourput equipment.

Modes of Operation

Bus-Pak II has two (2) modes of operation. A "Run" mode which is used for normal execution of the user's program and a "Single Instruction" mode for use in debugging Bus-Pak II programs. The control of the mode of operation is by the AC switch zero on the console. When AC switch zero is in the down position, Bus-Pak II operates in the "Run" mode. When AC switch zero is in the up position, Bus-Pak II operates in the "Single Instruction" mode.

In the Single Instruction mode of operation, Bus-Pak II halts after the execution of each Bus-Pak II instruction and indicates in the AC lights on the console the address of the next Bus-Pak II instruction to be executed. When a "GOTO" instruction is executed, Bus-Pak II will not stop until the instruction at the location indicated by the GOTO instruction is executed.

Addressing

Both instructions and data essential for processing are contained in core storage. Each core storage location is completely addressable. Bus-Pak II instructions are variable length type instructions in that not all the instructions take up the same number of core storage locations.

Data fields being processed are also of the variable length type. A data field length is determined by the "N" (number of characters) field in a specific instruction. All data is processed from left to right, for as many characters specified by the instruction being executed. Both instructions and data may be intermixed as long as the data does not interfere with the normal flow of the program.

Input/Output Storage Assignments

No specific input/output areas have been assigned to any input/output device in the Bus-Pak II system. The assignment of these areas has been left entirely up to the programmer. In this way, more efficient and less core consuming programs may be written. Care, though, must be taken so that an area defined for a specific input/output device is large enough for the particular device.

Editing

In the printing of reports, it is sometimes necessary to punctuate numeric data by dollar signs, commas, and decimal points. This punctuation would take many instructions of testing and shifting the data and inserting the correct punctuation characters. The editing feature provides this punctuation of data automatically, based on a control word specified by the user. Floating dollar sign and asterisk protection is also available for check writing. Multiple sequential data fields may be edited in one editing operation.

Indexing

Indexing is a means of address modification without disturbing the original data address in an instruction. Bus-Pak II makes available two modes of indexing, single indexing and double indexing. An effective address is calculated for every "TO," "FROM," and "BY" address field specified by an instruction. In single indexing, the contents of the index register specified by an address field are added to the data address, and this new effective address is used in the execution of the instruction. In double indexing, the contents of the index register specified by the double index register are also added to the data address and this new address used in the execution of the instruction.

Indirect Addressing

When indirect addressing is specified, the address is interpreted as the address of the register which contains the address of the data to be processed. Multiple levels of indirect addressing are available, and each level of a "TO" or "FROM" address field may use single and/or double indexing.

Double Precision Accumulators

All arithmetic operations on numeric data must be done by the use of one of the fifteen (15) double precision accumulators available in Bus-Pak II. Each accumulator is capable of containing a magnitude not exceeding ±3 4 3 5 9 7 3 8 3 6 7 ($\pm 2^{35}-1$). An overflow indicator is associated with each of the 15 available accumulators. The signs of the accumulators are computed algebraically depending on the signs of the data being calculated. Arithmetic (add, subtract, multiply, divide) can be performed directly to memory. See ADDMEM instruction example.

Program Counters

Fifteen (15) program counters are available for controlling multiple execution of a particular sequence of instructions.

Sense Switches

Fifteen (15) sense switches are available through the use of the AC switches on the console for manual control of program execution.

Program Switches

Fifteen (15) program switches are available for internal control of program execution.

Bus-Pak Example 1

MOVE CHARACTERS MV

op-code	mnemonic	variable operands
17411	MV	N FROM TO

The "N" consecutive characters starting address "FROM" are moved from left to right to the "N" consecutive character positions starting address "TO."

NOTE: 1. The original "N" consecutive characters with the starting address "TO" are replaced by the "N" consecutive characters with the starting address "FROM." The "N" consecutive characters with the starting address "FROM" are left undisturbed.

EXAMPLE:

MV 3 47Ø 473

CORE STORAGE	before	A	B	C	D	E	F
CONTENTS	after	A	B	C	A	B	C
CORE STORAGE		4	4		4		
ADDRESSES		7	7		7		
		Ø	1		3		

The MOVE instruction is typical of the generalized data manipulating instructions contained in Bus-Pak II.

Bus-Pak Example 2

ADD TO MEMORY — ADDMEM

op-code	mnemonic	variable operands
17523	ADDMEM	AC N TO

The contents of accumulator "AC" are algebraically added to the "N" consecutive characters with the starting address "TO." The results are placed into the "N" consecutive character positions with the starting address "TO."

NOTE:

1. The contents of accumulator "AC" are left undisturbed.
2. The sign over the units position of the "N" consecutive characters with the starting address "TO" is taken into consideration.
3. The original "N" consecutive characters with the starting address "TO" are lost.
4. If the result of the addition produced a value whose magnitude exceeded the capacity of the accumulator, the associated overflow indicator will be set. The result itself is worthless.
5. The sign of the result is placed over the units position of the "N" consecutive characters with the starting address "TO."

EXAMPLE:

ADDMEM 3 5Ø1

CONTENTS OF CORE STORAGE	before	Ø	7	5	3	9
	after	1	1	Ø	3	9
CORE STORAGE ADDRESSES		5 Ø 1				5 Ø 5

CONTENTS OF SPECIFIED ACCUMULATOR	before	+35
	after	+35

CHAPTER 7

INPUT/OUTPUT EQUIPMENT

This chapter contains descriptions of standard DEC input/output equipment. Included with each description are the iot instructions used with the device when it is connected to the PDP-7.

Peripheral equipment may either be asynchronous with no timed transfer rates or synchronous with a timed transfer rate. Devices such as the CRT displays, printer-keyboard, and the line printer may be operated at any speed up to a maximum without loss of efficiency. These asynchronous devices are kept on and ready to accept data; they do not turn themselves off between transfers. Devices such as magnetic tape, DECtape, the Serial Drum, and card equipment are timed-transfer devices and must operate at or very near their maximum speeds to be efficient.

Some of the timed-transfer devices can operate independently of the central processor once they have been set in operation by transferring a continuous block of data words through the PDP-7 Data Interrupt. Once the program has supplied information about the location and size of the block of data to be transferred, the device itself takes over the work of actually performing the transfer.

MECHANICAL CONFIGURATION

The basic PDP-7 is housed in two DEC standard (22-1/4" x 27-1/16" x 69-1/8") metal cabinets. Equipment is mounted within the bays (FRONT) and on the rear doors (REAR) in a layout shown in Figure 7-1. Doors on the front of Bay 1 provide access to the memory, central processor, and EAE wiring panels. The punch is housed in a pull-out drawer. Short doors beneath the removable table provide access to the I/O Control wiring panel. Power supplies for the central processor and memory (up to 32K) are mounted on the rear door of Bay 1. Additional bays are easily added to provide space for I/O options and other equipment.

I/O BUFFERING

Separate parallel buffers are provided on each input/output device attached to the basic PDP-7. The high-speed paper tape reader control contains an 18-bit buffer and binary word assembler. The high-speed paper tape punch, the teleprinter, and the teleprinter keyboard each contain separate 8-bit buffers.

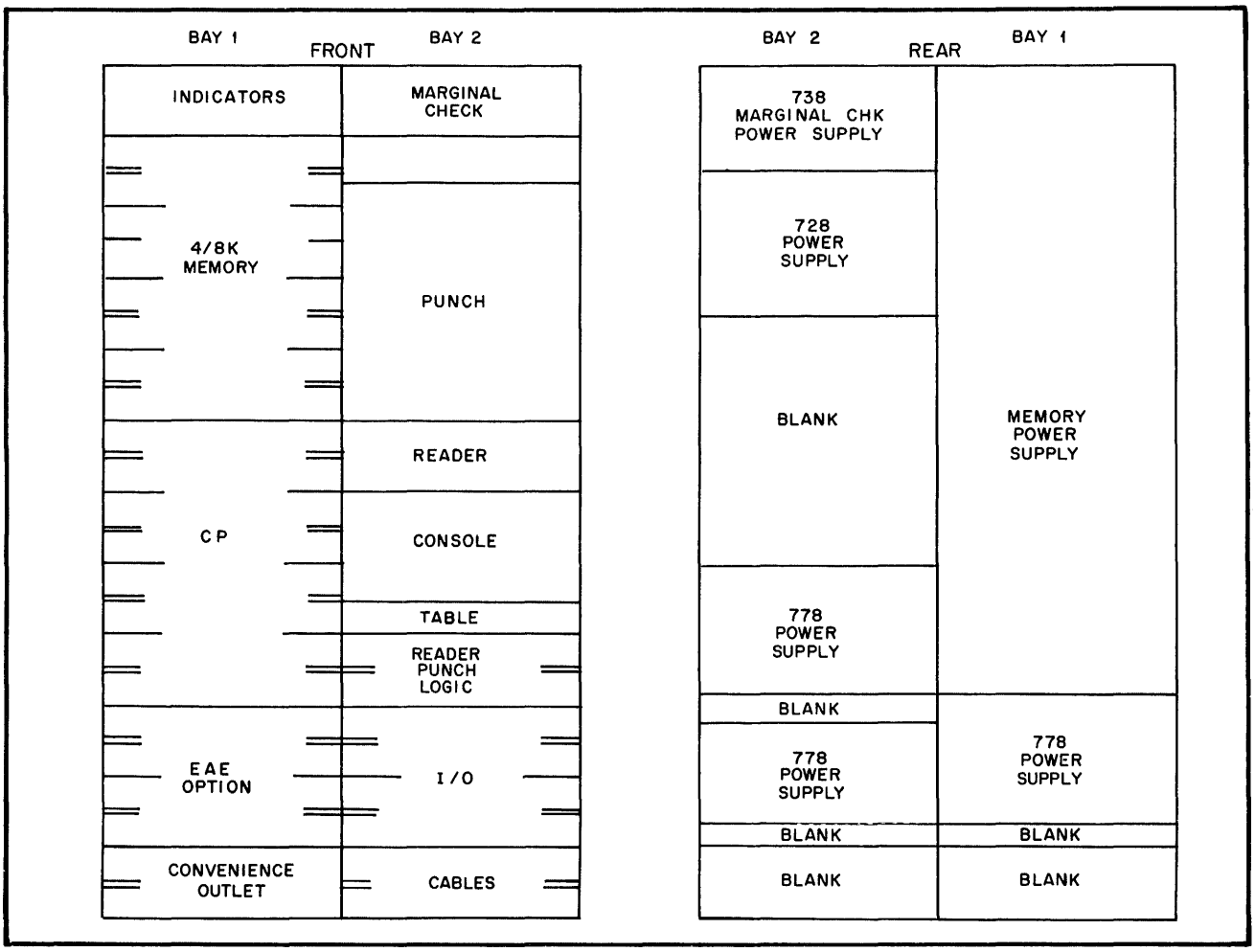


Figure 7-1 Cabinet Layout

Separate parallel buffers are also incorporated as part of DEC Standard I/O peripheral equipment. Information is transferred between the accumulator and a device buffer during the execution time of a single cycle iot instruction. Because the maximum time the accumulator is tied to any one external buffer is 1.75 microseconds, many standard I/O devices can operate simultaneously under control of the PDP-7.

Figure 7-2 shows the data path between device buffers and the AC through the Information Collector or Information Distributor.

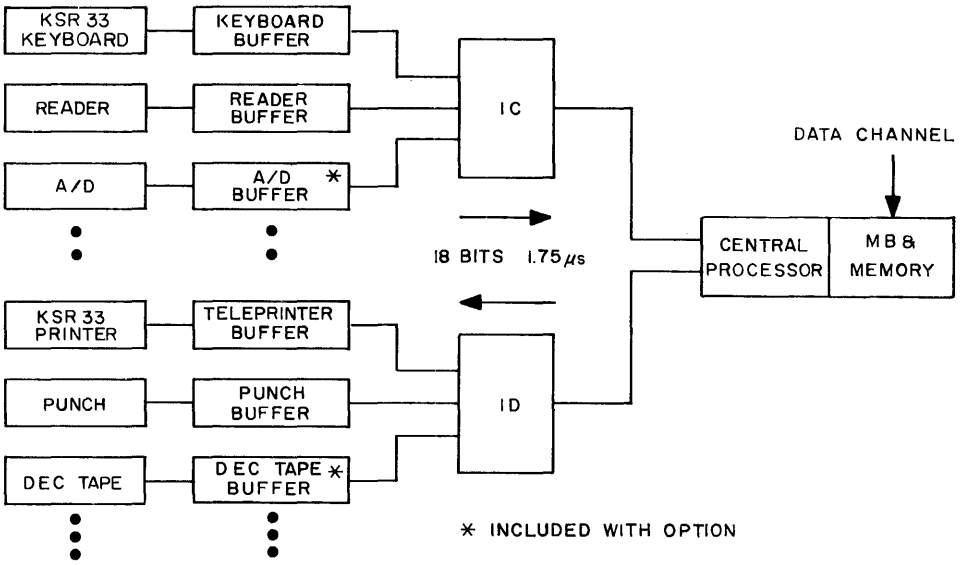


Figure 7-2 Input/Output Flow

TELETYPE MODEL 33 KSR

(Standard Equipment with the PDP-7)

The Teletype Model 33 KSR (keyboard-send-receive) can be used to type in or print out information at a rate of up to ten characters per second. Signals transferred between the 33 KSR and the keyboard printer 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 zeros and ones in the control and computer. The start mark and subsequent eight character bits are one unit of time duration and are followed by a two unit stop mark.

Each of the (64 type) characters and 32 control characters are represented by an 8-bit standard ASCII code. The Teletype eight-level code is listed in the Appendix. The teleprinter input and output functions are logically separate, and the programmer may think of the printer and keyboard as individual devices.

Keyboard

The keyboard control contains an 8-bit buffer (LUI) which assembles and holds the code for the last character struck on the keyboard. The keyboard flag becomes a 1 to signify that a character has been assembled and is ready for transfer to the accumulator. This flag is connected to the computer program interrupt and input/output skip facility and may be cleared by command. Instructions for use in controlling the keyboard are:

ksf	700301	Skip if the keyboard flag is set to 1. If the flag is 0, the next instruction is executed. If it is 1, the next instruction is skipped. The flag is set only when a character has been completely assembled by the buffer.
krb	700312	Read the keyboard buffer. The contents of buffer are placed in bits 13-17 of the AC and the keyboard flag is cleared.

