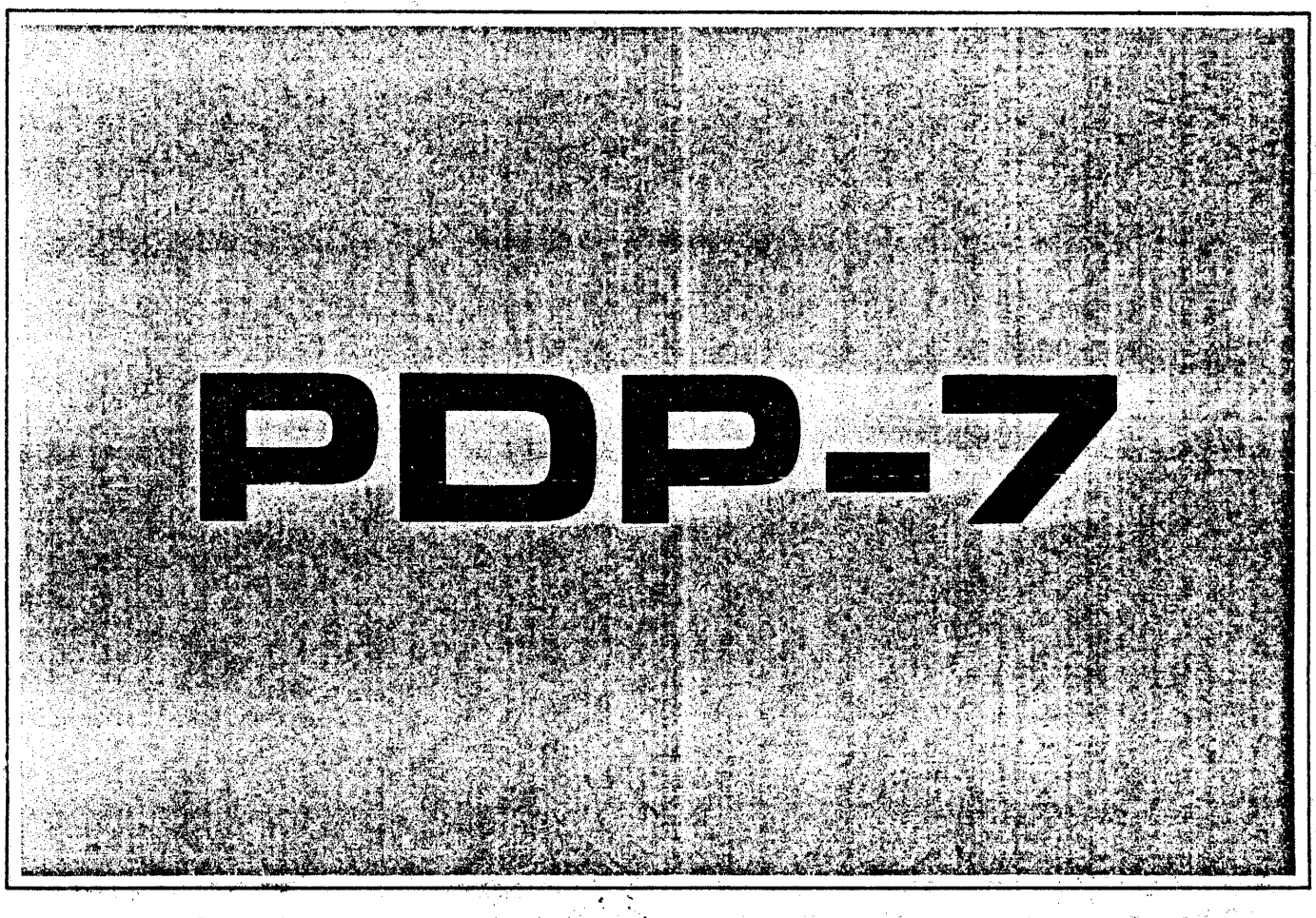
F-75

USERS HANDBOOK



DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

PROGRAMMED DATA PROCESSOR -7 USERS HANDBOOK

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

PREFACE

This handbook concerns programming and operating the Programmed Data Processor-7, a high speed, stored program, digital computer manufactured by the Digital Equipment Corporation. Section 1 presents a summary of the standard computer system and the available options, as well as information on the notation used throughout the handbook. Section 2 presents a brief block-diagram discussion of the standard computer system and its major logic elements. Sections 3 and 4 are devoted to explaining the structure and organization of the instructions, and providing information on the basic use of the equipment by features. Sections 5 throughout 12 describe the standard peripheral equipment and the optional equipment in functional groups. Section 13 serves as a resume of the basic software provided with the hardware system. Section 14 summarizes the operating procedures used with the software and in manual operation of the computer. Appendixes at the end of this handbook provide tables of reference data and detailed information which may be helpful in specific programming assignments. Although program examples are given in this document, no attempt has been made to teach programming techniques. The meaning and use of special characters employed in the programming examples are explained in the description of the PDP-7 Symbolic Assembler program, available from the Digital Program Library.

CONTENTS

Section		Page
1	SYSTEM INTRODUCTION	1
	Computer Organization	4
	Processor	4
	Core Memory	5
	Interface	5
	Input/Output	6
	Processor Options	6
	Core Memory Options	7
	Input/Output Options	8
	Programming System	12
	FORTRAN Compiler	12
	Symbolic Assembler	13
	Digital Debugging Tape (DDT)	13
	Symbolic Tape Editor	13
	Bus-Pak II	13
	Symbols	14
2	FUNCTIONAL DESCRIPTION	15
	Processor	15
	Accumulator (AC)	15
	Link (L)	16
	Memory Address Register (MA)	16
	Program Counter (PC)	16
	Memory Buffer Register (MB)	16
	Instruction Register (IR)	16
	Major State Generator	17
	Detailed Processor Block Diagram Discussion	18
	Core Memory	20
	Interface	21

CONTENTS (continued)

Section		Page
2 (continued)	Device Selector (DS)	23
	Information Distributor (ID)	25
	Information Collector (IC)	25
	Input/Output	26
3	INSTRUCTIONS	27
	Memory Reference Instructions	27
	Augmented Instructions	31
	Input/Output Transfer Instructions	31
	Operate Instructions	32
4	BASIC MACHINE LANGUAGE PROGRAMMING	39
	Memory Addressing	39
	Indirect Addressing	39
	Auto-Indexing	40
	Arithmetic Operations	42
	Complement Arithmetic	42
	Addition	43
	Subtraction	44
	Multiplication and Division	44
	Input/Output Fundamentals	44
	Program Flags	44
	Input/Output Status	45
	Input/Output Skip Facility (I/OS)	45
	Input/Output Trap	46
	Program Interrupt Control (PIC)	47
	Real Time Clock	49
	Data Break Channel	50

CONTENTS (continued)

Section		Page
5	PROCESSOR OPTIONS	52
	Extended Arithmetic Element Type 177	52
	EAE Microprogramming	58
	EAE Programming Examples	61
	Automatic Priority Interrupt Type 172	64
	The Multi-Instruction Subroutine Mode	65
	The Single Instruction Subroutine Mode	66
	Priority Interrupt Instructions	67
	Data Interrupt Multiplexer Control Type 173	68
6	CORE MEMORY OPTIONS	71
	Memory Extension Control Type 148	71
	Core Memory Modules Type 147 and 149	73
7	STANDARD INPUT/OUTPUT EQUIPMENT	75
	Teletype Model 33 KSR and Control Type 649	75
	Keyboard	75
	Teleprinter	76
	Perforated Tape Reader and Control Type 444B	76
	Perforated Tape Punch Type 75D	78
8	CARD EQUIPMENT AND LINE PRINTER OPTIONS	80
	Card Reader and Control Type CR01B	80
	Card Reader Operation	80
	Card Reader and Control Type 421	82
	Card Reader Operation	82
	Programming	85
	Card Punch Control Type 40	87
	Automatic Line Printer Type 647	88

CONTENTS (continued)

Section		Page
8 (continued)	Interface	88
	Printing	88
	Vertical Format Control	88
	Operating Controls and Indicators	89
	Programming	90
9	MAGNETIC TAPE AND DRUM OPTIONS	92
	DECtape 550/555 System	92
	DECtape Dual Transport Type 555	92
	DECtape Control Type 550	94
	DECtape Programming	94
	Automatic Magnetic Tape Control Type 57A	97
	Magnetic Tape Transport Type 570	101
	Magnetic Tape Transport Type 545	101
	Specifications	102
	Magnetic Tape Transport Type 50	102
	Serial Drum Type 24	102
10	PLOTTER AND DISPLAY OPTIONS	106
	Incremental Plotter and Control Type 350	106
	Oscilloscope Display Type 34A	107
	Precision CRT Display Type 30D	107
	Symbol Generator Type 33	109
	Precision Incremental Display Type 340	110
	Incremental Display Options	112
	Photomultiplier Light Pen Type 370	113
11	ANALOG/DIGITAL CONVERSION OPTIONS	114
	General Purpose Analog-to-Digital Converter Type 138E	114

CONTENTS (continued)

Section		Page
11 (continued)	Converter Specifications	115
	High Speed Analog-to-Digital Converter Type 142	115
	Multiplexer Control Type 139E	117
	Multiplexer Specifications	117
12	DATA AND COMMUNICATION EQUIPMENT OPTIONS	119
	Data Control Type 174	119
	Data Communication System Type 630	121
	Eight-Channel DCS	122
	Relay Buffer Type 140	123
	Inter Processor Buffer Type 195	123
	Programming	125
13	PROGRAMMING SYSTEM	127
	Symbolic Assembler	127
	Source Language	128
	Source Language Tapes	131
	Digital Debugging Tape (DDT)	131
	Symbolic Tape Editor	133
	Operating Modes	133
	FORTRAN II	135
	BUS-PAK II	136
14	OPERATING PROCEDURES	141
	Controls and Indicators	141
	Manual Data Storage and Modification	147
	Storing the RIM Loader	147
	Loading Data Under Program Control	149
	Checking and Modifying a Stored Program	149

CONTENTS (continued)

Section		Page
14 (continued)	FORTRAN Operating Procedures	149
	Procedure for Using FORTRAN With a PDP-7 Paper Tape System	150
	Diagnostics	153
	FORTRAN Assembler Error Messages	156
	Data Organization	158
	The FORTRAN Assembly System	158
	Operating the PDP-7 Assembler (Basic or Extended)	159
	Operating Instructions	159
	Loading a Symbol Punch	160
	Halts During Assembly	162
	Error Messages	163
	Summary of Symbolic Tape Editor Operations	165
	Special Key Functions	166
	Digital Debugging Tape (DDT)	167

Appendix		
1	PDP-7 PROGRAM LIBRARY	171
2	CODES	173
3	SCALES OF NOTATION	179
4	INSTRUCTION SUMMARY	188

ILLUSTRATIONS

Figure		
1	A Basic PDP-7	xiv
2	Basic PDP-7 Component Locations	2
3	Basic PDP-7 Installation Dimensions	3
4	Computer System Block Diagram	4
5	Major Register Block Diagram of the Processor	15
6	Detailed Block Diagram of the Processor	18

ILLUSTRATIONS (continued)

Figure		Page
7	Core Memory Block Diagram	21
8	Interface Block Diagram	22
9	Input/Output Transfer Timing Diagram	22
10	Device Selector Logic Diagram	24
11	Input/Output Information Flow	26
12	Memory Reference Instruction Bit Assignments	27
13	I/OT Instruction Bit Assignments	32
14	Group 1 Operate Instruction Bit Assignments	33
15	Group 2 (LAW) Operate Instruction Bit Assignments	37
16	I/ORS Instruction Status Bit Assignments	45
17	Information Stored in Address 000000 During a Program Interrupt	48
18	EAE Instruction Bit Assignment	52
19	EAE Setup Instruction Bit Assignments	56
20	EAE Multiply Instruction Bit Assignments	56
21	EAE Divide Instruction Bit Assignments	57
22	EAE Normalize Instruction Bit Assignments	57
23	EAE Shift Instruction Bit Assignments	57
24	EAE Example 1, Problem	61
25	EAE Example 1, Approach	61
26	EAE Example 2, Approach	62
27	EAE Example 3, Approach Algorithm	63
28	Data Interrupt Multiplexer Type 173	70
29	Tape Format and Reader Buffer Register Bit Assignments in Alphanumeric Mode	77
30	Tape Format and Reader Buffer Register Bit Assignments in Binary Mode	77
31	Type 421 Card Reader Console	83
32	Card Reader Control Panel	83
33	DECtape Format	93
34	Serial Drum Block Diagram and Interface Connections	103

ILLUSTRATIONS (continued)

Figure		Page
35	Type 142 A-to-D Converter, Block Diagram	116
36	Operator Console	141
37	Indicator Panel	145
38	Assembler Flow Diagram	161

TABLES

Table		
1	Memory Reference Instructions	28
2	Operate Instructions	33
3	Auto-Index Registers in each Memory Field	40
4	EAE Bit Assignments and Operations	53
5	EAE Instruction List	59
6	Priority Interrupt Instructions	67
7	Tape Reader Instructions	78
8	Tape Punch Instructions	79
9	Card Reader CR01B Controls and Indicators	80
10	Card Reader CR01B Instructions	81
11	Card Reader 421 Controls and Indicators	84
12	Card Reader 421 Instructions	86
13	Card Punch Instructions	87
14	Line Printer Controls and Indicators	89
15	Automatic Line Printer Instructions	91
16	DECtape Instructions	95
17	Transports and Interfaces used with Tape Control 57A	97
18	Automatic Magnetic Tape Control Basic Instructions	99
19	Automatic Magnetic Tape Control Flag Instructions	100
20	Serial Drum Instructions	104
21	Incremental Plotter and Control Instructions	106

TABLES (continued)

Table		Page
22	Oscilloscope and Precision Display Instructions	108
23	Symbol Generator Instructions	109
24	Precision Incremental Display Instructions	111
25	General Purpose A-to-D Converter Characteristics	114
26	Data Control instructions	120
27	Inter Processor Buffer Instructions	124
28	Operator Console Controls and Indicators	142
29	Indicator Panel Functions	146
30	RIM Loader (8K)	148
31	FORTRAN Diagnostic Error Printouts	153
32	Definition of a Physical Record for I/O Devices	158
33	Accumulator Switch Settings	165
34	Editor Command Summary	166
35	Summary of DDT Commands	167

Figure 1 A Basic PDP-7

SECTION 1

SYSTEM INTRODUCTION

The Digital Equipment Corporation (DEC) 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 notation to facilitate multiprecision operations. Cycle time of the 4096-word random-access magnetic-core memory is 1.75 microseconds, providing a computation rate of 285,000 additions per second.

The basic PDP-7 includes the processor (with operator 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.

Interface to the PDP-7 allows fast parallel information transfer between the computer and 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 magnetic drum storage, 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 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 all required circuit operating dc voltages. Total power consumption is 2200 watts. Built-in provisions for marginal checking allow the +10 and -15 volt logic power supplied to logic circuits to be varied, thereby providing a powerful maintenance tool for forestalling failure of the system or for rapid troubleshooting. The computer is constructed with standard DEC FLIP CHIP™ modules and power supplies. These solid-state components and built-in marginal checking facilities insure reliable machine operation.

The basic PDP-7 is housed in three metal DEC computer cabinets bolted together to form an integrated console. Double doors at the front allow access to the wiring side of all module-mounting panels. Double rear doors provide access to a plenum door on which the power supplies are mounted. Opening the plenum door yields access to the modules. Logical component locations are shown on Figure 2 and dimensions are indicated on Figure 3. For additional physical data refer to the PDP-7 Installation Manual F-78 or the PDP-7 Maintenance Manual F-77.

™ FLIP CHIP is a trademark of the Digital Equipment Corporation

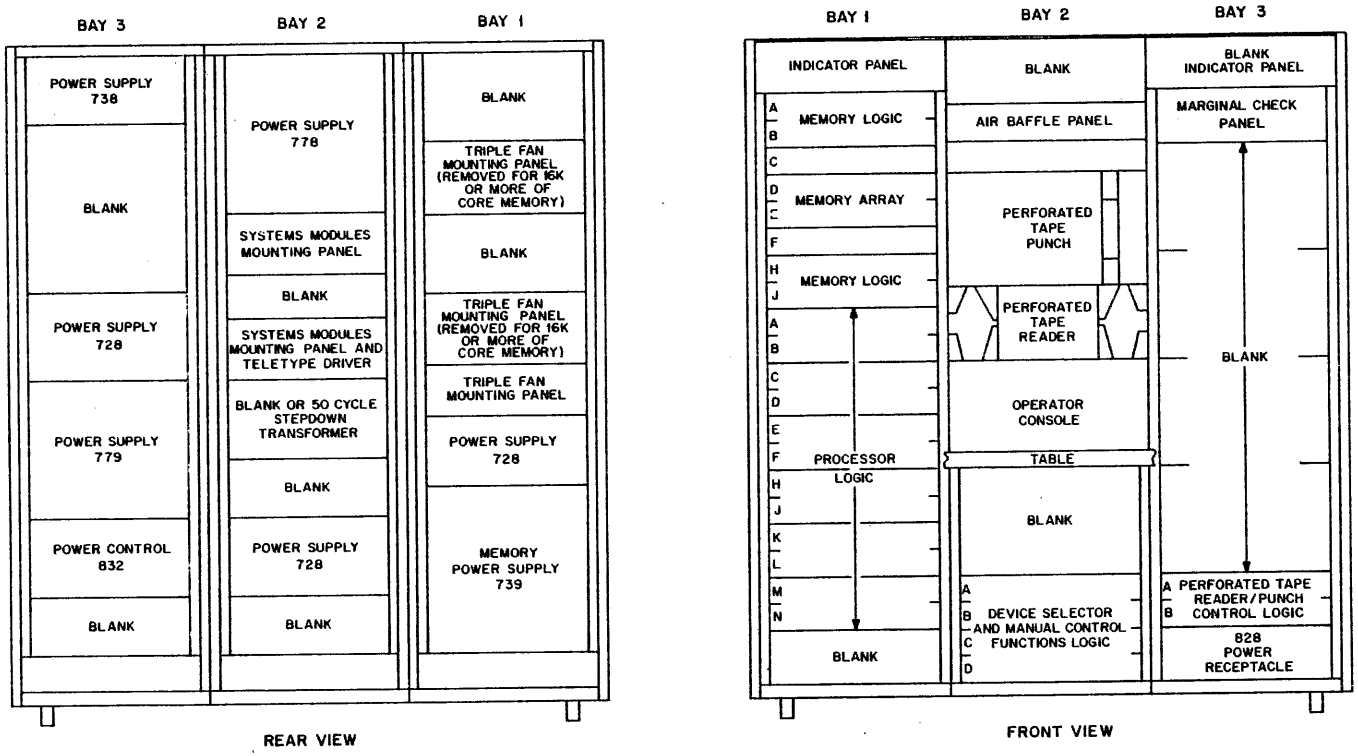


Figure 2 Basic PDP-7 Component Locations

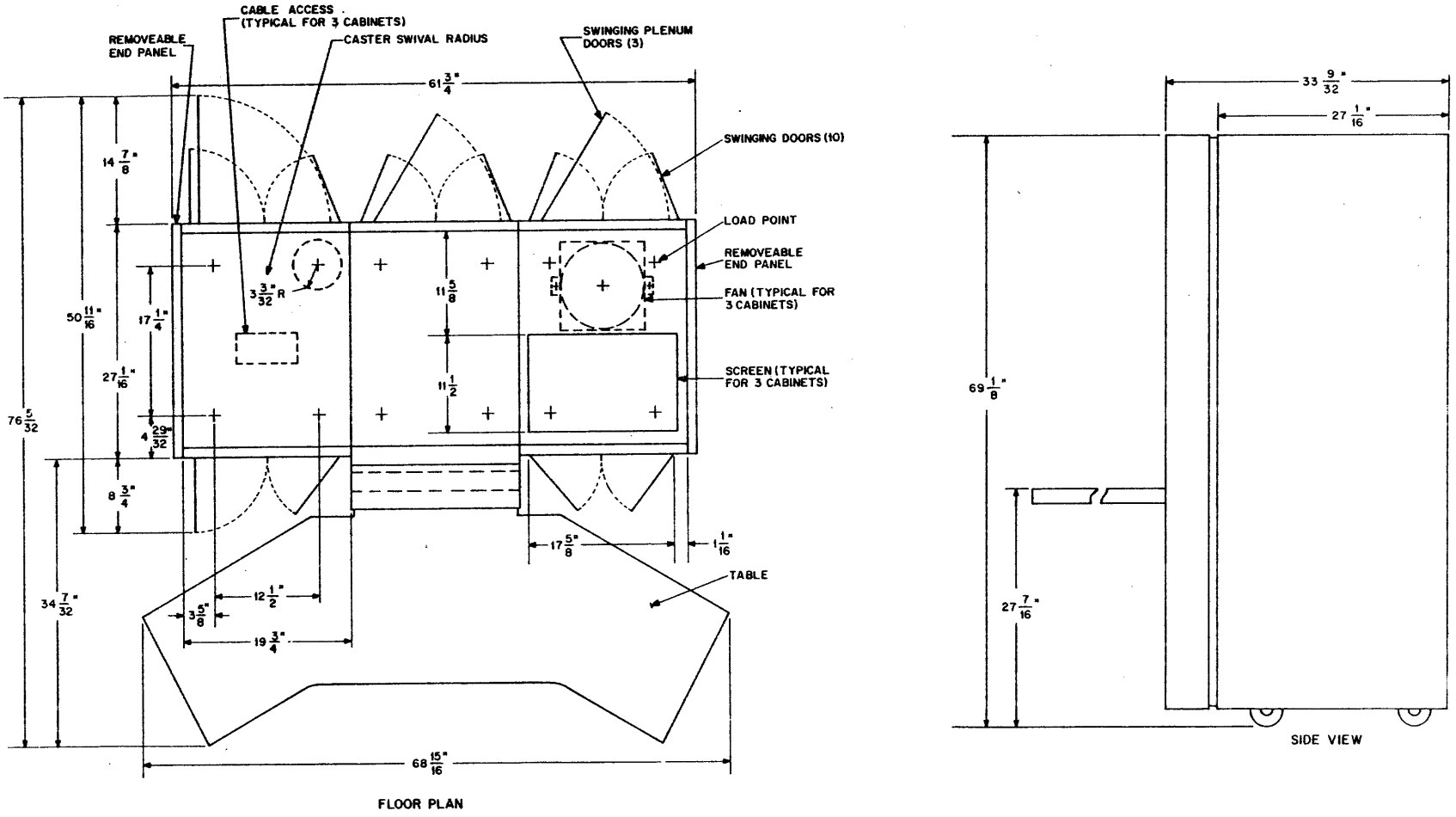


Figure 3 Basic PDP-7 Installation Dimensions

COMPUTER ORGANIZATION

The PDP-7 system is organized into a processor, core memory, interface, and input/output equipment and facilities as shown in Figure 4. All arithmetic, logic, and system control operations of the standard PDP-7 are performed by the processor. Permanent (longer than one instruction time) local information storage and retrieval operations are performed by the core memory. The memory is continuously cycling, automatically performing a read and write operation during each computer cycle. Input and output address and data buffering for the core memory is performed by registers of the processor, and operation of the memory is under control of timing signals produced by the processor.

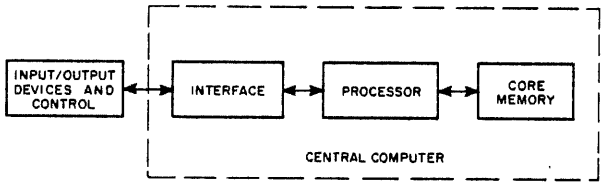


Figure 4 Computer System Block Diagram

Interface circuits allow connections to a variety of peripheral equipment, are responsible for detecting all I/O select codes, and for providing any necessary input or output gating. Individually programmed data transfers between the processor and peripheral equipment take place through the processor accumulator. Single or multiple data transfers can be initiated by peripheral equipment rather than by the program, by means of the data break facilities. Standard features of the PDP-7 also allow peripheral equipment to perform certain control functions such as instruction skipping, and transfer of program control initiated by a program interrupt.

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, and six active major registers.

Accumulator (AC)

This 18-bit register 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.

Link (L)

This 1-bit register extends the arithmetic facility of the accumulator and greatly simplifies programming of arithmetic operations.

Memory Address Register (MA)

This 13-bit register holds the address of the core memory location currently being used.

Memory Buffer Register (MB)

This 18-bit register serves as a buffer for all information sent to or received from core memory.

Instruction Register (IR)

This 4-bit register holds the operation code of the program instruction currently being performed.

Program Counter (PC)

This 13-bit register holds the address of the next memory location from which an instruction is to be taken.

Core Memory

The high-speed random-access core 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 location 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 address.

Interface

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

No additional interface equipment is required to connect standard DEC peripheral equipment to the standard PDP-7. Word buffers are included within each standard I/O optional equipment so that the basic PDP-7 can simultaneously operate many I/O devices at their maximum rates. 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

Standard input/output equipment provided with each PDP-7 consists of a Teleprinter and Control Type 649, Perforated Tape Reader and Control Type 444B, and a Perforated Tape Punch and Control Type 75D.

Teleprinter and Control Type 649

A Teletype Model 33 Keyboard Send Receive (KSR) set and an appropriate DEC control constitute this equipment. The Teletype unit is a standard machine operating from serial 11-unit-code characters at a rate of ten characters per second. The Teletype provides a means of supplying data to the computer by means of a keyboard and supplies data as an output from the computer in the form of typed copy. The Teletype control serves as a serial-to-parallel converter for Teletype inputs to the computer and serves as a parallel-to-serial converter for computer output signals to the Teletype unit.

Perforated Tape Reader and Control Type 444B

This equipment senses eight-channel, fan-fold perforated Mylar or paper tape photoelectrically at 300 characters per second. The reader is a Digitronics 2500 and the control is a DEC data register, flag, and associated logic circuits.

Perforated Tape Punch and Control Type 75D

This equipment consists of a control unit and a Teletype BRPE punch that perforates eight-channel, fan-fold paper tape at 63.3 lines per second.

Processor Options

Extended Arithmetic Element Type 177

The Extended Arithmetic Element (EAE) is a standard option for the PDP-7 which facilitates high-speed multiplication, division, shifting, normalizing, and register manipulation. Installation of the EAE adds an 18-bit multiplier quotient register (MQ) to the computer as well as a 6-bit step counter register (SC). The content of the MQ is continuously displayed on the operator console. The Type 177 option and the basic computer cycle operate asynchronously, permitting computations to be performed in the minimum possible time. Further, the EAE instructions are microcoded so that several operations can be performed by one instruction to simplify arithmetic 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 by identifying an interrupting device directly, without the need for flag searching. Multilevel program 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 sequence 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 I/O 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.

Data Interrupt Multiplexer Type 173

The single PDP-7 data break interrupt channel is expanded to handle information transfers with three high-speed I/O devices by addition of the Type 173 option. This option provides multiplex control for simultaneous operation of three high-speed devices such as magnetic tape or drum devices. Maximum combined transfer rate is 570,000 18-bit words per second.

Memory Increment Type 197

This option allows an external condition or signal from an I/O device to increment the content of any core memory location. The peripheral device initiates a break cycle so that the content of a core memory address specified by the device is read into the memory buffer register, incremented by one, and written back into the same address in one computer cycle.

Boundary Register and Control Type KA70A

This option establishes core memory address boundaries that can be assigned to specific users when the system is used for real time computing with simultaneous multiuser program execution.

Core Memory Options

Memory Extension Control Type 148

Memory expansion beyond a total capacity of 8K words requires addition of the Type 148 option to extend the program counter, memory address register, and mode control. Any memory size from 4096 to 32,768 words can be obtained by addition of Type 147 and 149B modules.

Core Memory Module Type 147

This option extends the capacity of the standard 4096-word memory to 8192 words.

Core Memory Module Type 149B

This option extends the capacity of the PDP-7 core memory by one field of 8192 words. The 149B option can be added only to memories of 8K, 16K, or 24K capacity (not to 4K, 12K, etc. without also adding a Type 147 module).

Memory Parity Type 176

This option assures reliability of all core memory data storage and retrieval operations by generating, storing, and checking parity on every transfer. An odd parity bit is generated and written in the same core location as the word being written. Upon reading, a word drawn from core memory is checked for parity and if odd parity is detected a program interrupt is initiated or the program is halted.

Input/Output Options

Card Reader and Control Type CR01B

Standard 12-row, 80-column punched cards are read by this device in either alphanumeric or binary mode. Reading is accomplished by mechanical sensors at a maximum rate of 100 cards per minute.

Card Reader and Control Type 421

Standard punched cards are read optically at up to 200 cards per minute on the Type 421A, or up to 800 cards per minute on the Type 421B. Information punched on the cards is read column by column in binary or alphanumeric modes.

Card Punch Control Type 40

This device controls on-line buffered operation of a standard card punch machine. Cards are punched one row at a time at 40 millisecond intervals, providing a punching rate of 100 cards per minute. Any or all positions can be punched in any format.

Automatic Line Printer and Control Type 647

This machine prints a selection of 64 characters on a line of 120 characters at a rate of 300, 600, or 1000 lines per minute. Printing is performed by solenoid-actuated print hammers. Loading, printing, and format are under program control. Format is program selected from a punched format tape in the printer.

DECtape Dual Transport Type 555 and Control Type 550

The DECtape system provides a unique fixed address magnetic-tape facility for high-speed loading, readout, and program updating. Each DECtape transport contains two independent tape drives. Up to four transports (eight drives) can be used with one control.

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. Phase recording, rather than amplitude recording; redundant, nonadjacent data tracks; and a prerecorded timing and mark track are features of this system. The control 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

Up to eight IBM or IBM-compatible tape transports can be operated automatically by the Type 57A to transfer information through the PDP-7 data break interrupt facility. Magnetic tape transports are controlled to read or write at densities of 200, 556, or 800 characters per inch at speeds of 75 or 112.5 inches per second.

Magnetic Tape Transport Type 570

The Type 570 is a highly sophisticated tape transport that reads and writes at 75 or 112.5 inches per second at program-selected densities of 200, 556, or 800 characters per inch. Tape motion is controlled by pneumatic capstans and brakes, eliminating conventional pinch rollers, clamps, and mechanical arms. 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. The Type 570 contains a multiplex interface which permits time-shared use of the transport by two Type 57A tape control on the same or different computers.

Magnetic Tape Transport Type 545

The Type 545 tape unit operates at a speed of 45 ips and has three selectable densities, 200, 556, 800 bpi. The 545 is controlled by the Type 57A with a Type 521 Interface. Standard 7-channel, IBM-compatible tape format is used. The transport mechanism uses a pinch roller drive with vacuum column tension.

Magnetic Tape Transport Type 50

The Type 50 can be used with the Type 57A to read or write IBM-compatible magnetic tapes at transfer rates of 15,000 or 41,700 characters per second. Tape speed is 75 inches per second at densities of 200 or 556 characters per inch.

Block Transfer Serial Drum System Type 24

Drum transfers operate through the computer 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.

Incremental Plotter Control Type 350

One California Computer Products Digital Incremental Recorder can be operated from a DEC Increment Plotter Control Type 350 to provide high-speed plotting of points, continuous curves, points connected by curves, curve identification symbols, letters, and numerals under program control. The recorder can be selected from four models, that differ in speed (12,000 or 18,000 steps per minute), step size (0.01 or 0.005 inches per step), and paper width (12 or 51 inches).

Oscilloscope Display Type 34A

Computer data can be plotted point-by-point on a 5-inch oscilloscope, such as the Tektronix Model RM503, by the option. The horizontal axis of each point is determined by 10 binary bits, and the vertical axis is determined by another 10 binary bits. This option can be obtained with or without the oscilloscope.

Precision CRT Display Type 30D

The Type 30D is a random-position point-plotting display with a self-contained, 16-inch CRT using magnetic deflection and focusing. Data is plotted point by point in a raster 9-3/8 inches square having 1024 points on a side according to separately variable 10-bit X and Y coordinates. The display includes program intensity control. Plotting rate is 35 microseconds per point.

Symbol Generators Type 33 and Type 342

The Type 33 is an option used with the Type 30D display that simplifies the programming required to present character and symbols on the face of the display tube. The Type 342 serves a similar purpose for plotting characters on the Type 340 display. Two 64-character sets are available for the Type 342.

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.

Photomultiplier Light Pen Type 370

A fiber optic light pipe and photomultiplier in the light pen allow high-speed detection of information displayed on the Type 34A, 30D, or 340 displays. Detection of information by the Type 370 can be sampled by the computer to alter the program.

General Purpose Analog-to-Digital Converter Type 138E

The Type 138E is a high-speed successive approximation converter with analog input signal range from 0 to 10 volts. The analog voltage is converted to a binary number, selectable from 6 to 12 bits. Conversion time varies, depending on the number of bits and the accuracy required. Combinations of switching point accuracy and number of bits can be selected on a front panel switch.

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 ±0.15% ±1/2 least significant bit.

General Purpose Multiplexer and Control Type 139E

Up to 64 analog input channels can be selected for application to the input of the Type 142 or Type 138E by the Type 139E. Channels can be program selected in sequence or by individual address. The number of channels that can be selected is determined by the number of optional Multiplexer Switches Type A100 series used in the Type 139E. Each Type A100 can select two channels.

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.

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 Communication Systems Type 630

This system is a real-time interface between Teletype stations and the PDP-7 and is ideal for multi-user computer time-sharing message switching systems, and data collection-processing systems. A variety of Type 630 systems are available for half-duplex and full-duplex operation with up to 64 stations.

Relay Output Buffer Type 140

A data buffer register loaded from the computer accumulator actuates 18 relays, each having mercury-wetted single-pole double throw contacts. These contacts can be used for direct digital control or signal generation for external equipment.

Inter Processor Buffer Type 195

This device serves as an interface between a PDP-7 and another computer to permit bi-directional data communication with an asynchronous processor.

PROGRAMMING SYSTEM

The PDP-7 Programming System includes an advanced FORTRAN compiler, a symbolic assembler, symbolic tape editor, Digital debugging system (DDT), maintenance routines and a library of arithmetic, utility and programming aids developed on the program-compatible PDP-4. Both the symbolic tape 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.

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 computer 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.

Symbolic Assembler

The symbolic assembler allows the programmer to 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. This 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 single quote (') character. The same symbol followed by the slash (/) character, causes the content 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 on tape immediately in the form of loadable patch tapes, eliminating the necessity of creating new symbolic tapes and reassembling each time an error is found.

Symbolic Tape Editor

The editor program 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.

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 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.

SYMBOLS

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

Symbol	Explanation
A = > B	The content of register A is transferred into register B
0 = > A	Register A is cleared to contain all binary zeros
Aj	Any given bit in A
A5	The content of bit 5 of register A
A5(1)	Bit 5 of register A contains a 1
A6 - 11	The content of bits 6 through 11 of register A
A6 - 11 = > B0 - 5	The content of bits 6 through 11 of register A is transferred into bits 0 through 5 of register B
Y	The content of any core memory location
V	Inclusive OR
$\not\vee$	Exclusive OR
$\wedge$	AND
$\overline{A}$	One's complement of the content of A
+ 1 = > A	The content of A is incremented by 1

SECTION 2

FUNCTIONAL DESCRIPTION

The standard PDP-7, as mentioned in Section 1, can be analyzed into a processor, a core memory, interface elements, and input/output equipment.

PROCESSOR

To perform logical, arithmetic, data processing, and control functions the processor employs seven active registers. Intelligence flow among these registers and between them and other major elements of the computer system is shown on Figure 5.

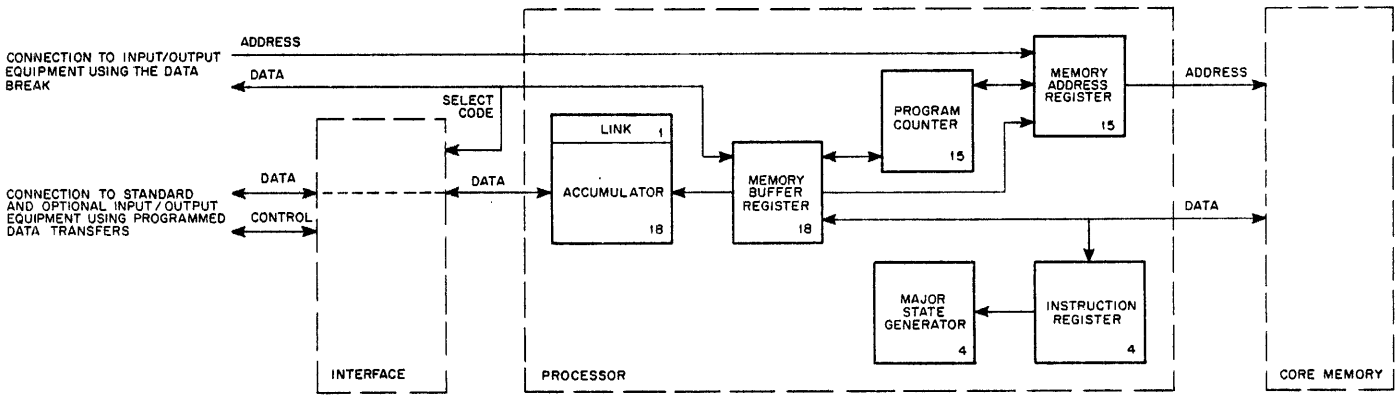


Figure 5 Major Register Block Diagram of the Processor

Accumulator (AC)

Arithmetic operations are performed in this 18-bit register. The AC can be cleared and complemented. Its content can be rotated right or left with the link. The content of the memory buffer register can be added to the content of the AC with the result left in the AC. The content of both registers can be combined by the logical operations AND and exclusive OR, the result remaining in the AC. The Inclusive OR can be formed between the AC and the ACCUMULATOR switches on the operator console 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.

Link (L)

This one-bit register is used to extend the arithmetic capability of the accumulator. In 1's complement arithmetic, the link is an overflow indicator; in 2's complement it logically extends the AC to 19 bits and functions as a carry register. Overflow into the link from the accumulator can be checked by the program to greatly simplify and speed up single and multiple precision arithmetic routines. The link can be cleared and complemented and its state sensed independent of the AC. It is included with the AC in rotate operations.

Memory Address Register (MA)

The address of the core memory cell currently being accessed is contained in the 13-bit MA. Information enters the MA from the memory buffer register program counter, or from external device operating in a data interrupt. Addition of the Memory Extension Control Type 148 option expands the MA to 15 bits.

Program Counter (PC)

The program sequence, that is the order in which instructions are performed, is determined by the PC. This 13-bit register contains the address of the memory cell from which the next instruction is to be taken. Information enters the PC from the MA, MB, or the ADDRESS switches of the operator console. Addition of the Memory Extension Control Type 148 option expands the PC to 15 bits.

Memory Buffer Register (MB)

All information transferred into or out of core memory passes through the MB. Information is read from a memory cell into the MB and rewritten into the cell in one cycle time (1.75 μsec). Instructions and data are brought from core memory into the MB for processing. The MB serves also as a buffer for information transferred between core memory and an external device in a data interrupt. The content of the MB may be incremented by one.

Instruction Register (IR)

This 4-bit register contains the operation code of the instruction currently being performed by the computer. The four most significant bits of the current instruction are loaded into the IR directly from core memory during the Fetch cycle. The content of the IR is decoded to determine the functions performed and the major states entered in execution of the instruction.

Major State Generator

The computer operates in one of four major control states during each machine timing cycle. One or more states are entered to execute an instruction. The states are Fetch, Execute, Defer, and Break and are determined by the major state generator. Only one state exists at a time and all states, except Break, are determined by the programmed instruction being executed.

Fetch (F)

A new instruction is obtained when this state is entered. The content of the memory cell specified by the PC is placed in the MB, and the operation code (bits 0-3) of this instruction word are placed in the IR. The content of the PC is then incremented by one. If a single-cycle instruction is fetched, the operations specified are performed during the last part of the fetch cycle, then the next state is fetch for the next instruction. If a two-cycle instruction is fetched, the succeeding control state is either defer or execute.

Defer (D)

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 by the instruction contains the address of the operand, and access to the operand is deferred to the next memory cycle.

Execute (E)

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

Break (B)

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 break interrupt allows information to be transferred between core 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 content of the PC and the content of the Link are stored in core memory location 0000, and the program continues from location 0001.

Detailed Processor Block Diagram Discussion

All logic circuit elements of the processor are shown on Figure 6. These elements consist of controls for the major registers, the essential timing generator for the computer system, the manual controls, and the special program feature controls (program interrupt, data break interrupt, I/O skip, I/O trap, etc.).

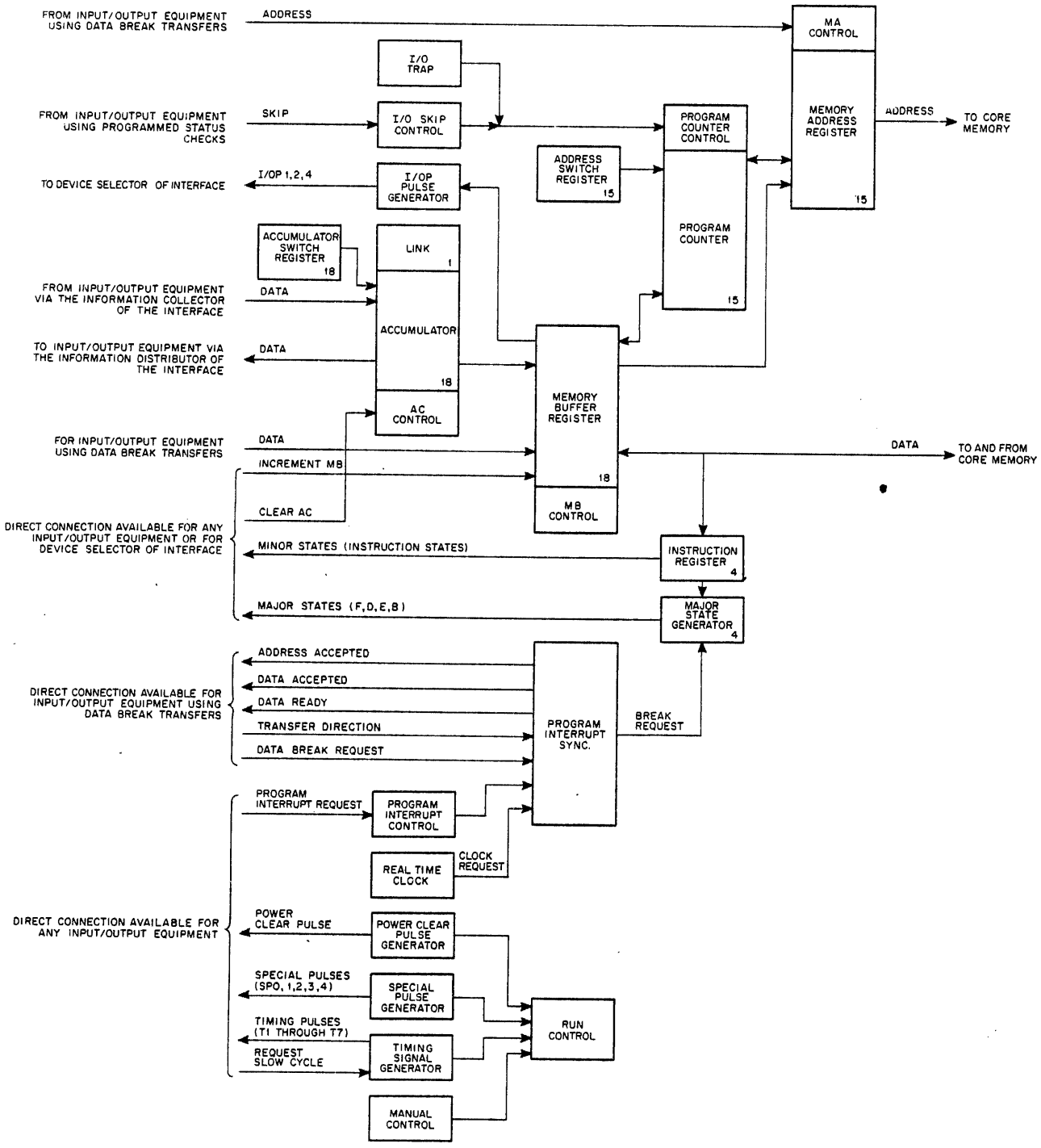


Figure 6 Detailed Block Diagram of the Processor

Timing Generators

The power clear pulse generator produces pulses to clear processor and I/O device register when the computer is energized or de-energized. Special pulses produced to enact computer function initiated by manual controls are also available for use by peripheral equipment. The timing signal generator produces the basic timing pulses that control all processor operations. The timing of these pulses can be in a normal fast cycle or in a slow cycle requested by a signal from an I/O device. The slow cycle is used to execute input/output transfer instructions that communicate with equipment that is not fast enough to act upon rapid successive command pulses.

Manual Controls

Control keys and switches allow the normal start/stop/continue control, control special single-cycle and single-instruction operation for maintenance of the machine, control the use of the special program features, and provide a means of loading data directly into the computer. The switch register is used to store an 18-bit word in the AC and the ADDRESS switch register is used to load a 15-bit core memory address into the PC. Use of the manual controls provides rapid entry of program information on perforated tapes and a broad means of modifying programs or data in core memory.

Input/Output Skip

This facility allows the program to skip or not skip the next instruction according to the condition of an external device flag. Skipping is accomplished by incrementing the content of the PC so that execution of an instruction is by-passed. This facility simplifies program branching or decision-making based upon the condition of device status lines.

Input/Output Trap

Basic hardware that allows the PDP-7 to serve in time-sharing applications is provided as the I/O trap facility. These circuits monitor instructions being performed and initiate a program interrupt if an instruction is encountered that will interfere with the execution of other programs. The interrupt program can then determine the cause of the trap and take appropriate action. The I/O trap feature can be enabled or disabled by a switch on the operator console or by programmed instruction.

Program Interrupt Control and Synchronization

Peripheral equipment can initiate an interrupt of the program that, effectively, transfers program control to a subroutine that services the initiating device. The interrupt is requested by a signal from the device. The synchronization element grants the interrupt only upon completion of the current instruction. The interrupt is accomplished by entering the Break state to store

the content of the link, the condition of the memory extend mode control, the condition of the I/O trap, the content of the extend PC, and the content of the PC in core memory location 0. Memory location 1 is then placed in the PC to transfer program control to the subroutine. Upon conclusion of the subroutine, the initial program is returned to its previous state by using the information stored in location 0.

Data Break Interrupt

Peripheral equipment can cause temporary suspension of the main program to "steal" a cycle of computer time for the transfer of information with core memory. A data break request received by the program interrupt synchronization element is honored at the end of execution of the current instruction. The Break state is then entered and controlled by the peripheral device. The device specifies a core memory address at which the transfer is to occur, and designates the direction of the transfer as into or out of core memory. The address is supplied to the MA and the data transfer is enacted between the device and the MB. At completion of the Break cycle the interrupted program is continued, unless the device has requested another data break. One I/O device can be connected to cause data breaks, or up to four devices can be connected through the Data Interrupt Multiplexer Control Type 173.

Real Time Clock

A clock within the standard computer causes a clock data break interrupt (in the same manner as described previously for an external device data break) every 1/60 second, so that program events can be related to real time. When this interrupt occurs the content of core memory location 7 is incremented by 1 during the Break cycle and serves as a counter. Any number can be stored in address 7 so that it overflows (counts up to zero) at a time multiple of 1/60 second and this stored number. When overflow occurs the clock flag is set and initiates a program interrupt.

CORE MEMORY

The core memory provides storage for instructions to be performed and information to be processed or distributed. This random-address ferrite-core memory holds 4096 18-bit words in the standard PDP-7. Optional equipment extends the storage capacity in blocks of 4096 or fields of 8192 words, or expands the word length to 19 bits to provide parity checking. Memory location 0_8 is used to store the content of the PC following a program interrupt, and location 1_8 is used to store the first instruction to be executed following a program interrupt. (When a program interrupt occurs, the content of the PC is stored in location 0_8, and program control is transferred to location 1 automatically.) Location 7_8 is used with the real time clock and locations 10_8 through 17_8 are used for auto-indexing. Location 20 is used to store information during a call subroutine (CAL) instruction and location 20 is used as the starting address of the CAL handling subroutine. All other locations can be used to store instructions or data.

Core memory contains numerous circuits such as read-write address selection switches, address decoders, inhibit drivers, and sense amplifiers as shown in Figure 7. These circuits perform

the electrical conversions necessary to transfer information into or out of the core array and perform no arithmetic or logic operations upon the data. Since their operation is not discernible by the programmer or operator of the PDP-7, these circuits are not described here in detail.

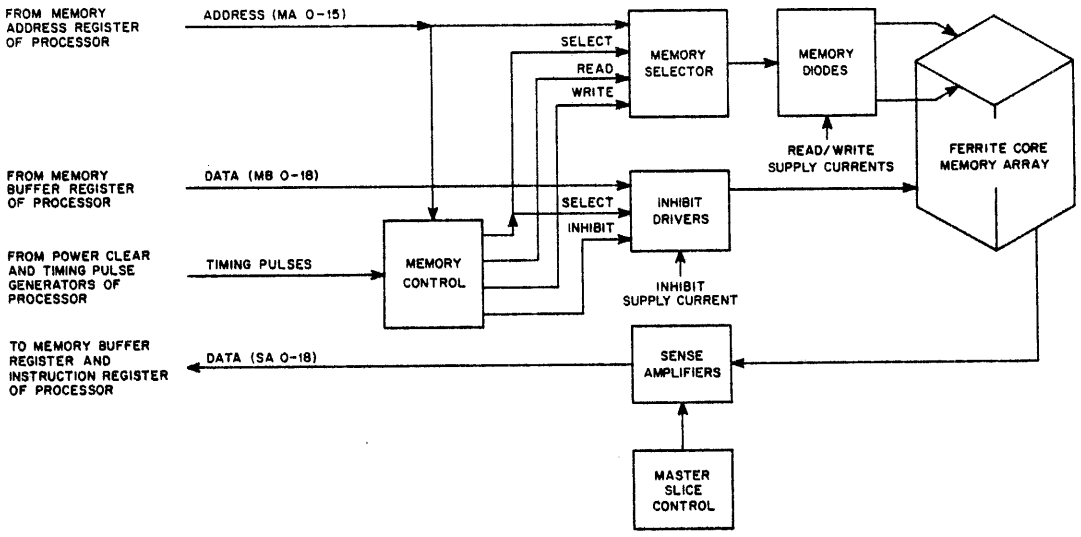


Figure 7 Core Memory Block Diagram

INTERFACE

Information is transferred to peripheral equipment from the processor by means of an information distributor. Information is transferred to the processor from I/O devices by the information collector. Addressing of a device to receive programmed command signals is accomplished by means of the device selector. These three logic circuit elements constitute the major PDP-7 data interface and are shown in Figure 8. Control interface is effected by elements of the processor, as described previously in this section. Timing of the processor as related to interface and input/output operations is shown on Figure 9.

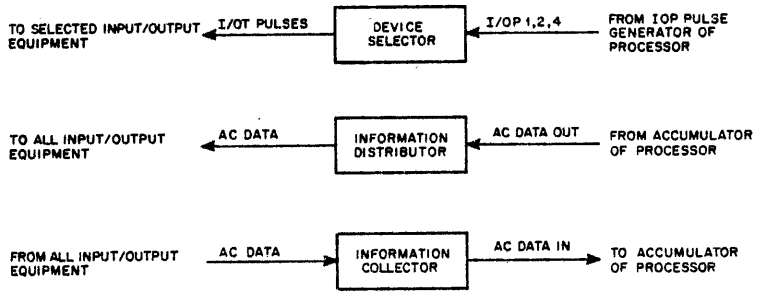
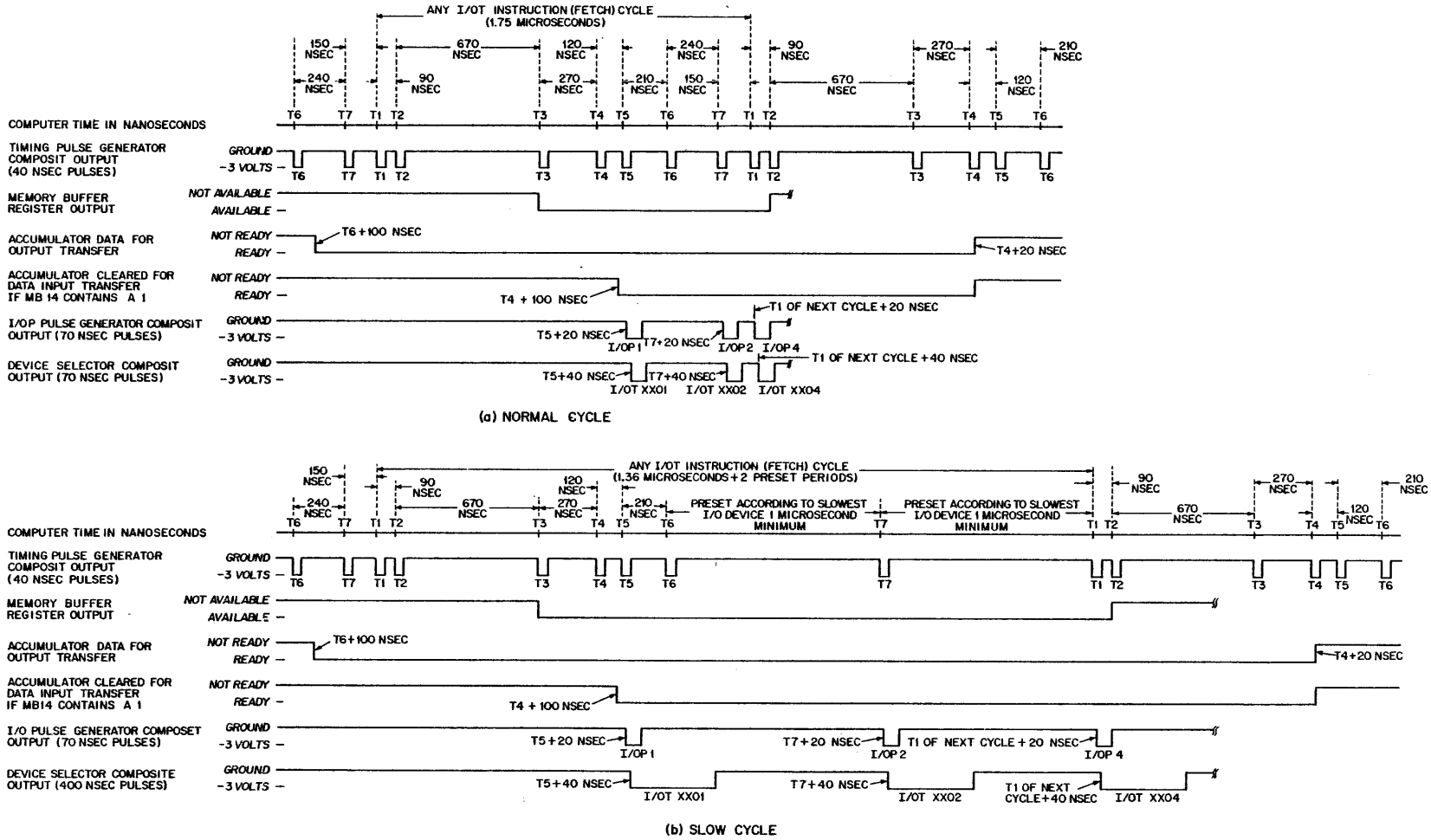


Figure 8 Interface Block Diagram



(a) NORMAL CYCLE

(b) SLOW CYCLE

Figure 9 Input/Output Transfer Timing Diagram

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 input/output transfer (I/OT) instruction. It then generates an I/OP pulse at event time 1 if memory buffer bit 17 is a 1, at 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 microprogrammed operate group event times. A complete list of the I/OP pulses and corresponding times is given below and shown in Figure 9.

Event Time	Computer Cycle Time	Instruction (Bit (in MB)	I/OP Number
1	T5	17	1
2	T7	16	2
3	T1 (next cycle)	15	4

Upon execution of an I/OT instruction, the device selector determines which device has been selected, and then generates I/OT pulses according to the content of bits 15 through 17 of the instruction. These I/OT pulses are sent to the appropriate device as command signals. Generally the I/OT commands perform one or all of the following functions:

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

b. I/OP 2 is used to clear the flag or flags associated with a device and to read the content of the device buffer into the information collector.

c. 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 or subdevice selection bits 4, 5, 12, and 13. For extremely expanded mode selection, a device may sense the state of the accumulator bits loaded prior to the I/OT instruction.

One channel of the device selector (decoding of one select code) is shown in Figure 10. The six-bit device selection numbers, memory buffer bits 6-11, are decoded by a Diode Gate module Type B171. The select code, therefore, produces an enabling level for the selected device. This level can be used to request a slow cycle and enables three gates of a Diode Gate module Type R111. The device selector pulse amplifiers transmit pulses to the selected device according to bits 15, 16, and 17 of the I/OT 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:

a. 2.5 volts, positive or negative pulses of 70-nanosecond duration

b. 2.5 volts, positive or negative pulses of 400-nanosecond duration

The standard device selector contains selector modules for the standard devices and has provisions for up to six additional select codes. 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 10 shows the B171 with the clipping point marked with a ⊗ symbol. Either the 1 or 0 diode must be clipped from each of the six bit inputs of the module to establish the select code.

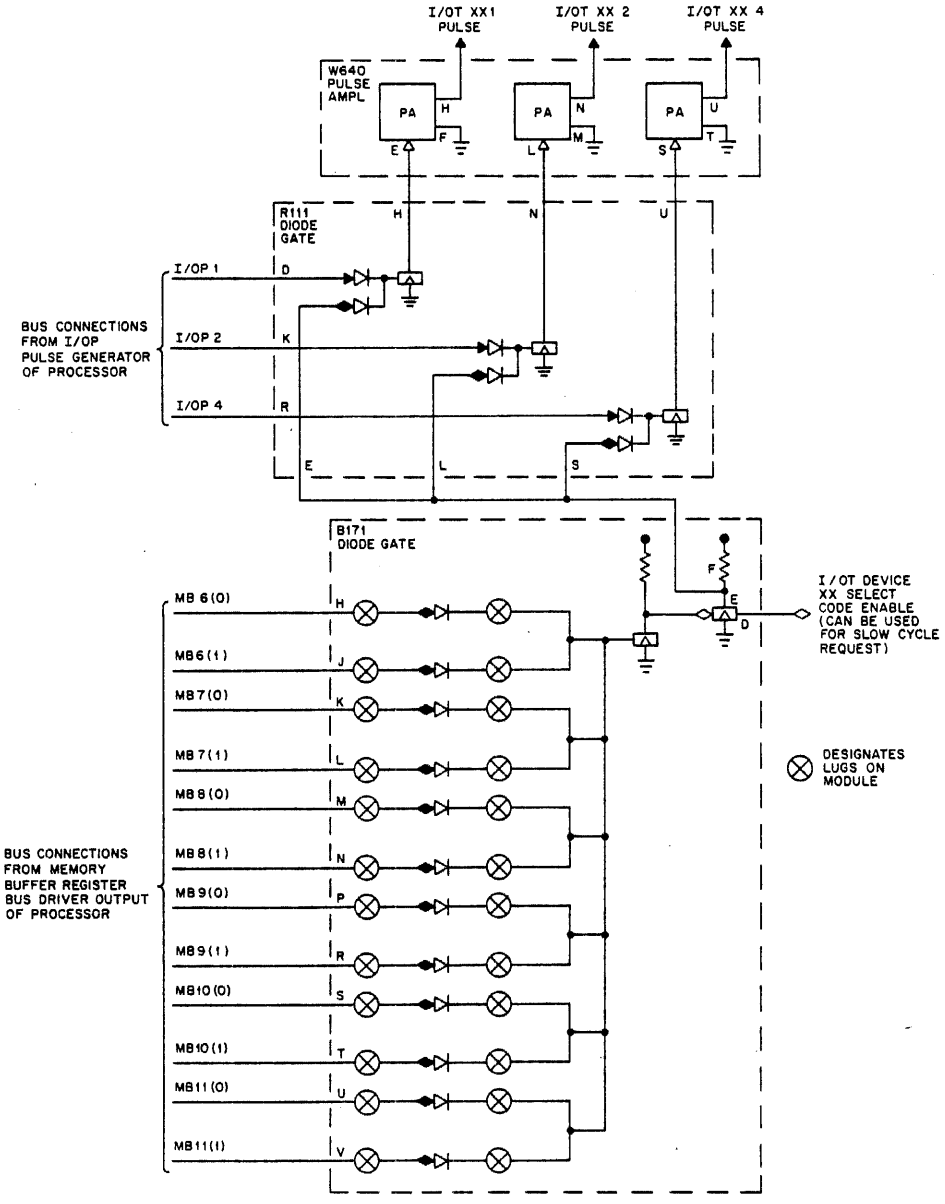


Figure 10 Device Selector Logic Diagram

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 connectors are used to transfer memory buffer and device control bits. The Bus Driver Type R650 modules are used in the ID. Nine 18-bit connection points to the bussed output are standard on the PDP-7. The paper tape punch and teleprinter use two of six channels. A third channel is used for expanded ID connection. If all of the standard channels are used, the ID can be expanded to any number of output channels by adding circuit blocks similar to the standard ID and suitable buffering.

Other external devices are easily connected to the information distributor. Each device receives pulses from the device selector to gate bus information into the receiving buffer register.

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

Information Collector (IC)

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

In the basic PDP-7, seven channels of IC are provided. 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 or more channels of the IC (e.g. Extended Arithmetic Element Type 177, A-D Converter Type 138, DECtape Control 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 two-input Diode Gates Type R141. The R141 accepts standard levels of 0 and −3 volts or standard 70-nanosecond or wider pulses. The input load is 1.0 milliampere per grounded input (using a Type 175 option).