Teleprinter

The teleprinter control contains an 8-bit buffer (LUO) which receives a character to be printed from AC bits 10 through 17. The LUO receives the 8-bit code from the AC in parallel and transmits it to the teleprinter serially. When the last bit has been transmitted, the teleprinter flag is set to 1. This flag is connected to the computer program interrupt and input/output skip facility. It is cleared by programmed command. The instructions for printing are:

tsf	700401	Skip if the teleprinter flag is set.
tls	700406	Load printer buffer and select. The contents of AC_{10-17} are placed in the buffer and printed. The flag is cleared before transmission takes place and is set when the character has been printed.

PERFORATED TAPE READER TYPE 444

(Standard Equipment with the PDP-7)

The tape reader is a timed-transfer device which senses the holes punched in 5, 7, or 8-channel paper (or Mylar-base) tape. The standard input medium is 8-channel tape. The maximum reading rate is 300 characters (lines) per second. A power switch is provided on the reader. This switch is usually left on, however, as the reader power is removed when the computer is turned off.

Operation of the tape reader is controlled entirely by the program. When the reader is selected, the brake is released and the clutch engages the drive capstan to move the tape past the photocells which sense the holes punched in the tape. For each hole present in a given line of tape, a corresponding bit of the reader buffer is set to 1.

Information can be read from tape and assembled in the reader buffer in one of two modes:

ALPHANUMERIC MODE: Each select instruction causes one line of tape, consisting of eight bits, to be read and placed in the buffer. Blank tape is ignored. The absence of a feed hole causes the character punched in that line to be ignored. See Figure 7-3.

BINARY MODE: In the binary mode, select the instruction causes three lines of tape to be read. The first six bits of each line are assembled in the buffer, thus three lines form a single 18-bit word. The seventh bit is ignored. However, a character is not read unless the eighth bit is punched. See Figure 7-4.

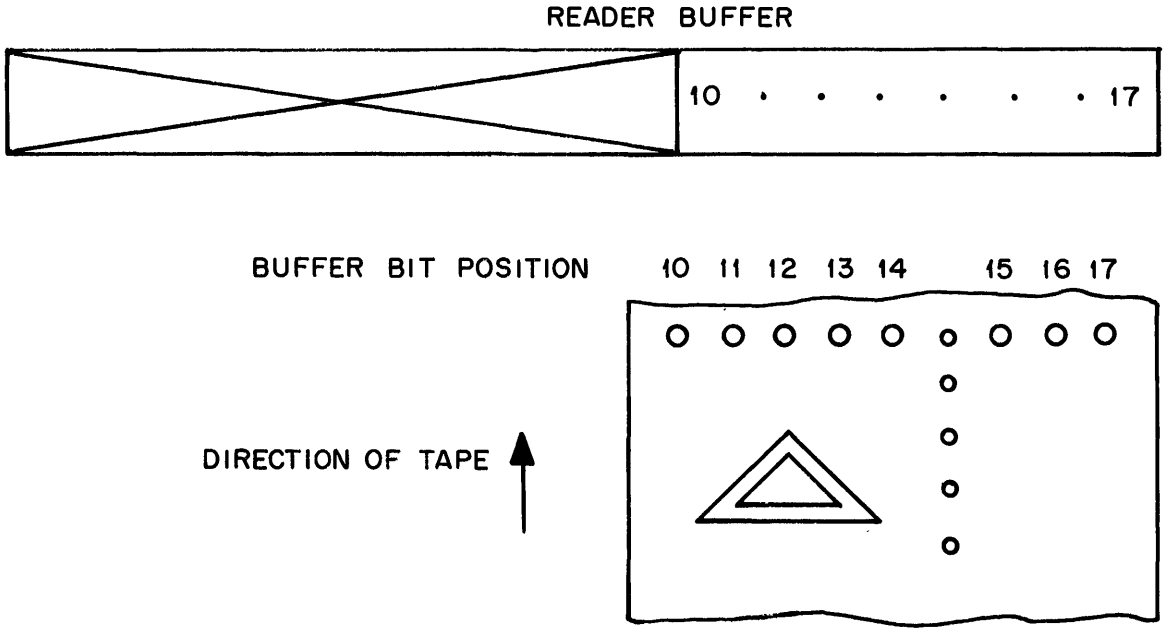


Figure 7-3 Alphanumeric Perforated Tape Format and Reader Buffer Bit Assignment

PAPER TAPE READER INSTRUCTIONS

rsa	700104	Select reader in alphanumeric mode. One 8-bit character is read and placed in the reader buffer. The reader flag is cleared before the character is read. When transmission is complete, the flag is set.
rsb	700144	Select reader in binary mode. Three 6-bit characters are read and assembled in the reader buffer. The flag is immediately cleared and later set when character assembly is completed.

rsf	700101	Skip if reader flag is set.
rcf	700102	Clear reader flag then inclusively OR reader buffer into AC. C(RBj) V C(ACj) =>C(A Cj).
rrb	700112	Clear reader flag. Clear AC and then transfer contents of reader buffer to AC. C(RB) = > C(AC).

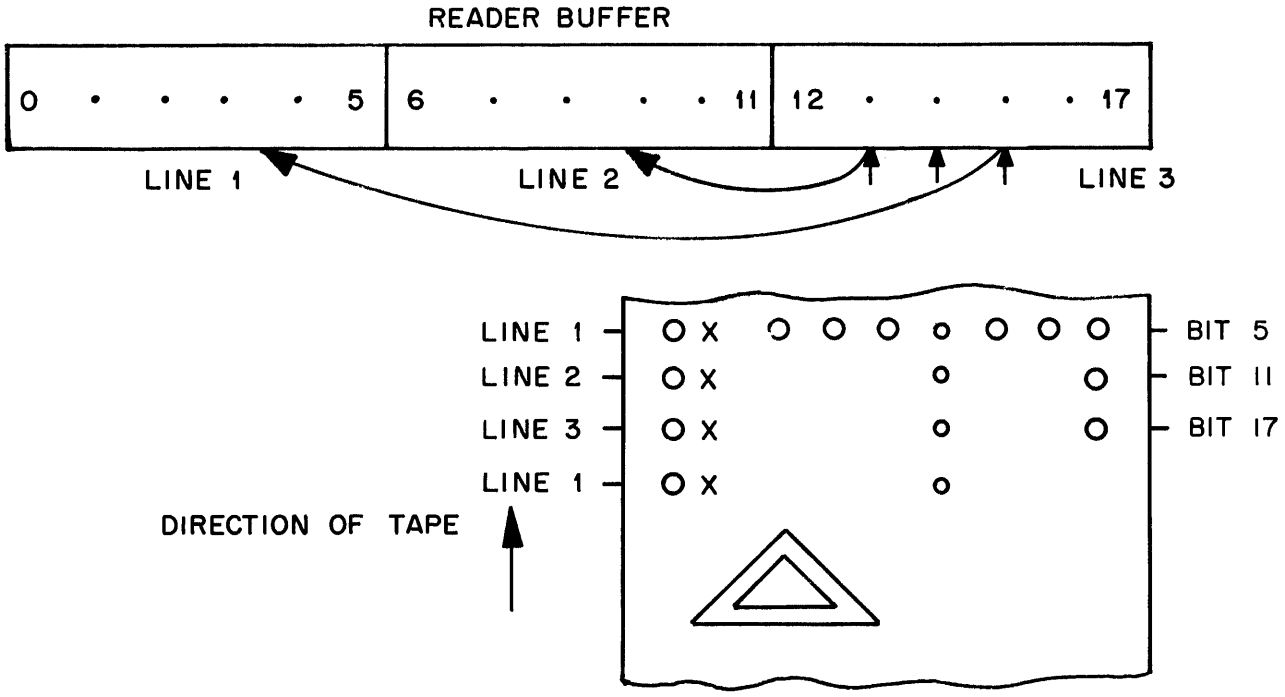


Figure 7-4 Binary Perforated Tape Format and Reader Buffer Bit Assignment

PERFORATED TAPE PUNCH TYPE 75

(Standard Equipment with the PDP-7)

The Tape Punch is a timed-transfer device capable of punching 5, 7, or 8 channel tape at a maximum rate of 63.3 characters per second. The standard input medium is 8 channel tape.

Operation of the Tape Punch is controlled either by the program or by the computer operator. The operator may punch blank tape (feed hole only punched) by depressing the punch feed button on the console or he may force on the punch power by turning on the console punch switch. Normally, the punch is left completely under program control. An instruction to punch when the punch is turned off causes the punch to be turned on and the actual

punching takes place approximately one second later when the punch motor is up to speed. Note that the central processor is never delayed by this instruction nor any other IOT instruction. Subsequent punching follows at normal punch speed. The motor remains on for five seconds after the last punch command is given.

When the punch is selected, the contents of AC_{10-17} are sent to the punch buffer and then subsequently placed on tape. If a bit in the AC is a 1, the corresponding bit in the buffer is set. Since the punch buffer is automatically cleared after punching a character, it is impossible to OR into it. Information is handled by the punch logic in one of two modes:

ALPHANUMERIC MODE: Each select instruction causes one line of tape, consisting of eight bits, to be punched. A hole is punched in a tape channel if the corresponding punch buffer bit is a one. A feed hole is always punched.

BINARY MODE: Each select instruction causes one line of tape, consisting of eight bits, to be punched. Holes are punched corresponding to bits 12-17 of the punch buffer. Bit 11 is never punched and bit 10 is always punched. This forces the standard format for binary information on tape.

TAPE PUNCH INSTRUCTIONS

psa	700204	Punch a line of tape in alphanumeric mode. The punch flag is immediately cleared and then set when punching is complete.
psb	700244	Punch a line of tape in binary mode. The punch flag is immediately cleared and then set when punching is complete.
psf	700201	Skip the following instruction if the punch flag is set.
pcf	700202	Clear the punch flag.

The following instruction will cause a line of blank tape (except for feed hole) to be punched. The accumulator is also cleared.

psa +10	700214	Clear AC and punch.

The following instruction as used on the PDP-4 is also available, but is generally replaced with the more direct psa.

pls	700206	Same as psa.

DECTAPE

DECtape (Digital's microtape system) is a bidirectional magnetic tape system which uses a ten-track recording head to read and write five duplexed channels. The DECtape system incorporates the Type 555 DECtape Dual Transport and the Type 550 DECtape Control.

Dual DECtape Transport Type 555

The Type 555 Transport consists of two logically independent bidirectional tape drives capable of handling 260 foot reels of 3/4 inch, 1.0 mil Mylar tape. The bits are recorded at a density of 375 (±60) bits per track inch. Since the tape moves at a speed of 80 inches per second, the effective information transfer rate is 90,000 bits per second, or one 18-bit word every 200 microseconds. Traverse time for a reel of tape is approximately 40 seconds.

The 3-1/2 inch reels are loaded simply by pressing onto the hub, bringing the loose end of the tape across the tape head, attaching it to the take up reel, and spinning a few times. Individual controls on the transport enable the user to manipulate the tape in either direction manually. The units can be "dialed" into a particular selection address.

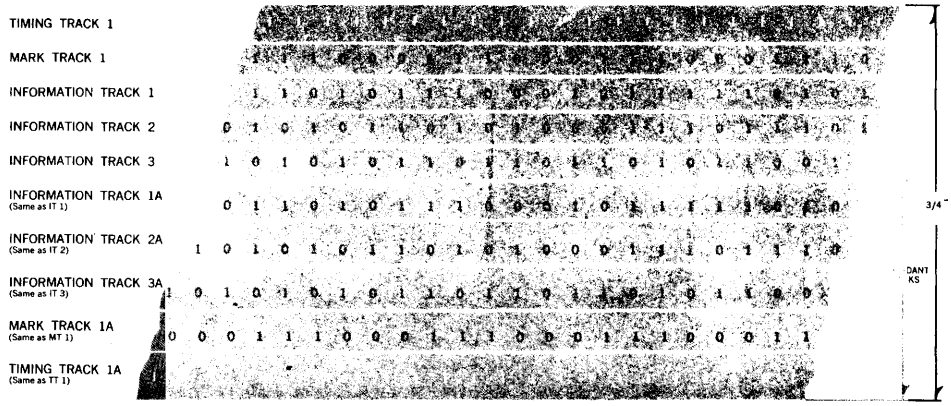
There is no capstan or pinch-roller arrangement on the transport, and movement of the tape is accomplished by increasing the voltage (and thereby the torque) on one motor, while decreasing it on the other. Braking is accomplished by a torque pulse applied to the trailing motor. Start and stop time average 0.15-0.2 seconds and turn around takes approximately 0.3 seconds.

Recording Technique

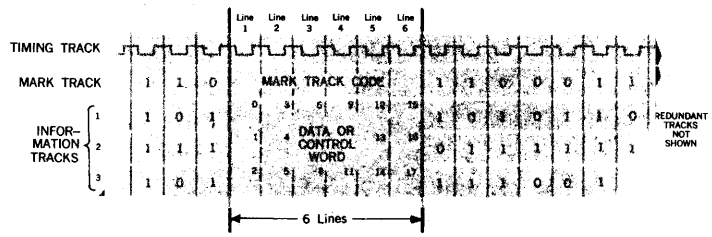
The DECtape system uses the Manchester type polarity sensed (or phase modulated) recording technique. This differs from other standard types of tape recording where, for example, a flux reversal might be placed on the tape every time a one is desired. In the polarity sensed scheme a flux reversal of a particular direction indicates a zero while a flux reversal in the opposite direction indicates a one. A timing track, recorded separately in quadrature phase, is used to strobe the data tracks. Thus, the polarity of the signal at strobe time indicates the presence of a zero or one. Using the timing track on the tape as the strobe also negates the problems caused by variations in the speed of the tape. See Figure 7-5.

With this type of recording only the polarity, not the amplitude of the signal, need be considered, thus removing some of the signal to noise problems and allowing the use of read amplifiers with high uncontrolled gain. This recording also allows the changing of individual bits on the tape without changing the adjacent bits.