Bits transferred to the AC correspond to the incoming polarities:

0 volts = binary 0 transmitted to AC
−3 volts = binary 1 transmitted to AC

INPUT/OUTPUT

Peripheral equipment may either be asynchronous with no timed transfer rates or synchronous with a timed transfer rate. Devices such as the CRT displays, teleprinter-keyboard, and the line printer can 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 after they have been set in operation by transferring a continuous block of data words through the PDP-7 data interrupt facility. 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.

Separate parallel buffers are provided on each input/output device attached to the basic PDP-7. The high speed perforated Tape Reader Control Type 444B contains an 18-bit buffer and binary word assembler. The high speed perforated Tape Punch Type 75D, and the teleprinter and the keyboard of the Teletype and Control Type 649 each contain separate 8-bit buffers. These devices are described briefly in Section 1 and are discussed in detail in Section 7 of this handbook.

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

Figure 11 shows the data path between device buffers and the AC through the information collector or information distributor.

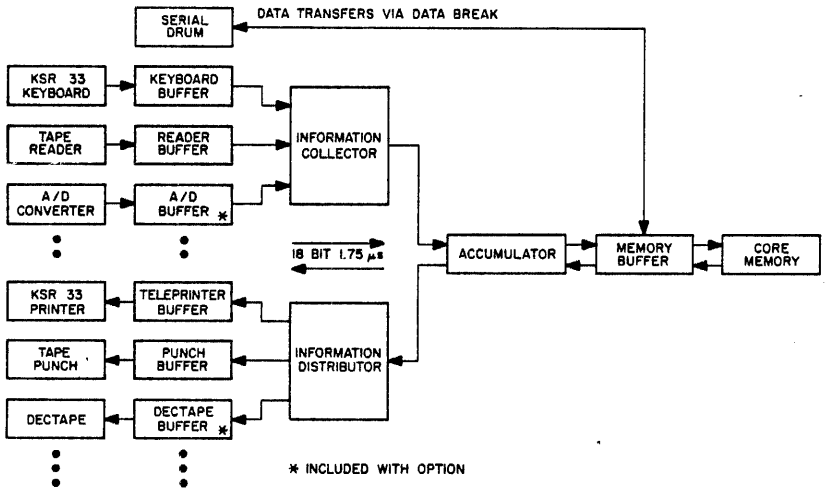


Figure 11 Input/Output Information Flow

SECTION 3

INSTRUCTIONS

Instruction words are of two types: memory reference and augmented. Memory reference instructions store or retrieve data from core memory, while augmented instructions do not. All instructions utilize bits 0 through 3 to specify the operation code. Operation codes of 00_8, through 60_8 specify memory reference instructions, and codes of 70_8 and 74_8 specify augmented instructions. Memory reference instruction execution times are multiples of the 1.75-microsecond memory cycle. Indirect addressing increases the execution time of a memory reference instruction by 1.75 microseconds. The augmented instructions, input/output transfer and operate, are performed in 1.75 microseconds.

MEMORY REFERENCE INSTRUCTIONS

Memory reference instructions require a Fetch cycle to interpret the operation and determine the memory address, and most (all except the jump) require an Execute cycle to carry out the operation. When indirect addressing is specified, an extra (Defer) cycle is entered to determine the effective address. Information is transferred from the AC into core memory through the MB. When an operand is to be retrieved from core memory it is transferred into the MB; and the specified operation is then performed (usually between the AC and the MB). When information in the accumulator is to be stored in core memory, it passes through the MB to the instruction-specified address.

The jump instruction contains an address but does not require an operand. An Execute cycle is not needed, and the instruction is completed in one (Fetch) cycle.

The bit assignments of the memory reference instruction are shown in Figure 12 and listed in Table 1. Bits 0-3 determine the operation to be performed. Bit 4 is used to specify direct or indirect addressing. Bits 5-17 specify the memory address of the operand.

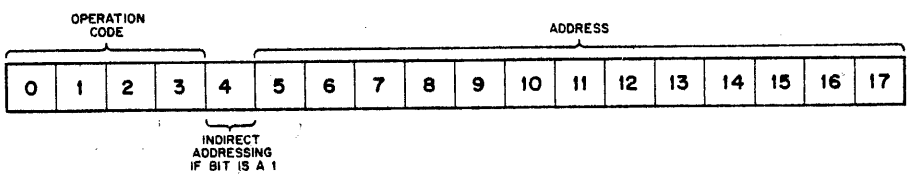


Figure 12 Memory Reference Instruction Bit Assignments

TABLE 1 MEMORY REFERENCE INSTRUCTIONS

Mnemonic Symbol	Octal Code	Machine Cycles	Operation Executed
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. L, PC = > 20 21 = > PC
DAC Y	04	2	Deposit AC. The content of the AC is deposited in the memory cell at location Y. The previous content of Y is lost; the content of the AC is unchanged. AC = > Y
JMS Y	10	2	Jump to subroutine. The content of the PC and the content of the L is deposited in memory cell Y. The next instruction is taken from cell Y + 1. L = > Y_0 0 = > Y1-4 PC = > Y5-17 Y + 1 = > PC
DZM Y	14	2	Deposit zero in memory. Zero is deposited in memory cell Y. The original content of Y is lost. The AC is unaffected by this operation. 0 = > Y
LAC Y	20	2	Load AC. The content of Y is loaded into the AC. The previous content of the AC is lost; the content of Y is unchanged. Y = > AC

TABLE 1 MEMORY REFERENCE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Machine Cycles	Operation Executed
XOR Y	24	2	Exclusive OR. The logical operation exclusive OR is performed between the content of Y and the content of the AC. The result is left in the AC and the original content of the AC is lost. The content of Y is changed. Corresponding bits are compared independently. Yj ⊻ ACj = > ACj Example ACj original / Yj / ACj final 0 / 0 / 0 0 / 1 / 1 1 / 0 / 1 1 / 1 / 0
ADD Y	30	2	Add (1's complement). The content of Y is added to the content of the AC in 1's complement arithmetic. The result is left in the AC and the original content of the AC is lost. The content of Y is unchanged. The link is set to 1 on overflow. Y + AC = > AC
TAD Y	34	2	Two's complement add. The content of Y is added to the content of the AC in 2's complement arithmetic. The result is left in the AC and the original content of the AC is lost. The content of Y is unchanged. A carry out of bit 0 complements the link. Y + AC = > AC
XCT Y	40	1+*	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.

*This instruction requires one cycle plus the number of cycles required for execution of the instruction at the specified address.

TABLE 1 MEMORY REFERENCE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Machine Cycles	Operation Executed
ISZ Y	44	2	Increment and skip if zero. The content of Y is 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 content of the AC is unaffected. Y + 1 = > Y If result = 0, PC + 1 = > PC
AND Y	50	2	AND. The logical operation AND is performed between the content of Y and the content of the AC. The result is left in the AC, and the original content of the AC is lost. The content of Y is unchanged. Corresponding bits are compared independently. $Y_j \wedge AC_j = > AC_j$ Example AC_j original / Y_j / AC_j final 0 / 0 / 0 0 / 1 / 0 1 / 0 / 0 1 / 1 / 1
SAD Y	54	2	Skip if AC is different from Y. The content of Y is compared with the content of the 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 content of the AC and the content of Y are unchanged. If AC $\neq$ Y then PC + 1 = > PC
JMP Y	60	1	Jump to Y. The next instruction to be executed is taken from memory cell Y. Y = > PC

AUGMENTED INSTRUCTIONS

Augmented instructions do not require a memory reference. Since no address is required, the least significant bits are decoded to initiate various operations to extend, or augment, the operation code. Because no call on core memory is required for execution of instructions in this class, no Execute cycle is needed and the instructions are performed in one 1.75-microsecond Fetch cycle. Microprogramming of bits 4 through 17 can be used to specify an operation in each three sequential event times within the cycle.

The augmented instructions are divided into two classes:

a. Instructions having an operation code of 70_8 are Input/Output Transfer (I/OT) commands and are used to control, test, or transfer information with input/output devices.

b. Instructions having an operation code of 74_8 are Operate (OPR) commands and are used for basic processor data manipulation such as skipping, shifting, rotating, etc. A large group of operate instructions having an operation code of 76_8 is added to the computer with addition of the Type 177 Extended Arithmetic Element. This group of instructions is described in Section 5 of this handbook.

Input/Output Transfer Instructions

Input/Output Transfer (I/OT) instructions are used to control peripheral devices, to sense their status, and to transfer information between them and the processor. Three instructions initiate generation of time-sequenced I/OP pulses and a device-select code, both of which are applied to the device selector interface circuits. Upon receipt of these signals, the device selector generates I/OT pulses that effect the operations specified by the I/OT instruction. The three event times of the I/OT instructions are identified with an I/OP pulse, and generation of the I/OP pulses is determined by the content of bits 15, 16, and 17 of the I/OT instruction. 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. The relationship between pulses, event times, and instruction bits is as follows:

Instruction Bit	I/OT Pulse	I/OP Pulse	Processor Time	Event Time
17	I/OT 1	I/OP 1	T5	1
16	I/OT 2	I/OP 2	T7	2
15	I/OT 4	I/OP 4	T1 of next cycle	3

During a normal computer cycle the I/OT pulses are standard DEC 70-nanosecond pulses and I/OT 2 occurs approximately 450 nanoseconds after I/OT 1, and I/OT 4 occurs approximately 150 nanoseconds after I/OT 2. During a slow cycle the I/OT pulses are standard DEC 400-nanosecond pulses occuring at a minimum of 1 microsecond apart. Slow-cycle timing is adjusted to accommodate the slowest device connected to the computer. This timing is indicated in Figure 9.

The I/OT instruction format is shown in Figure 13. Bits 0-3 carry the I/OT 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 transmission electrical pulses to the device for direct control of the information transfer or device operation or to the information collector. Descriptions of I/OT instructions are given along with the I/O equipment descriptions in succeeding sections of this handbook.

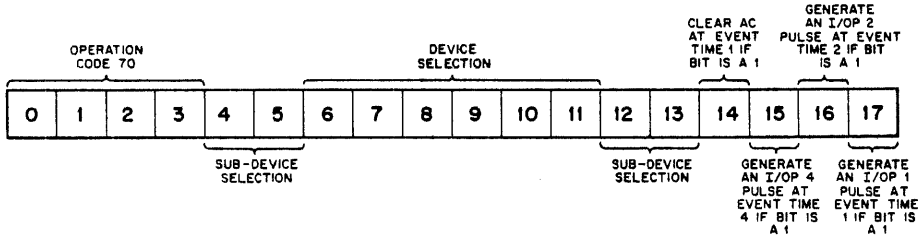


Figure 13 I/OT Instruction Bit Assignments

Operate Instructions

Operate (OPR) instructions are divided into two groups, designated by the condition of bit 4. Group 1 (OPR 1) instructions are identified by bit 4 containing a 0, and are used to complement, shift, rotate, skip, etc. Group 2 (OPR 2) instructions are identified by bit 4 containing a 1, and are used to load a program-specified number into the accumulator (literal or immediate data) without storing the number in core memory.

Group 1 Operate Instructions

The group of augmented instructions with operation code 74$_8$ and containing a 0 in bit 4, is used to manipulate and sense information in the link and accumulator. These instructions can be combined to cause several operations to occur, by microprogramming bits associated with functions that occur in each of the three event times. The three event times are numbered according to the sequence in which they occur, and all three occur within the single 1.75 microsecond Fetch cycle required for execution of the instruction. An operation which takes place at event time 1 is completed before event time 2 begins, and events specified for enactment during event time 2 are completed before event time 3 begins. No two operations can be specified in one event time of the same instruction if they logically conflict.

Illegal example: 740014 RAL OAS
Legal example: 740003 CML CMA

Format of the Group 1 Operate instructions is shown in Figure 14 and the instructions are listed in Table 2.

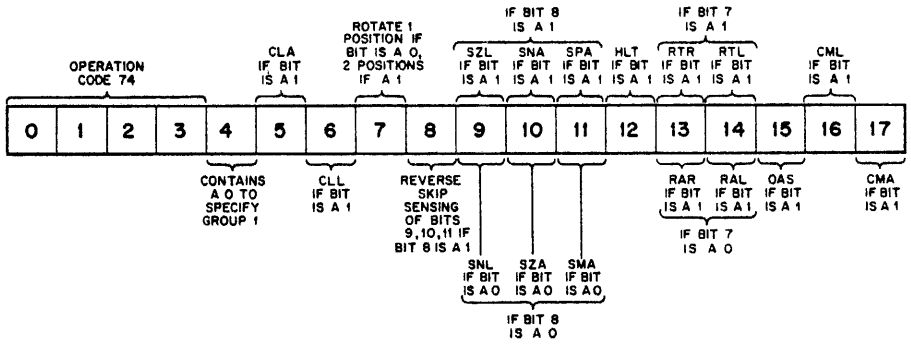


Figure 14 Group 1 Operate Instruction Bit Assignments

TABLE 2 OPERATE INSTRUCTIONS

Mnemonic Symbol	Octal Code	Event Time	Operation Executed
OPR or NOP	740000	---	Operate or No Operation. Indicates the operate class. When used alone, performs no operation; the computer proceeds to the next instruction after one memory cycle.
CMA	740001	3	Complement Accumulator. Each bit of the AC is complemented. $\overline{AC}j => ACj$
CML	740002	3	Complement Link. $\overline{L} => L$
OAS	740004	3	Inclusive OR ACCUMULATOR Switches. The word set into the ACCUMULATOR switches is OR combined with the content of the AC, the result remains in the AC, the original content of the AC is lost, and the switches are unaffected. AC V ACCUMULATOR Switches = > AC
RAL	740010	3	Rotate Accummulator (and link) Left. The content of the AC and L are rotated one position to the left. $ACj => ACj-1$ $AC0 => L$ $L => AC\ 17$

TABLE 2 OPERATE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Event Time	Operation Executed
RLR	740020	2	Rotate accumulator (and link) Right. The content of the AC and L are rotated one position to the right. ACj = > ACj + 1 L = > AC0 AC17 = > L
HLT	740040	---	Halt. The program is stopped at the conclusion of the cycle, so that HLT can be combined with other operations to be performed in any event time. 0 = > RUN flip-flop
SMA	740100	1	Skip on minus accumulator. If the content of the AC is a negative (2's complement) number, the content of the PC is incremented to skip the next successive instruction. If AC0 = 1, then PC + 1 = > PC
SZA	740200	1	Skip on zero accumulator. If the content of the AC equals zero (2's complement), the next instruction is skipped. If AC0-17 = 0, then PC + 1 = > PC
SNL	740400	1	Skip on non-zero link . If the L contains a 1, the next instruction is skipped. If L = 1, then PC + 1 = > PC
SKP	741000	1	Skip. The next instruction is unconditionally skipped. PC + 1 = > PC
SPA	741100	1	Skip on positive accumulator.. If the content of the AC is zero (2's complement) or a positive number, the next instruction is skipped. SPA = $\overline{\text{SMA}}$ If AC0 = 0, then PC + 1 = > PC
SNA	741200	1	Skip on non-zero accumulator. If the content of the AC is not zero (2's complement), the next instruction is skipped. SNA = $\overline{\text{SZA}}$ If AC0-17 $\neq$ 0, then PC + 1 = > PC

TABLE 2 OPERATE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Event Time	Operation Executed
SZL	741400	1	Skip on zero link. If the L contains a 0, the next instruction is skipped. SZL = $\overline{\text{SNL}}$ If L = 0, then PC + 1 = > PC
RTL	742010	2,3	Rotate two left. The content of the AC and the L are rotated two positions to the left. RTL is equivalent to two successive RAL instructions. ACj = > ACj − 2 AC1 = > L AC0 = > AC17 L = > AC16
RTR	742020	2,3	Rotate two right. The content of the AC and L are rotated two positions to the right. RTR is equivalent to two successive RAR instructions. ACj = > ACj + 2 L = > AC1 AC17 = > AC0 AC16 = > L
CLL	744000	2	Clear link. The L is cleared to contain a binary 0. 0 = > L
STL	744002	2,3	Set link. The L is set to contain a binary 1. 0 = > L, then $\overline{\text{L}}$ = > L, ∴ 1 = > L
RCL	744010	2,3	Clear link, then rotate Left. The L is cleared, then the L and AC are rotated one position left. RCL = CLL RAL
RCR	744020	2,3	Clear link, then rotate Right. The L is cleared, then the L and AC are rotated one position right. RCR = CLL RAR
CLA	750000	2	Clear accumulator. Each bit of the AC is cleared to contain a binary 0. 0 = > AC0-17

TABLE 2 OPERATE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Event Time	Operation Executed
CLC	750001	2,3	Clear and complement accumulator. Each bit of the AC is set to contain a binary 1. CLC = CLA CMA
LAS	750004	2,3	Load accumulator from switches. The word set into the ACCUMULATOR switches is loaded into the AC. LAS = CLA OAS
GLK	750010	2,3	Get link. The content of L is set into AC17. GLK = CLA RAL

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 content of the AC = 0, the content of the L = 0, or both. When the sense of the skip is inverted (bit 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 AC = 0, the L = 1, or both. The skip occurs only if both AC $\neq$ 0 and 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.

Group 2 Operate Instructions (LAW)

The group of augmented instructions with an operation code of 74_8 and containing a 1 in bit 4, is used to load the entire instruction word into the accumulator. Since this group performs the same operation regardless of the content of bits 5 through 17, it can be considered as one instruction. Format of this instruction is shown in Figure 15.

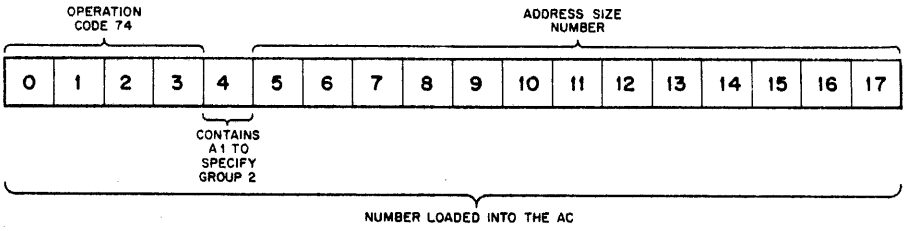


Figure 15 Group 2 (LAW) Operate Instruction Bit Assignments

This instruction can be defined as follows:

Mnemonic Symbol	Octal Code	Machine Cycles	Operation Executed
LAW	76XXXX	1	Load Accumulator With. The AC is loaded with the entire instruction word contained in the MB. MB => AC

Use of this instruction can be applied to load an address-size number into the accumulator without using an extra core memory location. The LAW instruction is used to:

a. load memory addresses for use in indirect addressing

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

c. initialize word count in I/O devices, such as magnetic tape equipment

d. preset the real time clock counter

As used in these examples, only bits 5-17 of the LAW instruction are regarded as the addresses, characters, and counts, although the entire word is contained in the AC. This instruction should be used with care on machines having extended core memory capacity when operating in the extend mode, since bits 3 and 4 are used to select addresses above 8K.

Example:

```
LAW     1234    /OCTAL NUMBER 761234 IS ENTERED
                /INTO THE AC

DAC     15      /THE CONTENT OF THE AC IS STORED
                /IN MEMORY LOCATION 15
```

To initialize a core 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 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 17023. Since 760000 is the operation code for LAW, the resulting program sequence is used:

```
LAW     17023
DAC     Y
```

In actual practice this operation is seldom used, since the PDP-7 Symbolic Assembler has defined the special character LAM to load negative numbers for counting or masking purposes. The character LAM is a special case of the LAW instruction, equal to LAW 17777.

SECTION 4

BASIC MACHINE LANGUAGE PROGRAMMING

MEMORY ADDRESSING

When planning the location of instructions and data in core memory, remember that the following locations are reserved for special purposes:

Address	Purpose
0_8	Stores the content of the program counter, extended program counter, TRAP flip-flop, EXTEND flip-flop, and link following a program interrupt.
1_8	Stores the first instruction to be executed following a program interrupt.
7_8	Stores real time clock count.
10_8 through 17_8	Auto-indexing registers.
20_8	Stores the content of the program counter, extended program counter, TRAP flip-flop, EXTEND flip-flop, and link during execution of a CAL instruction.
21_8	Stores the first instruction of the subroutine entered through the CAL instruction.

Usually addresses 0_8 through 77_8 are used to store special control words, address, or word counts and data and routines are stored from address 100 through the rest of core memory.

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 to another location. The indirect instruction appears as:

ADD I 100 where, the content of location 100 = 001357

where I signifies indirect addressing. The processor interprets the content of register 100 as the address of the instruction operand and in the next memory cycle adds the content of location 1357 to the content of 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 core memory field of a PDP-7 computer system contains eight auto-indexing memory registers in addresses specified in Table 3. When one of these locations is used as an indirect address, the content of that location is automatically incremented by one, and the result is taken as the effective address of the instruction. The incrementing is done with no added instruction time. Note that incrementing of an auto-index location occurs only on an indirect reference; for direct addressing the auto-index locations are identical to other memory locations.

TABLE 3 AUTO-INDEX REGISTERS IN EACH MEMORY FIELD

Memory Word Capacity	Memory Fields	Relative Address	Addresses of Auto-Indexing Registers
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, 20010-20017_8
24K	0,1,2	10-17	10_8-17_8, 20010-20017_8 40010-40017_8
32K	0,1,2,3	10-17	10_8-17_8, 20010-20017_8 40010-40017_8, 60010-60017_8

Example:

Assume four memory locations initially have the following content:

Location	Content
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, the content of location 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 $Y = \sum_{i=1}^{N} Xi$

Tag	Location	Content	Remarks
	10/	FIRST -1	/LOCATION OF FIRST WORD-1
	COUNT	-N + 1	/TWO'S COMPLEMENT OF NUMBER OF /ADDITIONS
ENTRY,	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: $Ci = Ai + Bi$ 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.

Tag	Location	Content	Remarks
	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: $Cj = Cj + K \quad j = 1, 2, \ldots N$

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 the extend mode, only the address bits are used in generating the effective address.

Tag	Location	Content	Remarks
	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 IN TABLE, GO BACK
	CONTINUE		/CONTINUE WITH PROGRAM

ARITHMETIC OPERATIONS

Two arithmetic instructions are included in the PDP-7 order code, the one's complement add: ADD Y, and the two's complement add: TAD Y. Using these instructions, routines can easily be written to perform addition, subtraction, multiplication, and division in either one's complement or two's complement arithmetic.

Complement Arithmetic

In complement arithmetic addition, subtraction, multiplication, and division of binary numbers is performed in accordance with the common rules of binary arithmetic. In PDP-7 as in other machines utilizing complementation techniques, negative numbers are represented as the complement of positive numbers, and subtraction is achieved by complement addition. Representation of negative values in one's complement arithmetic is slightly different from that in two's complement arithmetic.

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

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

A one's complement representation of a negative number is always one less than the two's complement representation of the same number. Differences between one's and two's complement representations are indicated in the following list.

Number	1's Complement	2's Complement
+5	000000000101	000000000101
+4	000000000100	000000000100
+3	000000000011	000000000011
+2	000000000010	000000000010
+1	000000000001	000000000001
+0	000000000000	000000000000
-0	111111111111	Nonexistent
-1	111111111110	111111111111
-2	111111111101	111111111110
-3	111111111100	111111111101
-4	111111111011	111111111100
-5	111111111010	111111111011

Note that in two's complement there is only one representation for the number which has the value zero, while in one's complement there are two representations. Note also that complementation does not interfere with sign notation in either one's complement or two's complement arithmetic; bit 0 remains a 0 for positive numbers and a 1 for negative numbers.

To form the two's complement of any number, the one's complement is formed, and the result is incremented by one. This is accomplished by the instruction CMA followed by an ISZ instruction for a number in a known core memory location as follows:

```
LAC Y
CMA
DAC Y
ISZ Y
NOP
```

Addition

The addition of a number contained in a core memory location and the number contained in the accumulator is performed directly by using the ADD Y or the TAD Y instruction, assuming that the binary point is in the same position and that both numbers are properly represented in the appropriate complement arithmetic. Addition can be performed without regard for the sign of either the augend or the addend. Overflow is possible, in which case the result will have an incorrect sign, although the 17 least significant bits will be correct. Following the addition a test for overflow can be made by using the SZL command.

Subtraction

Subtraction is performed by complementing the subtrahend and adding the minuend. As in addition, if both numbers are represented by their one's or two's complement, subtraction can be performed without regard for the sign of either number. Assuming that both numbers are stored in core memory, a routine to find the value of A-B follows:

1's Complement

```
LAC B     /LOAD SUBTRAHEND
CMA       /FORM 1'S COMPLEMENT
ADD A     /-B PLUS A = RESULT IN AC
```

2's Complement

```
ONE,  0001      /CONSTANT
      LAC B     /LOAD SUBTRAHEND
      CMA       /FORM 1'S COMPLEMENT
      TAD ONE   /FORM 2'S COMPLEMENT
      TAD A     /-B PLUS A = RESULT
                /IN AC
```

Multiplication and Division

The nature of the algorithms for multiplication and division make their explanation here impractical. An understanding of these operations is best gained by studying the program descriptions and listings in the Digital Program Library.

INPUT/OUTPUT FUNDAMENTALS

Program Flags

The status of each I/O device is indicated to the processor by flag signals. A program reads the flag status of a device and initiates appropriate action. In this way, input/output transfers and program operations are easily coordinated. Flags are connected to the program interrupt control, status bits, and the input/output skip facility. A flag is an electrical level which indicates the status of part or all of an I/O device. 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 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 can indicate:

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

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

3. Connected to the status register, a flag can 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

Input/Output Status

The status of each I/O device, as indicated by its flags, can be read into assigned bits of the AC. Figure 16 shows the standard assignment for the commonly used devices. An asterisk 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 can be read into the AC at one time using the I/ORS I/OT instruction.

I/ORS	700314	Input/Output Read Status. The content of given flags replace the content of the assigned AC bits.

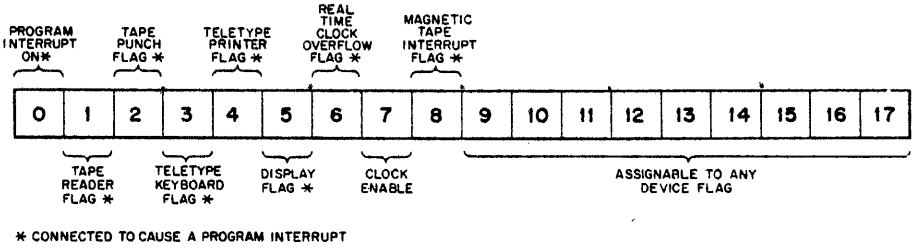


Figure 16 I/ORS Instruction Status Bit Assignments

Input/Output Skip Facility (I/OS)

The input/output skip facility enables the program to branch according to the status of an external device. The I/OS has fourteen flag inputs and is expandable to any number, seven of which are used by the basic computer equipment. When an input/output skip instruction is executed, the DS sends I/OT 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 a 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 Type R141 modules, each of which contains seven additional skip inputs. A -3 volt signal indicates the presence of a flag.

Commonly used skip instructions are:

CLSF	700001	Skip if real time clock has overflowed.
RSF	700101	Skip if perforated tape reader buffer holds a character.
PSF	700201	Skip if perforated tape punch is ready.
KSF	700301	Skip if teleprinter keyboard buffer holds a character.
TSF	700401	Skip if teleprinter is ready to receive a character.
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 holds 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 interrupt break in place of the I/OT instruction. Two other conditions are also trapped, an XCT instruction whose subject instruction is also 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 operator console TRAP switch on. The trap mode is disabled by any program interrupt 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 interrupt 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.

I/OT	An I/OT 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 interrupt break in place of the trapped instruction increments the program counter and stores its content 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 interrupt is turned off. The next instructions are taken from the appropriate I/O service routine, which begins in location 2.

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 interrupt 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 I/OT instructions control the PIC:

Mnemonic Symbol	Octal Code	Operation Executed
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 been effectively executed. The interrupt is then disabled and the extend mode is turned off. The format of the word stored in location 0 is indicated in Figure 17.

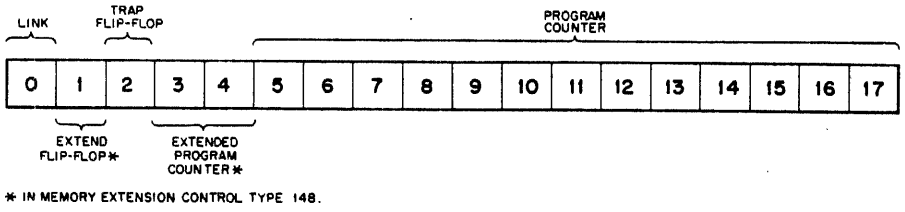


Figure 17 Information Stored in Address 000000 During a Program Interrupt

Example:

When the program interrupt is used to free the 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	Mnemonic	Tag	Remarks
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 perforated tape reader (300 cps), perforated tape punch (63 cps) and teleprinter (10 cps) were all operating at full speed simultaneously through the PIC, the percent of computer time taken for I/O servicing is roughly:

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

In this case,

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

or the time required to service the perforated tape reader, punch, and teleprinter operating simultaneously is 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 PIE (program interrupt enabled) indicator.

Real Time Clock

The clock produces a pulse every 1/60 second (6.7 milliseconds). When the clock is enabled, every clock pulse causes a clock break. The clock break interrupt is similar to a data break in that the content of the active registers are not changed. This interrupt has priority over a program interrupt but is of lower priority than a data break. During the interrupt the content memory location 7 are incremented by 1. If the content of location 7 overflows, the clock flag is set to 1. The clock flag is connected to the program interrupt system and causes a program interrupt.

Three I/OT instructions are associated with the clock:

Mnemonic Symbol	Octal Code	Operation Executed
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. Pressing the START key on the operator console clears the clock flag and disables the clock. Memory location 7 is not incremented unless the program is running, thus a halt prevents a clock interrupt but does not disable the flag.

Since the clock register is in core memory loacation 7, it can be loaded or deposited by a program. A standard technique for using the clock is to preset the content 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 Break Channel

This facility allows one high-speed input/output device, such as a magnetic tape or drum unit, to operate independently of the computer program (after a short initializing sequence) and transfer data with core memory at device-determined times on a cycle-stealing basis. When the device needs to transfer data into or out of core memory it provides a request to the computer. Since the data break has priority over all other breaks or interrupts, this request is granted at the completion of the current instruction (within three machine cycles, maximum). When the break occurs, the program is suspended for one cycle while the data is transferred between the device and the MB, then the program is resumed. The break does not affect the AC, PC, or IR so program conditions are not stored, as in program interrupts, but are held static for a one-cycle delay. The core memory address of each break and the direction of the transfer (into or out of core memory) is specified by signals from the device. A transfer rate of 570,000 18-bit words/second (1,710,000 6-bit characters/second) is possible.

The external device requesting the break must supply 15 address lines, 18 input/output data lines, a break request line, and a transfer direction signal. All signals are −3 volts for assertion, ground for 0. To accomodate slow I/O devices, the external device may request the computer to slow its cycle for the duration of the transfer.

*LAM is a pseudo-instruction to the assembler which generates the equivalent negative number in machine language using a LAW instruction.

The optional Type 173 Data Interrupt Multiplexer increases the data break facility to four channels arranged in a priority sequence. Thus, several high-speed devices such as a Type 57A tape control, a Type 24 Serial Drum, etc., can 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 Automatic Magnetic Tape Control and Type 24 Serial Drum do not require this data control. Maximum transfer rate is 570 KC words/second.

SECTION 5

PROCESSOR OPTIONS

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 (SC), two sign registers and the EAE control logic. The two panels of EAE logic are installed just below the operator console in bay 2 of the PDP-7 computer. The content of the MQ register is continually displayed on the operator 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. The bit configuration for the EAE instructions is shown in Figure 18 and defined in Table 4. The set-up phase of the instruction is broken up into three event times. Microprogramming for all but the set-up commands uses only the first two event times. The bits corresponding to the third event time then specify the step count of commands such as multiply, divide, and the shifts. The unassigned operation code (010) should not be used as it is reserved for future EAE expansion.

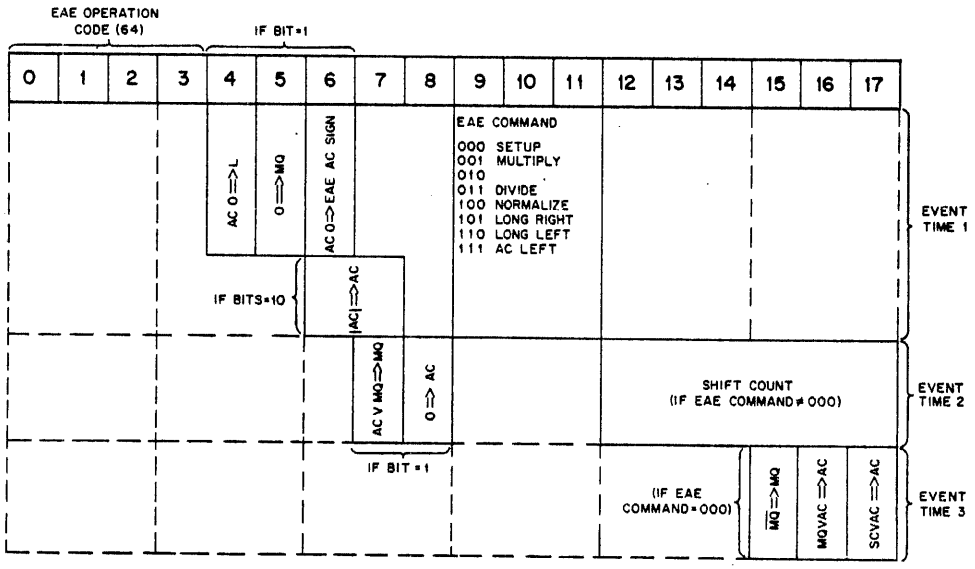


Figure 18 EAE Instruction Bit Assignment

TABLE 4 EAE BIT ASSIGNMENTS AND OPERATIONS

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.
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

TABLE 4 EAE BIT ASSIGNMENTS AND OPERATIONS (continued)

Bit Positions	Bits	Function
9, 10, 11	001 (continued)	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 μsec, 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 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 a 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 μsec in the case of divide overflow or from 9.0-12.6 μsec 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 μsec, where n is the step count.

TABLE 4 EAE BIT ASSIGNMENT AND OPERATIONS (continued)

Bit Positions	Bits	Function
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 ≠ 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 μsec, where n is the number of steps in the shift counter or the number required to effect normalization, whichever is less.
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 μsec, 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).

Bit assignments for EAE setup, multiply, divide, normalize and shift instructions are shown in Figures 19 through 23.

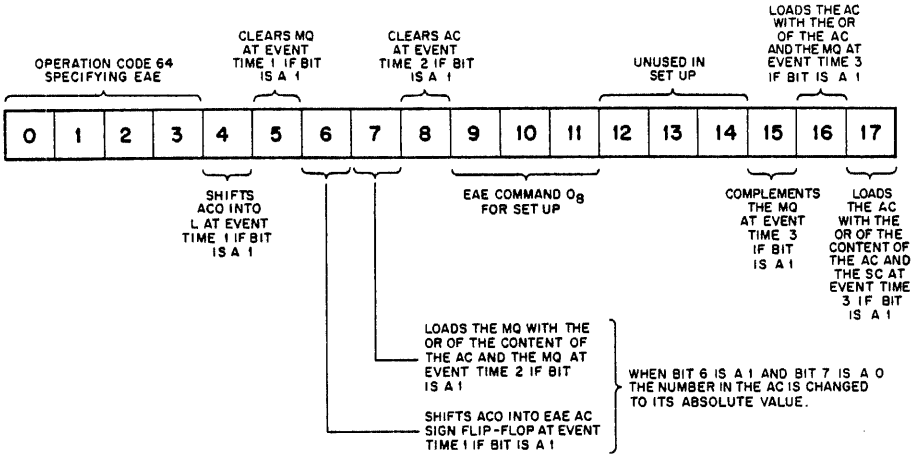


Figure 19 EAE Setup Instruction Bit Assignments

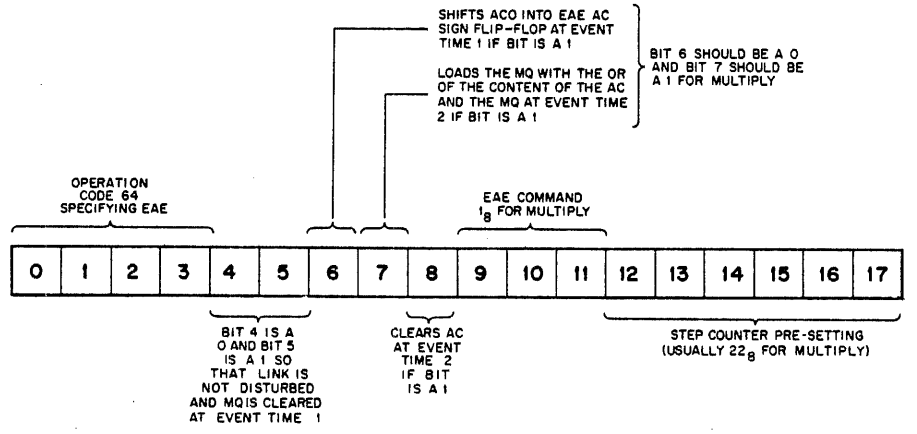


Figure 20 EAE Multiply Instruction Bit Assignments

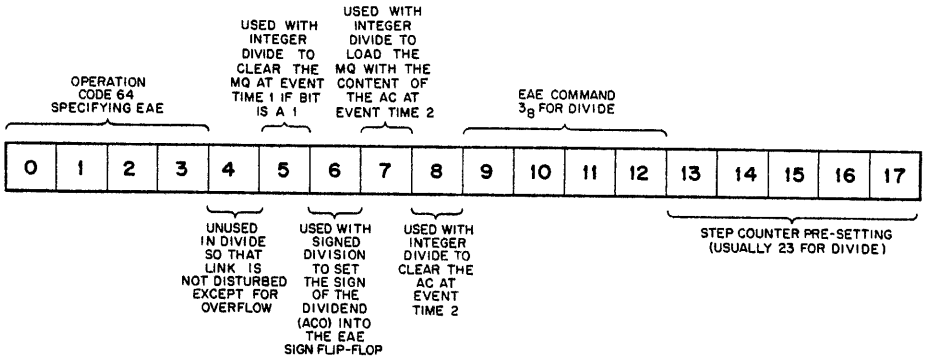


Figure 21 EAE Divide Instruction Bit Assignments

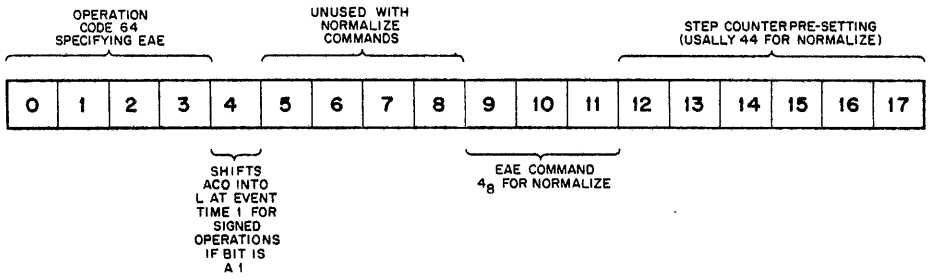


Figure 22 EAE Normalize Instruction Bit Assignments

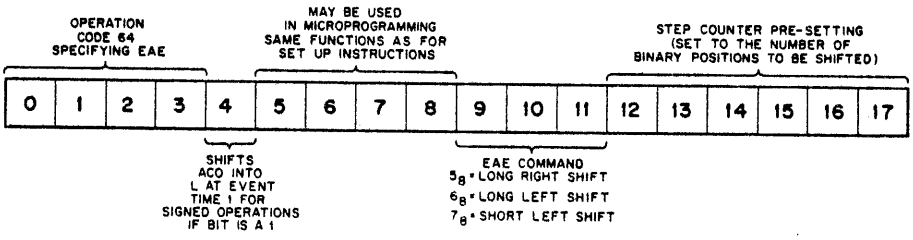


Figure 23 EAE Shift Instruction Bit Assignments

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 μsec plus 0.1 μsec/step.

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

Since the EAE expects to find the multiplier or the divisor in the location following the multiply or divide instruction, a short subroutine is usually used to setup 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 EAC AC sign register. If this register contains a 1, the MQ 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 AC 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_{10}. 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 contain a 0.

A list of microprogrammed EAE register manipulation instructions is given in Table 5. 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.

TABLE 5 EAE INSTRUCTION LIST

Mnemonic Symbol	Octal Code	Operation Executed
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 unsigned. Maximum shift is 44_8.
NORMS	660444	Normalize, signed (AC sign = L).
MUL	653122	Multiply the number in the AC by the number in the core memory addressed by the PC as 18-bit unsigned numbers, leave result in AC and MQ. The link must be 0.
MULS	657122	Multiply signed, the number in the AC by the number in the core memory address currently designated by the PC. The multiplier must be positive and its original sign must be in the link. The signed result appears in AC and MQ right adjusted.
DIV	640323	Divide the content of both the AC and MQ as a 36-bit unsigned number by the number in the core memory location currently specified by the PC. Leave quotient in MQ and remainder in AC. The link must be 0.
DIVS	644323	Divide the content of both the AC and MQ as a 1's complement signed number by the

TABLE 5 EAE INSTRUCTION LIST (continued)

Mnemonic Symbol	Octal Code	Operation Executed
DIVS (continued)		number in the core memory location currently specified by the PC. The divisor 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 the number in the AC as an 18-bit unsigned integer by the number in the core memory location currently specified by the 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 the content of the AC is a 17-bit signed and the usual convention on the divisor and link apply.
FRDIV	650323	Fraction divide. Divide the 18-bit fraction in the AC by the 18-bit fraction in the number in the core memory location currently specified by the 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 the content of the AC 17-bit signed and the usual conventions of the divisor and link apply.
LACQ	641002	Replace the content of the AC with the content of the MQ.
LACS	641001	Replace the content of the AC with the content of the 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

TABLE 5 EAE INSTRUCTION LIST (continued)

Mnemonic Symbol	Octal Code	Operation Executed
GSM (continued)		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.
LMQ	652000	Load MQ from AC, leave AC unchanged.

EAE Programming Examples

Example 1:

Exchange right and left halves of the accumulator as shown in Figure 24.

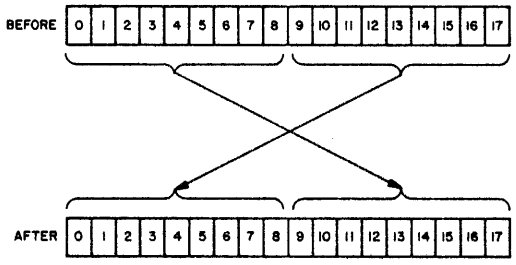


Figure 24 EAE Example 1, Problem

Approach: Since the EAE may be represented as shown in Figure 25, the right half of the accumulator may be shifted into the left half of the MQ.

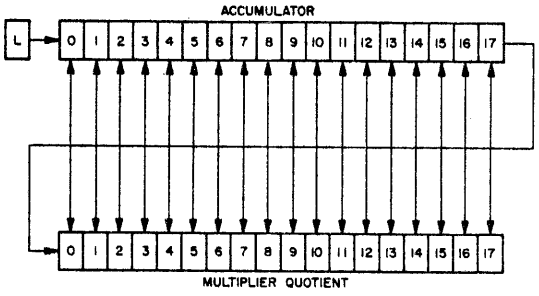


Figure 25 EAE Example 1, Approach

```
CLL                     /CLEAR LINK
CLQ V LRS 11            /CLEAR MQ
                        /RIGHT SHIFT 9 DECIMAL
OMQ                     /OR MQ INTO AC
```

Timing = 1.75 + 1.6 + (0.1) (9) + 1.75
= 6.0 microseconds

Example 2:

Given two 18-bit words in memory locations A and B, pack the five high-order bits of each word into the right hand ten bits of the accumulator.

Approach: As shown in Figure 26.

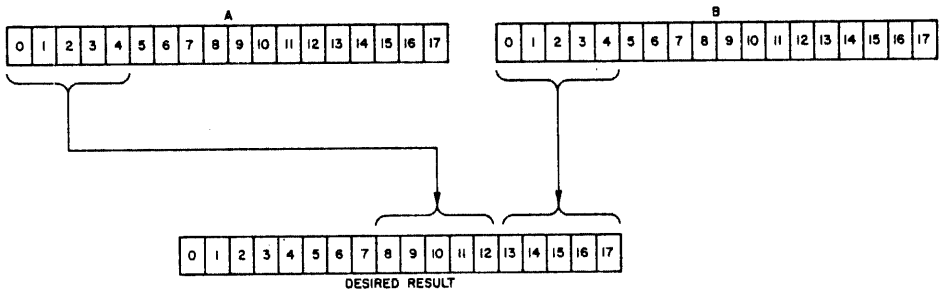


Figure 26 EAE Example 2, Approach

```
CLL                     /CLEAR LINK TO INITIALIZE SHIFTING
LAC B                   /LOAD AC WITH SECOND WORD
LRS 27                  /SHIFT CHARACTER INTO MQ, FILL
                        /VACATED BITS WITH ZEROS
LAC A                   /LOAD AC WITH FIRST WORD
AND (760000             /MASK OFF UNUSED BITS
OMQ                     /OR MQ INTO AC
LRS 10                  /LONG RIGHT SHIFT TO JUSTIFY
                        /CHARACTER, FILLING VACATED BITS
                        /WITH ZEROS
```

Timing = 20.3 microseconds

Example 3:

Multiplication of a constant by a 3-bit number. Result in AC.

Since the multiplier, in this case considered a constant, is fetched from memory; it will appear in the memory buffer register during the multiplication. The multiplicand, in this case a 3-bit number, is placed in the MQ and the AC is cleared. The multiply algorithm may be represented as shown in Figure 27.

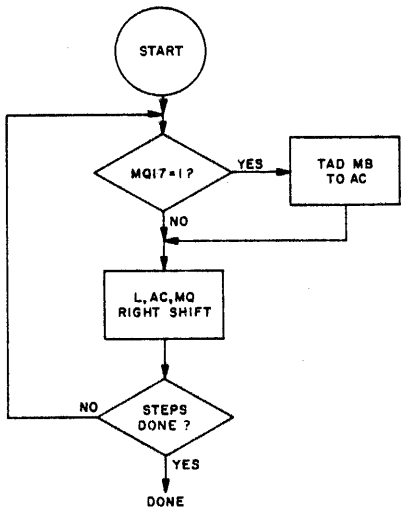


Figure 27 EAE Example 3, Approach Algorithm

```
                        /3 BIT NUMBER IN AC
        MUL -17         /3 STEPS 22₈ - 17₈ = 3
        XY              /CONSTANT
        LLS +3          /LEFT SHIFT 3 PLACES TO PUT
                        /RESULT IN AC
```

If the constant is 000071 and the AC contains 000005, the result at point XY is 000043 in t AC and 500000 in the MQ. After the shift the AC contains 000435.

Example 4: Signed Divide Closed Subroutine

```
/SIGNED DIVIDE SUBROUTINE                    37-42 μsec
/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
```

Example 5: Signed Multiply Closed Subroutine

```
/SIGNED MULTIPLY SUBROUTINE          25-31 μsec
/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
                LAC I MPY
                MULS
                0                    /LOCATION OF MULTIPLICAND
                ISZ MPY              /INDEX RETURN
                JMP I MPY
```

AUTOMATIC PRIORITY INTERRUPT TYPE 172

The 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 option identifies an interrupting device directly without the need for flag searching. Multilevel interrupts are permissible where a device at 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. A priority just below that of the data break channel is assigned to the Type 172. This is the priority normally held by the real time clock in PDP-7 systems not having the automatic priority interrupt option. The clock flag and clock overflow flag are usually assigned to channels 17_8 and 16_8 of the Type 172 option. The priority interrupt system operates in either 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:

a. The 172 is in the enabled condition (by program control).

b. There is no data interrupt request present.

c. The requesting channel is in the enabled condition (by program control).

d. There is no interrupt in progress on a channel of higher priority.

e. There is no interrupt in progress on the requesting channel.

When an interrupt is granted, the content of the channel memory location is transferred to the MB and executed. If the instruction executed is JMS Y, the system operates in the multi-instruction subroutine mode. The content 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 content 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 172, but can be cleared by an I/OT 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 I/OT and a jump indirect to location Y, where the content of the PC prior to interrupt are stored. The debreaking I/OT requires no channel designator, since the interrupt priority chain automatically releases the correct channel and returns it to the receptive state. This I/OT 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 I/OT releases channel.
3011	JMP I 3000	Return to main program sequence.
1001	---	Next instruction executed from here unless another priority interrupt is waiting:

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 instructions listed in Table 6 are added to the PDP-7 with the installation of the Type 172 option. Some instructions for example CAC and ASC, require that a channel number be contained in the AC for execution.

TABLE 6 PRIORITY INTERRUPT INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
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	700044	Enable automatic priority interrupt system. Same as real time clock CLON.
DPI	700004	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.

TABLE 6 PRIORITY INTERRUPT INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
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.

DATA INTERRUPT MULTIPLEXER CONTROL TYPE 173

The Data Interrupt Multiplexer Type 173 permits four high-speed input/output devices to operate with 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 break-interrupt facility, the following events occur:

1. The multiplexer switches to the device.
2. The 1.75-microsecond data break cycle begins.
   a. At time 1, the computer samples the 15 address lines.
   b. At time 3, data is transferred in or out of core memory through the multiplexer.

The following control signal lines pass from the 173 to each external device and are gated to only the transferring device:

Signal	Time	Characteristics
a. Address Accepted	T 1	Standard 70-nsec negative pulse

Signal	Time	Characteristics
b. Data Accept	T3	Standard 70-nsec negative pulse used only when transfer direction is into PDP-7.
c. Data Ready	T3	Standard 70-nsec negative pulse used only when transfer direction is out of PDP-7.
d. MPXB Select	When device is selected.	Standard negative level (−3 volts) when selected, ground when not selected.

The following control signal lines pass from each device to the 173 (these signals are delayed approximately 25 nsec before being applied to the computer):

Signal	Time	Characteristics
a. Data Break Request	Must be received by processor before T5	Standard negative level (−3 volts) for request, ground for no request.
b. Transfer Direction	Must be received by processor before T5	Standard negative level (−3 volts) for into PDP-7, ground for out of PDP-7.

The 18-bit data signals (−3 volts for assertion) and 15-bit address signals (−3 volts for assertion) should be set up at the time the device makes the request.

The major elements of the Type 173 Data Interrupt Multiplexer and the interface signal flow between it and both the processor and the devices are shown in Figure 28.

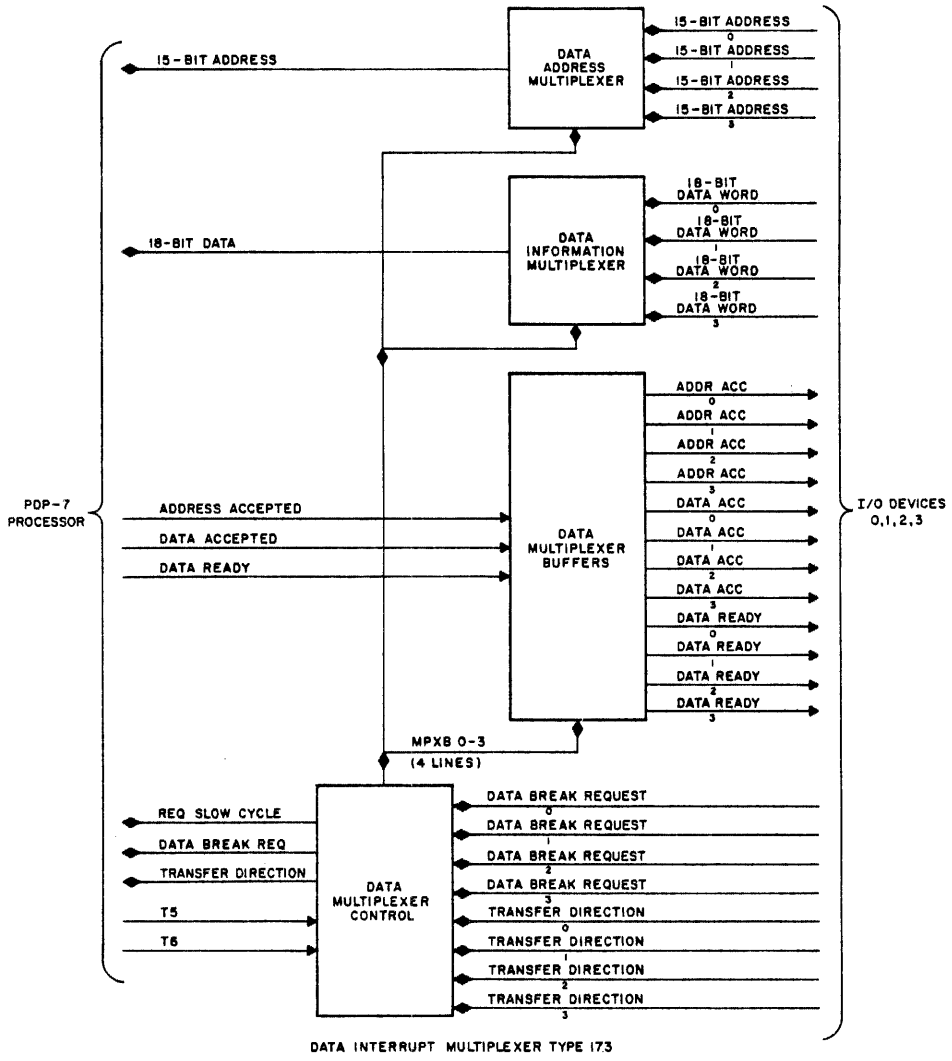


Figure 28 Data Interrupt Multiplexer Type 173

SECTION 6

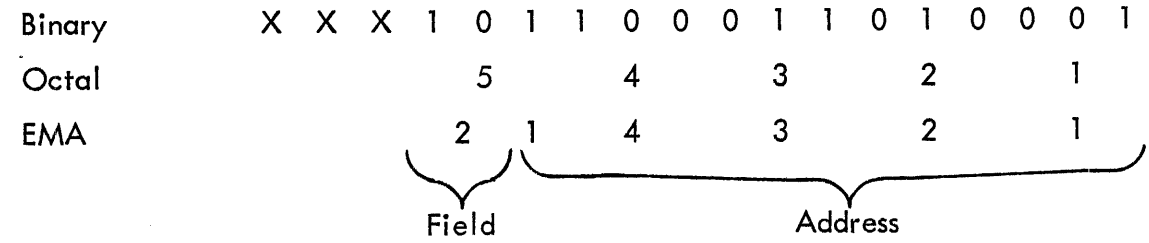
CORE MEMORY OPTIONS

MEMORY EXTENSION CONTROL TYPE 148

The Type 148 Memory Extension Control allows expansion of the PDP-7 core 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 a 2-bit extended program counter, a 2-bit 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 at 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 addresses location 00020 of field 0. When not in the extend mode, it addresses 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. Figure 17 illustrates the configuration of bits which are stored in location 0 on a program interrupt.

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 through 00017

20010 through 20017

40010 through 40017

60010 through 60017

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

Mnemonic Symbol	Octal Code	Operation Executed
SEM	707701	Skip if in Extend Mode
EEM	707702	Enter Extend Mode
LEM	707704	Leave Extend Mode
EMIR	707742	Extend Mode Interrupt Restore. The EMIR turns on the extend mode and sets a flip-flop which restores a prior condition of that mode during the next JMP I instruction.

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 INTERRUPT ON
JMP I 0   /RETURN
```

The acutal effect on 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 and real time clock 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.

```
Tag             Instruction             Remarks
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.

CORE MEMORY MODULES TYPE 147 AND 149

The 4096-word memory in the standard PDP-7 is a Type 149A Core Memory Module. The 149A is a 8192-word memory implemented to 4096 words. Addition of a Type 147 Core Memory Module option is required to fully implement the Type 149A in the standard machine. One Type 147 option is required for any memory size above 4K words. The Type 149B Core Memory Module is a fully implemented 8192-word core memory that can be added only to a memory of 8K, 16K, or 24K capacity. The options required to obtain various storage capacities are as follows:

Word Capacity	Options Required
4096	None
8192	one 147
12288	one 147 and one 149A
16384	one 147 and one 149B
20480	one 147, one 149B, and one 149A
24576	one 147 and two 149B
28672	one 147, two 149B, and one 149A
32768	one 147 and three 149B

Any core memory size above 8192 words requires addition of a Type 148 Memory Extension Control option. Addressing a core memory up to 8192 words is accomplished as explained in Section 4 of this handbook. Addressing above 8K is accomplished as described for the memory extension control in the previous portion of this section.

SECTION 7

STANDARD INPUT/OUTPUT EQUIPMENT

Standard input/output equipment supplied with each PDP-7 system consists of the Teletype and Control Type 649, Perforated Tape Reader and Control Type 444B, and a Perforated Tape Punch and Control Type 75D.

TELETYPE MODEL 33 KSR AND CONTROL TYPE 649

The Teletype Model 33 Keyboard Send Receive (KSR) set 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 character code is listed in the Appendix 2. The teleprinter input and output functions are logically separate, and the programmer can consider the printer and keyboard as individual devices.

Keyboard

The keyboard control contains an 8-bit buffer line unit in (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:

Mnemonic Symbol	Octal Code	Operation Executed
KSF	700301	Skip if the keyboard flag is set to 1. If the flag is a 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 content of the buffer is placed in bits 10-17 of the AC and the keyboard flag is cleared.

Teleprinter

The teleprinter control contains an 8-bit buffer line unit out (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:

Mnemonic Symbol	Octal Code	Operation Executed
TSF	700401	Skip if the telepringer flag is set.
TCF	700402	Clear the teleprinter flag.
TLS	700406	Load printer buffer and select. The content of AC bits 10 through 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 AND CONTROL TYPE 444B

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 either alphanumeric or binary modes. In alphanumeric mode each select instruction causes one line of tape, consisting of eight bits, to be read and placed in the buffer. Blank tape lines are ignored. The absence of a feed hole causes the character punched in that line to be ignored. Alphanumeric tape format and character bit transfer assignments when loaded into the AC are shown in Figure 29. In the binary mode, the select instruction causes three lines of tape to be read. The first six bits of each line are assembled in the buffer, thus three tape characters form a single 18-bit computer word. The seventh bit is ignored. However, a character is not read unless the eighth bit is punched. Binary tape format and character bit assignments when loaded into the AC are shown in Figure 30. Tape reader instructions are listed in Table 7.

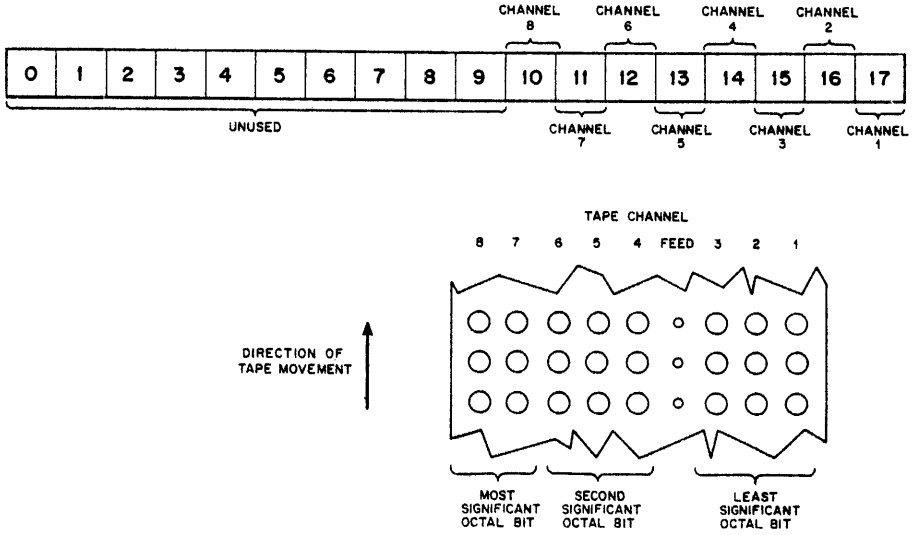


Figure 29 Tape Format and Reader Buffer Register Bit Assignments in Alphanumeric Mode

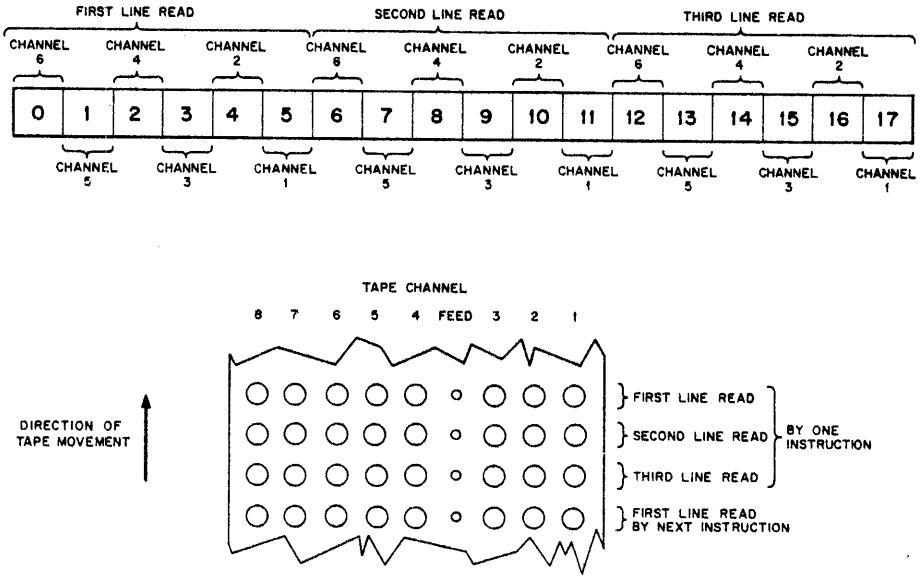


Figure 30 Tape Format and Reader Buffer Register Bit Assignments In Binary Mode

TABLE 7 TAPE READER INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
RSF	700101	Skip if reader flag is a 1.
RCF	700102	Clear reader flag then inclusively OR content of reader buffer into the AC. RBj V ACj ⇒ ACj
RRB	700112	Clear reader flag. Clear AC and then transfer contents of reader buffer to AC. RB ⇒ AC
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 to 1.
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.