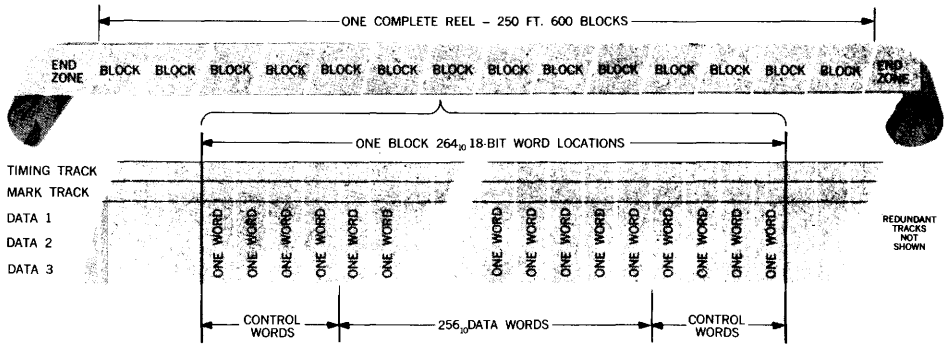
Reliability is further increased by redundantly recording all five of the information tracks on the tape. Figure 7-5 shows the placement of this track. This is accomplished by wiring the two heads for each information track in series. On reading, the analog sum of the two heads is used to detect the correct value of the bit. Therefore, a bit cannot be misread until the noise on the tape is sufficient to change the polarity of the sum of the signals being read. Noise which reduces the amplitude would have no effect.



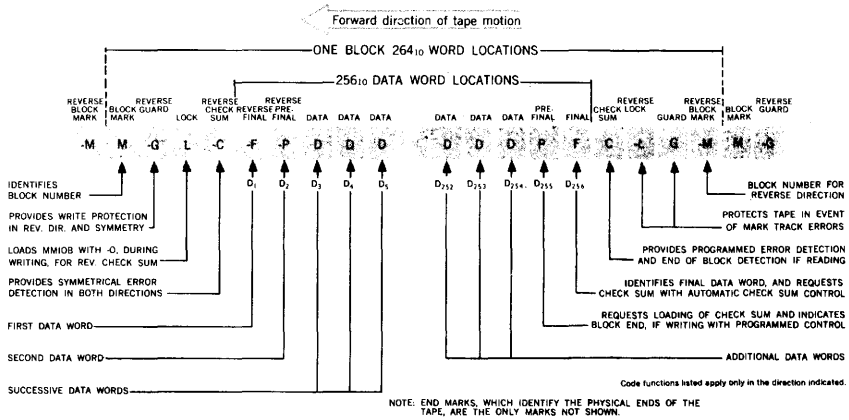
Track Allocation Showing Redundantly Paired Tracks



Basic Six Line Tape Unit



Control and Data Word Assignments



DECtape Mark Track Format (Assumes 256_{10} Data Words Per Block)

Figure 7-5 DECtape Recording

DECtape Control Type 550

The DECtape Control Type 550 operates up to four Type 555 Dual Tape Transports (8 drives) transferring binary data between tape and computer. By using the automatic Mark track decoding of the control and the program interrupt facility of the computer to signal the occurrence of data words, errors, or block ends, computation in the main program can continue during tape operations. Information can be transferred with programmed checking by using the subroutines which are provided with the equipment. Format control tracks, tailored to individual use by establishing any desired block lengths, can also be written with the subroutines provided. The Control allows reading and writing of any number of words at one mode command irrespective of the block length. Assembly of lines on the tape into 18 bit computer words in either direction is performed automatically by the Control. Status bits available to the program specify the current condition of the Control and error indications.

DECtape Programming

Three main groups of Programs are provided with the DECtape Systems: a basic set of subroutines for searching, reading and writing; a set of maintenance and diagnostic routines (DECTOG); and a program for easy storage and retrieval of information via the computer console (DECTRIEVE).

The basic PDP-7 subroutines for reading, writing, or searching allow the user to specify the total number of words to be transferred irrespective of the block format on the tape. Searching can occur in either direction, and the search routine can be used independently to position the tape or is used automatically by the read and write subroutines. Transfer of data in this program, however, will occur only with the tape moving in the forward direction. If the number of words specified is not a multiple of the aggregate block lengths, the final block is filled with zeroes which are ignored upon reading. The subroutines use the program interrupt during searching but will pre-empt the computer during the actual transfer of data. One auto-index register is used and must be defined by the main program, and "DISMIS" must be defined as a jump to the routine which dismisses the interrupt. When the transfer is completed, a programmed status register is set and a return is made to the main program with the tape stopped. Errors are detected, coded numerically, saved in status bits and indicated by a predesignated error return. The programmer can decode the error and proceed in any manner desired. Approximately 400_8 words of storage are used. A sample sequence of instructions for transferring core locations 1000 through 1777 beginning with block 100 on tape unit 1 would appear as follows:

```
jms     MMWRS     /Or MMRDS for Reading
law     100       /Or LAC (100) Block Number
jmp     ERR       /Error Return
10000             /Unit Selection
law     1000      /Or 1000, Core Starting Address
law     1777      /Or 1777, Core Final Address
```

DECTOG for the PDP-7 is a collection of short programs which allow the user to perform various DECtape functions using the Accumulator Switches on the console. Programs available include those which create the Mark track and block format, read or write designated portions of the tape, write specified patterns on designated blocks in either direction, sum check designated blocks in either direction, "rock" the tape in various modes for specified times or distances, and an exerciser which writes and sum checks designated areas of the tape in both directions with changing patterns. Errors are completely analyzed and typed out together with the number of the block causing the error and the status of the DECtape system at the time of the error. Detailed descriptions of the various sub-programs are available. For a more complete description of DECTOG refer to Digital 7-20-I/O.

DECTRIEVE for the PDP-7 allows the user to save or retrieve data using the Accumulator Switches on the console. To store data the user specifies the unit, block number and starting and ending core locations. The data will be saved together with appropriate control information and sum checked. To retrieve the data only the unit and starting block need be specified. The control information is used to insure the correct starting block, the starting core location, and the amount of data to be read. Messages typed after reading or writing indicate the operation, tape blocks used, and the total check sum for verification purposes. All errors are fully analyzed as in DECTOG. Tapes are available for 4K or 8K memories and for the first or second DECtape controls. For a more complete description of DECTRIEVE refer to Digital 7-21-I/O.

DECtape INSTRUCTION LIST

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
mmrd	707512	READ. Clears IO or AC and transfers one word from MMIOB to bits 0-17 of AC. **
mmwr	707504	WRITE. Transfers one word from bits 0-17 of AC to MMIOB. **
mmse	707644	SELECT. Connects the unit designated in bits 2-5 of AC to the DECtape Control.**
mmlc	707604	LOAD CONTROL. Sets the DECtape Control to the proper mode and direction from bits 12-17 of the AC, as follows: **

**mmse and mmlc clear the Error Flag and error status bits (EOT, TIMING MTE, UNAB) and mmse, mmlc, mmrd, and mmwr clear the Data and Block End Flags.

DECtape INSTRUCTION LIST (continued)

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
		Bit 12 = Go ($\overline{\text{Go}}$ = Stop) Bit 13 = Reverse Bit 14 = In-motion Read Bits 15-17 = Mode: 0 = Move 1 = Search 2 = Read 3 = Write 4 = Spare 5 = Read through block ends 6 = Write through block ends 7 = Write timing and mark track i.e. 42 = Read Forward 62 = Read Reverse 43 = Write Forward 41 = Search Forward 61 = Search Reverse
mmrs	707612	READ STATUS. Clears the IO or AC and transfers the DECtape status conditions into bits 0-8 of the AC as follows: Bit 0 = Data Flag Bit 1 = Block End Flag Bit 2 = Error Flag Bit 3 = End of Tape Bit 4 = Timing Error Bit 5 = Reverse Bit 6 = Go Bit 7 = Mark Track Error Bit 8 = Tape Unable
mmdf	707501	Skip on DECtape Data Flag. In Search Mode: Block mark number should be unloaded via mmrd instruction. In Read Mode: Data or Reverse Check Sum should be unloaded via mmrd instruction. In Write Mode: Data should be loaded via mmwr instruction.

DECtape INSTRUCTION LIST (continued)

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
mmbf	707601	Skip on DECtape Block End Flag. In Read Mode: Unload forward Check Sum via mmrd instruction. In Write Mode: Load calculated forward Check Sum via mmwr instruction.
mmef	707541	Skip on DECtape Error Flag. Timing Error, Mark Track Error, End Tape, or Tape Unable Condition has occurred. Use mmrs instruction to detect specific error.

AUTOMATIC MAGNETIC TAPE CONTROL TYPE 57A

The Automatic Magnetic Tape Control transfers information between the PDP-7 and up to eight magnetic tape transports, using the data interrupt control and a data channel supplied with the Type 57A. A number of different tape unit configurations may be attached to the tape control, using one of three interfaces as follows:

Interface	Units Controlled	Densities Available
Type 520	DEC Type 50	200 bpi
Type 521	DEC Type 570	200 bpi, 556 bpi, 800 bpi
Type 522	IBM Model 729II, IV	200 bpi, 556 bpi
	IBM Model 7330	200 bpi, 556 bpi
	IBM Model 729V, VI	200 bpi, 556 bpi, 800 bpi

Tape format is standard and IBM-compatible, in odd or even parity modes.

The following functions controlled by various combinations of iot (in-out transfer) command are all possible.

Write
Write End of File
Write Blank Tape
Read
Read Compare
Space Forward
Space Backward
Rewind

Rewind/Unload
Gather Write
Scatter Read
Write Continuous
Read Continuous
Read Compare/Read
Read/Read Compare

Tape transport motion is governed by one of two control modes: Normal, in which tape motion starts upon command and stops automatically at the end of the record; and Continuous, in which tape motion starts on command and continues until stopped by the program when synchronizing flags or status conditions appear.

The tape control contains the following registers:

DATA ACCUMULATOR (DA): 18-bits. Characters read from tape are assembled in the DA and are taken, one 6-bit character at a time, from the DA to be written on tape.

DATA BUFFER (DB): 18-bits. A secondary buffer between the DA and the MB in the PDP-7. Under the data interrupt control, information is transferred between the MB and the DB.

COMMAND REGISTER (CR): 3-bits. Contains the tape operation to be performed, as specified by $C(AC_{9-11})$.

UNIT REGISTER (UR): 3-bits. Contains the number (0-7) of the tape unit addressed for the current operation, as specified by $C(AC_{15-17})$.

CURRENT ADDRESS REGISTER (CA): 13-bits. Contains the address of the memory cell involved in the next data transfer. The initial contents of the CA are specified by bits 5-17 of the AC.

WORD COUNT REGISTER (WC): 13-bits. Contains the 2's complement of the number of words involved in the transfer. The C(WC) are incremented by one after each word transfer. The initial contents of the WC are specified by AC_{5-17}.

Tape operations, modes, and unit numbers are specified by the contents of bits 7-17 of the AC. Tape control iot instructions transfer this information to the proper registers in the control. A set of mnemonics has been defined to place any desired combination of specifications in the AC by means of the law instruction. Data transfers are executed through the Data Interrupt, thereby permitting simultaneous computation and data transfer.

The iot instructions used to perform these operations are briefly described below. For detailed instructions on using the Type 57A Control, along with programming examples, refer to Digital's publication F-13(57A).

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
mscr	707001	Skip if the tape control is ready. This senses the tape control flag, which is set when an operation has been completed and the control is ready to perform another task. This flag is connected to the program interrupt.
msur	707101	Skip if the tape unit is ready. This senses the tape unit flag, which is set when the specified unit is ready for another operation. This flag is connected to the PIC.
mccw	707401	Clear CA and WC.
mlca	707405	Clear CA and WC, and transfer C(AC $_{5-17}$) to the CA. Loads the CA.
mlwc	707402	Load WC. Transfers C(AC$_{5-17}$) to the WC.
mrca	707414	Transfer the C(CA) to AC$_{5-17}$.
mdcc	707042	Disable TCR and clear CR. Clear WCO and EOR flags (see below).
mctu	707006	Disable TCR, clear CR, and WCO and EOR flags. Transmit unit, parity, and density to tape control.
mtcs	707106	Transmit tape command and start. This initiates the transfer.
mncm	707152	End continuous mode. Clears the AC; the operation terminates at the end of the current record.
mrrc	707204	Switch mode from read to read/compare. Allows mode switching during the operation.
mrcr	707244	Switch from read/compare to read.

The following commands deal with the two tape flags which determine when a transfer is complete. The WCO flag (word count overflow) is set when the WC becomes 0 after incrementing. The end of record (EOR) flag is set when the EOR mark is sensed. Both flags are connected to the PIC.

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
msef	707301	Skip if EOR flag is set.
mdef	707302	Disable EOR flag. This disconnects it from the program interrupt
mcef	707322	Clear EOR flag.
meef	707242	Enable EOR flag. This connects it to the PIC.
mief	707362	Initialize EOR flag. Clears and enables the flag.
mswf	707201	Skip if WCO flag is set.
mdwf	707202	Disable WCO flag.
mcwf	707222	Clear WCO flag.
mewf	707242	Enable WCO flag.
miwf	707262	Initialize WCO flag.

There are 11 status indicators associated with the Type 57A Tape Control. The states of all indicators may be observed by placing their contents into the AC. This is done by an instruction similar to iors, but applying only to the tape control. The instruction is given below; the AC bit assignment is on the following page.

mtrs	707314	Read tape status

AC bit	Indication is bit =1
0	Data request late
1	Tape parity error
2	Read /compare error
3	End-of-file flag is set
4	Write lock ring is out
5	Tape is at load point
6	Tape is at end point
7	(Type 520) Tape is near end point (Type 521 and 522) Last operation was writing
8	(Type 520) Tape is near load point (Type 521) B Control in use with multiplexed transport (Type 522) Write echo check OK
9	Transport is rewinding
10	Missed a character

MAGNETIC TAPE TRANSPORT TYPE 570

The Type 570 Tape Transport may be connected to the PDP-7 using the Type 57A Tape Control and the Type 521 Interface. It operates at speeds of 75 or 112.5 inches per second, and densities of either 200, 556, or 800 characters (bits) per inch.

The Type 570 includes a multiplexing interface that permits time-shared use of the transport by two tape controls connected to the same or different computers. This facilitates the pooling of tape units and allows two computers to exchange information via magnetic tape. Programming is described in the section on the Type 57A Tape Control.

MAGNETIC TAPE TRANSPORT TYPE 50

The Type 50 tape unit may be connected to the Type 57A Control using the Type 520 interface. It operates at a speed of 75 inches per second and records information in low density (200 characters per inch). Standard 7-channel, IBM-compatible tape format is used.

SERIAL DRUM TYPE 24

The serial drum system provides auxiliary data storage for the PDP-7 in any of three capacities: 32,768 words, 65,536 words, 131,072 words. Each word consists of 18 information bits and a parity bit (generated by the drum system control; the parity bit is not transferred to core memory).