PERFORATED TAPE PUNCH TYPE 75D

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 can punch blank tape (feed hole only punched) by pressing the punch FEED push-button on the console, or he can force the punch power on by setting the console PUNCH feed 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 processor is never delayed by this instruction nor any other I/O instruction. Subsequent punching follows at normal punch speed. The motor remains energized for five seconds after the last punch command is given.

When the punch is selected, the content of AC bits 10 through 17 are loaded into 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 always loaded by direct data transfer and can not be loaded by a logical OR transfer.

Information is handled by the punch logic in either alphanumeric or binary modes. In the alphanumeric mode each select instruction causes one line of tape, consisting of eight bits is to be punched. Each hole punched in a tape channel corresponds to a binary 1 in the appropriate bit of the punch buffer. A feed hole is punched for each punch command, even if the punch buffer contains all zeros. The correlation between tape channels and accumulator bits shown in Figure 28 applies to the tape punch in alphanumeric mode. In the binary mode each select instruction causes one line of tape, consisting of eight bits to be punched. Holes are punched in channels 6 through 1 as a function of binary ones in bits 12 through 17 of the accumulator, respectively. Channel 10 is always punched and channel 11 is never punched, thereby conforming to standard binary tape information format. The instructions for the tape punch are listed in Table 8.

TABLE 8 TAPE PUNCH INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
PSF	700201	Skip the following instruction if the punch flag is set to 1.
PCF	700202	Clear the punch flag.
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.

The following instruction causes a line of blank tape (except for feed hole) to be punched and clears the accumulator:

PSA +10	700214	Clear AC and punch.

The following instruction as used on the PDP-4 is also available on the PDP-7, but is generally replaced with the more direct PSA command:

PLS	700206	Same as PSA.

SECTION 8

CARD EQUIPMENT AND LINE PRINTER OPTIONS

CARD READER AND CONTROL TYPE CR01B

This device reads standard 12-row, 80-column punched cards at a maximum rate of 100 cards per minute. The 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. A punched hole is interpreted as a one and no hole as a zero. Column information is read in either alphanumeric or binary modes. In the alphanumeric mode holes (bits) in a column are interpreted as a Hollerith character code (see Appendix 2). These bits are translated into a 6-bit card reader code for the character which is read by the read data instruction. In the binary mode the 12 bits of each column are accepted directly as a 12-digit binary number and are transferred into AC bits 6 through 17.

Card Reader Operation

Holes in a card column are sensed by mechanical star wheels in the reader. When a column of data is ready to be transferred into the computer, a data ready flag is set. Data should be strobed into the computer with the CRRB instruction within 1.5 milliseconds after the ready flag is raised. A reader not ready flag indicates that the reader is energized but no card is in the read station.

The function of controls and indicators on this card reader is described in Table 9 and the instructions are listed in Table 10.

TABLE 9 CARD READER CR01B CONTROLS AND INDICATORS

Control or Indicator	Function
ON/OFF switch	This switch controls the application of primary power to the reader. When the power is applied, the reader is ready to respond to operation of the other keys or computer command signals.
AUTO-MAN switch	In the manual position this switch mechanically disables the card feed mechanism. In the auto position card eading under program control is enabled.
REG switch	The register key on a reader is used to feed the first card to the read station manually.

TABLE 9 CARD READER CR01B
CONTROLS AND INDICATORS (continued)

Control or Indicator	Function
SKIP switch	This key is not connected on the CR01B and has no effect on equipment operation.
CARD RELEASE pushbutton	When pressed, this pushbutton adjacent to the read station, releases a card already in the read station.
READY indicator	This indicator lights when the reader is energized and cards are present in the card hopper. The plastic card cover should always be used on top of a deck of cards to assure that the ready switch and indicator is activated.

TABLE 10 CARD READER CR01B INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
CRSF	706701	Skip on card reader flag. When the card reader flag is set, indicating that a card column is present and ready to be read, the next instruction is skipped. This flag is connected to the program interrupt facility.
CRSA	706704	Select alphanumeric. This instruction enables the reader logic to code punched data in BCD form, so that it can be presented to bits 12-17 of the AC during a CRRB command.
CRSB	706744	Select binary. This instruction enables the reader logic to present the bits read from the card in binary form so that it can be transferred into AC bits 6-17 during a CRRB command. Each column is read as a 12-bit binary number.
CRRB	706712	Read data. This instruction clears the AC, strobes data from the card reader into the AC in a format specified by the previous select instruction, and clears the data ready flag.

The logical program sequence for reading cards is:

1. Select the mode of reading; the select instruction also moves a new card into position at the read station.
2. Respond to the setting of the card reader flag. This may be done by permitting the flag to interrupt the computer program or by program looping on the status of the flag.
3. Read the data called for by the select instruction. The data ready flag is cleared as the data is read.

When reading the card columns, the program responds to the ready flag which is set as each new column is in place. The instruction CRRB must be given within 1.5 milliseconds following the setting of the ready flag.

CARD READER AND CONTROL TYPE 421

The card reader reads standard 12-row, 80-column punched cards at a maximum rate of 200 (for Type 421A) or 800 (for Type 421B) 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 I/OT 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 either alphanumeric or binary mode. In the alphanumeric mode the holes (binary ones) in a column are interpreted as a Hollerith character code (see Appendix 2). This character 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. In the binary mode 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

The card reader is shown in Figure 31. The feed hopper is at the right and the run-out stacker is at the left. Cards to be read are placed face down in the hopper, with the tops of the cards (12 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 shown in Figure 32 contains indicating switches which control the operation of the device and 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 indicator on the console is lit. The NOT READY condition exists until the START button is pressed, at which time the NOT READY indicator is extinguished. If a malfunction exists, the RESET control must be pressed first. The function of the console controls and indicators is presented in Table 11.

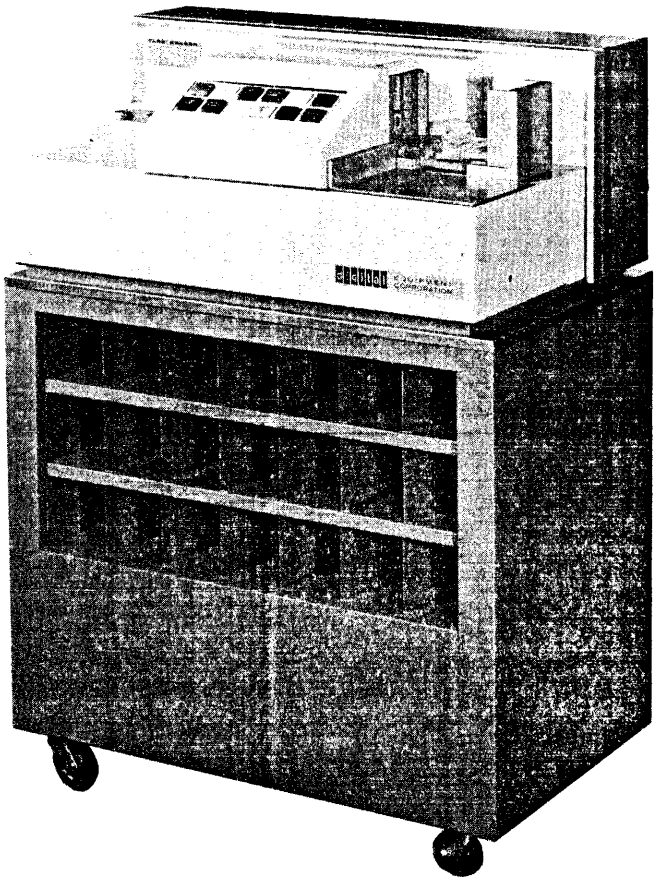


Figure 31 Type 421 Card Reader Console

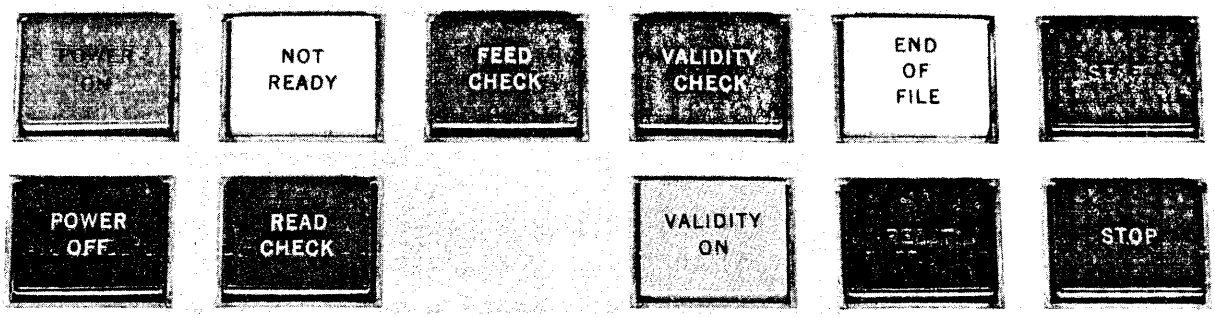


Figure 32 Card Reader Control Panel

TABLE 11 CARD READER 421 CONTROLS AND INDICATORS

Control or Indicator	Function
POWER ON and POWER OFF pushbuttons	These devices control the application of primary power to the reader. When the POWER ON pushbutton is pressed, its green indicator lights; the motors are started, and the drive rollers which move a card through the reader are set in motion.
START pushbutton	This pushbutton must be pressed to clear the NOT READY condition. Only then does the card reader accept a select instruction.
STOP pushbutton	If the reader is in operation when this pushbutton is pressed, the reading of the currently selected card is completed, the readers stops, and the NOT READY indicator lights. The START pushbutton must be pressed to make the reader available again.
RESET pushbutton	After a malfunction (see below) has occurred, the RESET pushbutton must be pressed to turn off the check indicator and clear the reader logic of the condition which caused the error. It does not turn off the NOT READY indicator.
END OF FILE pushbutton	To signal the program that no more cards are expected, press the END OF FILE pushbutton when the hopper is empty. The white indicator lights when this happens. If the hopper is not empty, pressing this pushbutton has no effect. The end-of-file condition is removed and the indicator is extinguished when cards are placed in the hopper.
VALIDITY ON pushbutton	If this pushbutton 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 control, which lights yellow when pressed, has no effect when reading in binary mode.
NOT READY indicator	When one of the conditions described above exists, this white indicator lights. As long as it is lit, the reader is not available to the program. The NOT READY indicator is turned off only by pressing the START pushbutton.

TABLE 11 CARD READER 421 CONTROLS AND INDICATORS (continued)

Control or Indicator	Function
READ CHECK, FEED CHECK, VALIDITY CHECK indicators	Each of these red malfunction indicators light whenever the corresponding error condition exists. In each case, the NOT READY indicator lights 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 indicator, after which, pressing START clears the NOT READY indicator and makes the reader available.

The card reader check indicators function as follows:

Read Check When a read check error occurs, it indicates that something is wrong in the reading circuits. 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 defective 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 pushbutton has been pressed, the NOT READY indicator lights and the reader stops. The card in error should be checked for improper punches before a second attempt is made to read it.

Appendix 2 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 I/ORS instruction is executed, the status of the flags is read into these bits (see Figure 16). These flags function as follows:

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 indicator on the reader console and is set or cleared by the same operations.

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

The card reader instructions are listed in Table 12.

TABLE 12 CARD READER 421 INSTRUCTIONS

Mnemonic Symbol	Octal Code	Function Executed
CRSF	706701	Skip if the card column flag is set.
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 indicator is lit, a validity check causes the reader to stop.
CRRB	706712	Read the card reader buffer. The content of the CRB is placed in bits 6-17 of the AC. The card column flag is cleared.
CRSB	706744	Select and read a card in binary mode. A card is started through the reader and 80 columns are read as 12-bit numbers. The VALIDITY ON pushbutton has no effect since validity checking is not performed during this mode.

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.

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 content 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.

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 is automatically shut down.

The card punch control contains an 80-bit punch buffer (CPB) into which information is placed for punching. 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 I/OT skip instruction and is connected to the PIC.

The card punch instructions are listed in Table 13.

TABLE 13 CARD PUNCH INSTRUCTIONS

Mnemonic Symbol	Octal Code	Function Executed
CPSF	706401	Skip if card flag is set. This flag is set when the CPB is ready to accept a new row.
CPLR	706406	Load the punch buffer, clear punch flag.
CPCF	706442	Clear the card row flag.
CPSE	706444	Select the card punch. This starts a card moving from the hopper to the punch station. Load the card punch buffer. This command transmits the content of the AC into the CPB. Five CPSE commands are required to fill the CPB.

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 content 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 Automatic Line Printer prints text in lines of up to 120 characters at a maximum rate of 300 lines per minute for the 647A, 600 lines per minute for the 647B, or 1000 lines per minute for the 647C. 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 a printer 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 pushbuttons, 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. The function of line printer controls and indicators is specified in Table 14.

TABLE 14 LINE PRINTER CONTROLS AND INDICATORS

Control or Indicator	Function
TRACTOR INDEX	Used for aligning the forms with the format tape when new paper is loaded. This pushbutton works only when the printer is off line.
PAPER LOW ALERT	This red indicator lights 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 red indicator lights. An interlock prevents all but the TOP OF FORM and TRACTOR INDEX controls from operating.
ALARM STATUS	Whenever an alarm signal is generated, this red indicator lights.
ON, OFF	These pushbuttons control application of primary power to the functioning parts of the printer. The main power switch must be turned on for these switches 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.

TABLE 14 LINE PRINTER CONTROLS AND INDICATORS (continued)

Control or Indicator	Function
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 pushbutton is used for maintenance at the printer and is not used in normal operation.
TOP OF FORM	Moves the paper to the top of the next page. This pushbutton works only when the printer is off line.

In addition to the above paper low alert, no paper, and yoke open alarms, 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 characters in the most significant bits of the AC are transferred before characters in less significant bits.

The buffer loading instructions perform the inclusive OR transfer of the content 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 occurs and can be reset only when the alarm condition is removed. Both flags are connected to the program interrupt control.

The instructions listed in Table 15 are added with the Type 647 Automatic Line Printer.

TABLE 15 AUTOMATIC LINE PRINTER INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
LPSF	706501	Skip if the printing done flag is a 1.
LPCF	706502	Clear the printing done flag.
LPL1	706562	Load the printing buffer with one character. The printing done flag is cleared, the content of AC bits 12-17 is transferred into the printing buffer, then the printing done flag is set.
LPL2	706522	Load the printing buffer with two characters. The printing done flag is cleared, the two character codes represented by bits 6-11 and 12-17 of the AC are transferred into the printing buffer in that order, then the printing done flag is set.
LPLD	706542	Load the printing buffer with three characters. The printing done flag is cleared; the three character codes represented by bits 0-5, 6-11, and 12-17 of the AC are transferred into the printing buffer in that order; then the printing done flag is set.
LPSE	706506	Select the printer and print. The printing done flag is cleared, the characters contained in the printing buffer are printed, then the printing done flag is cleared.
LSSF	706601	Skip if spacing flag is a 1. The spacing flag is set when spacing is completed.
LSLS	706606	Load the spacing buffer. The spacing flag is cleared, the space channel number contained in bits 15-17 of the AC is transferred into the spacing buffer, the spacing performed, then the spacing flag is cleared.

The status of the buffer and error flag is read into AC bits 15 and 16, respectively, by the I/ORS instruction.

SECTION 9

MAGNETIC TAPE AND DRUM OPTIONS

DECTAPE 550/555 SYSTEM

DECtape (DEC'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 one or more Type 555 DECtape Dual Transport and the Type 550 DECtape Control.

DECtape Dual Transport Type 555

The Type 555 DECtape Dual 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.

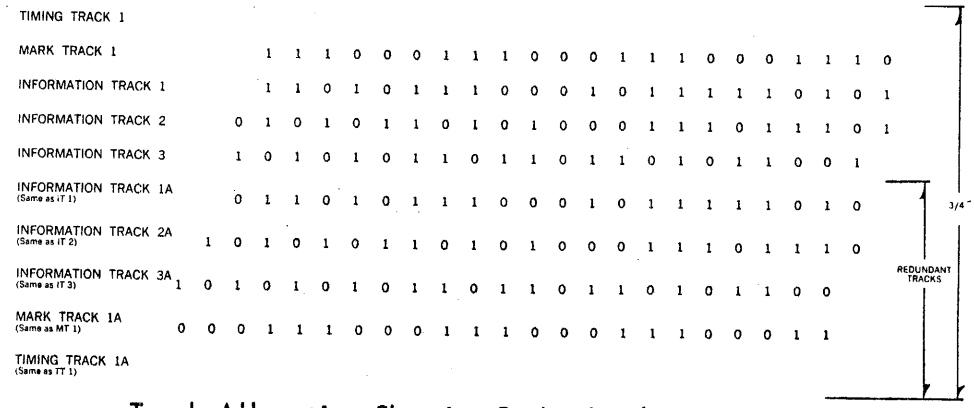
The 3-1/2 inch reels are loaded simply by pressing onto the hub, bringing the loose end of the tape across the tape head, attaching it to the take up reel, and spinning a few times. Individual controls on the transport enable the user to manipulate the tape in either direction manually. The units selection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

There are no capstans or pinch-rollers 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 times average 0.15-0.2 seconds and turn around takes approximately 0.3 seconds.

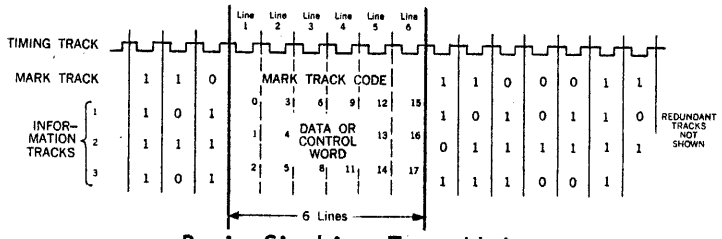
Recording Technique

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 binary 1 is desired. In the polarity sensed system a flux reversal of a particular direction indicates a binary 0 while a flux reversal in the opposite direction indicates a binary 1. A timing track, recorded separately in quadrature phase, is used to control strobing of the data tracks. Thus, the polarity of the signal at strobe time indicates the presence of a 0 or a 1. Using the timing track on the tape as the strobe also negates the problems caused by variations in the speed of the tape.

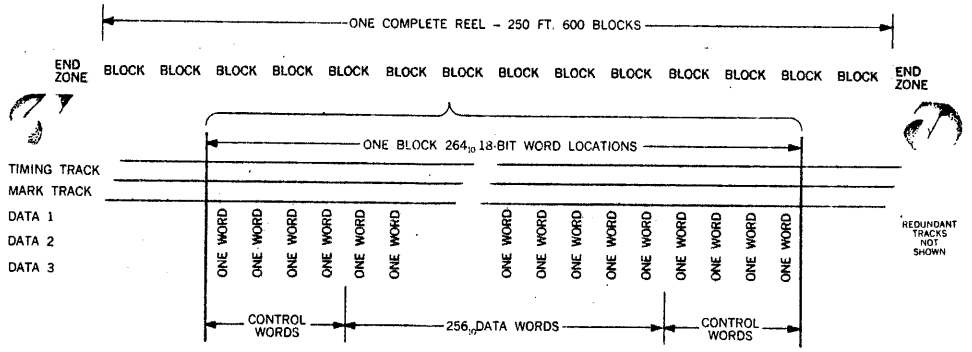
With this type of recording only the polarity, not the amplitude of the signal, need be considered, thus removing some of the signal to noise problems and allowing the use of read amplifiers with high uncontrolled gain. This recording also allows the changing of individual bits on the tape without changing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



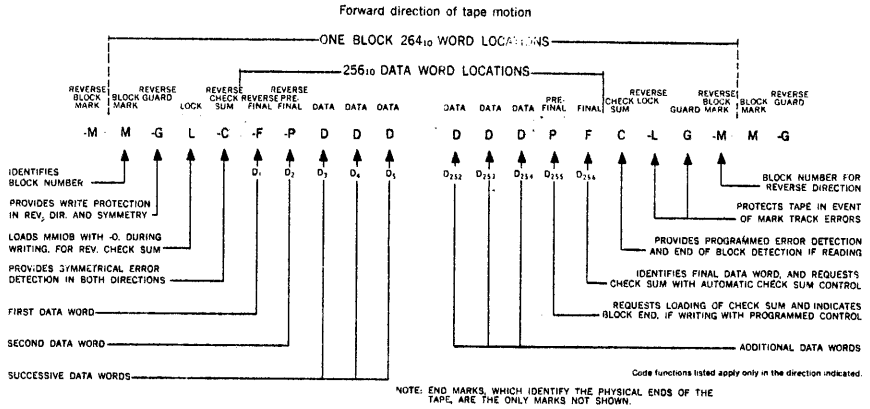
Track Allocation Showing Redundantly Paired Tracks



Basic Six Line Tape Unit



Control and Data Word Assignments



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

Figure 33 DECtape Format

Reliability is further increased by redundantly recording all five of the information tracks on the tape. This is accomplished by wiring the two head windings for each information track in series. On reading, the analog sum of voltage induced in 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 has no effect. Track, data block, and mark track information format is shown in Figure 33.

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 operator 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 acutal 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 operator 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 Program Library document Digital 7-20-I/O.

DECTRIEVE for the PDP-7 allows the user to save or retrieve data using the ACCUMULATOR switches on the operator console. To store data the user specifies the unit, block number, starting and ending core memory locations to be used. The data is saved together with appropriate control information and is 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 checksum 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 Program Library document Digital 7-21-I/O.

Table 16 lists the DECtape system instructions.

TABLE 16 DECTAPE INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
MMRD	707512	Read. Clears the AC and transfers one word the data buffer in the control to bits 0-17 of the AC.**
MMWR	707504	Write. Transfers one word from bits 0-17 of the AC to the data buffer in the control.**

**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.

TABLE 16 DECTAPE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
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:** 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. Clear the 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

**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.

TABLE 16 DECTAPE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
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 checksum should be unloaded via MMRD instruction. In write mode: data should be loaded via MMWR instruction.
MMBF	707601	Skip on DECtape block end flag. In read mode: unload forward checksum via MMRD instruction. In write mode: load calculated forward checksum 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 Type 57A tape control buffers, compiles, synchronizes, and controls data transfers between up to eight magnetic tape transports and the PDP-7, using the program interrupt control and the data break channel. Each transport type requires a small interface circuit for connection to the control. The interface required and the characteristics of the six types of transports that can be connected to the Type 57A Tape Control are listed in Table 17.

TABLE 17 TRANSPORTS AND INTERFACES USED WITH TAPE CONTROL TYPE 57A

Transport		Tape Speed (ips)	Densities (bpi)	Interface
Maker	Designation			
DEC	Type 50	75	200/556	Type 520
DEC	Type 545	45	200/556/800	Type 521
DEC	Type 570	75/112.5	200/556/800	Type 522
IBM	Model 729II,IV	75	200/556	Type 552
IBM	Model 7330	36	200/556	Type 552
IBM	Model 729V,VI	112.5	200/556/800	Type 552

Format of all of these tape mechanisms is IBM-compatible in odd or even parity. The following transport functions are controlled by the Type 57A, as a function of I/OT commands:

Write	Rewind
Write end of file	Rewind unload
Write blank tape	Gather write
Read	Scatter read
Read compare	Write continuous
Space forward	Read continuous
Space backward	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) Characters read from tape are assembled in the 18-bit DA and are taken, one 6-bit character at a time, from the DA to be written on tape.

Data Buffer (DB) This 18-bit secondary buffer transfers data between the DA and the MB in the PDP-7, under data break control.

Command Register (CR) Contains the 3-bit tape operation to be performed, as specified by the content of bits 9 through 11 of the AC.

Unit Register (UR) Contains the 3-bit select number (0-7) of the tape unit addressed for the current operation, as specified by the content of bits 15 through 17 of the AC.

Current Address Register (CA) Contains the 13-bit address of the memory cell involved in the next data transfer. The initial content of the CA is specified by bits 5 through 17 of the AC.

Word Count Register (WC) Contains the 13-bit 2's complement of the number of words involved in the transfer. The content of the WC is incremented by one after each word transfer. The initial content of the WC is specified by the content of bits 15 through 17 of the AC.

Tape operations, modes, and unit numbers are specified by the content of bits 7 through 17 of the AC. Tape control I/OT 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 break channel, effectively permitting simultaneous computation and data transfer.

The I/OT instructions used to perform these operations are briefly described in Table 18. For detailed instructions on using the Type 57A control, along with programming examples, refer to DEC's publication F-13(57A).

TABLE 18 AUTOMATIC MAGNETIC TAPE CONTROL
BASIC INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
MSCR	707001	Skip if the tape control is ready. This command 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 command senses the tape unit flag, which is set when the specified unit is ready for another operation. This flag is connected to the PIC.
* MCC	707401	Clear CA and WC.
MCA	707405	Clear CA and WC, and transfer the content of AC 5-17 into the CA. Loads the CA.
MWC	707402	Load WC. Transfers the content of AC 5-17 into the WC.
MRCA	707414	Transfer the content of the CA into AC 5-17.
MCD	707042	Disable TCR and clear CR. Clear WCO and EOR flags (see Table 16.)
MTS	707006	Clear control register (CR) and clear job done, WCO, and EOR flags. Transmit unit, parity, and density to tape control.
MTC	707106	Transmit tape command and start. This command initiates the transfer.
MNC	707152	End continuous mode. Clears the AC; the operation terminates at the end of the current record.
MRD	707204	Switch mode from read to read/compare. Allows mode switching during the operation.
MRCR	707244	Switch from read/compare to read.

*This basic instruction is described here as an aid to understanding the programming of the Type 57A, however it is not recognized by the PDP-7 Symbolic Assembler and is usually combined with other commands.

The commands listed in Table 19 deal with the two tape flags that determine when a transfer is complete. The word count overflow flag (WCO) is set when the WC becomes 0 after incrementing. The end of record (EOR) flag is set when the end of record (EOR) mark is sensed. Both flags are connected to the PIC.

TABLE 19 AUTOMATIC MAGNETIC TAPE CONTROL FLAG INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
MSEF	707301	Skip if EOR flag is set.
*MDEF	707302	Disable EOR flag. This command disconnects the EOR flag from the program interrupt.
*MCEF	707322	Clear EOR flag.
*MEEF	707342	Enable EOR flag. This command connects the EOR flag 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.

*These basic instructions are not recognized by the PDP-7 Symbolic Assembler and are always used in combination with other commands.

There are 11 status indicators associated with the Type 57A tape control. The states of all indicators can be observed by placing their content into the AC. This is done by an instruction similar to I/ORS, but applying only to the tape control. The instruction serves as follows:

MTRS 707314 Read tape status

AC bit	Indication when 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 can be connected to the PDP-7 using the Type 57A Automatic Magnetic 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 Automatic Magnetic Tape Control:

MAGNETIC TAPE TRANSPORT TYPE 545

The Type 545 is a digital magnetic tape transport designed for use with the Type 57A Automatic Magnetic Tape Control or Type 581 Tape System.

Specifications

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

Tape Width, 0.5 inch; length 2400 feet (1.5 mil); reels, 10-1/2 inch; IBM compatible with file protect ring.

Heads Write-read gap, 0.300 inch. Dynamic and static skew <20 microseconds.

Recording 45 ips. Rewind less than 3 minutes maximum. Start time <5 milliseconds. Start distance 0.080 inch + 0.035, −0.025 inch. Stop time <5 milliseconds. Stop distance 0.045 inch ±0.015 inch.

Density 200, 556, and 800 bpi. Maximum transfer rate is 36 kc.

Transport Mechanism Pinch roller drive, vacuum column tension.

Controls ON/OFF, REMOTE/LOCAL, FORWARD, REVERSE, REWIND.

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 the computer).

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 on one drum track. Depending on the drum capacity, there are 64, 128, or 256 tracks. From the programming point of view, the track may be ignored; the logical storage unit is the sector. Transfers are effected through the data break control with the drum system providing the data channel.

Two I/OT instructions are required to initiate the transfer of a block of data. The first I/OT specifies the core 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 I/OT 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.

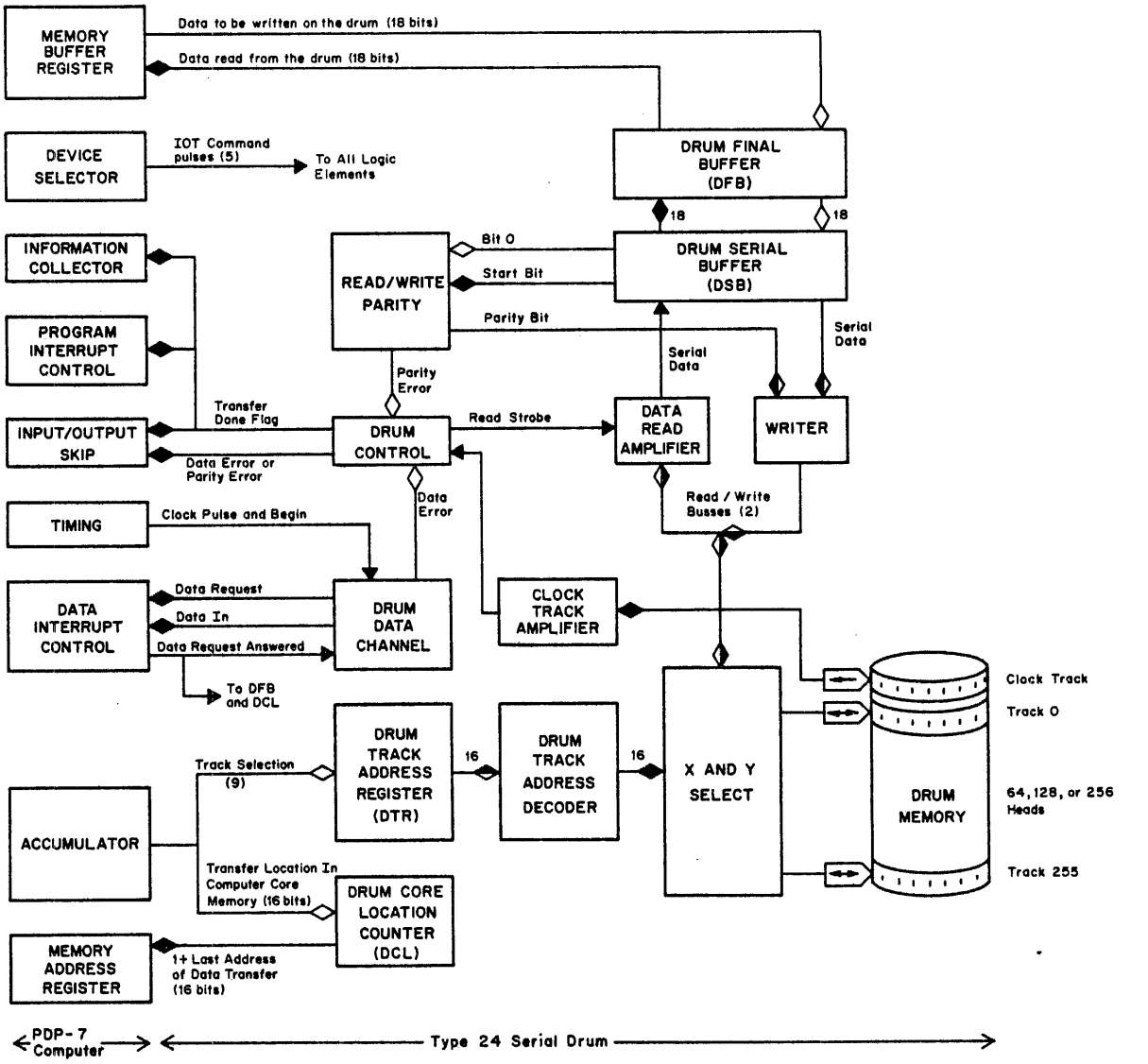


Figure 34 Serial Drum Block Diagram and Interface Connections

The logic elements that compose the serial drum and their interface with the computer are shown in Figure 34. The four major registers function as follows:

Drum Core Location Counter (DCL) The 16-bit 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 content of the DCL is incremented by 1.

Drum Track Address Register (DTR) The 9-bit DTR contains the address of the sector currently involved in a block transfer. At the completion of a successful transfer, the content of the DTR is incremented by 1.

Drum Final Buffer (DFB) This 18-bit register is a secondary buffer between the memory buffer and the drum serial buffer. In writing, a word taken from the MB is placed in the DFB to

await storage on the drum. In reading, the word assembled in the serial buffer is placed in the DFB; the next data break interrupt transfers it to the MB and stores it in core memory.

Drum Serial Buffer (DSB) On reading, a word is read serially and assembled in the 18-bit 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:

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

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

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

The I/OT instructions added with the drum system are listed in Table 20.

TABLE 20 SERIAL DRUM INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
DRLR	706006	Load counter and read. Places the content 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 to be received from core memory.
DRSF	706101	Skip if drum transfer flag is set. This flag is set when a block transfer is completed.
DRCF	706102	Clear both drum flags.
DRSS	706106	Load sector and select. Places the content of bits 9-17 of the AC in the DTR, clears both drum flags, and initiates the block transfer (read or write, as specified by the load counter instruction).

TABLE 20 SERIAL DRUM INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
DRSN	706201	Skip if drum error flag is not set.
DRCS	706204	Continue select. Clears the flags and initiates a transfer as specified by the content of the DCL and DTR.

SECTION 10

PLOTTER AND DISPLAY OPTIONS

INCREMENTAL PLOTTER AND CONTROL TYPE 350

Four models of California Computer Products Digital Incremental Recorder can be operated from a DEC Type 350 Increment Plotter Control. Characteristics of the four recorders are:

CCP Model	Step Size (inches)	Speed (steps/minute)	Paper Width (inches)
563	0.01	12,000	31
564	0.005	18,000	31
565	0.01	18,000	12
566	0.005	18,000	12

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

The instructions for this equipment are listed in Table 21.

TABLE 21 INCREMENTAL PLOTTER AND CONTROL INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
PLSF	702401	Skip if plotter flag is a 1.
PLCF	702402	Clear plotter flag.
PLPU	702404	Plotter pen up. Raise pen off of paper.
PLPR	702501	Plotter pen right.
PLDU	702502	Plotter drum (paper) upward.

TABLE 21 INCREMENTAL PLOTTER
AND CONTROL INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
PLDD	702504	Plotter drum (paper) downward.
PLPL	702601	Plotter pen left.
PLUD	702602	Plotter drum (paper) upward. (Same as 702502)
PLPD	702604	Plotter pen down. Lower pen on to paper.

Program sequence must assume that the pen location is known at the start of a routine since there is no means of specifying an absolute pen location in an incremental plotter. Pen location can be preset by the manual controls on the recorder. During a subroutine, the computer can track the location of the pen on the paper by counting the instructions that increment position of the pen and the drum.

OSCILLOSCOPE DISPLAY TYPE 34A

Type 34A is a two-axis digital-to-analog converter and an intensifying circuit, which provides the Deflection and Intensify signals needed to plot data on an oscilloscope. Coordinate data is loaded into an X buffer (XB) or a Y buffer (YB) from bits 8 through 17 of the accumulator. The binary data in these buffers is converted to a −10 to 0 volt Analog Deflection signal. The 30-volt, 10-microsecond Intensify signal is connected to the grid of the oscilloscope CRT. Points can be plotted at approximately a 30-kilocycle rate. The instructions for this display are identical to those of the Precision CRT Display Type 30D described under the following heading, except that the 34A does not have a brightness register so the DLB command is not applicable.

PRECISION CRT DISPLAY TYPE 30D

The Type 30D 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 CRT screen. The array contains 1024 points on a side and measures 9-1/4 by 9-1/4 inches square.

The X- and Y-coordinates each have a 10-bit buffer which is 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 content of this buffer specifies the brightness of the point being displayed as designated on the following scale. The five brightest intensities are easily visible in a normally lighted room; the dimmest can be seen in a darkened room.

BR Content	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 intensifying the CRT. The usual procedure is to load one buffer, then load the second buffer and select in one instruction. The Type 30D requires 50 microseconds to display a point. No flag is associated with this operation.

The I/OT instructions for the Type 30D display are listed in Table 22.

TABLE 22 OSCILLOSCOPE AND PRECISION DISPLAY INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
DXL	700506	Load the X-coordinate buffer from AC8-17. AC8-17 ⇒ XB
DXS	700546	Load the X-coordinate buffer and display the point specified by the XB and YB.
DYL	700606	Load the Y-coordinate buffer from AC8-17. AC8-17 ⇒ YB
DYS	700646	Load the Y-coordinate buffer and display the point specified by the XB and YB.
DXC	700502	Clear the X-coordinate buffer.
DYC	700602	Clear the Y-coordinate buffer.
DLB	700706	Load the brightness register from bits 15-17 of the AC. Note: This instruction clears the display flag associated with the light pen.
DSF	700701	Skip if display (light pen) flag is a 1.
DCF	700702	Clear display (light pen) flag.

SYMBOL GENERATOR TYPE 33

The symbol generator is an option available for use with the Type 30D display. It allows the programmer to plot text on the face of a Type 30D 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. Table 23 lists the instructions for the symbol generator.

TABLE 23 SYMBOL GENERATOR INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
GSF	701001	Skip on display done flag. The next instruction is skipped if display done flag in the generator is a 1, indicating that a word has been processed or a point has been plotted.
GPL	701002	Generator plot left. The content of the AC is transferred into the generator shift register and plotting of the first 17 points is initiated. Bit 17 of this word controls the subscript flip-flop.
GPR	701042	Generator plot right. The content of the AC is transferred into the generator shift register and plotting of the last 18 points is initiated. Bit 12 of the instruction controls the clear flag to prevent loosing the count contained in the horizontal and verticle counter that determines point position.
GLF	701004	Load format. The content of bits 15-17 of the AC is transferred into the character size register. A completion pulse is not generated by the display when this instruction is performed. Bits 15 specifies automatic spacing between symbols when it is a 1. Bits 16 and 17 specify the symbol size. Matrix size, and hence character size, is determined by the number of increments separating the dots on the matrix, when an increment is defined as 1/1024th of the width or height of the display area. The relationship between character size and incremental separation of dots is as follows:

TABLE 23 SYMBOL GENERATOR INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
GLF (continued)		Character Size / Bit 16 / Bit 17 / Number of Increments
		1 / 0 / 0 / 2
		2 / 0 / 1 / 3
		3 / 1 / 0 / 4
		4 / 1 / 1 / 5
GSP	701084	Plot a space. The content of the X-buffer counter is incremented to position the point one character position to the right. Since the content of the AC is transferred into the generator shift register during this instruction the AC must be cleared when this command is given.
GCL	700641	Clear done flag. (This operation is also accomplished by the GPL and GPR commands.)

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. When the coordinates of the starting point of the matrix are given, two I/OT instructions suffice to plot the whole symbol. When the plot is complete, the content of the X-coordinate buffer is 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 I/OT 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).

PRECISION INCREMENTAL DISPLAY TYPE 340

The Type 340 Precision 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 microseconds per point. Point locations are specified on a 9-3/8 inch square raster by any of the 1024 X and 1024 Y 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 microseconds 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 DEC's publication "Precision Incremental CRT Display Type 340" F-13(340).

Instructions added to the computer with the Type 340 are designated in Table 24.

TABLE 24 PRECISION INCREMENTAL DISPLAY INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
IDLA	700606	Load address and select. The content of bits 5-17 of the AC are placed in the display address counter (DAC) and the display is started.
IDVE	700501	Skip on verticle edge violation. If the right or left edge of the grid is encountered (except in vector continue mode), the display stops and an interrupt occurs if the PIC is enabled.

TABLE 24 PRECISION INCREMENTAL DISPLAY INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
IDHE	701001	Skip on horizontal edge violation. If the top or bottom edge of the grid is encountered (except in vector continue mode), the display stops and an interrupt occurs if the PIC is enabled.
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 photo-multiplier light 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 command causes the display to resume at the point indicated by the content of the DAC.
IDRD	700614	Restart display. After a stop code interrupt, this command causes the display to resume at the point indicated by the content of the DAC.
IDRA	700512	Read display address. Transfers the address in the DAC to AC bits 5-17.
IDRC	700712	Read X and Y coordinates. The content of bits 0-8 of the XB is transferred into AC0-8 and the content of bits 0-8 of the YB is transferred into AC9-17.
IDCF	700704	Clear display control. All flags and interrupts are cleared.

Incremental Display Options

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

Type 341 Direct Data Channel Interface This equipment is a complete computer-display interf 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 This device 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 Monitor Display This display is used for remote observation of data displayed on the Type 340 display.

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

PHOTOMULTIPLIER 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 I/OT instructions for the light pen are listed with the display option instruction lists.

SECTION 11

ANALOG/DIGITAL CONVERSION OPTIONS

GENERAL PURPOSE ANALOG-TO-DIGITAL CONVERTER TYPE 138E

The Type 138E is a successive approximation converter that measures a 0 to 10 volt analog input signal and provides a binary output indication of the amplitude of the input signal. Output indication accuracy is a function of the conversion time, and is determined by a switch on the front panel. Each of the seven rotary switch positions establishes an output word length, conversion accuracy, and conversion time for operation of the converter. Overall conversion error equals switching point error plus a quantization error of ±1/2 the digital value of the least significant bit (LSB). Converter characteristics selected for each switch position are specified in Table 25.

TABLE 25 GENERAL PURPOSE A-TO-D CONVERTER CHARACTERISTICS

Word Length (In Bits)	Maximum Switching Point Error*	Conversion Time (In μsec)	Conversion Rate (In kc)
6	±1.6%	9.0	110.0
7	±0.8%	10.5	95.0
8	±0.4%	12.0	83.0
9	±0.2%	13.5	74.0
10	±0.1%	17.0	58.5
11	±0.05%	25.0	40.0
12	±0.025%	35.0	28.5

*±1/2 LSB for quantizing error.

The converter circuits are constructed entirely of FLIP CHIP modules. Both the Type 138E converter and the Type 139E Multiplex Control (implemented to 24 input channels) circuits can be contained in one standard 64-connector module mounting panel.

The instructions for the Type 138E converter are:

Mnemonic Symbol	Octal Code	Operation Executed
ADSF	701301	Skip if converter flag is set. This flag is connected to the program interrupt.

Mnemonic Symbol	Octal Code	Operation Executed
ADSC	701304	Select and convert. The convertrer flag is cleared and a conversion of an incoming voltage is initiated. When the conversion is complete, the converter flag is set.
ADRB	701312	Read converter buffer. Places the content of the buffer in the AC, left adjusted. The remaining AC bits are cleared. The converter flag ss cleared.

Converter Specifications

Monotonicity Guaranteed for all settings

Aperture Time Same as conversion time

Converter Recovery Time None

Analog Input 0 to -10 volts is standard. Bipolar or specific amplitude range input can be accommodated on special request. 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 the possibilities of noise pickup between the source and the converter.

Input Loading ±1 microampere and 125 picofarads for the standard 0 to -10 volt input.

Digital Output A signed 6- to 12-bit binary number in 2's complement notation. A 0 volt input yields a digital output number of 4000_8; a -5 volt input produces 0000_8; and a -10 volt input gives an output of 3777_8. Unsigned and 1's complement outputs are available on special order. Binary ones are represented by DEC standard negative logic level signals (-3 volts) and binary zeros are represented by DEC standard ground logic level signals.

Controls Binary readout indicators and a seven position rotary switch for selecting word length and converter accuracy are provided on the front panel.

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 in 2's complement representation for negative numbers. Extremely high rates of conversion are possible with this unit; five microseconds are 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 microseconds 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 the digital value of the LSB. Instructions for the Type 142 are usually identical to those listed previously for the Type 138E converter. If a system contains both types of converter, different mnemonic symbols and octal codes are assigned for the commands used for the Type 142.

Electrical and logical elements of the Type 142 are shown in block-diagram form in Figure 35. In this illustration the voltage scales for each step are produced by the circuits shown at the left. Two digital-to-analog converters define maximum and minimum voltage levels; 14 precision resistors generate a voltage scale between these parameters. These resistances, plus the D to A converters, define 16 equal voltage levels. The 15 mode 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 putput register holds the 10-bit binary word at the end of conversion.

The Type 142 Analog-to-Digital Converter uses DEC System Modules entirely and is constructed in two 25-position DEC module mounting panels.

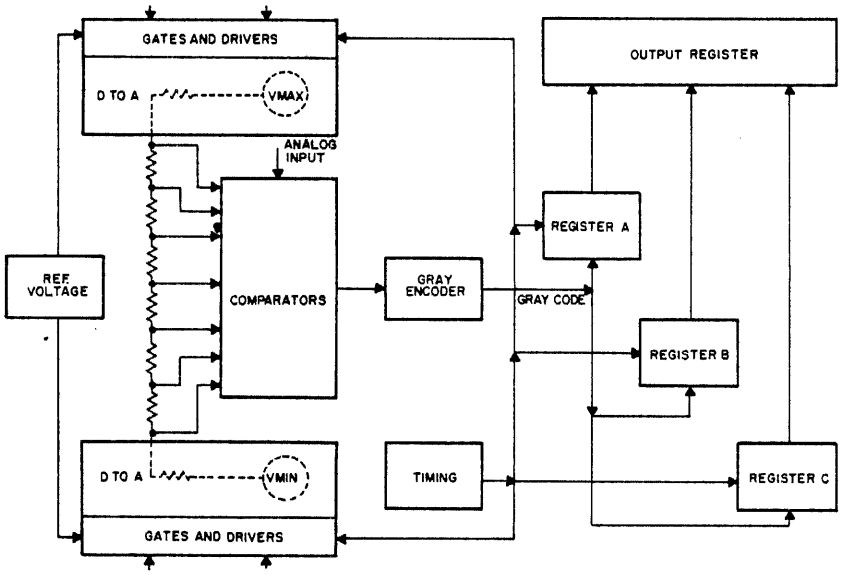


Figure 35 Type 142 A-to-D Converter, Block Diagram

Converter Specifications

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

Input The analog signal can vary between 0 and -9 volts. The input presents 3 units of pulse load. Maximum current is 25 microamperes.

The Convert pulse is the only digital input required. It should be a negative-going signal with a swing 0f 2.5 to 4 volts, a fall time less than 0.5 μsec, and a width greater than 60 nanoseconds.

Output Ten binary bits in 2's complement notation. When read intothe PDP-7 this output is transferred as the ten most significant bits of a computer word. When used with other equipment the output bit signals are -3 volt levels binary ones and ground levels of binary zeros. The converter output is available from 2 microseconds after the end of the conversion until 3 microseconds after the start of the next conversion.

MULTIPLEXER CONTROL TYPE 139E

The Type 139E is intednded for use with the Type 138E or 142 analog-to-digital conversion systems in applications where the PDP-7 must process sampled analog data from multiple sources of high speeds. Under program control the multiplexer can select from 2 to 64 analog input signal channels for connection to the input of an analog-to-digital converter. Channel selection is provided by Type A100, A101, A102, or A103 Multiplex Switch FLIP CHIP modules. These module types each have slightly different timing, impedance, and power characteristics so that multiplexers can be built for wide differences in application by selecting the appropriate module type. Each module contains two indepenent, floating, transistor switches letting the user select any multiple of two channels to a maximum of 64. In the individual address mode, the Type 139E routes the analog data from any program-selected channel to the converter input. In the sequential address mode, the multiplexer advances the channel address by one each time it receives an incrementing 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 0-77$_8$. A channel address may be chosen in one of two ways. It can be specified by the content of bits 12-17 of the AC by incrementing the content of the MAR. The following I/OT instructions are used:

Mnemonic Symbol	Octal Code	Operation Executed
ADSM	701103	Select MX channel. The content of AC12-17 are placed in the MAR.
ADIM	701201	Increment channel address. The content of the MAR is incremented by 1. Channel 0 follows channel 77$_8$.

Multiplexer Specifications

Indicators Six binary indicators on the front panel give visual indication of the selected channel.

Multiplexer Switching Time The time required to switch from one channel to any program-specified channel, or to select the next adjacent channel when the content of the MAR is incremented is 2.5 microseconds. This time is measured from when either a select or increment command is received.

Multiplex Channel Input Six signal lines accept DEC standard logic levels of 0 and -3 volts, with 0 volts for assertion. A pulse or level change clears and strobes the channel data lines to load the MAR. The input accepts DEC standard 70-nanosecond or 400-nanosecond positive pulses (referenced to -3 volts). Readin occurs at the positive transition ot the pulse and should not occur until 400 nanoseconds after the channel address lines have settled.

Increment Channel Input The increment MAR input accepts standard DEC 70-nanosecond or 400-nanosecond positive pulses (referenced to -3 volts).

SECTION 12

DATA AND COMMUNICATION EQUIPMENT OPTIONS

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 high-speed external devices. Interface between the data control and the processor can be by direct connection or through the Data Interrupt Multiplexer Type 173. Block transfers are made from consecutive core 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 memory buffer register. The data control includes four hardware registers: two data buffer registers, one word count register (WC), and one initial address register (AR). The word counter contains the 2's complement of the number of words to be transferred in a block and is incremented on each transfer. The location register contains the address of the next data word to be transferred and is incremented on each transfer.

A block transfer is set up by an initializing sequence of I/OT instructions. Microprogrammed commands of this sequence perform the following operations:

a. Load the starting address into the AR from the AC.

b. Load the block length into the WC from the AC.

c. Load the transfer direction.

d. Initiate the transfer.

The data control operates in either a burst mode or an interlace mode. In the burst mode a data break is entered and maintained, so that consecutive Break cycles are used to transfer words until the entire data block is completed. This mode is used only with devices that can synchronize with the computer timing cycle and can transfer a word every 1.75 microseconds. In the interleave mode a data break is entered, one word is transferred, and the Break cycle is released to allow continuation of the main program. This mode is used where device timing determines the transfer rate and each transfer is interleaved with execution of instructions in the main program.

A done flag in the data control signals the processor when a transfer with the selected device is complete. 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 Type 174 Data Controls can draw information through the data interrupt multiplexer.

The instructions for the data control are listed in Table 26. All data control instructions use bits 12 and 13 to select one of the four (1 through 3) associated devices.

TABLE 26 DATA CONTROL INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
STC	704001	Skip on transfer complete. The next instruction is skipped if the done flag in the data control is set to 1.
LWC	704006	Load word count. The WC register is cleared then loaded from the content of bits 3 through 17 of the AC. When this command is given the 2's complement of the number of words to be transferred in the next block should be contained in the AC.
SEC	704101	Skip on error condition. If an error signal (such as end of tape) has been received by the data control from the selected device, the next instruction is skipped. The data control error flag is connected to the program interrupt facility.
LAR	704106	Load address register. The AR is cleared then loaded by OR transfer from the content of bits 3-17 of the AC. The core memory address of the first word in the next data block should be in the AC when this command is given.
CDC	704201	Clear data control. All flags and registers of the data control are cleared, and the busy status is set.
LCW	704205	Load control word. The control word contained in the AC is transferred into the data control status register. Bit configuration of the control word is determined by the requirements of the devices connected to the data control. One bit determines transfer direction, two select a channel, and the remaining bits are assigned according to programmable control states or other requirements of the device.
RWC	704212	Read word count. The AC is cleared, then the content of the WC is transferred into bits 3-17 of the AC.

DATA COMMUNICATION SYSTEM TYPE 630

The Type 630 Data Communication System (DCS) is a real time interface between Teletype stations and the PDP-7. It is used for multi-user 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; converts the characters from serial to parallel form, and forwards them 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 transmitted to the Teletype stations.

Modularity and plugability of the Type 630 DCS simplify 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 DCS.

The Type 630 system consists of the Type 631 Data Line Interfaces, Type 632 Send/Receive Groups, and a Type 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 Type 631 is dependent upon the type of station signals to be converted. The Type 631 is plug connected to the Type 632 Send/Receive Group.

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 eight Teletype stations onto a bus for presentation to the Type 633 Flag Scanner and notifies the scanner when service is required.

When a character has been received or transmitted, a flag (indicator) is activated. The flag in turn notifies the Type 633 Flag Scanner that service is required for that particular station. The manual OFF/ON switch on the handle of the receiver and transmitter modules can 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 depends upon the data rate, unit code, and the number of data bits processed. 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 Type 632 Send/Receive Group 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 Type 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 Type 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 is 32 microseconds; the maximum total time to search, notify the computer, and continue to search for 64 simultaneously active stations is 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 is 6 microseconds; and the maximum time required to find the next active station (station being serviced minus one) upon being released by the computer is 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:

a. Decodes and interprets computer instructions.

b. Forwards receive characters to the computer upon request from the computer.

c. Sends characters to the transmitter modules when instructed by the computer.

d. Requests computer service when notified by the Type 632 Send/Receive Group 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 stations tends to be the practical limit for this method of station service request detection. For more than eight stations, a high-speed built-in flag scanner is recommended.

When the system is used in-house, the function of the Type 631 Data Line Interface can be included in the Type 634 Control. The Type 634 is a totally pluggable unit.

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

RELAY BUFFER TYPE 140

The Type 140 is a computer output device that allows data in the computer to control external electrical equipment through relays. The relay buffer consists of an 18-bit flip-flop register, an 18-bit relay register, filters to reduce noise due to contact bounce, and a patchboard. Under program control the flip-flop register can be set to correspond to the content of the accumulator and can be cleared. Each bit of the flip-flop register in the binary 1 condition energizes an associated relay in the relay register. Each relay has single-pole double-throw mercury-wetted contacts rated at 2 amperes at 500 volts. External connection to the relay contacts is accomplished by two 50-pin connectors at the back of the relay mounting panel. Connections to system ground or any relay contact at the cable connectors can be modified by means of four banana jacks per bit on the front panel. An indicator on this panel for each bit, lights to denote the energized state of the associated relay. The commands for the Relay Buffer Type 140 are as follows:

Mnemonic Symbol	Octal Code	Operation Executed
ORC	702101	Clear output relay buffer flip-flop register.
ORS	702104	Set output relay buffer flip-flop register to correspond with the contents of the accumulator.

INTER PROCESSOR BUFFER TYPE 195

The inter processor buffer (IPB) controls the flow of information between two asynchronous processors (one or both are assumed to be PDP-7s), interconnecting the processors through their program controlled (I/OT) information channels. The buffer contains an information flag and a buffer available flag, for each processor. These flags interrupt their respective processor when the buffer has been loaded by the other processor or when the buffer has been emptied by the other processor and is ready for another word.

The I/OT commands listed in Table 27 are used to control the buffer. They are identical for both processors since the control is completely symmetrical.

TABLE 27 INTER PROCESSOR BUFFER INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
IPSI	702201	Skip on IPB information flag. The next instruction in the program sequence is skipped if the other processor has loaded the buffer for information transfer.
IPRB	702212	Read IPB buffer. The content of the IPB is read into the accumulator, replacing the previous content. The information flag, which indicated that the data is available to be read, is cleared. The buffer becomes available to one of the processors.
IPLB	702204	Load IPD buffer. The content of the accumulator is loaded into the IPB. The information flag of the other processor is set, indicating to it that data for it is in the buffer. The available flag, which indicates that the buffer can be loaded, is cleared.
IPSA	702301	Skip on IPB available. The next instruction in the program sequence is skipped if the buffer is ready to accept data from the accumulator. The EIA instruction must have been given sometime prior to this instruction. When the LIB instruction is given this flag is cleared.
IPDA	702302	Disable IPB available. The available flag of the processor issuing the command is unconditionally cleared. It will neither interrupt nor cause a skip on the SIA instruction. The EIA instruction must be given before the available flag may be turned on.
IPEA	702304	Enable IPB available (request use of buffer).* The available flag is connected (it may still be off) to the program interrupt and I/O skip facility. The flag is set when the buffer is available for loading by the processor (and if enabled an interrupt will occur).

*When the buffer becomes available to a processor, the processor must either use the buffer (LIB) or dismiss it (DIA). Until either occurs, the buffer cannot become available to the other processor.

Programming

When the buffer is requested for transmitting (EIA) it becomes available immediately unless:

a. It is available to the other processor.

b. It still contains information the other processor has loaded into it (this processor must read the data).

c. It still contains information this processor has loaded into it (the other processor must read the data).

Thus, in unidirectional applications, the transmitting processor tests its available flag and the receiving processor tests its information flag. In bidirectional applications, the buffer becomes available to the processor requesting it at that time. That is, if processor A has given the DIA instruction, and processor B is executing the EIA instruction, the buffer will become available to processor B whenever data is read out of it. If both processors select the buffer, it will become available to each alternately.

To transfer a group of words, a processor gives the EIA and DIA instructions once for the whole group.

Example 1:

Single direction, interrupt off, single-word transfer, word in AC.

```
/TRANSMITTING ROUTINE
        IPEA      /REQUEST BUFFER
        IPSA      /TEST FOR AVAILABILITY
        JMP .-1   /WAIT FOR AVAILABILITY
        IPLB      /LOAD BUFFER FROM AC WITH WORD TO BE TRANS-
                  /MITTED
        IPDA      /DISMISS BUFFER

/RECEIVING ROUTINE**
        IPSI      /TEST FOR INFORMATION READY
        JMP .-1   /WAIT
        IPRB      /READ IN INFORMATION
```

**Since this routine monopolizes the receiving processor, the program interrupt is used to signal the information ready status.

Example 2:

Bidirectional, interrupt on, multi-word transfer with both processors having identical routines.

```
        /RECEIVE INITIALIZE ROUTINE
                LAM     -XX         /SET UP WORD COUNT
                DAC     CNTRR
                LAW     TABLR -1    /SET UP DATA TABLE
                DAC     10

        /TRANSMIT INITIALIZE ROUTINE
                LAM     -XX         /SET UP WORD COUNT
                DAC     CNTRT
                LAW     TABLT -1    /SET UP DATA TABLE
                DAC     11
                IPEA                /REQUEST BUFFER

        /RECEIVE ROUTINE
        /SII INSTRUCTION USED IN INTERRUPT TEST TO ENTER THIS ROUTINE
        /ASSUME AC SAVED EXTERNALLY TO THIS ROUTINE

ENTRY R 0
                IPRB                /READ IN INFORMATION
                DAC     I 10        /STORE IN DATA TABLE
                ISZ     CNTRR       /INDEX WORD COUNT, TEST
                JMP I   ENTRYR      /RETURN FOR END
                JMP     X           /GO TO ROUTINE TO DE-INITIALIZE
                                    /INPUTTING AND DISMISS INTERRUPT

        /TRANSMIT ROUTINE
        /SIA INSTRUCTION USED IN TEST TO ENTER THIS ROUTINE
        /ASSUME AC SAVED EXTERNALLY TO THIS ROUTINE

ENTRY T 0
                LAC     I 11        /GET INFORMATION FROM DATA TABLE
                IPLB                /LOAD INTO BUFFER
                ISZ     CNTRT       /INCREMENT WORD COUNT, TEST FOR END
                SKP
                IPDA                /DONE, DISABLED, BUFFER
                JMP     I ENTRY T   /RETURN
```

SECTION 13

PROGRAMMING SYSTEM

The following programming aids are provided with each PDP-7 system. Each of these programs is described completely in a separate programming manual and are described here in condensed form only as a reference data.

Program	Manual
Symbolic Assembler	Digital-7-3-S
Digital Debugging Tape (DDT)	Digital-7-4-S
Symbolic Tape Editor	Digital-7-1-S
FORTRAN II	Digital-7-2-S
Bus Pak II	Digital-7-5-S

SYMBOLIC 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 instructions. Instructions for the processor and standard input/output 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 Symbolic Assembler Programming Manual.