Information is transferred between core memory and the drum in 256-word blocks. Each block is stored on one sector of the drum. Two sectors are interleaved to one drum track; depending on the drum size, there are 64, 128, or 256 tracks. From the programmer's point of view, the track may be ignored; the logical storage unit is the sector. Transfers are effected through the data interrupt control with the drum system providing the data channel.

Two iot instructions are required to initiate the transfer of a block of data. The first iot specifies the memory location of the first word of the block and determines the direction of the transfer; that is, drum to core or core to drum. The second iot instruction specifies the drum sector address and initiates the transfer, which then proceeds under data interrupt control. The drum transfer flag is set to 1 when a block transfer is successfully completed. The flag is connected to the program interrupt.

Four registers are used with the drum: (See Figure 7-6)

DRUM CORE LOCATION COUNTER (DCL) 16 bits. The DCL contains the core memory location of the next cell into or out of which a word is to be transferred. When a word transfer is complete, the C(DCL) are incremented by 1.

DRUM TRACK ADDRESS REGISTER (DTR) 9 bits. The DTR contains the address of the sector currently involved in a block transfer. At the completion of a successful transfer, the C(DTR) are incremented by 1.

DRUM FINAL BUFFER (DFB) 18 bits. This is a secondary buffer between the memory buffer and the drum serial buffer (see below). In writing, a word taken from the MB is placed in the DFB to await transfer to the drum. In reading, the word assembled in the serial buffer is placed in the DFB. The next data interrupt takes it to the MB and puts it in core.

DRUM SERIAL BUFFER (DSB) 18 bits. On reading, a word is read serially and assembled in the DSB. On writing, a word in the DSB is written serially around the drum track.

In addition to the drum transfer flag, an error flag is used with the drum system. It may be sensed by a skip instruction and should be checked at the completion of each block transfer. The error flag indicates one of two conditions:

1. A parity error has been detected after reading from drum to core.

2. The data interrupt request signal from the drum was not answered within the word-transfer period.

Because the DCL and DTR are automatically incremented (the DCL after each word transfer and the DTR after each successful block transfer), contiguous blocks of core may be written on successive sectors of the drum, and conversely. The contents of one core load (4096 words) may be transferred in either direction and would occupy eight successive tracks (16 successive sectors) on the drum.

The iot instructions added with the drum system are:

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
drlr	706006	Load counter and read. Places the contents of bits 2-17 of the AC in the DCL and prepares the drum system for reading a block into core memory.
drlw	706046	Load counter and write. Loads the DCL as above and prepares the drum system for writing a block from memory.
drss	706106	Load sector and select. Places the contents of AC_{9-17} in the DTR, clears both drum flags, and initiates the block transfer (read or write, as specified by the load counter instruction).
drcs	706204	Continue select. Clears the flags and initiates a transfer as specified by the contents of the DCL and DTR.
drsf	706101	Skip if drum transfer flag is set. This flag is set when a block transfer is completed.
drsn	706201	Skip if drum error flag is not set.
drcf	706102	Clear both drum flags.

For a complete description of drum timing, refer to the Digital publication, F-03(24A).

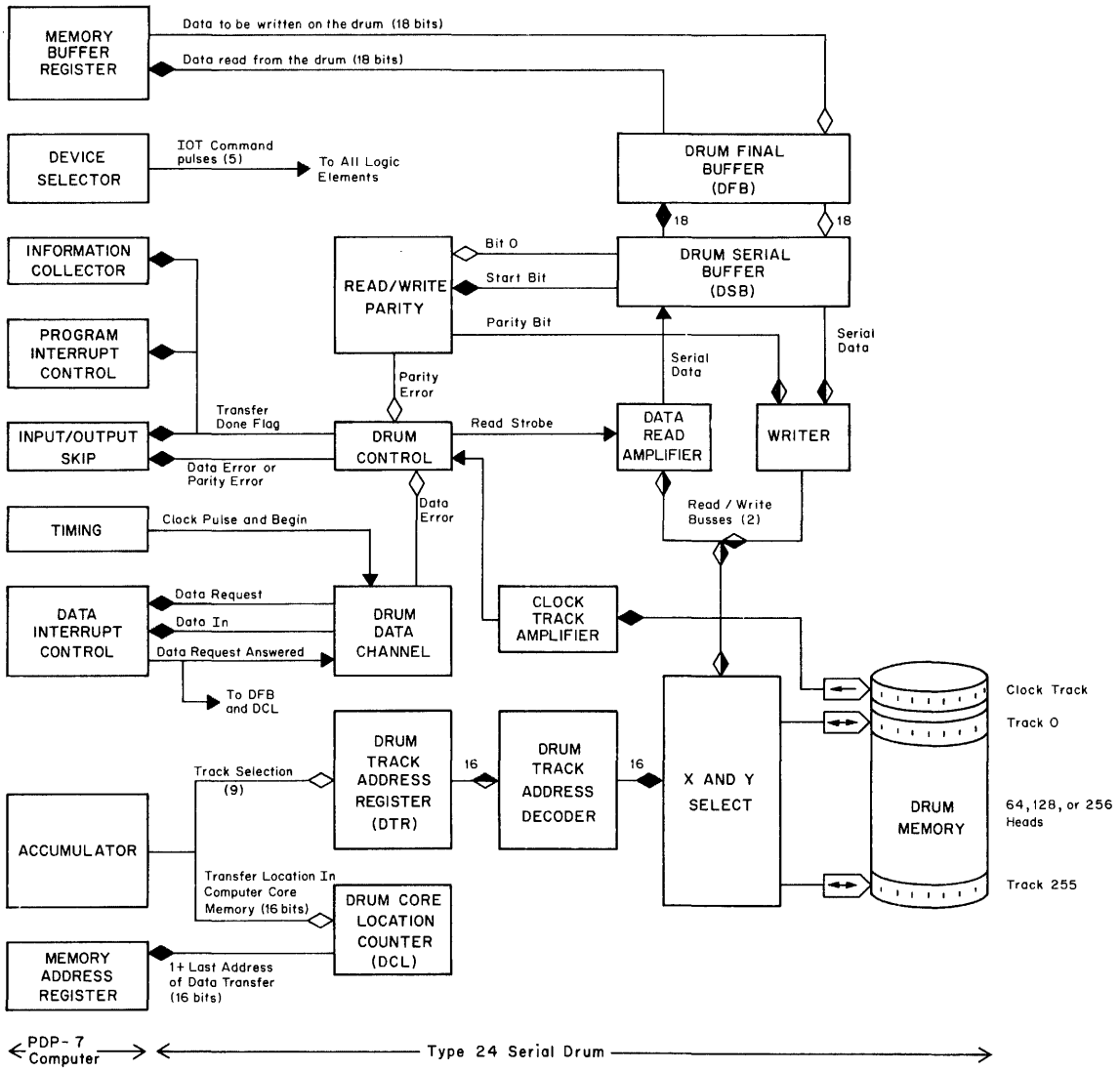


Figure 7-6 Drum Logic and Interface Connections

PRECISION CRT DISPLAY TYPE 30

The Type 30 displays points on the face of a cathode ray tube. Each point is located by its x- and y-coordinates in a square array whose origin is in the lower left corner of the tube face. The array contains 1024 points on a side and measures 9-1/4" x 9-1/4".

The x- and y-coordinates have their own 10-bit buffers which are loaded from bits 8-17 of the AC. In addition, there is a 3-bit brightness register (BR) which is loaded from bits 15-17 of the AC. The contents of this buffer specify the brightness of the point being displayed on the scale below. The five brightest intensities are easily visible in a normally lighted room; the dimmest can be seen in a darkened room.

C(BR)	Intensity
3	brightest
2	
1	
0	average
7	
6	
5	
4	dimmest

The x- and y-coordinate buffers (XB and YB) are loaded separately. Either may be loaded without selecting the CRT. The usual procedure is to load one buffer, then load the second buffer and select in one instruction. The Type 30 requires 50 microseconds to display a point. No flag is associated with this operation.

The iot instructions for the Type 30 are as follows:

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
dxl	700506	Load the x-coordinate buffer from AC_{8-17}. $C(AC_{8-17}) => C(XB)$
dxs	700546	Load the x-coordinate buffer and select. The point specified by the C(XB) and C(YB) is displayed.
dyl	700606	Load the y-coordinate buffer. $C(AC_{8-17}) => C(YB)$
dys	700646	Load the y-coordinate buffer and select. The point specified by the C(XB) and C(YB) is displayed.

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
dxc	700502	Clear the x-coordinate buffer.
dyc	700602	Clear the y-coordinate buffer.
dlb	700706	Load the brightness register from AC_{15-17}. Note: This instruction clears the display flag connected to the light pen.

PRECISION INCREMENTAL DISPLAY TYPE 340

The Type 340 Incremental Display is designed to permit rapid plotting of adjacent points, as in vectors and geometric figures. Adjacent points are plotted at a rate of 1.5 μs per point. Point locations are specified on a 9-3/8 inch square raster by any of the 1024X and 1024Y coordinate addresses. The origin is at the lower left corner of the raster. Plotting information is taken from sequential locations of core memory. Five word formats are used to display data in one of four modes. The location of the first word of the data is specified by the contents of bits 5-17 of the AC. The five word formats are as follows:

PARAMETER WORD Specifies the mode of display of the next word in sequence, the scale and intensity of the display, and status of the light pen.

POINT MODE WORD Specifies an x- or y-coordinate, light pen status, and the mode of the following word. Used for displaying random (non-sequential) points. Random points are displayed at the slower rate of 35 μs per point.

VECTOR MODE WORD Specifies the magnitude and direction of the x- and y-components of a vector. An escape bit determines whether or not the following word will be a parameter word.

VECTOR CONTINUE MODE WORD As in the vector mode, this format specifies magnitude and direction of components, but the vector is continued until the edge of the grid is encountered.

INCREMENT MODE WORD From a currently displayed point, this word specifies the direction in which the next adjacent point is to be displayed. Four increments are specified by a single word.

Detailed description of the Type 340 operation and the structure of the word formats are given in Digital's publication "Precision Incremental CRT Display Type 340" F-13(340). A list of the instructions added with the Type 340 follows:

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
idla	700606	Load address and select. The contents of AC_{5-17} are placed in the display address counter (DAC) and the display is started.
idse	700501	Skip on edge. If the edge of the grid is encountered (except in vector continue mode), the display stops and an interrupt occurs if the PIC is enabled. Skip on stop code. If a stop code is encountered in a parameter word, the stop flag is set. This flag is connected to the program interrupt.
idsi	700601	Skip on stop interrupt. This flag is connected to the program interrupt.
idsp	700701	Skip if light pen flag is set. When the pen senses a displayed point, the pen flag is set. This flag is connected to the program interrupt.
idrs	700504	Continue display. After a light pen interrupt, this causes the display to resume at the point indicated by the C(DAC).
idrd	700614	Restart display. After a stop code interrupt, this causes the display to resume at the point indicated by the C(DAC).
idra	700512	Read display address. Places the C(DAC) in AC_{5-17}.
idrc	700712	Read x and y coordinates. The $C(XB_{0-8})$ are placed in AC_{0-8}, the $C(YB_{0-8})$ are placed in AC_{9-17}.
idcf	700704	Clear display control. All flags and interrupts are cleared.

Display Options

Additional equipment is available for use with the Precision Incremental Display Type 340.

Type 341 Interface to the PDP-7 is a complete computer-display interface to the PDP-7 providing automatic, high-speed address control, data communication, data feedback, program interrupt, and skip capability. The interface provides sequential access to a single block of data in the computer core memory.

Type 342 Character Generator plots standard ASCII code characters on a 35-dot matrix in one of four sizes on the Type 340 Display. Average plotting time is 35 μsec per character. Two 64-character sets are available.

Type 343 Slave Display is used for remote observation of data displayed on the Type 340 Display.

Type 347 Subroutine Option permits data display from arbitrarily located and non-consecutive display tables within the PDP-7 memory.

HIGH SPEED LIGHT PEN TYPE 370

The high-speed light pen is a photosensitive device which senses displayed points on the face of the CRT. The Type 370 uses a fiber optic light pipe and photomultiplier system, which gives the pen a response time approximately five times faster than that of a photodiode. If the pen is held in front of a point displayed on the face of the CRT, it transmits a signal which sets the display flag to 1. The Type 370 is equipped with a mechanical shutter which prevents the sensing of unwanted information while positioning the pen. Variable fields of view are obtained by means of a series of interchangeable tips with fixed apertures. The iot instructions for the light pen are:

dsf	700501	Skip if the display flag is set.
dcf	700502	Clear the display flag.

Operation and programming of the Type 32, a photodiode light pen, are the same as for the Type 370.

SYMBOL GENERATOR TYPE 33

The symbol generator allows the programmer to plot text on the face of a Type 30 Display without having to specify every point of each character. This capability increases the speed of text display by a factor of about ten and reduces flicker proportionally.

Each symbol is plotted on a matrix of 35 dots (5 dots wide and 7 dots high) in one of four character sizes. The information is supplied in the form of two 18-bit data words. Once the coordinates of the starting point of the matrix are given, two iot instructions suffice to plot the whole symbol. When the plot is complete, the contents of the x-coordinate buffer are incremented automatically to provide a space between characters.

To plot a line of text, the coordinates of the starting point are given, using the two iot instructions, dxl and dyl. This point is the lower left dot of the matrix for the first symbol. Second, the format must be specified. Bits 15-17 of the AC specify the character size and whether automatic spacing is to be employed. Finally, the two plot instructions are given to display the symbol.

Detailed descriptions of the Type 33 operation and word format are given in the publication Digital Symbol Generator Type 33, F-13(33B).

MULTIPURPOSE ANALOG-TO-DIGITAL CONVERTER TYPE 138B

Twenty-four combinations of conversion accuracy and word length are possible with the Type 138B. Rotary switches on the front panel allow the converter characteristics to be changed to suit the application. One switch varies the word length from 6 to 11 bits. A second switch varies the switching point error from ±1.6% to ±0.05%. Conversion time varies with the switch settings as shown in the table.

The left-hand parameter is the maximum switching point error. Overall conversion error equals this error plus a quantization error of ±1/2 LSB (least significant bit). At the top of the table is the resolution in terms of the total number of binary bits. The table shows the total time required to perform a conversion. (See p. 7-26)

The six circled values in the table are for general purpose applications. The 11 lower settings are for use when accuracy, repeatability, and differential linearity are more important than resolution (as in histograms). The seven upper settings may be used when resolution is more important than accuracy, repeatability, and differential linearity (as in averaging applications).