Character Set

All characters of the alphabet are used along with numerals and certain punctuation characters. These punctuation characters and their meaning to the assembler are:

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

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 →|

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

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
         SZA! SNL
```

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.

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
```

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 immediatley 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
```

Variables

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

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 indicators denote 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 pressing the CONTINUE key.

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 Symbolic Tape Editor program.

DIGITAL DEBUGGING TAPE (DDT)

DDT is a debugging program for the PDP-7 computer. In computers with 4K or 8K core memory capacity, 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 selected 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 content 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.

One of the most useful features of DDT is the breakpoint. A simplified way of understanding a breakpoint is to think of halts being inserted in a program at critical points. The breakpoint control characters are:

" (double quote)	DDT inserts a breakpoint at the address specified before the ". DDT removes the instruction at the break location and saves 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 causes the instruction at the break location to be executed and controls 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 the register which was specified as a breakpoint is encountered.

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 parenthesis, some spaces, and the current content of the AC. At this point, the programmer is free to change registers or just examine them, change the content of the AC, or use any other DDT features.

SYMBOLIC TAPE 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 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 a machine 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 content of the text buffer. To help the user keep track of the mode, a visual indication is provided by the LINK lamp on the PDP-7 operator console. In command mode, this lamp is off; in test mode, it is lit.

Five of the special functions which are part of the Editor are:

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 text 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 by 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 or 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 RUB OUT key has another function. Typed immediately after a carriage return, it signals the Editor to return to command mode. If the programmer 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 the command mode, the LINK indicator which has been lit while in text mode, goes out.

Colon (:)

When this symbol is typed in command mode, the Editor prints 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

FORTRAN II

Based on the field proven FORTRAN used with the PDP-4, the 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 DECtape units. 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	Object Time System
Fortran Assembler	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 ready to execute a program, the user loads the main program and any subprogram, followed by any built-in functions called from the library. With the total program in core memory, the object time system is then loaded and the program is executed. 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. Assembly language coding may be introduced within FORTRAN programs or subprograms simply by prefixing each line of code with a special character. Thus, a complete set of machine language instructions, not normally provided, are made available.

The FORTRAN compiler 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 through 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 four dimensions, either fixed or floating may be defined.

BUS-PAK II

Bus-Pak II is a program assembly system designed for data processing operations. By operating on a character-by-character basis, Bus-Pak II 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/output equipment.

Modes of Operation

Bus-Pak II has two 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. Control of the mode of operation is accomplished by AC switch zero on the operator 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 the address of the next Bus-Pak II instruction to be executed is denoted by the ACCUMULATOR indicators the operator console. 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 memory. 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 variable length. 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 the number of 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 to the programmer. In this way, more efficient and less core-memory consuming programs may be written. Care must be taken that an area defined for a specific input/output device is large enough for that 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 content of the index register specified by the double index register is also added to the data address and this new address is 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 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) operations can be performed as drawn directly from memory. See ADDMEM instruction example.

Program Counters

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

Sense Switches

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

Program Switches

Fifteen program switches are available for internal control of program execution.

Bus-Pak Example 1, Move Characters

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

The N consecutive characters with starting address FROM are moved from left to right to the N consecutive character positions at starting address TO. 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. This instruction can be illustrated as follows:

Operation Code	Mnemonic	Variable Operands
17411	MV	N FROM TO

MV 3 47Ø 473

Core Storage	before	A	B	C	D	E	F
Content	after	A	B	C	A	B	C
Core Storage		4	4		4		
Addresses		7	7		7		
		Ø	1		3		

Bus-Pak Example 2, Add to Memory

The content of accumulator AC is 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. The content of accumulator AC is undisturbed. The sign over the units position of the N consecutive characters with the starting address TO is taken into consideration. The original N consecutive characters with the starting address TO are lost. If the result of the addition produces a value whose magnitude exceedes the capacity of the accumulator, the associated overflow indicator will be set. The result itself is worthless. The sign of the result is placed over the units position of the N consecutive characters with the starting address TO. This instruction can be illustrated as follows:

Operation Code	Mnemonic	Variable Operands
17523	ADDMEM	AC N TO

Core Storage Content	before	Ø	7	5	3	9
	after	1	1	Ø	3	9
Core Storage Addresses		5				5
		Ø				Ø
		1				5

Content of Specified Accumulator	before	+35
	after	+35

SECTION 14

OPERATING PROCEDURES

CONTROLS AND INDICATORS

Manual control of the PDP-7 is exercised by means of keys and switches on the operator console. Visual indications of the machine status and the content of major registers and control flip-flops is also given on both the operator console and on the indicator panel at the top of bay 1. Indicator lamps light to denote the presence of a binary 1 in specific register bits and in control flip-flops. Lighted control indicators denote activation of the associated control functions. The function of controls and indicators on the operator console is listed in Table 28, and their location is shown in Figure 36. The functions of all lamps on the indicator panel are described in Table 29 and shown in Figure 37.

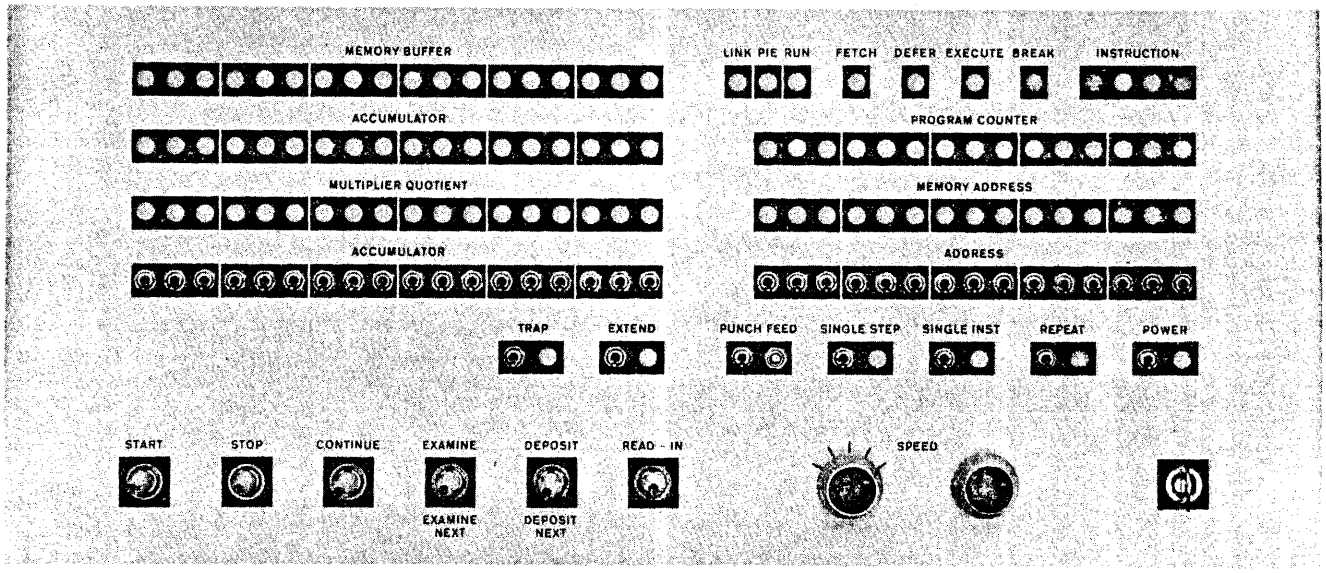


Figure 36 Operator Console

TABLE 28 OPERATOR CONSOLE CONTROLS AND INDICATORS

Control or Indicator	Function
START key	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 key	Stops the processor at the completion of the memory cycle in progress at the time of key operation.
CONTINUE key	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.
EXAMINE key	Places the content of the memory cell specified by the ADDRESS switches into the AC and MB. This operation is accomplished by automatically performing a LAC instruction when the EXAMINE key is pressed. At the completion of the operation, the content of the ADDRESS switches appears in the MA, and the PC contains the address of the next cell.
EXAMINE NEXT key	Places the content of the cell specified by the PC into the AC and MB. The content of the PC is incremented by 1, and the MA contains the address of the register examined.
DEPOSIT key	Deposits the content of the ACCUMULATOR switches into the memory cell specified by the ADDRESS switches. This operation is accomplished by automatically performing tasks similar to the combination of the CLA, OAS, and DAC instructions when the DEPOSIT key is pressed. The content of the ACCUMULATOR switches remains in the AC and MB. The content of the ADDRESS switches appears in the MA. The PC contains the address of the next cell.
DEPOSIT NEXT key	Deposits the content of the ADDRESS switches into the memory cell specified by the PC. The content of the PC is then incremented by 1. At the completion of the operation, the content of the AC and MA are the same as for DEPOSIT key operation.

TABLE 28 OPERATOR CONSOLE CONTROLS AND INDICATORS (continued)

Control or Indicator	Function
READ-IN key	Punched paper tape is read in binary mode and stored in a core memory block when this key is pressed and released. The first address of the memory block is taken from the ADDRESS switches. After reading the tape, program control transfers to the processor which executes the last instruction word stored in the block. To indicate that this last computer word on tape is the instruction to be executed next, a hole must be punched in channel 7 of the first line of the three binary lines that constitute the last word.
SPEED switch and control	These two controls vary the repetition rate of manual operations from approximately 40 microseconds to 8 seconds. The switch (left) is a 5-position coarse control, the control (right) is a continuously variable fine control. Slowest speed is obtained with both controls in the fully counterclockwise position.
Console Lock	This key-operated, 2-position lock switch can be used to prevent inadvertent key operation from disturbing a program in progress. 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 INST or REPEAT switches has no effect on the running of the computer. The status of the ACCUMULATOR switches can be monitored by the program even with the keys disabled by the lock.
TRAP switch and indicator	Permits the trap mode to be entered by the program.
EXTEND switch and indicator	In the raised position, the switch enables operation of any of the console keys except STOP and CONTINUE to turn on the extend mode. The indicator lights to denote enabling of the extend mode control. This switch and indicator are operable only on systems containing a Type 148 Memory Extension Control.

TABLE 28 OPERATOR CONSOLE CONTROLS AND INDICATORS (continued)

Control or Indicator	Function
PUNCH toggle switch and FEED pushbutton	The toggle switch controls application of primary power to the perforated tape punch. When the switch is down, punch power is under program control; when up, punch power is on. The pushbutton causes the perforated tape punch to punch tape leader. Punch power remains on for an additional 5 seconds as it does under program control.
SINGLE STEP switch and indicator	The switch causes the computer to stop at the completion of each memory cycle. Repeated operation of the CONTINUE key while this switch is on steps the program one memory cycle at a time. The indicator lights to denote operation in the single-step mode.
SINGLE INST switch and indicator	The switch causes the computer to stop at the completion of each instruction. Repeated operation of CONTINUE key while this switch is on steps the program one instruction at a time. When both switches are on, SINGLE STEP takes precedence over SINGLE INST. The indicator lights to denote operation in the single-instruction mode.
REPEAT switch and indicator	The switch causes the operations initiated by pressing CONTINUE, EXAMINE NEXT, or DEPOSIT NEXT keys to be repeated as long as the key is held on. The rate of repetition is controlled by the SPEED controls. The indicator lights to denote activation of the repeat controls.
POWER switch and indicator	The switch controls the application of primary power to the computer and to all external devices attached to it. The energized state of the equipment is indicated by lighting of the lamp.
ACCUMULATOR switches	Used to establish the 18-bit word to be placed in core memory by the DEPOSIT and DEPOSIT NEXT keys, or the word to be placed in the AC by a program. These switches are also used for program sense control.

TABLE 28 OPERATOR CONSOLE CONTROLS AND INDICATORS (continued)

Control or Indicator	Function
ADDRESS switches	Used to establish the 15-bit core memory address loaded into the PC by operation of the START, EXAMINE, or DEPOSIT keys.
MULTIPLIER QUOTIENT indicators*	Denote the content of the MQ
ACCUMULATOR indicators	Denote the content of the AC
MEMORY BUFFER indicators	Denote the content of the MB
MEMORY ADDRESS indicators	Denote the content of the MA
PROGRAM COUNTER indicators	Denote the content of the PC
LINK indicator	Denotes the content of the link
PIE indicator	Lights when the program interrupt is enabled
RUN indicator	Lights when the computer is executing instructions
FETCH, DEFER, EXECUTE, BREAK indicators	Light to denote the major control state of the next memory cycle

*These indicators function only when the computer is equipped with a Type 177 Extended Arithmetic Element option.

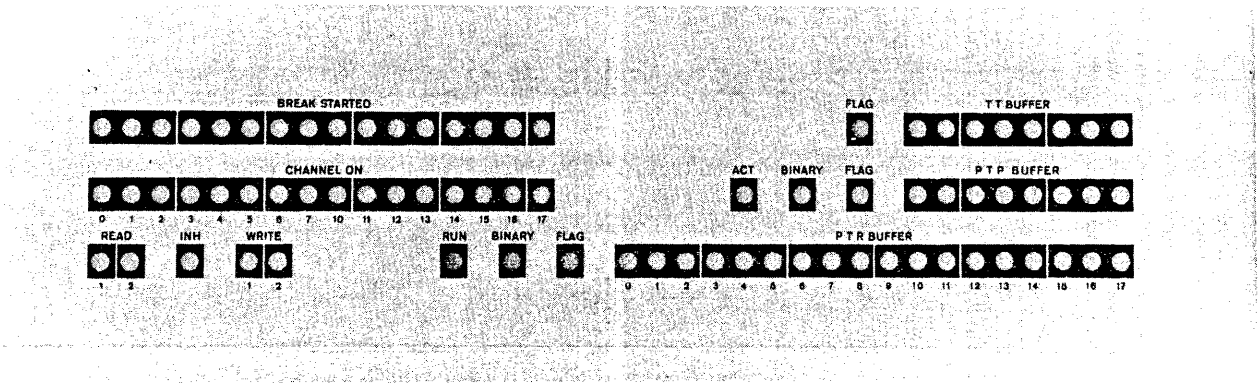


Figure 37 Indicator Panel

TABLE 29 INDICATOR PANEL FUNCTIONS

Indicator	Function
READ 1, 2	Designate the status of timing control flip-flops in the memory control and indicate that the core memory is in a read cycle.
INH	Designates the status of the INH (inhibit) flip-flop in the memory control and indicates the memory is in a write cycle.
WRITE 1, 2	Designate the status of timing control flip-flops in the memory control and indicate that the core memory is in a write cycle.
CHANNEL ON	Lighted indicators denote enabled priority interrupt channels of the Type 172 Automatic Priority Interrupt.
BREAK STARTED	Lighted indicators denote program interrupt channels that are active (requesting or using a break cycle).
RUN	Denotes the status of the RUN flip-flop in the Type 444B Perforated Tape Reader and Control, and indicates operation of this device.
BINARY (bottom row)	Lights to designate that the perforated tape reader is in the binary mode. If this lamp is not lit the reader is in the alphanumeric mode.
FLAG (bottom row)	Denotes the status of the perforated tape reader flag.
PTR BUFFER	Indicates the content of the data buffer register of the perforated tape reader (last character or binary word read).
ACT	Denotes the active, or operating, status of the Type 75D Perforated Tape Punch and Control.
BINARY (center row)	Lights to designate that the perforated tape punch is in the binary mode.
FLAG (center row)	Denotes the status of the perforated tape punch flag.
PTP BUFFER	Indicates the content of the data buffer register of the perforated tape punch (last character punched).

TABLE 29 INDICATOR PANEL FUNCTIONS (continued)

Indicator	Function
FLAG (top row)	Denotes the status of the Type 649 Teleprinter and Control line unit in (LUI) flag. (The in direction is referenced to the computer, not to the Teletype equipment.)
TT BUFFER	Indicates the content of the Teletype control line unit in (LUI) data register (code of the last keyboard character struck).

A POWER ON switch is provided on the perforated tape reader and is used to disable this device for maintenance regardless of the condition of computer primary power. Normally this switch remains in the ON position so that the reader is energized as a function of the POWER switch on the operator console.

MANUAL DATA STORAGE AND MODIFICATION

Programs and data can be stored or modified manually by means of the facilities on the operator console. Chief use of the manual data storage facilities is made to load the readin mode (RIM) loader program and all other programs in readin mode format.

Storing the RIM Loader

The RIM loader is a program used to automatically load any program that is in RIM format into the computer core memory. The initial operation of the computer should always be to load the RIM loader. If a prepared RIM loader tape is available it can be stored as follows:

1. Turn the lock switch counterclockwise and set the POWER switch to the up position.

2. Load the tape into the perforated tape reader by placing the title end of the tape beneath the read head mechanism so that the tape feed holes are on the inside and the tape is positioned to move from right to left as it is read in. The lever at the center of the tape reader is turned clockwise to disengage the tape feed mechanism and allow entrance of the program tape. This lever must be turned counterclockwise to engage the tape or the READ-IN key will be disabled.

3. Set the ADDRESS switches to correspond with the starting address to be used for the program as it is stored in core memory. In a computer having a 4K word core memory this address is 7763, in computers having a larger core storage facility the starting address is 17763.

4. Press and release the READ-IN key. When the key is pressed three lines on tape are read, then when the key is released the tape is completely read into the machine.

To manually load data or a program into the core memory, or if a prepared tape containing the RIM loader is not available this program can be loaded manually as follows:

1. Turn the lock switch counterclockwise and set the POWER switch to the up position.

2. Set the ADDRESS switches to correspond with the core location of the first word to be stored.

3. Set the ACCUMULATOR switches to correspond with the bits of the instruction word or data to be stored at the address determined by the ADDRESS switches. Lift the DEPOSIT key and observe that the MEMORY BUFFER indicators, and hence the core memory, hold the word contained in the ACCUMULATOR switches and that the PROGRAM COUNTER indicators correspond to the setting of the ADDRESS switches.

4. Set the ACCUMULATOR switches to correspond with the next data word or instruction to be stored, then press the DEPOSIT NEXT key and observe that the content of the ACCUMULATOR switches is stored (as indicated by the MEMORY BUFFER indicators) and observe that the content of the program counter has been incremented by one.

5. Repeat step 4 until the entire program or data block has been loaded into the sequential memory locations.

The RIM loader program is listed in Table 30. Note that the RIM and FF loader programs are described in the DEC Program Library and are available to all PDP-7 users.

TABLE 30 RIM LOADER (8K)

Location	Octal Code	Tag	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

Loading Data Under Program Control

Information can be stored or modified in the computer automatically only by enacting programs previously stored in core memory. For example, having the RIM loader programs stored in core memory allows RIM format tapes (including the FF loader program tape) to be loaded as follows:

1. Turn the lock switch counterclockwise and lift the POWER switch.

2. Assure that the perforated tape reader is energized by observing that the lamp which illumines the photo diodes is lit. If this lamp is not lit, set the POWER ON toggle switch on the reader to the up position.

3. Load the tape in the reader as specified in step 2 of the procedure for storing the RIM loader.

4. To load data set the starting address into the ADDRESS switches and press and release the READ-IN key. To load a program set the RIM loader starting address into the ADDRESS switches and press the START key.

Tapes being loaded into core memory by means of the FF loader can be made self-starting by having a hole punched in channel 7 of any of the last three lines of the tape that constitute the last instruction of the program. Usually this instruction is a JMP to the starting address of the program contained on the tape. Under these conditions when the last block on tape is read it is interpreted as the current instruction to be executed and the program is started. If the tape is not self-starting the last instruction is a HLT command. To initiate programs that are not self-starting, set the starting address into the ADDRESS switches and press the START key.

Checking and Modifying a Stored Program

To check the content of an address in core memory, set the address into the ADDRESS switches, lift the EXAMINE key, and observe the data displayed in the MEMORY BUFFER indicators. Note also that the address established by the switches is contained in the MEMORY ADDRESS indicators. Examination of sequential core memory locations can then be performed by repeated pressing of the EXAMINE NEXT key. A data or instruction word can be stored at any core memory location by specifying the location in the ADDRESS switches, setting the word in the ACCUMULATOR switches, and lifting the DEPOSIT key. Blocks of words can be stored at sequential addresses by repeatedly pressing the DEPOSIT NEXT key, without specifying each address in the ADDRESS switches.

FORTRAN OPERATING PROCEDURES

The PDP-7 FORTRAN compiler is written for a machine having a minimum of 8K of memory but significantly different hardware configurations; one an exclusively paper-tape configuration and the other a configuration which includes a dual DECtape transport. In an 8K system about 4600_{10} locations are available for the user's program and data.

The principal subsections of the FORTRAN system for paper tape are:

Compiler
Assembler
Operating System
Library

The compiler accepts input in the FORTRAN language and produces an object program output in computer source 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. To run the program, load the main program and any subprograms followed by any functions called from the library tape. When the program and library routines are stored in memory, load the operating system and execute the program. The operating system contains an interpreter for floating-point arithmetic, an interpreter for FORMAT statements, bookkeeping routines such as fix a floating number etc., and the I/O routines. The operating system must be in memory when a FORTRAN program is executed.

Procedure for Using FORTRAN With a PDP-7 Paper Tape System

The RIM loader is used with starting address 17770_8 (for 8K machines). Pressing the START key with 17770_8 in the ADDRESS switches is referred to as RIM start.

1. Prepare the programs to be compiled in accordance with the conventions described in the preceding paragraphs. Each program or subprogram on paper tape must be followed by the three-character sequence:

   carriage return, line feed
   carriage return, line feed
   form feed

2. Place the paper tape labeled FORTRAN Compiler in the reader and RIM start.

3. Set ACCUMULATOR switch 9 up to allow ASCII code input or set this switch down for FIODEC code input. Set ACCUMULATOR switch 10 up for ASCII code output or down for FIODEC code output.

4. Place the program to be compiled in the reader and press the CONTINUE key. FORTRAN will punch out the intermediate object program tape.

5. If other programs are to be compiled, repeat step 3. If an accidental error should occur (e.g. the punch running out of paper tape before compilation is completed), the compilation procedure may be restarted by repositioning the source tape in the reader, placing 22_8 in the ADDRESS switches, and pressing the START key.

6. If an error occurs in the source language, the compiler will type a three-letter plus two-digit code on the teleprinter followed by the current (last encountered) statement number. The compiler then prints the offending line to the point where the error is encountered. A line feed is given and the rest of the statement is then printed. See the description of diagnostics for the associated error conditions. As a rule, a source language error will prevent proper execution of the compiled program. The error must be corrected and the program compiled again. However, compilation should be completed to uncover all errors in the same program.

7. When all necessary compilations have been successfully completed, remove the output tape(s) from the punch.

8. Load the tape labeled FORTRAN assembler through RIM start.

NOTE: The normal usage of the FORTRAN assembler and linking loader is described in steps 9 through 18.

9. Set ACCUMULATOR switch 10 up for ASCII code input; down for FIODEC code input. Place the first program to be assembled in the reader. If several programs were compiled together they will be separated from each other by a short length of blank tape. Press the CONTINUE key. The assembler will punch a partial binary output, displaying all ACCUMULATOR indicators lit when finished. Should an error occur during the assembly procedure, the assembler will print a message on the teleprinter. For a summary see FORTRAN Assembler Error Messages. An error printed by the assembler is either the result of an original program error which was not detected by FORTRAN or by a punching error.

10. Press the CONTINUE key to finish punching the binary output. Undefined symbols used in the source program (symbols which never appear on the left-hand side of an arithmetic statement or in an input statement or as the argument of a subroutine call) will be printed with a relative location automatically assigned by the assembler. Any statement number which is referred to but never used as a statement label also will be printed. When finished, all ACCUMULATOR indicators will again be lit. The aforementioned errors prevent program execution.

11. If a printout of the relative locations of program symbols is desired, set the least significant switch of the ACCUMULATOR switches (bit 17) to the up position and press the CONTINUE key. If the printout is not desired, leave the switch in the down position and press the CONTINUE key to restore the assembler for the next assembly. No ACCUMULATOR indicators will be lit at this time.

This step applies to the main program. If a subroutine was assembled, pressing the CONTINUE key will readin and assemble the next program.

12. When starting a new assembly be sure to start at the beginning. If more programs are to be assembled, place the next tape in the reader and return to step 9. If several programs were compiled together, be sure that the blank tape area separating them is under the reader head before continuing. Since the assembler uses a buffered loader, the end of one program and the beginning of the next program are read into the same buffer. It is usually necessary to withdraw a portion of tape which has already been read in order to start reading at the beginning of the second and succeeding programs on the same paper tape.

13. Remove the assembled programs from the punch. Each program will have the title punched in readable format at the beginning. Since the FORTRAN assembler is a one-pass assembler, the title will be the last item punched on the tape. (The last program to be assembled will be the first program on the binary tape. For this reason, user subroutines should be assembled before the main program.)

> NOTE: The following steps describe the loading process. After each tape is loaded into core memory the ACCUMULATOR indicators will display the last memory address used.

14. Load the main program through RIM start. It is important that the main program be loaded first since the linking loader is punched on the main program tape only. The loader is a lengthy strip of tape immediately following the title with the eighth hole punched in every line of the paper tape.

15. Place any subprograms in the reader (readable title is always in the leader), and load through RIM start. The linking loader which is punched at the beginning of the main program binary tape will handle the problem of linking between programs.

16. To obtain a printout of the absolute locations in core memory of subprogram symbols and/or to determine if library subroutines are required, place 5_8 in the ADDRESS switches and press the START key. If a subroutine or library function has been called but is not yet loaded, its symbol will be preceded on the line by a minus sign followed by the address of the first reference to this symbol. If further user subprograms are requested by the main program, they should be loaded as in step 15.

17. Load the I/O library tape. Place the library tape in the reader, put 6_8 in the ADDRESS switches and press the START key. When all the called functions have been loaded, the loader will halt. If the tape is entirely read, it is possible that certain requested (arithmetic) routines were not encountered. To determine this, return to step 16. The loaded routines will not be preceded by the minus sign. Load either the six-decimal digit (6DD) or the nine-decimal digit (9DD) library tape.