Input

Analog signal may vary between 0 and -10 volts. Input load is ±1 microampere and 125 picofarads. If a different voltage range is desired, it is recommended that an amplifier be used at the source, since this will also provide a low driving impedance and reduce possibilities of noise pickup between the source and the converter.

Output

A signed binary number of 6 to 11 bits, left justified, with negative numbers represented in 2's complement notation. A 0 volt input gives the digital number 10000. ...A -5 volt input produces 0000.....A -10 volt input produces 01111.....Ones are represented by DEC Standard -3 volt Logic Levels; zeros, by ground levels. (Unsigned output is available on special order.)

Instructions

The iot instructions for the converter are:

adsc	701304	Select and convert. The converter flag is cleared and a conversion of an incoming voltage is initiated on the channel specified by the multiplexer address register. When the conversion is complete, the converter flag is set.
adrb	701312	Read converter buffer. Places the contents of the buffer in the AC left adjusted. The remaining AC bits are cleared. The converter flag is cleared.
adsf	701301	Skip if converter flag is set. This flag is connected to the program interrupt.

A-to-D CONVERTER TYPE 138B

CONVERSION TIMES
(Microseconds)

Max. Switching Point Error	Number of Bits 6	7	8	9	10	11
0.8%	(6)	(7)	8	9	10	11
0.2%	10	13	(14)	(15)	17	19
0.1%	16	19	22	24	(27)	29
0.05%	25	29	33	37	41	(45)

MULTIPLEXER CONTROL TYPE 139

The Type 139 is intended for use with the Type 138 A-to-D systems in applications where the PDP-7 must process sampled analog data from multiple sources at high speeds. For each new point selected, the multiplexer adds approximately 2.5 μsec to the Type 138 A-to-D conversion time. For example the combined time to switch to a point and convert with 10-bit accuracy is 2.5 μsec + 27 μsec = 29.5 μsec. Switching point accuracy for this example is 99.90 percent with an additional quantization error of half the least significant bit.

The Type 139 Multiplexer Control can include from one to 16 Type 15780 Multiplexer Modules (each module contains 4 independent transistorized floating switches), letting the user select any multiple of four channels to a maximum of 64. In the Individual Address mode, the Type 139 routes the data from any selected channel to the Type 138 converter input. In the Sequential Address mode, the multiplexer advances its channel address by one each time it receives an indexing command, returning to channel zero after scanning the last channel. Sequenced operations can be short-cycled when the number of channels in use is less than the maximum available.

A 6-bit multiplexer address register (MAR) specifies a channel number from $0\text{-}77_8$. A channel address may be chosen in one of two ways. It can be specified by the contents of bits 12-17 of the AC or by indexing the contents of the MAR. The following iot instructions are used:

adsm	701103	Select MX channel. The contents of AC_{12-17} are placed in the MAR.
adim	701201	Index channel address. The contents of the MAR are incremented by L. Channel 0 follows channel 77_8.

The channel address select instructions do not initiate a conversion. This can be done only by an adsc instruction to the converter (see Type 138).

Multiplexer Specifications

Multiplexer Switching time

2.3 microseconds for source resistance (R) $\leq$ 50 ohms 2.27 + 0.006 R (to be specified on order) for source resistance > 50 ohms.

Multiplex

Six lines accept DEC standard levels of 0 and -3 volts, with 0 volts for assertion.

Individual Address Control

Two pulses or level changes for clear and readin. The clear input accepts negative going signals with a swing of 2.5 to 4 volts, a fall time less than 0.5 microseconds, and a width greater than 60 nanoseconds. The readin terminal should receive a similar positive-going signal 1 microsecond later. Address inputs should be brought to final value at least 1 microsecond before clear.

Sequential Address Control

One pulse or level change for indexing should be negative going as above. Multiplex address inputs must be returned to -3 volts at least 2 microseconds before indexing.

Convert

One negative pulse, -2.5 volts amplitude and 0.2 to 0.4 microseconds duration. This may occur 2.3 microseconds after a clear or index pulse (2.27 + 0.006R microseconds for source impedance greater than 50 ohms).

HIGH SPEED ANALOG-TO-DIGITAL CONVERTER TYPE 142

The Type 142 Analog-to-Digital Converter transforms an analog voltage to a signed, 10-digit binary number with two's complement representation for negative numbers. Extremely high rates of conversion are possible with this unit; five microseconds is needed for one conversion. The sampling technique, a series of simultaneous comparisons, is responsible for the speed with which conversions take place; other methods used in similar conversion applications require 20 μsec or more for a 10-bit conversion. The new method simultaneously compares the amplitude of an analog signal with 16 digital values. Conversion accuracy is ±0.15% ±1/2 LSB (least significant bit).

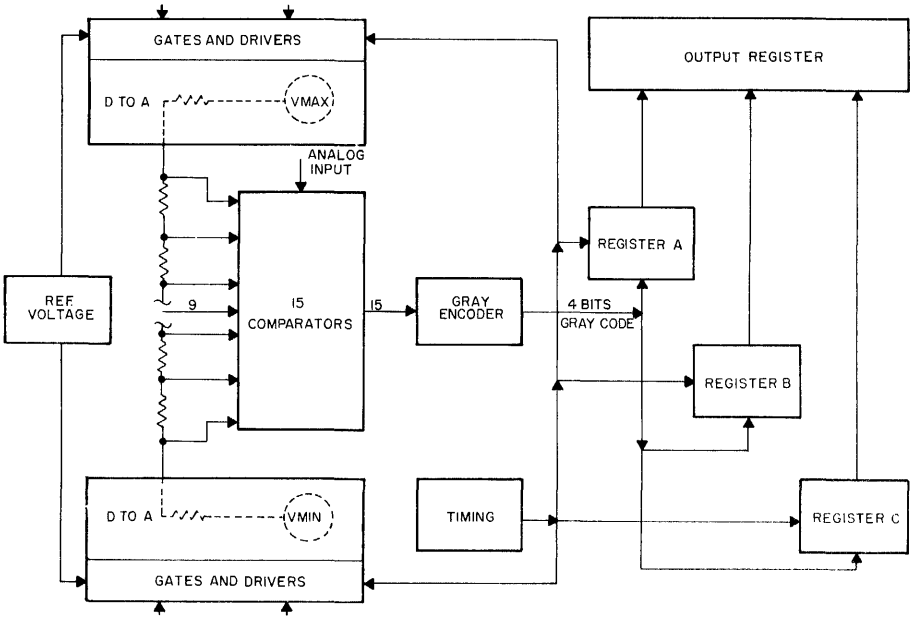


Figure 7-7 Type 142 Simplified Block Diagram

In the simplified block diagram, the voltage scales for each step are produced by the equipment shown at left. Two D-A converters define maximum and minimum voltage levels; 14 precision resistors generate a voltage scale between them. These resistances, plus the D-A converters, define 16 equal voltage levels. The 15 node points are applied to 15 comparators and are compared to the analog input. Registers A, B, and C are loaded with a Gray-coded word after each comparison during the conversion. The output register holds the 10-bit binary word at the end of conversion.

The Type 142 A-D Converter uses DEC System Modules throughout and is housed in two 25-position DEC mounting panels.

Specifications

Indicators

Indicators for the system are included on a standard 5-1/4 inch rack-mounting panel. The contents of the output register and Gray code registers are shown by the indicators.

Input

The analog signal can vary between 0 and -9 volts. The CONVERT pulse is the only digital input required. It should be a negative-going signal with a swing of 2.5 to 4 volts, a fall time less than 0.5 μsec, and a width greater than 60 nsec. The input presents a pulse load of 3 units. Maximum current is 25 microamperes.

Output

10 binary bits in two's complement notation. When used with PDP-7, these transfer to most significant bits of computer words. When used separately, signals are -3 volts for a one, ground for a zero. Available from 2 μsec after end of conversion until 3 μsec after start of next conversion.

DATA COMMUNICATION SYSTEM TYPE 630

The 630 Data Communication System (DCS) is a real time interface between Teletype stations and PDP-7. It is used for multiuser time sharing systems, message switching systems, and data collection-processing systems. Its basic function is to receive and transmit characters. When receiving, characters of different data rates and unit codes arrive from the Teletype stations in serial form. The DCS converts the signals to Digital voltage levels; the characters are converted from serial to parallel form and are forwarded to the computer. When transmitting, characters in parallel form are presented to the DCS by the computer. The characters are converted to serial Teletype form of the correct data rate and unit code; they are converted from Digital voltage levels to Teletype station signal levels, and they are sent to the Teletype stations.

The modularity and plugability of the 630 DCS simplify the expansion of the system from one station to 64 stations. Various combinations of data rates, unit codes, station types, and station signal levels can be accommodated in one 630 DCS.

The 630 System consists of the 631 Data Line Interfaces, 632 Send/Receive Groups, and a 633 Flag Scanner. It has a maximum capacity of 8 groups (8 stations per group) or 64 stations (128 pairs of wires for full duplex operation).

The Type 631 Data Line Interface converts Teletype station signal levels to Digital voltage levels and converts Digital voltage levels to Teletype station signal levels. The extent of modularity of the 631 is dependent upon the type of station signals to be converted. The 631 is plug connected to the 632.

The Type 632 Send/Receive Group converts parallel characters to serial Teletype characters or converts serial Teletype characters to parallel characters. It mixes the received characters of the eight Teletype stations onto a bus for presentation to the 633 and notifies the 633 when service is required.

When a character has been received or transmitted, a flag (indicator) is activated. The flag in turn notifies the 633 that service is required for that particular station. The manual OFF-ON switch mounted on the handle of the receiver and transmitter modules may be turned off to inhibit the flag from requesting service.

The Type 632 can accommodate a maximum of eight receiver modules and eight transmitter modules. The quantity required is dependent upon the number of Teletype stations. (If four half duplex stations are to be interfaced, only four receiver and four transmitter modules are required.) The type of each module required is dependent upon the data rate, unit code, and the number of data bits. Teletype stations requiring different data rates, unit codes, and data bits can be intermixed in the Type 632. The receiver module disregards hits (noise) less than one-half of a unit in length on an idle line. The 632 is completely pluggable.

The Type 633 Flag Scanner decodes and interprets computer instructions, forwards received characters to the computer upon request from the computer, sends characters to the transmitter modules when instructed by the computer, scans each 632 in search of activated flags, notifies the computer when an activated flag has been found, and forwards the station number requiring service to the computer upon request from the computer.

The Type 633 contains a precision crystal-controlled clock that generates highly accurate timing pulses. The transmitter and receiver modules use the pulses to sample the serial Teletype signals. An additional crystal clock can be added to accommodate multiple Teletype speeds. A crystal clock is also used to generate timing pulses that control the search logic of the scanner. The scanning mechanism of the 633 is modular. Each expansion permits eight additional stations (1 group) to be scanned.

A rotating priority scanner notifies the computer when an active flag has been found. The computer program requests the station number and then handles the character. Programmed priority of the stations is permitted.

The flag scanner operates at the following speeds: the maximum total time required to examine 64 inactive stations, 32 microseconds; the maximum total time to search, notify the computer and continue to search for 64 simultaneously active stations, 544 microseconds (exclusive of computer interrupt and programming cycles); the minimum time required to find the next active station upon being released by the computer, 6 microseconds; and the maximum time required to find the next active station (station being serviced minus one) upon being released by the computer, 92 microseconds.

Eight-Channel DCS

For smaller, lower-cost Data Communication Systems, programmed flag scanning can be used in place of the hardware Type 633 Flag Scanner. Up to eight remote teletype stations can be interfaced to the PDP-7 using the Type 634 Control.

The Type 634 Control:

1. decodes and interprets computer instructions.
2. forwards receive characters to the computer upon request from the computer.
3. sends characters to the transmitter modules when instructed by the computer.
4. requests computer service when notified by the 632 that service is required.

The Type 634 contains a precision crystal-controlled clock which generates highly accurate timing pulses. The transmitter and receiver modules use these pulses to sample the serial Teletype signals. An additional crystal clock can be added to accommodate intermixed Teletype speeds.

A computer program tests each flag to determine the station requesting service. A system of eight (8) stations tends to be the practical limit for this method of station service request detection. For more than 8 stations, a high-speed built-in flag scanner is recommended.

When the system is used in-house, the function of the 631 may be included in the 634. The 634 is a totally pluggable unit.

For a complete description of the DCS interface characteristics, operation, and instruction sets, refer to the Digital publication F-03 (630A).

CARD READER AND CONTROL TYPE 421A

The card reader reads standard 12-row, 80-column punched cards at a maximum rate of 200 cards per minute. Cards are read by columns 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. The information obtained from each column is placed in a 12-bit card reader buffer (CRB) from which it is transferred to the AC by the read buffer iot instruction.

The card reader buffer is a 12-bit register into which the information obtained from reading a card column is placed. Cards may be read in one of two modes:

Alphanumeric: The holes (bits) in a column are interpreted as a Hollerith character code (see Appendix). This is translated into a 6-bit card reader code for that character, which is then placed in bits 6-11 of the CRB. Bits 0-5 of the CRB are cleared.

Binary: The 12 bits of each column are accepted literally as a 12-digit binary number and placed directly into the CRB. A punch is interpreted as a 1; no punch, as a 0.

Card Reader Operation

Figure 7-8 is a photograph of the card reader console. The feed hopper is at the right, the run-out stacker at the left. Cards to be read are placed face down in the hopper, with the tops of the cards (12's edge) facing the operator. The plastic "hat" is placed on top of the desk to insure that enough weight is provided to prevent jamming as the last few cards are read.

The card reader console contains the buttons which control the operation of the device and the lights which indicate its availability. From the standpoint of the program, the card reader has two states, READY and NOT READY. In the READY condition, the card reader accepts a select instruction and moves a card through the read station. The NOT READY condition is caused by one of the following: power off, cover (of the console) not in place, empty hopper, full stacker, malfunction (read check, feed check, validity check), or end-of-file condition.

In each of these cases, the NOT READY light on the console is lit. The NOT READY condition exists until the START button is pressed, at which time the NOT READY light goes out. If a malfunction exists, the reset button must be pressed first.

The control buttons function as follows:

Button	Function
POWER ON POWER OFF	These buttons control the primary power to the reader. When the POWER ON button is pressed, it lights green; the motors are started, and the drive rollers which move a card through the reader are set in motion.
START	This button must be pressed to clear the not ready condition. Only then does the card reader accept a select instruction.
STOP	If the reader is in operation when this button is pressed, the reading of the currently selected card is completed, the reader stops, and the NOT READY light goes on. The START button must be pressed to make the reader available again.
RESET	After a malfunction (see below) has occurred, the RESET button must be pressed to turn off the check light and clear the reader logic of the condition which caused the error. It does not turn off the NOT READY light.
END OF FILE	When the operator wishes to signal the program that no more cards are to be expected, he presses the END OF FILE button when the hopper is empty. The button lights white when this happens. If the hopper is not empty, pressing this button has no effect. The end-of-file condition is removed and the light extinguished when cards are placed in the hopper.
VALIDITY ON	If this button is pressed, validity errors (see below) that occur when reading in the alphanumeric mode cause the not ready condition to occur. The card reading is completed, and the reader stops. This button, which lights yellow when pressed, has no effect when reading in binary mode.

The state of the reader is indicated by the console lights.

Light	Indication
NOT READY	When one of the conditions described above exists, this white light is lit. As long as it is on, the reader is not available to the program. The NOT READY light is turned off only by pressing the START button.

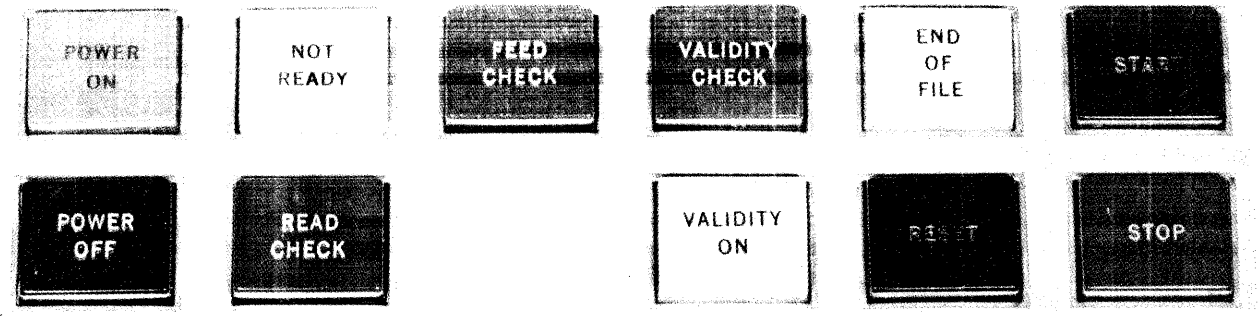


Figure 7-9 Card Reader Control Panel

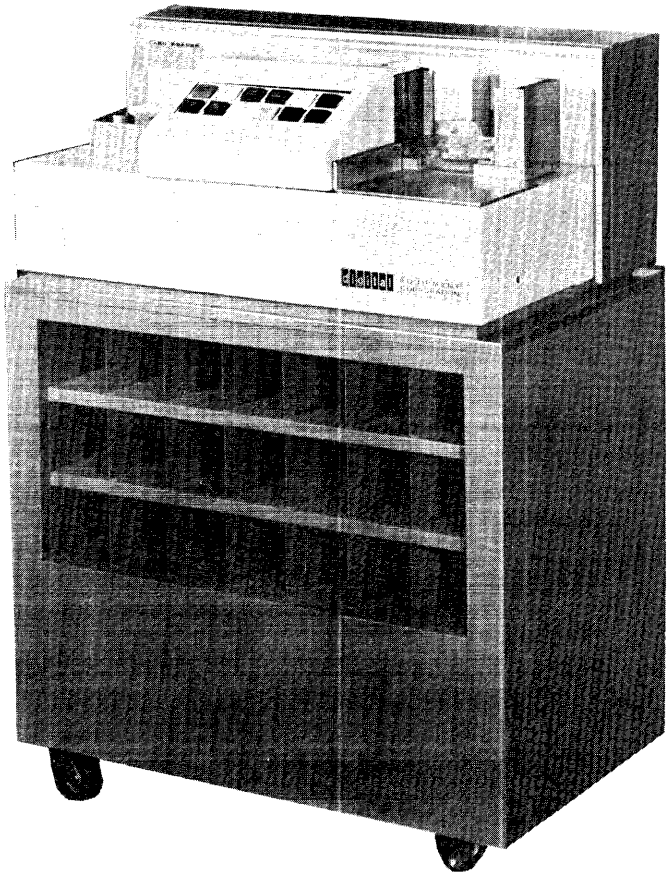


Figure 7-8 Card Reader Console

Light	Indication
READ CHECK FEED CHECK VALIDITY CHECK	Each of these red malfunction lights is lit whenever the corresponding error condition exists. In each case, the NOT READY light goes on at the same time, the current card is passed out of the reader, and reading stops. To make another attempt to read the card causing the error, take it from the top of the stacker and place it on the bottom of the deck in the hopper. Pressing RESET clears the malfunction and turns off the corresponding light, after which, pressing START clears the NOT READY light and makes the reader available.

Card Reader Check Indicators

Read Check When a read check error occurs, it indicates that something is wrong in the reading circuitry. If the condition is temporary, a second attempt to read the card should be successful. More likely, however, a read check indicates a failure of some part of the circuit, such as a bad read lamp or photocell. In this case, the reader probably requires technical attention.

Feed Check This error occurs when a card fails to move properly through the feed ways from the hopper into the stacker. If the card is bent, it may jam in the feed ways. If the trailing edge has been damaged by frequent handling, the pickup knife on the bottom of the hopper may not move the card to the drive rollers. When the card fails to appear at the read station in the prescribed time, a feed check occurs. In any case, the card in error should not be put back into the deck for a second read attempt, but a duplicate should be made and put in its place.

Validity Check When reading in alphanumeric mode, every column is checked to see if the punches correspond to a valid Hollerith character. If they do not, a validity check occurs and the CRB is cleared to 0. If the VALIDITY ON button has been pressed, the NOT READY light goes on and the reader stops. The card in error should be checked for improper punches before a second attempt is made to read it.

The appendix gives a table of Hollerith character codes. Any punch combination which does not appear in this table is invalid.

Programming

There are four flags associated with the card reader. Each of the flags is associated with a bit in the AC. When an iors instruction is executed, the status of the flags is read into these bits.

Card Column This flag signals the presence of information in the CRB. It is sensed by a skip instruction and is connected to the program interrupt.

Card Done As soon as the trailing edge of the card has begun to pass the reading station, this flag is set. It is cleared as soon as the next select instruction is given.

Not Ready Whenever the reader is not available, this flag is set. It corresponds exactly to the NOT READY light on the reader console and is set or cleared by the same operations.

End of File This flag corresponds to the END OF FILE light and button on the reader console. It is set when the EOF button is pressed and the hopper is empty; it is cleared when more cards are placed in the hopper.

Instructions

MNEMONIC SYMBOL	OCTAL CODE	FUNCTION
crsa	706704	Select and read a card in alphanumeric mode. A card is started through the reader and 80 columns are read, interpreted, and translated into 6-bit character codes. If the VALIDITY ON button is lit, a validity check causes the reader to stop.
crsb	706714	Select and read a card in binary mode. A card is started through the reader and 80 columns are read as 12-bit numbers. VALIDITY ON has no effect since validity checking is not performed during this mode.
crrb	706712	Read the card reader buffer. The C(CRB) are placed in bits 6-17 of the AC. The card column flag is cleared.
crsf	706701	Skip if the card column flag is set.

Because a validity error causes the CRB to be cleared, the program can easily detect such errors and take the appropriate action. For example, the number of the column or columns in error can be typed on the printer to help the operator in checking the card.

Timing

When a card is selected, the card done flag is cleared. A minimum time of 83 microseconds elapses before the first column is present in the CRB, at which time the card column flag is set. The program then has 2.3 milliseconds to read the contents of the CRB into the AC. At the end of that time, the information from the next column is present. A column is ready every 2.3 milliseconds until the 80th column is encountered. The card done flag is set 600 to 1200 microseconds after the last column is read. If a select instruction is given within the next 20 microseconds, the reader continues at its maximum reading rate.

OUTPUT RELAY BUFFER (18 BITS) TYPE 140

The Type 140 Relay Buffer consists of an 18-bit relay register and 18 HGS-1009 relays all mounted within a standard 18 x 5-1/4" computer rack. Each relay is rated at 2 amperes for 500 volts and may be used to directly control external operations or transfer sensing signals to and from external equipment.

The status of the 18 relays corresponds to the status of the 18 bits of the PDP-7 accumulator. Two IOT instructions control the contents of the relay buffer. One IOT instruction reads the contents of the accumulator into the reader buffer; the other IOT clears the reader buffer.

The front panel of the 140 Relay Buffer contains 18 indicator lights to display the status of each relay. The panel also contains 4 banana jack plugs for each relay output: a ground, the 1 side of the relay, the 0 side of the relay, and a common point.

CARD PUNCH CONTROL TYPE 40

The card punch control is designed to allow the operation of a device such as the IBM Model 523 Summary Punch. This type of punch requires one select instruction for each card. Once the card is in motion, the 12 rows are punched at fixed intervals. If a select instruction has not been given within a maximum time after the punching of the previous card is completed, the punch shuts itself off.

The card punch control contains an 80-bit punch buffer (CPB) into which information is placed for output. When a row has been punched and the CPB is ready to accept new information, the card row flag is set. This flag is sensed by an iot skip instruction and is connected to the PIC.

Instructions

The four iot instructions associated with the card punch are:

cpse	706442	Select the card punch. This starts a card moving from the hopper to the punch station. Load the card punch buffer. This transmits the contents of the AC to the CPB. Five are required to fill the CPB (see below).
cpsf	706401	Skip if card flag is set. This flag is set when the CPB is ready to accept a new row.
cpcf	706402	Clear the card row flag.
cplb	706406	Load the punch buffer, clear punch flag.

The 80-bit CPB is loaded from the 18-bit AC. Five cplb instructions are required to assemble a complete row. The first four fill up the first 72 bits of the CPB (corresponding to the first 72 columns of the card). The fifth cplb places the contents of bits 10-17 of the AC in the last eight bits of the CPB and clears the card row flag.

AUTOMATIC LINE PRINTER TYPE 647

The Type 647 Line Printer prints lines of text of up to 120 characters at a maximum rate of 300 lines per minute. Printing is performed by solenoid-actuated hammers. The typeface is engraved on the surface of the continuously rotating drum. A 64-character set is provided.

Interface

Information is transferred from computer to printer through the interface, which contains a core buffer in which a line to be printed is assembled character by character. Each character is represented by a 6-bit binary code. When a print cycle is initiated, the core buffer is scanned each time a row on the drum comes up to the print station. As the characters are printed, the corresponding core buffer positions are cleared so that at the completion of the print cycle the buffer is clear and ready for the next line.

Printing

A print cycle is initiated by a command from the program. Depending on the distribution and number of different characters in the line to be printed, a print cycle may take from about 48 to 180 milliseconds, not including vertical spacing of the paper.

Vertical Format Control

Vertical movement of the paper is under control of a punched format tape. Eight program-selectable channels determine the amount of vertical spacing by sensing the punches in the tape. Spacing is performed at the completion of a print cycle, at which time the contents of bits 15-17 of the AC cause one of the eight channels to be selected. The paper and tape then move until a hole in the tape is sensed. The table below shows the increments punched on the standard format tape. The user may also create his own formats for which a special punch is available.

AC Bits 15-17	Tape Channel	Spacing Increment
0	2	Every line
1	3	Every 2nd line
2	4	Every 3rd line
3	5	Every 6th line
4	6	Every 11th line (1/6 page)
5	7	Every 22nd line (1/3 page)
6	8	Every 33rd line (1/2 page)
7	1	Top of next form

Note that spacing is referenced from the top of the form. A space of one line requires 18 milliseconds. Longer skips vary in time; a full-page skip to the top of the next form takes about 610 milliseconds.

Operating Controls and Indicators

With the exception of the main power switch and certain test buttons, all of the operating controls are located on two panels. The main panel is at the left on the front of the printer; the auxiliary panel is at the rear on the same side of the machine.

ON, OFF: These buttons control primary power to the functioning parts of the printer. The main power switch must be turned on for these buttons to function. The rest of the controls operate only after ON has been pressed.

START: Places the printer on-line; it is then ready to receive information and print it.

STOP: Takes the printer off-line as soon as the buffer is clear. If there is information in the buffer, the printer remains on line until after the next clear buffer instruction or the completion of the next print cycle. When the printer goes off line, an alarm signal is sent to the computer.

TEST PRINT: This button is for checking purposes; it is not used in normal operation.

TOP OF FORM: Moves the paper to the top of the next page. This button works only when the printer is off line.

TRACTOR INDEX: Used for aligning the forms with the format tape when new paper is loaded. This button works only when the printer is off line.

PAPER LOW ALERT: This indicator lights red when the end of the paper is about to pass through the drag devices below the printer yoke. An alarm signal is sent to the computer at the same time.

NO PAPER: When the end of the paper has passed out of the forms tractors, this indicator lights red, and an alarm signal is sent to the computer.

YOKE OPEN: When the printer yoke is open, this indicator lights red. An interlock prevents all but the TOP OF FORM and TRACTOR INDEX controls from operating.

ALARM STATUS: Whenever an alarm signal is generated, this indicator lights red.

In addition to the above ways, an alarm can be generated by a failure in any part of the printer; such a failure automatically takes the printer off line.

Programming

A line to be printed is assembled in the printer buffer character by character from left to right. When the line is complete, a program command initiates the print cycle. When the cycle is finished, the paper may or may not be spaced vertically. Suppressing vertical movement makes underscoring and overbarring possible. When spacing is performed, the printer buffer becomes available 6 to 8 milliseconds before the paper comes to a stop. The program may begin assembling the next line during this time.

Three loading instructions allow the program to transfer one, two, or three characters at a time from the AC to the printer buffer. If more than one character is transferred, the leftmost one is taken first.

The buffer loading instructions perform the Inclusive OR of the contents of the AC and the current positions of the printer buffer. Thus, the buffer must be clear before a new line is loaded. Clearing is done automatically during the print cycle, and an instruction is provided for initializing the interface and clearing the buffer before starting to print.

The capacity of the printer buffer is 120 characters. The program must keep track of the number of characters transferred; if more than 120 are sent, the extra codes are ignored.

Two flags are associated with the Type 647. The buffer flag is set when the buffer is cleared; this occurs at the end of the print cycle or as the result of a clear instruction. The error flag is set when an alarm signal is sent and can be reset only when the alarm condition is removed. Both flags are connected to the program interrupt control.

The following instructions are added with the Type 647 Automatic Line Printer:

MNEMONIC SYMBOL	OCTAL CODE	OPERATION
lpb-1	706504	Load printer buffer with 1 character. The contents of AC_{12-17} are transferred to the buffer. A minimum of 10 μsec must elapse before the next load instruction may be given.
lpb-2	706524	Load printer buffer with 2 characters. The two character codes represented by the middle (bits 6-11) and right (bits 12-17) portions of the AC are transferred in that order to the buffer. A minimum of 20 μsec must elapse before the next load instruction.
lpb-3	706544	Load printer buffer with 3 characters. The character codes in the left, middle, and right portions of the AC are transferred to the buffer in that order. A minimum of 30 μsec must elapse before the next load instruction.
cpb	706502	Clear printer buffer. The contents of the buffer are set to 0. The buffer flag is set on completion of this operation.
pri	706604	Print. The contents of the printer buffer are printed, but the paper is not moved. The next line of characters will be printed in the same space. This makes underlining and overbarring possible. The buffer flag is set when this operation is complete.
pas	706624	Print and space. The contents of the buffer are printed, and the paper is spaced vertically. The spacing increment is specified by $C(AC_{15-17})$. The buffer flag is set 6 to 8 microseconds before the spacing is completed.
cbf	706602	Clear buffer flag.
sbf	706501	Skip if buffer flag is set.
sef	706601	Skip if error flag is set.
-	706522	Spare.

The status of the buffer and error flags is read into AC bits 15 and 16, respectively, by the iors instruction.

APPENDIX 1

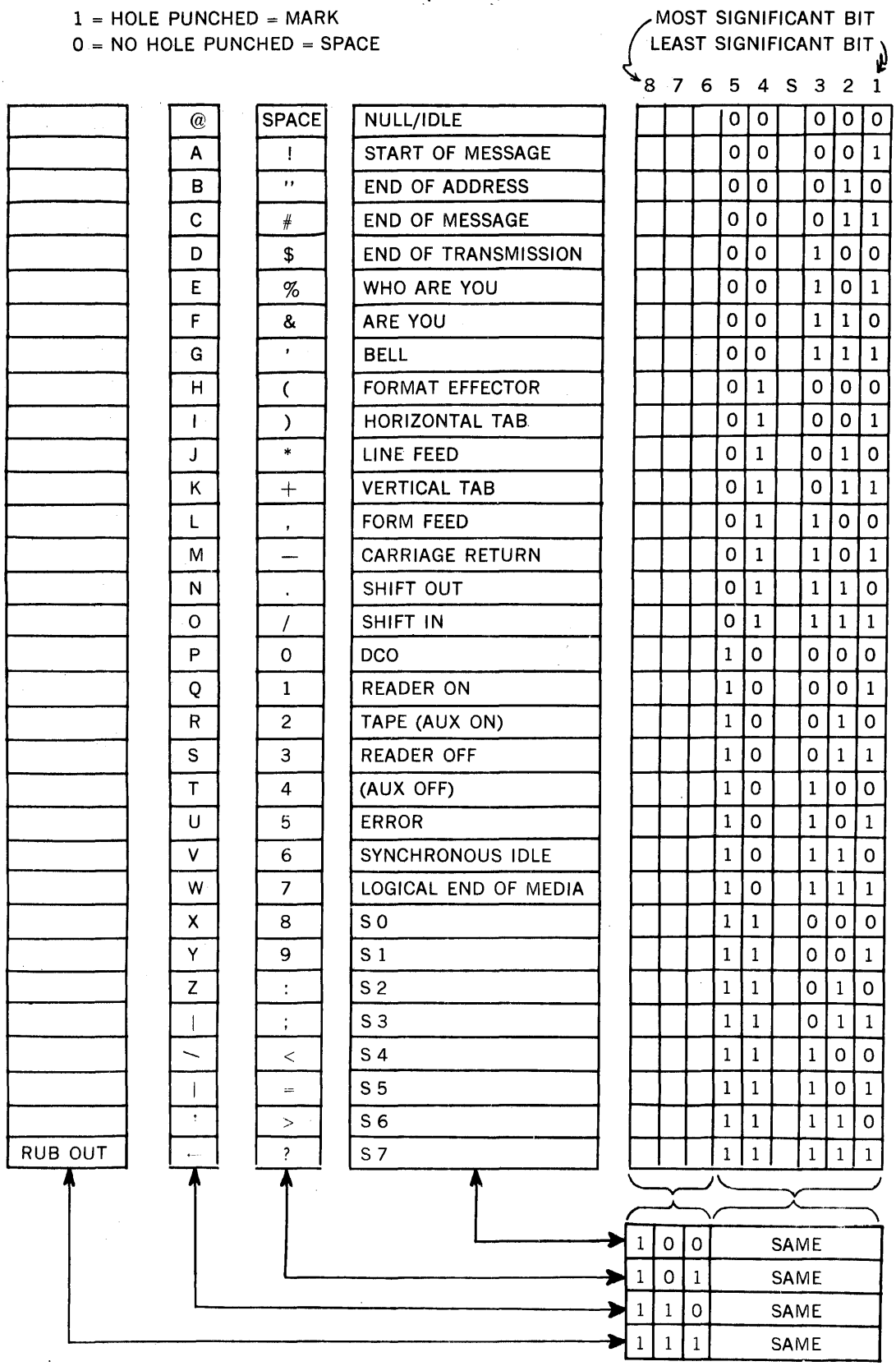
TELEPRINTER CODES

	FIO-DEC	Baudot (28 KSR)	ASCII (33 KSR)
	0-9 a-z A-Z	0-9 A-Z $A-$Z	0-9 A-Z A-Z
(period)	.	.	.
	,	,	,
(minus sign)	-	-	-
	/	/	/
(center dot, period)	:	:	:
(center dot, comma)	;	;	;
	(	(	(
	)	)	)
	+	&	+
	↑	!	↑
(multiply)	x	#	*
	"	$"	"
	'	$'	'
	=	$:	=
	⌜	$(	⌜
	⌟	$)	⌟
	<	$-	<
	>	$&	>
	)	$?	←
	∪	$,	%
	∨	$/	!
	∧	$#	&
	→	$;	@
(vertical stroke)	|	$!	\
(underbar)	_	"	$
(center dot)	·	$.	n.e.
(overbar)	¯	'	#
	Stop Code ↓ Tab	n.e. ↓ ↓ bell	Form Feed ↓ ↓ Tab

APPENDIX 2

TELETYPE EIGHT-LEVEL CODE
(ASCII)

1 = HOLE PUNCHED = MARK
0 = NO HOLE PUNCHED = SPACE



APPENDIX 3

CARD READER; HOLLERITH CODE

A	61
B	62
C	63
D	64
E	65
F	66
G	67
H	70
I	71
J	41
K	42
L	43
M	44
N	45
O	46
P	47
Q	50
R	51
S	22
T	23
U	24
V	25
W	26
X	27
Y	30
Z	31
0	12
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	10
9	11
+	60
—	40
/	21
=	13
,	33
$	53
.	73
'	14
(	34
*	54
)	74
blank	00

Low order bits	High order bits 00	01	10	11
0000		blank	—	+ [&]
0001	1	/	J	A
0010	2	S	K	B
0011	3	T	L	C
0100	4	U	M	D
0101	5	V	N	E
0110	6	W	O	F
0111	7	X	P	G
1000	8	Y	Q	H
1001	9	Z	R	I
1010	0			
1011	= [#]	,	$	.
1100	' [@]	([%]	*	) [□]

HOLLERITH CARD CODE

digit	Zone: no zone	12	11	0
no punch	blank	+ [&]	—	0
1	1	A	J	/
2	2	B	K	S
3	3	C	L	T
4	4	D	M	U
5	5	E	N	V
6	6	F	O	W
7	7	G	P	X
8	8	H	Q	Y
9	9	I	R	Z
8-3	= [#]	.	$	,
8-4	' [@]	) [□]	*	([%]

APPENDIX 4

LINE PRINTER CODE

A	61
B	62
C	63
D	64
E	65
F	66
G	67
H	70
I	71
J	41
K	42
L	43
M	44
N	45
O	46
P	47
Q	50
R	51
S	22
T	23
U	24
V	25
W	26
X	27
Y	30
Z	31
0	20
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	10
9	11
°	40
/	21
′	12
⌣	13
⊃	14
∨	15
∧	16
<	17
$	52
=	53
−	54
)	55
(	57
—	56

space	00
—	60
″	32
,	33
>	34
↑	35
→	36
?	37
×	72
.	73
+	74
]	75
[	77
|	76

Low order bits	High order bits 00	01	10	11
0000	space	0	°	—
0001	1	/	J	A
0010	2	S	K	B
0011	3	T	L	C
0100	4	U	M	D
0101	5	V	N	E
0110	6	W	O	F
0111	7	X	P	G
1000	8	Y	Q	H
1001	9	Z	R	I
1010	′	″	$	×
1011	⌣	,	=	.
1100	⊃	>	−	+
1101	∨	↑	)	]
1110	∧	→	—	|
1111	<	?	(	[

APPENDIX 5

DIGITAL'S SERVICE PRACTICE

Digital's reputation for reliability owes a great deal to rigid quality control and customer field service. Before delivery all Digital products are thoroughly tested by trained check-out teams. Each module and every piece of accessory equipment is subjected to rigorous tests, many of them conducted by specially designed, automatic check-out devices. Computers and special systems are checked electrically and logically by numerous programmed routines.

During system checkout, customers are invited to visit the Maynard manufacturing facility to inspect and become familiar with the equipment. Computer customers may also send personnel to instruction courses in computer operation and maintenance at the Maynard headquarters.

Digital's engineers are available during installation and test for assistance or consultation. Further technical assistance in the field is provided by home office design engineers or branch office application engineers in New York, Washington, Pittsburgh, Chicago, Huntsville, Los Angeles, San Francisco, Ottawa, Sydney (Australia), Reading (U.K.), and Munich.

APPENDIX 6

RIM LOADER

To load the RIM loader, place the RIM loader tape in the reader, set the ADDRESS switches to 17763, and press the READ-IN switch.

The RIM loader contains the following program:

Location	Octal Code		Mnemonic	Remarks
17762/	0	r,	0	/read one binary word
17763/	700101		rsf	
17764/	617763		jmp .-1	/wait for word to come in
17765/	700112		rrb	/read buffer
17766/	700144		rsb	/read another word
17767/	637762		jmp i r	/exit subroutine
17770/	700144	go,	rsb	/enter here, start reader going
17771/	117762	g,	jms r	/get next binary word
17772/	057775		dac out	
17773/	417775		xct out	/execute control word
17774/	117762		jms r	/get data word
17775/	0	out,	0	/store data word
17776/	617771		jmp g	/continue

To load system tapes and other normal binary tapes, place the tape in the reader, set the ADDRESS switches to 17770, and press START.

APPENDIX 7

INSTRUCTION SUMMARY

MEMORY REFERENCE INSTRUCTIONS

MNEMONIC	OCTAL	MACHINE CYCLES*	OPERATION
cal Y	00	2	Call subroutine. Y is ignored jms 20 if bit 4 = 0, jms i 20 if bit 4 = 1.
dac Y	04	2	Deposit AC. C(AC) = > C(Y)
jms Y	10	2	Jump to subroutine. C(PC) = > $C(Y_{5-17})$, C(L) => $C(Y_0)$, Y + 1 = > C(PC)
dzm Y	14	2	Deposit zero in memory. 0 = > C(Y)
lac Y	20	2	Load AC. C(Y) = > C(AC)
xor Y	24	2	Exclusive OR. C(AC) V C(Y) = > C(AC)
add Y	30	2	Add (1's complement). C(AC) + C(Y) = > C(AC)
tad Y	34	2	2's complement add. C(AC) + C(Y) = > C(AC)
xct Y	40	1+	Execute.
isz Y	44	2	Index and skip if 0. C(Y) + 1 = > C(Y), if C(Y) + 1 = 0, then C(PC) + 1 = > C(PC)
and Y	50	2	AND. C(AC) $\wedge$ C(Y) = > C(AC)
sad Y	54	2	Skip if AC and Y differ. If C(AC) $\neq$ C(Y), then C(PC) + 1 = > C(PC)
jmp Y	60	1	Jump. Y = > C(PC)
law N	76	1	Load AC with law N. 1 = > $C(AC_{0-4})$, N = > $C(AC_{5-17})$

* 1 machine cycle = 1.75 μsec.

OPERATE INSTRUCTIONS

MNEMONIC	OCTAL	EVENT TIME	OPERATION
opr	740000	-	Operate.
nop	740000	-	No operation.
cma	740001	3	Complement, $C(\overline{AC}) => C(AC)$
cml	740002	3	Complement link, $C(\overline{L}) => C(L)$
oas	740004	3	Inclusive OR AC switches. C(ACS) V C(AC) => C(AC)
las	750004	2,3	Load AC from switches. C(ACS) => C(AC)
ral	740010	3	Rotate AC + link left one place. $C(AC_i) => C(AC_{i-1})$, C(L) => $C(AC_{17})$, $C(AC_0)$ => C(L)
rcl	744010	2,3	Clear link, then ral. 0 => C(L), then ral.
rtl	742010	2,3	Rotate AC left twice. Same as two ral instructions.
rar	740020	2	Rotate AC + link right one place. $C(AC_i) => C(AC_{i+1})$, C(L) => $C(AC_0)$, $C(AC_{17})$ => C(L)
rcr	744020	2,3	Clear link, then rar. 0 => C(L), then rar.
rtr	742020	2,3	Rotate AC right twice. Same as two rar instructions.
hlt	740040	4	Halt. 0 => RUN.
sza	740200	1	Skip on zero AC. Skip if C(AC) = positive zero.
sna	741200	1	Skip on non-zero AC. Skip if C(AC) ≠ positive zero.
spa	741100	1	Skip on positive AC. Skip if $C(AC_0) = 0$.

OPERATE INSTRUCTIONS (continued)

MNEMONIC	OCTAL	EVENT TIME	OPERATION
sma	740100	1	Skip on negative AC. Skip if $C(AC_0) = 1$.
szl	741400	1	Skip on zero link. Skip if C(L) = 0.
snl	740400	1	Skip on non-zero link. Skip if C(L) = 1.
skp	471000	1	Skip, unconditional. Always skip.
cll	744000	2	Clear link. 0 => C(L).
stl	744002	2,3	Set the link. 1 => L.
cla	750000	2	Clear AC. 0 => C(AC).
clc	750001	2,3	Clear and complement AC. -0 => C(AC).
glk	750020	2,3	Get link. 0 => C(AC), C(L) => $C(AC_{17})$.

EAE INSTRUCTIONS

MNEMONIC	OCTAL	OPERATION
eae	640000	Basic EAE command - no operation.
lrs	640500	Long right shift.
lrss	660500	Long right shift, signed.
lls	640600	Long left shift.
llss	660600	Long left shift, signed.
als	640700	Accumulator left shift.
alss	660700	Accumulator left shift, signed.
norm	640444	Normalize: max. shift is 44.
norms	660444	Normalize, signed.

EAE INSTRUCTIONS (continued)

MNEMONIC	OCTAL	OPERATION
mul	653122	Multiply unsigned.
muls	657122	Multiply signed.
div	640323	Divide C(AC and MQ) as a 36-bit unsigned number.
divs	644323	Divide C(AC and MQ) as a 34-bit 1's complement signed number.
idiv	653323	Integer divide unsigned.
idivs	657323	Integer divide, signed.
frdiv	650323	Fraction divide unsigned.
frdivs	654323	Fraction divide, signed.
lacq	641002	Replace the C(AC) with the C(MQ).
lacs	641001	Replace the C(AC) with the C(SC).
clq	650000	Clear MQ.
abs	644000	Place absolute value of AC in the AC.
gsm	664000	Place AC sign in link and take absolute value of AC.
osc	640001	Inclusive OR the SC into the AC.
omq	640002	Inclusive OR AC with MQ and place results in AC.
cmq	640004	Complement the MQ.

PRIORITY INTERRUPT INSTRUCTIONS

MNEMONIC	OCTAL	OPERATION
cac	705501	Clear and reset all channels.
asc	705502	Enable selected channel(s).
dsc	705604	Disable selected channel(s).
epi	700004	Enable automatic priority interrupt system.
dpi	700044	Disable automatic priority interrupt system.
isc	705504	Initiate break on selected channel (for maintenance purposes).
dbr	705601	Debreak....Returns highest priority channel to receptive state.

IOT INSTRUCTIONS

MNEMONIC	OCTAL	OPERATION
		Program Interrupt
iof	700002	Turn off interrupt.
ion	700042	Turn on interrupt.
iton	700062	Turn on trap, also turns on program interrupt.
		I/O Equipment
iors	700314	Read status of I/O equipment.
		Clock
clsf	700001	Skip if clock flag is 1.
clof	700004	Turn off clock, clear clock flag.

IOT INSTRUCTIONS (continued)

MNEMONIC	OCTAL	OPERATION
		Clock (continued)
clon	700044	Turn on clock, clear clock flag.
		Paper Tape Reader
rsa	700104	Select reader for alphanumeric, clear reader flag.
rsb	700144	Select reader for binary, clear reader flag.
rsf	700101	Skip if reader flag is a 1.
rrb	700112	Read the reader buffer into AC, clear reader flag.
rcf	700102	Clear reader flag then inclusively OR reader buffer into AC.
		Paper Tape Punch
psa	700204	Punch a line of tape in alphanumeric mode. The punch flag is immediately cleared and then set when punching is complete.
psb	700244	Punch a line of tape in binary mode. The punch flag is immediately cleared and then set when punching is complete.
psf	700201	Skip the following instruction if the punch flag is set.
pcf	700202	Clear the punch flag.

IOT INSTRUCTION (continued)

MNEMONIC	OCTAL	OPERATION
		Keyboard Input from Teleprinter
ksf	700301	Skip if keyboard flag is a 1.
krb	700312	Read the keyboard buffer into the AC, clear keyboard flag.
		Teleprinter
tsf	700401	Skip if teleprinter flag is a 1.
tls	700406	Load teleprinter buffer and select, clear teleprinter flag.
tcf	700402	Clear the teleprinter flag.
		DECtape 551
mmrd	707512	Read. Transfers one word from mmiob to the AC.
mmwr	707504	Write. Transfers one word from the AC to mmiob.
mmse	707644	Select. Connects the unit designated to the DECtape control.
mmlc	707604	Load control. Sets the DECtape control to the proper mode and direction.
mmrs	707612	Read status. Reads the DECtape status conditions into bits 0-8 of the AC.
mmdf	707501	Skip on DECtape data flag.
mmbf	707601	Skip on DECtape block end flag.
mmef	707541	Skip on DECtape error flag.

MNEMONIC	OCTAL	OPERATION
		Tape Control 57A
mscr	707001	Skip if the tape control ready (TCR) level is 1.
msur	707101	Skip if the tape transport is ready (TTR).
mccw	707401	Clear CA and WC.
mlca	707405	Transfer AC bits 5-17 to CA and clear CA and WC.
mlwc	707402	Transfer AC bits 5-17 to WC.
mrca	707414	Transfer CA bits 5-17 to AC bits 5-17.
mdcc	707042	Disable the TCR flag from the program interrupt and clear command register.
mctu	707006	Disable the TCR flag from the program interrupt, turn off the WCO flag and EOR flag, and select the unit, the mode of parity, and the density from the AC.
mtcs	707106	Place AC bits 9-12 in the tape control command register and start tape motion.
mncm	707152	Terminate the continuous mode (the AC is cleared).
mrrc	707204	Switch mode from read to read compare.
mrcr	707244	Switch mode from read compare to read.
msef	707301	Skip if the EOR flag is a 1.
mdef	707302	Disable ERF.
mcef	707322	Clear ERF.
meef	707242	Enable ERF.
mief	707362	Initialize ERF, clear and enable.
mswf	707201	Skip if the WCO flag is a 1.

MNEMONIC	OCTAL	OPERATION
		Tape Control 57A (continued)
mdwf	707202	Disable WCO flag.
mcwf	707222	Clear WCO flag.
mewf	707242	Enable WCO flag.
miwf	707262	Initialize WCO flag.
mtrs	707314	Read tape status bits into the AC.
		Display 30D
dxl	700506	Load the x-coordinate buffer from AC_{8-17}.
dxs	700546	Load the x-coordinate buffer and select.
dyl	700606	Load the y-coordinate buffer.
dys	700646	Load the y-coordinate buffer and select.
dxc	700502	Clear the x-coordinate buffer.
dyc	700602	Clear the y-coordinate buffer.
dlb	700706	Load the brightness register from AC_{15-17}.
		Precision Incremental Display Type 340
idla	700606	Load address and select.
idse	700501	Skip on edge, skip on stop code.
idsi	700601	Skip on stop interrupt.
idsp	700701	Skip if light pen flag is set.
idrs	700504	Continue display following light pen interrupt.

MNEMONIC	OCTAL	OPERATION
		Precision Incremental Display Type 340 (continued)
idrd	700614	Restart display following stop code interrupt.
idra	700512	Read display address.
idrc	700712	Read x and y coordinates.
idcf	700704	Clear display control.
		Light Pen 370
dsf	700501	Skip if the display flag is set.
dcf	700502	Clear the display flag.
		Card Reader 421A
crsa	706704	Select card reader for alphanumeric.
crsb	706714	Select card reader for binary.
crrb	706712	Read card column buffer into AC.
crsf	706701	Skip if reader character flag is a 1.
		Card Punch 40
cpsf	706401	Skip if the card punch flag is a 1.
cpse	706444	Select a card, set card punch flag.
cplr	706406	Load row buffer, clear punch flag.
cpcf	706442	Clear punch flag.

IOT INSTRUCTION (continued)

MNEMONIC	OCTAL	OPERATION
		Line Printer 647
lpsf	706501	Skip if printing flag is a 1.
lpcf	706502	Clear printing flag.
lpld	706542	Load the printing buffer.
lpse	706506	Select printing, clear printing flag.
lssf	706601	Skip if spacing flag is a 1.
lscf	706602	Clear spacing flag.
lsls	706606	Load spacing buffer and select spacing, clear spacing flag.

APPENDIX 8

POWERS OF TWO

2^n	n	2^{-n}
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	0.000 000 007 450 580 596 923 828 125
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
2 147 483 648	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	0.000 000 000 029 103 830 456 733 703 613 281 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
1 125 899 906 842 624	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
2 251 799 813 685 248	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	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25
18 014 398 509 481 984	54	0.000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
36 028 797 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5
72 057 594 037 927 936	56	0.000 000 000 000 000 013 877 787 807 814 456 755 295 395 851 135 253 906 25
144 115 188 075 855 872	57	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 925 567 626 953 125
288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
1 152 921 504 606 846 976	60	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
2 305 843 009 213 693 952	61	0.000 000 000 000 000 000 433 680 868 994 201 773 602 981 120 347 976 684 570 312 5
4 611 686 018 427 387 904	62	0.000 000 000 000 000 000 216 840 434 497 100 886 801 490 560 173 988 342 285 156 25
9 223 372 036 854 775 808	63	0.000 000 000 000 000 000 108 420 217 248 550 443 400 745 280 086 994 171 142 578 125
18 446 744 073 709 551 616	64	0.000 000 000 000 000 000 054 210 108 624 275 221 700 372 640 043 497 085 571 289 062 5
36 893 488 147 419 103 232	65	0.000 000 000 000 000 000 027 105 054 312 137 610 850 186 320 021 748 542 785 644 531 25
73 786 976 294 838 206 464	66	0.000 000 000 000 000 000 013 552 527 156 068 805 425 093 160 010 874 271 392 822 265 625
147 573 952 589 676 412 928	67	0.000 000 000 000 000 000 006 776 263 578 034 402 712 546 580 005 437 135 696 411 132 812 5
295 147 905 179 352 825 856	68	0.000 000 000 000 000 000 003 388 131 789 017 201 356 273 290 002 718 567 848 205 566 406 25
590 295 810 358 705 651 712	69	0.000 000 000 000 000 000 001 694 065 894 508 600 678 136 645 001 359 283 924 102 783 203 125
1 180 591 620 717 411 303 424	70	0.000 000 000 000 000 000 000 847 032 947 254 300 339 068 322 500 679 641 962 051 391 601 562 5
2 361 183 241 434 822 606 848	71	0.000 000 000 000 000 000 000 423 516 473 627 150 169 534 161 250 339 820 981 025 695 800 781 25
4 722 366 482 869 645 213 696	72	0.000 000 000 000 000 000 000 211 758 236 813 575 084 767 080 625 169 910 490 512 847 900 390 625

DIGITAL SALES AND SERVICE

MAIN OFFICE AND PLANT

146 Main Street, Maynard, Massachusetts 01754
Telephone: From Metropolitan Boston: 646-8600
Elsewhere: AC617-897-8821
TWX: 710-347-0212 Cable: Digital Mayn. Telex: 092-027

DIGITAL SALES OFFICES

NORTHEAST OFFICE:
146 Main Street, Maynard, Massachusetts 01754
Telephone: AC617-646-8600 TWX: 710-347-0212

NEW YORK OFFICE:
1259 Route 46, Parsippany, New Jersey 07054
Telephone: AC201-335-0710 TWX: 510-235-8319

WASHINGTON OFFICE:
1430 K. Street, NW, Washington, D. C. 20005
Telephone: AC202-628-4262 TWX. 710-822-9435

SOUTHEAST OFFICE:
Suite 21, Holiday Office Center
3322 Memorial Parkway, S.W., Huntsville, Ala. 3580
Telephone AC205-881-7730 TWX: 205-533-1267

ORLANDO OFFICE:
1510 E. Colonial Drive, Orlando, Florida 32803
Telephone: AC305-422-4511

PITTSBURGH OFFICE:
300 Seco Road, Monroeville, Pennsylvania 15146
Telephone: AC412-351-0700 TWX: 412-372-4695

CHICAGO OFFICE:
910 North Busse Highway, Park Ridge, Illinois 60068
Telephone: AC312-825-6626 TWX: 312-823-3572

ANN ARBOR OFFICE:
3853 Research Park Drive, Ann Arbor, Mich. 48104
Telephone: AC313-761-1150 TWX: 810-223-6053

LOS ANGELES OFFICE:
8939 Sepulveda Boulevard, Los Angeles, Calif. 90045
Telephone: AC213-670-0690 TWX: 910-328-6121

SAN FRANCISCO OFFICE:
2450 Hanover, Palo Alto, California 94304
Telephone: AC415-326-5640 TWX: 910-373-1266

IN CANADA:
Digital Equipment of Canada, Ltd.,
150 Rosamund Street, Carleton Place, Ontario, Canada
Telephone: AC613-237-0772 TWX: 610-561-1650

IN EUROPE:
Digital Equipment GmbH, Theresienstrasse 29
Munich 2/West Germany
Telephone: 29 94 07, 29 25 66 Telex: 841-5-24226

Digital Equipment Corporation (UK) Ltd.
11 Castle Street
Reading, Berkshire, England
Telephone: Reading 57231 Telex: 851-84327

IN AUSTRALIA:
Digital Equipment Australia Pty. Ltd.,
89 Berry Street
North Sydney, New South Wales, Australia
Telephone: 92-0919 Telex: 790AA-20740
Cable: Digital, Sydney

DIGITAL SALES REPRESENTATIVES

IN THE SOUTHWEST:

DATRONICS INC.

7800 Westglen Drive, Houston, Texas 77042
Telephone: AC713-782-9851 TWX: 713-571-2154

7078 San Pedro Avenue, Suite 205,
San Antonio, Texas 78216
Telephone: AC512-824-6368 TWX: 512-571-0788

Post Office Box 782, Kenner, Louisiana 70062
Telephone: AC504-721-1410

Post Office Box 13384, Fort Worth, Texas 76118
Telephone: AC817-281-1284 TWX: 817-281-3120

IN THE NORTHWEST:

SHOWALTER-JUDD, INC.
1806 South Bush Place, Seattle, Washington 98144
Telephone: 206-324-7911 TWX: 206-998-0323

IN JAPAN:

RIKEI TRADING CO.,
12, 2-Chome, Shiba Tamura-cho, Minato-ku,
Tokyo, Japan
Telephone: 591-5246 Cable Rikeigood, Tokyo

IN SWEDEN:

TELARE AB
Industrigatan 4, Stockholm K, Sweden
Telephone: 54 33 24 Telex: 854-10178

5140 PRINTED IN U.S.A. 15-11/64