18. Load the tape labeled FORTRAN Operating System through the RIM loader.

19. Place 22_8 in the ADDRESS switches and press the START key to execute the program.

20. If paper tape input to the FORTRAN program is requested, the tape must be positioned in the reader which is conditioned for immediate operation upon program command.

NOTE: The linking loader will not detect when the user has loaded a program over common storage (assigned backward from the last address in memory). To guarantee an overlay has not occurred, the last program address used as indicated in the AC indicators after loading, should always be smaller than the lowest address in common storage necessary to store the arrays and common variables used in the program.

Diagnostics

The following diagnostics may be printed during compilations followed by the offending statement with a line feed after the last character processed. Each diagnostic is identified by a three-letter name, and a two-digit number. For all errors except those which indicate storage capacity exceeded, processing will continue. The diagnostic error prints listed in Table 31 will be followed by the current statement number. As previously noted the occurrence of an error will necessitate correction of the error and recompilation.

TABLE 31 FORTRAN DIAGNOSTIC ERROR PRINTOUTS

Error Name	Error Number	Reason for Error
CON		CONTROL STATEMENT
	1	Illegal control statement.
	2	Upper case character in control statement.
COM		COMMON STATEMENT
	1	Illegal entry in list.
	2	Symbol appears twice in COMMON.

TABLE 31 FORTRAN DIAGNOSTIC ERROR PRINTOUTS (continued)

Error Name	Error Number	Reason for Error
ASG		ASSIGN
	1	N not a fixed-point number.
	2	Number not followed by two.
	3	No fixed-point variable.
	4	Illegal format - variable.
SUB		SUBROUTINE AND FUNCTION
	1	Name not a variable.
	2	Dummy symbol not a variable.
	3	Dummy symbol used twice.
DIM		DIMENSION
	1	Array name not a variable.
	2	Array dimensioned twice.
	3	Dimension not a fixed-point number.
DO		DO STATEMENT
	1	First two letters not do.
	2	No statement number.
	3	No end test value specified.
	4	Too many characters.
ILF		ILLEGAL FORMAT
	1	Nonstatement number at left margin.
	2	Missing left parenthesis.
	3	Missing right parenthesis.
	4	Missing left parenthesis.
	5	Missing right parenthesis.
	6	Comma missing in goto.
	7	Variable missing in arithmetic statements.
	11	Illegal device number in input or output statement.
	12	Illegal format in accept statement.
	17	Extra right parenthesis.
	20	Extra characters in statement.
	22	Comma missing in repetitive element in I/O list.
	24	Illegal format in I/O list element.
	26	Illegal format statement number in an I/O statement.
ICH		ILLEGAL CHARACTER
	1	Illegal character.
	2	Illegal upper-case character.
	4	No more characters after an illegal one.

TABLE 31 FORTRAN DIAGNOSTIC ERROR PRINTOUTS (continued)

Error Name	Error Number	Reason for Error
DIT*		MISCELLANEOUS ERRORS. Cannot proceed.
	1	Logic error.
	2	Wrong place in table.
	3	Dispatch number too big.
	10	Too many cal's.
	11	Illegal cal.
	12	Too many exits.
UFX		UNSEEN FIXED POINT
	1	Fixed-point number expected; punctuation character or no character appeared.
	2	Floating point quantity appeared where fixed-point number expected.
	3	Fixed-point number expected; decimal number appeared.
FOR		FORMAT STATEMENT
	1	Character missing.
	2	Illegal format.
	3	Characters missing.
	4	Illegal control character.
	5	Illegal punctuation.
	6	Specification letter other than I, F, E, X, H.
	7	N too large in H format
IFU		ILLEGAL FUNCTION USAGE
	1	Function name on left side outside function definition.
SCE		STORAGE CAPACITY EXCEEDED Processing may not proceed.
	1	Polish stack exhausted.
	2	Table exceeded.
	3	Table exceeded.
	4	Symbol generator exhausted.
	5	Table exceeded.
	6	Statement too long.
	7	Push down stack exceeded (too many nested do's).

*If any of the errors labeled DIT occurs, correct all other errors and recompile; if DIT errors still occur, note any pertinent data and send to DEC Programming Group.

FORTRAN Assembler Error Messages

NOTE: The following error messages refer to the object program code generated by the compiler. Familiarity with this code is necessary for an understanding of this appendix. See the Assembler Program Description (Digital 7-3-S) for details.

With the exception of SCE (storage capacity exceeded) and ILP (illegal parity), assembly continues after the error message has been printed unless assembling a library tape. An error message may occur in one of three formats.

Format A

ERROR PREVIOUS VALUE SYMBOL NEW VALUE

Format A is used to indicate errors in the redefinition of symbols. ERROR represents a three-letter code for the particular error. Whether the symbol was redefined depends upon the particular error.

Error	Meaning
MDT	The symbol was redefined with a comma.
RSP	A permanent symbol was redefined.
RDA	An attempt to redefine a symbol was made. The symbol was not redefined.

Format B

ERROR OCTAL ADDRESS SYMBOLIC ADDRESS

The general error message is printed in Format B. It includes both the octal address and the symbolic address at which the error occurred.

Error	Meaning
IFP	Illegal format in parameter assignment.
IFC	Illegal format in a symbolic address tag.
IFQ	Illegal format in library list.
IFY	Illegal format in internal declaration.
IFZ	More than one symbol in internal declaration.
LIQ	Illegal term punctuation in library list.
MDT	The location counter and address disagree in an address assignment.

Error	Meaning
TUA	Too many undefined symbols in a symbolic address tag.
ILF	Illegal format in a pseudo-instruction.
LIT	Illegal terminator in a PUNDEF or EXTERNAL list.
IFL	Illegal format in a PUNDEF or EXTERNAL list.
IFS	Illegal format in a START.
IFI	Illegal format in an input pseudo-instruction.
SCE	Storage capacity exceeded.
INS	A nonsymbol appeared in a PUNDEF list.
IFX	External symbol preceeded external declaration.

Format C

ERROR OCTAL ADDRESS SYMBOLIC ADDRESS CAUSE

Format C is an expanded version of Format B. CAUSE is additional information to help the programmer ascertain the cause of the error. For example, in the case of an error caused by an undefined symbol, the symbol will be printed.

Error	Cause	Meaning
ILP	character	Illegal parity (place correct character in ACS and press the CONTINUE key). May also be caused by reading tape in backward order.
UST	symbol	Undefined symbol in a START or PAUSE.
UAA	symbol	Undefined symbol in an absolute address assignment.
UPA	symbol	Undefined symbol in a parameter assignment.
ICH	character	Illegal character.
SYS	symbol	Previously defined symbol in internal declaration.
UPN	symbol	Undefined symbol in a punch pseudo-instruction.

At the end of assembly, before the loader is punched, the undefined symbols and their definitions will be printed. Each undefined symbol which was used in a storage word will be defined as the address of a register at the end of the program, and the definition printed. If the symbol was not used in a storage word, then just the symbol will be printed and the symbol will not be defined. An example of the latter is a symbol which appears to the right in a parameter assignment only.

Data Organization

Records

On every I/O device, data is organized into physical groups called records. Because of the dissimilarity of the devices, the definition of a record varies. Table 32 lists the I/O devices and the definition of a record for each.

TABLE 32 DEFINITION OF A PHYSICAL RECORD FOR I/O DEVICES

Device	Physical Record
Keyboard	The information typed on a single line (maximum 72 characters).
Teleprinter	The information typed on a single line (maximum 72 characters).
Perforated Tape Reader, Punch	The information punched between two carriage returns (practical maximum 72 characters, for compatibility with other devices).
Magnetic Tape	The information contained between two record gaps (delimiters) (maximum record length is 576 characters or 256 binary words).
DECtape (microtape)	The information contained in one fixed-length record (256 18-bit words).

One FORMAT statement corresponds to one record. Consequently, the programmer must be careful that the total number of characters in the format specifications, including repetitions, does not exceed the maximum for one record on the respective device.

The FORTRAN Assembly System

The FORTRAN assembler is a modified version of the PDP-7 assembler. The FORTRAN assembler produces a relocatable object program unless absolute address assignments are used. Relocatable programs are loaded by the linking loader consecutively starting at location 22_8. The loader also joins programs by supplying definitions for symbols which are referenced in one program and defined in another. The differences present in the FORTRAN assembler are:

1. The addition of pseudo-instructions to define symbols used by the loader to link relocatable programs to each other. These pseudo-instructions are EXTERNAL, INTERNAL, and LIBFRM.

2. Error printouts associated with these three pseudo-instructions have been included.

3. The object programs produced by the FORTRAN assembler are relocatable; the programs are loaded into an area of memory determined by the position of other programs at load time (usually starting at address 22_8 for the first program).

4. DDT cannot be used with relocatable programs since symbol definitions are not established until loading, unless changes mentioned in the FORTRAN manual or the RELOCATABLE write-up (Digital-7-10-RE-1) are followed.

5. Execution of a program assembled and loaded by the FORTRAN system is accomplished by placing the starting address (22_8) in the ADS and pressing the START key.

NOTE: To avoid improper loading, all absolute parameter assignments should precede any references to them in the program.

OPERATING THE PDP-7 ASSEMBLER (BASIC OR EXTENDED)

Operating Instructions

1. Make sure that the RIM loader is in core.

2. Load the assembler by placing the binary tape of the assembler in the reader and starting the RIM (readin mode) loader in location 17770.

3. Place the symbolic source language tape in the reader, and set the ADDRESS switches to 20. Set ACCUMULATOR switch 10 up when using ASCII symbolic tapes or down when using FIODEC code.

4. The operator may choose, at this point, to begin a normal assembly or command the assembler to execute special functions as indicated by the AC switches.

a. Normal assembly (restores symbol table to permanent symbols): press the CONTINUE key.

b. Special functions: set ACS (as described in the succeeding summary of AC switch control) and press the START key. When the pseudo-instruction START or PAUSE in the source tape is encountered, the assembler stops with all ones in the AC.

NOTE 1: To assemble more than one symbolic tape into one binary output tape (a main program and subroutines, for example), the sequence of steps in assembly is altered. After step 4, the next symbolic tape is put in the reader. With 20 in the ADDRESS switches, press the START key. Repeat these steps for remaining symbolic tapes. The title of the first tape and the START from the last tape are incorporated into the binary output tape unless otherwise specified by ACS3. When all desired symbolic tapes have been assembled, continue with step 5.

5. To complete the assembly, press the CONTINUE key. The assembler punches the variables, the undefined symbols (listing these on the on-line Teletype), the starting block, and the loader and punches the title in readable form. Then the assembler stops with all ones in the AC. The assembly of a loadable object tape is complete at this point.

NOTE 2: To restore the assembler's symbol table to permanent symbols before beginning another assembly, put up AC switch 15. After completing step 6, return to step 3. If no symbol printouts are desired, press the CONTINUE key and return to step 3.

6. To obtain a printout of the symbol definitions, set AC switches 16 and/or 17 (see below) and press the CONTINUE key. When the printouts are completed, the computer halts displaying all zeros in the AC.

Figure 38 indicates the logical flow of the assembler program.

Loading a Symbol Punch

A symbol punch in assembler format can be loaded into the assembler at any time, but the suggested time is prior to assembling the first tape (before step 3). To load a symbol punch, place the tape in the tape reader, set ADDRESS switches to 4, and press the START key. The symbol definitions are added to the assembler's permanent symbol table; restoring the assembler's symbol table has no effect on them. To start an assembly, return to step 3 above.

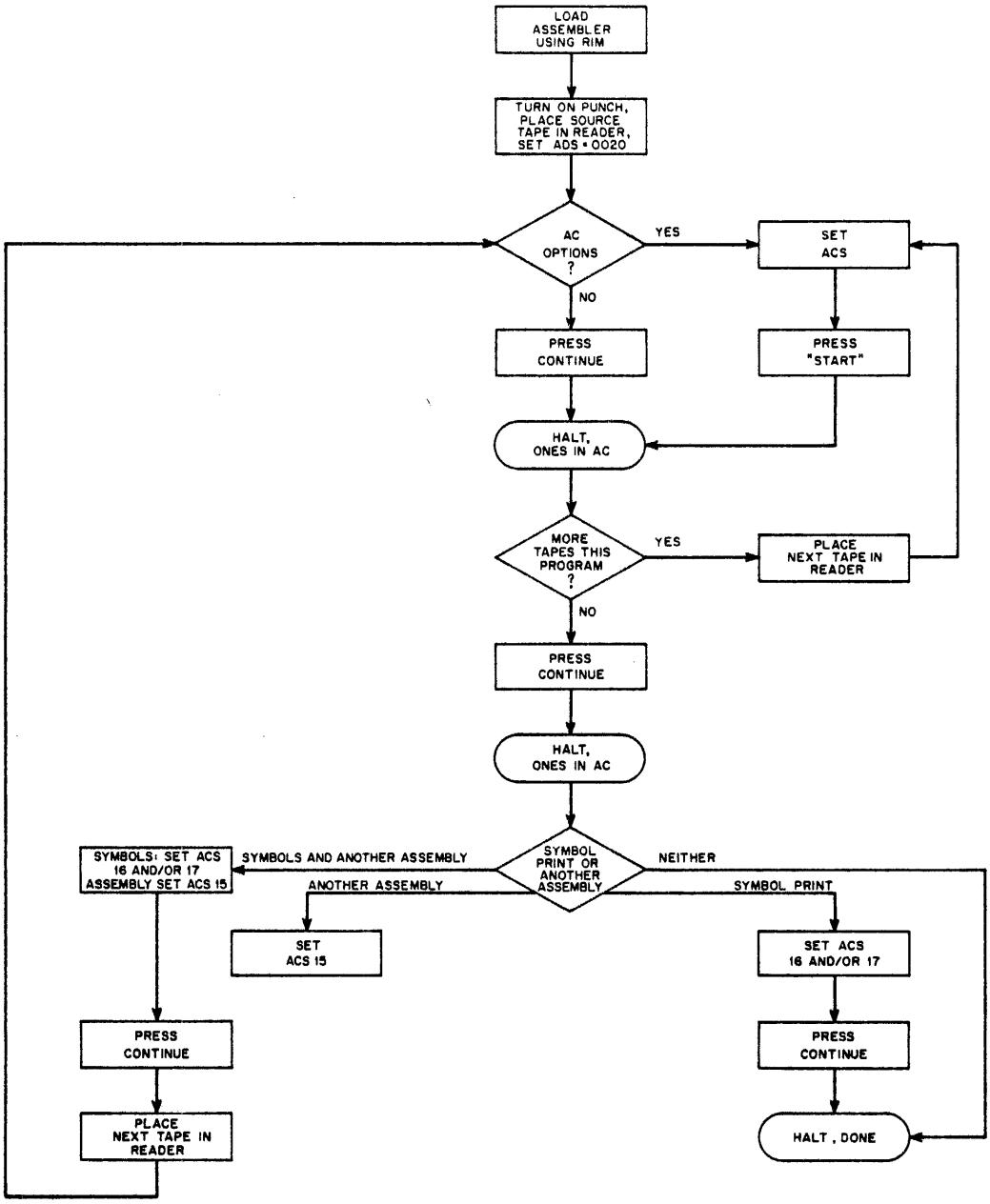


Figure 38 Assembler Flow Diagram

AC Switch Control

Throughout the assembly of a program, ACS 10 indicates the symbolic tape code: up for ASCII or down for FIODEC. This switch may be reset if necessary for each program or subprogram assembled. In step 4, the ACCUMULATOR switches perform the following functions:

AC Switches Up	Meaning
0 and 1	Suppress punching.
0 and 2	Suppress punching of symbols for DDT-7. Save space on tape unless needed for DDT work.
0 and 3	Take the title on this tape. The title from the current tape replaces the first tape's title on a single binary output tape (see Note 1).
0 and 4	Restore the assembler when restarting an unfinished assembly.

In step 6, the switches have the following meaning:

AC Switch Up	Meaning
15	Restore symbol table to permanent symbols (after symbol printouts if requested). Starting the next assembly with CONTINUE has the same effect.
16	Symbol printout, numerical order.
17	Symbol printout, alphabetical order.

The following switches have meaning throughout an assembly.

AC Switch Up	Meaning
10	ASCII symbolic tape input
11	Causes all printing to be done on the high speed line printer.

Halts During Assembly

The following are all possible abnormal halts during assembly, the cause, and the action which can be taken.

Cause	AC Content	Action
Illegal parity, ILP, using FIODEC code	Character with illegal parity	1. Place correct character in ACS. 2. Press CONTINUE.

Cause	AC Content	Action
Illegal parity, ILP, using ASCII code	- - -	1. Set ACS 10 up. 2. Restart assembly
Storage capacity exceeded, print-out SCE	- - -	Segment program and reassemble.

Error Messages

The error message appears in one of the following three formats. With the exception of SCE (storage capacity exceeded) and ILP (illegal parity), assembly continues automatically after the error message has been printed.

Format A

The appearance of a diagnostic printed in format A:

ERROR PREVIOUS VALUE SYMBOL NEW VALUE

Whether the new value was actually incorporated into the symbol table depends upon the particular error.

Error	Meaning
MDT	A previously defined symbol was redefined with a comma.
RDA	An attempt was made to redefine a permanent symbol with a comma. The symbol was not redefined.
RPS	A permanent symbol was redefined.

Format B

The appearance of a format B diagnostic is:

ERROR OCTAL ADDRESS SYMBOLIC ADDRESS

The general error message is printed in format B.

Error	Meaning
IFC	Illegal format in symbolic address tag. The tag is ignored.

Error	Meaning
IFI	An expression using CHAR or FLEX was formed improperly.
IFL	Illegal format in a PUNDEF list.
IFP	Illegal format in a parameter assignment. The assignment is ignored.
IFS	START or PAUSE used incorrectly. Assembly continues as if START or PAUSE had been used with no expression following.
ILF	Illegal format in a pseudo-instruction such as BAR. The pseudo-instruction is ignored.
INS	An illegal format in a PUNDEF list—two commas appeared in a row or a digit appeared.
LIT	An illegal character was found in a PUNDEF list. The character is taken as a terminator.
MDT	The value of the complex symbolic address assignment (tag) and the location counter disagree. The symbolic address tag is redefined if possible.
SCE	Storage capacity of the symbol table was exceeded. No recovery is possible.
TUA	Too many undefined symbols appeared in a symbolic address assignment (tag). Location counter remains unchanged.
UBR	An undefined symbol appeared in a BAR pseudo-instruction. The setting of BAR remains unchanged.

Format C

The appearance of a format C diagnostic is:

ERROR OCTAL ADDRESS SYMBOLIC ADDRESS CAUSE

Format C is an expanded version of format B. CAUSE is additional information to help the programmer ascertain the cause by an undefined symbol which will be printed. ASCII codes are printed when the cause is a character.

Error	Cause	Meaning
ICH	character	A character not part of the assembler's source language was used. The character is ignored.
ILP	character	A character read from tape did not have an odd number of holes across the line. Place the correct character (if possible) in bits 12 through 17 of the ACS and press the CONTINUE key.
UAA	symbol	An undefined symbol appeared in an absolute address assignment (/). The current address indicator remains unchanged.
UPA	symbol	An undefined symbol appeared in a parameter assignment. The assignment is ignored.
UPN	symbol	An undefined symbol appeared in a PUNCH pseudo-instruction. The symbol is ignored.
UST	symbol	An undefined symbol appeared in a START or PAUSE instruction. The symbol is ignored and the START or PAUSE taken alone.

SUMMARY OF SYMBOLIC TAPE EDITOR OPERATIONS

TABLE 33 ACCUMULATOR SWITCH SETTINGS

Switch	Position	Function
0	Down	Normal operation.
	Up	Stop printing or punching
1	Down	Teletype without automatic tab.
	Up	Teletype with automatic tabulation.
13	Down	ASCII tape feeds punched as 000.
	Up	ASCII tape feeds punched as 200.
14	Down	ASCII output: transmit tab characters.
	Up	Convert tabs into proper number of spaces.

TABLE 33 ACCUMULATOR SWITCH SETTINGS (continued)

Switch	Position	Function
15	Down	Punch output in FIODEC code.
	Up	Punch output in ASCII code.
16	Down	Input tape is in FIODEC code.
	Up	Input tape is in ASCII code.
17	Down	Check parity on FIODEC input tape when reading.
	Up	Ignore parity errors.

Special Key Functions

rub out key (text mode)	Leave text mode (if first character typed); otherwise, erase last character.
rub out key (y command)	Stop output.
rub out key (command mode)	Print next line.
line feed	Print next line (if first character typed); otherwise delete all typed input.
carriage return	Complete specified action.

TABLE 34 EDITOR COMMAND SUMMARY

Command	Arguments	Function
A	0	Append.
B	0	Back up and print.
nC	1	Change line n.
n,mC	2	Change lines n through m.
nD	1	Delete line n.
n,mD	2	Delete lines n through m.
nF	1	Feed n lines of tape.
nG	1	Get next location tag after line n.
nI	1	Insert text before line n.
K	0	Kill the text buffer.

TABLE 34 EDITOR COMMAND SUMMARY (continued)

Command	Arguments	Function
nL	1	Print line n.
n,mL	2	Print lines n through m.
N	0	Punch and Next page. Equivalent to P, S, K, R.
mN	1	Punch, duplicate, and read. Equivalent to P, S, m-1 (T), R.
O	0	Punch and kill. Equivalent to P, S, K.
P	0	Punch the contents of the buffer.
nP	1	Punch line n.
n,mP	2	Punch lines n through m.
Q	0	Print uncommented—entire buffer.
nQ	1	Print line n uncommented.
n,mQ	2	Print lines n through m uncommented.
R	0	Read one page of text.
nR	1	Read n lines of tape.
S	0	Punch form feed (FIODEC: stop code).
nT	1	Duplicate n pages of tape. Equivalent to K, n(R, P, S, K).
W	0	Write the entire buffer.
nW	1	Write n pages. Equivalent to K, n(R, W, K).
n,mX	2	External insert. Insert n lines of text from tape after line m.
nY	1	Individual character correction on line n.
Z	0	Skip one page of text.
nZ	1	Skip n pages of text.

DIGITAL DEBUGGING TAPE (DDT)

TABLE 35 SUMMARY OF DDT COMMANDS

Character	Action
space	Separation character meaning arithmetic plus.
-	Separation character meaning arithmetic minus.
/	Register examination character: when following the address of a register, it causes the register to be opened and its contents printed. Immediately following a register printout, slash will cause the register addressed therein to be opened.

TABLE 35 SUMMARY OF DDT COMMANDS (continued)

Character	Action
carriage return	Make modifications, if any, and close register.
line feed	Make modifications, if any, close register, and open next sequential register.
& (ampersand)	Make modifications, if any, and open addressed register. (Establishes a new sequence.)
: (colon)	Type last quantity as an octal integer.
. (period)	Current location.
k#	Execute the expression k as an instruction.
Q$	Last quantity typed out by DDT.
k,	Define the symbol k as the tag of the currently open register.
)	Make modification and open addressed register. (The sequence is not changed.)
(...)	Define the enclosed symbol as the value preceding the (.
SYMBO$	Sets the mode in which DDT types out words to symbolic.
CONST$	Sets the mode in which DDT types out words to octal constants.
ABSOL$	Sets the mode in which DDT types out words to absolute. That is, the instruction code is typed as symbolic while the address is typed in octal.
RELAT$	Sets the mode in which DDT types out locations to relative (symbolic).
OCTAL$	Sets the mode in which DDT types out locations to octal.
N WORD$	Search for all occurrences masked with M$ of the expression N.
N NOT$	Search for all words not equal to the expression N after masking with M$.

TABLE 35 SUMMARY OF DDT COMMANDS (continued)

Character	Action
N ADDRE$	Search for all words masked with M$ with the same effective address as N.
KILL$	Resets the symbol table to the initial list. Modified definitions are retained only if altered on line. Definitions added from a user's symbol table tape are restored to their original values.
ZERO$	Clears memory available to the user.
k" (double quote)	Insert a breakpoint at the location specified by k. If no address is specified, remove any breakpoint.
! (exclamation)	Proceed from a breakpoint.
k' (single quote)	Transfer control to the location specified by k, or to the address in the start block on tape if no address is specified.
LOAD$	Load a FF format tape (storage words only).
TABLE$	Load only the symbols from a FF format tape.
DEBUG$	Load both storage words and symbols.
N PUNCH$	Punch the contents of N.
N; M PUNCH$	Punch N to M, inclusive.
INPUT$	Punch the input block.
START$	Punch a start block.
TRAP$	Place a trap location at 21 for CAL.

The following symbols are the address tags of certain registers in DDT whose contents are available to the user.

A$	Accumulator storage (at breakpoints).
L$	Link storage (at breakpoints).
M$	Mask used in search; M$+1 and M$+2 contain first and last address of the area to be searched.

TABLE 35 SUMMARY OF DDT COMMANDS (continued)

Character	Action
F$	Contains the lower limit of DDT as the address part of an XOR instruction.
B$	Contains the current breakpoint location.

APPENDIX 1

PDP-7 PROGRAM LIBRARY

Programs in the following list are available to users and purchasers of the PDP-7. Forward your requests to the Digital Program Library.

BASIC SOFTWARE PACKAGE

Name	Number
Symbolic Tape Editor	Digital-7-1-S
FORTRAN II System - 8K	Digital-7-2-S
Assembler - Basic & Extended	Digital-7-3-S
DDT - Basic & Extended	Digital-7-4-S
Teletype Output Package	Digital-7-10-O
Tic-Toc	Digital-7-11-IO
FF Loader	Digital-7-12-I
Readin Mode Loader	Digital-7-13-I
Octal Print Subroutine	Digital-7-14-O
Decimal Integer Print	Digital-7-15-O
Floating Point Package	Digital-7-30-A
Multiply Subroutine	Digital-7-31-A
Divide Subroutine	Digital-7-32-A
Double Precision Integer Package	Digital-7-33-A
Unsigned Multiply	Digital-7-34-A
Unsigned Divide	Digital-7-35-A
Master Tape Duplicator	Digital-7-40-U
Tape Reproducer	Digital-7-41-U
RIM Puncher	Digital-7-42-U
CAL Handler Type II	Digital-7-43-U
CAL Handler Type III	Digital-7-44-U

BASIC SOFTWARE FOR SPECIAL EQUIPMENT

Name	Number
Machines with DECtape	
DECtog	Digital-7-20-IO
DECtrieve	Digital-7-21-IO
DECtape Subroutines	Digital-7-22-IO
Machines with Card Reader, Card Punch, Line Printer	
Buffered Input - Output Package	Digital-7-23-IO
Machines with 30G or 30D Display	
Pen Follow Subroutine	Digital-7-24-IO
Character Display Subroutine	Digital-7-25-IO
Machines with Magnetic Tape	
Type 57A Compiler	Digital-7-45-U

BASIC MAINTENANCE ROUTINES

Name	Number
Teleprinter Input-Output Test	Digital-7-50-M
Clock Interrupt Test	Digital-7-51-M
Contest II	Digital-7-52-M
Reader & Punch Test	Digital-7-53-M
MAINDEC 401 (Instruction Test)	Digital-7-54-M
MAINDEC 402 (Memory)	Digital-7-55-M
MAINDEC 403 (Address Test)	Digital-7-56-M
MAINDEC 410 (RPB Test)	Digital-7-57-M

APPENDIX 2

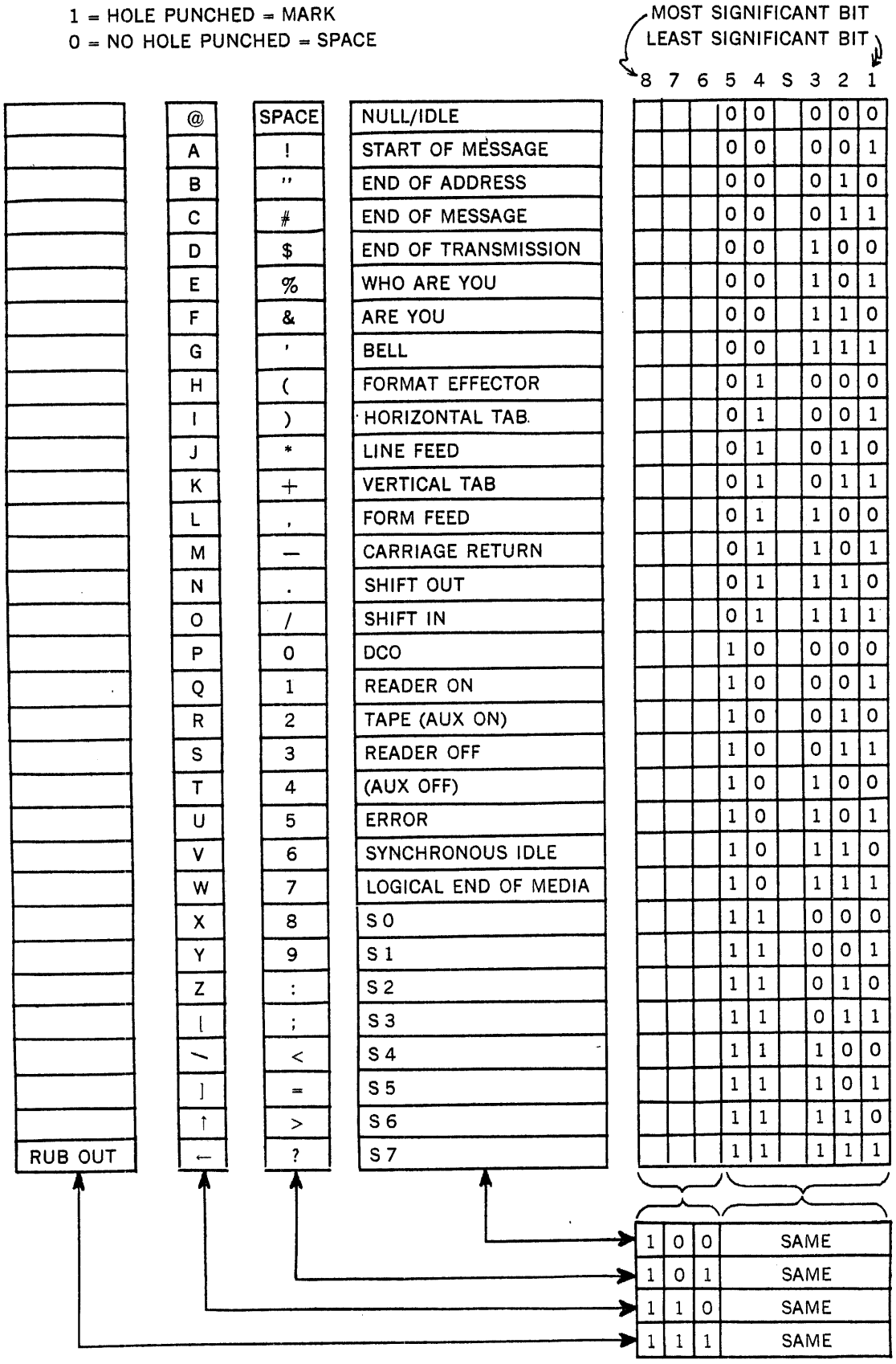
CODES

MODEL 33 ASR/KSR TELETYPE CODE (ASCII) IN OCTAL FORM

Character	8-Bit Code (in Octal)	Character	8-Bit Code (in Octal)
A	301	!	241
B	302	"	242
C	303	#	243
D	304	$	244
E	305	%	245
F	306	&	246
G	307	'	247
H	310	(	250
I	311	)	251
J	312	*	252
K	313	+	253
L	314	,	254
M	315	-	255
N	316	.	256
O	317	/	257
P	320	:	272
Q	321	;	273
R	322	<	274
S	323	=	275
T	324	>	276
U	325	?	277
V	326	@	300
W	327	[	333
X	330	/	334
Y	331	]	335
Z	332	↑	336
0	260	→	337
1	261	Leader/Trailer	200*
2	262	Line-Feed	212*
3	263	Carriage-Return	215
4	264	Space	240
5	265	Rub-out	377*
6	266	Blank	000*
7	267		
8	270	*Ignored by the operating system	
9	271		

MODEL 33 ASR/KSR TELETYPE CODE (ASCII) IN BINARY FORM

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



				8	7	6	5	4	S	3	2	1
	@	SPACE	NULL/IDLE				0	0		0	0	0
	A	!	START OF MESSAGE				0	0		0	0	1
	B	"	END OF ADDRESS				0	0		0	1	0
	C	#	END OF MESSAGE				0	0		0	1	1
	D	$	END OF TRANSMISSION				0	0		1	0	0
	E	%	WHO ARE YOU				0	0		1	0	1
	F	&	ARE YOU				0	0		1	1	0
	G	'	BELL				0	0		1	1	1
	H	(	FORMAT EFFECTOR				0	1		0	0	0
	I	)	HORIZONTAL TAB				0	1		0	0	1
	J	*	LINE FEED				0	1		0	1	0
	K	+	VERTICAL TAB				0	1		0	1	1
	L	,	FORM FEED				0	1		1	0	0
	M	—	CARRIAGE RETURN				0	1		1	0	1
	N	.	SHIFT OUT				0	1		1	1	0
	O	/	SHIFT IN				0	1		1	1	1
	P	0	DCO				1	0		0	0	0
	Q	1	READER ON				1	0		0	0	1
	R	2	TAPE (AUX ON)				1	0		0	1	0
	S	3	READER OFF				1	0		0	1	1
	T	4	(AUX OFF)				1	0		1	0	0
	U	5	ERROR				1	0		1	0	1
	V	6	SYNCHRONOUS IDLE				1	0		1	1	0
	W	7	LOGICAL END OF MEDIA				1	0		1	1	1
	X	8	S 0				1	1		0	0	0
	Y	9	S 1				1	1		0	0	1
	Z	:	S 2				1	1		0	1	0
	[	;	S 3				1	1		0	1	1
	\	<	S 4				1	1		1	0	0
	]	=	S 5				1	1		1	0	1
	↑	>	S 6				1	1		1	1	0
RUB OUT	←	?	S 7				1	1		1	1	1

1	0	0	SAME
1	0	1	SAME
1	1	0	SAME
1	1	1	SAME

The character codes of various machines are compared in the following table to simplify preperation of perforated program tapes off line.

TELETYPE CODE COMPARISON

Character Name	Flexowriter FIO DEC Code	28 KSR Baudot Code	33 KSR ASCII Code
	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	·	$.	n.e.
overbar	¯	'	#
	Stop Code ↓ Tab	n.e. ↓↓ bell	Form Feed ↓↓ Tab

CARD READER/PUNCH CODE (HOLLERITH) IN OCTAL FORM

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

CARD READER/PUNCH CODE (HOLLERITH) IN BINARY FORM

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	' [@]	) [□]	*	([%]

LINE PRINTER ASCII CODE IN OCTAL FORM

Character	6-Bit Trimmed Code (in octal)	Character	6-Bit Trimmed Code (in octal)
A	01	!	41
B	02	"	42
C	03	#	43
D	04	$	44
E	05	%	45
F	06	&	46
G	07	'	47
H	10	(	50
I	11	)	51
J	12	*	52
K	13	+	53
L	14	,	54
M	15	-	55
N	16	.	56
O	17	/	57
P	20	:	72
Q	21	;	73
R	22	<	74
S	23	=	75
T	24	>	76
U	25	?	77
V	26	@	00
W	27	[	33
X	30	\	34
Y	31	]	35
Z	32	↑	36
0	60	←	37
1	61	Space	40
2	62		
3	63		
4	64		
5	65		
6	66		
7	67		
8	70		
9	71		

LINE PRINTER FIODEC CODE IN OCTAL FORM

Octal Code	Line Printer Character	Octal Code	Line Printer Character	Octal Code	Line Printer Character
00	space	25	V	53	=
01	1	26	W	54	.
02	2	27	X	55	)
03	3	30	Y	56	—
04	4	31	Z	57	(
05	5	32	"	60	—
06	6	33	'	61	A
07	7	34	>	62	B
10	8	35	↑	63	C
11	9	36	→	64	D
12	'	37	?	65	E
13	~	40	°	66	F
14	....	41	J	67	G
15	....	42	K	70	H
16	....	43	L	71	I
17	<	44	M	72	×
20	0	45	N	73	.
21	/	46	O	74	+
22	S	47	P	75	]
23	T	50	Q	76	I
24	U	51	R	77	[

APPENDIX 3

SCALES OF NOTATION

2^x IN DECIMAL

x	2^x	x	2^x	x	2^x
0.001	1.00069 33874 62581	0.01	1.00695 55500 56719	0.1	1.07177 34625 36293
0.002	1.00138 72557 11335	0.02	1.01395 94797 90029	0.2	1.14869 83549 97035
0.003	1.00208 16050 79633	0.03	1.02101 21257 07193	0.3	1.23114 44133 44916
0.004	1.00277 64359 01078	0.04	1.02811 38266 56067	0.4	1.31950 79107 72894
0.005	1.00347 17485 09503	0.05	1.03526 49238 41377	0.5	1.41421 35623 73095
0.006	1.00416 75432 38973	0.06	1.04246 57608 41121	0.6	1.51571 65665 10398
0.007	1.00486 38204 23785	0.07	1.04971 66836 23067	0.7	1.62450 47927 12471
0.008	1.00556 05803 98468	0.08	1.05701 80405 61380	0.8	1.74110 11265 92248
0.009	1.00625 78234 97782	0.09	1.06437 01824 53360	0.9	1.86606 59830 73615

$10^{\pm n}$ IN OCTAL

10^n	n	10^{-n}	10^n	n	10^{-n}
1	0	1.000 000 000 000 000 000 000 00	112 402 762 000	10	0.000 000 000 006 676 337 66
12	1	0.063 146 314 631 463 146 31	1 351 035 564 000	11	0.000 000 000 000 537 657 77
144	2	0.005 075 341 217 270 243 66	16 432 451 210 000	12	0.000 000 000 000 043 136 32
1 750	3	0.000 406 111 564 570 651 77	221 411 634 520 000	13	0.000 000 000 000 003 411 35
23 420	4	0.000 032 155 613 530 704 15	2 657 142 036 440 000	14	0.000 000 000 000 000 264 11
303 240	5	0.000 002 476 132 610 706 64	34 327 724 461 500 000	15	0.000 000 000 000 000 022 01
3 641 100	6	0.000 000 206 157 364 055 37	434 157 115 760 200 000	16	0.000 000 000 000 000 001 63
46 113 200	7	0.000 000 015 327 745 152 75	5 432 127 413 542 400 000	17	0.000 000 000 000 000 000 14
575 360 400	8	0.000 000 001 257 143 561 06	67 405 553 164 731 000 000	18	0.000 000 000 000 000 000 01
7 346 545 000	9	0.000 000 000 104 560 276 41			

$n \log_{10} 2$, $n \log_2 10$ IN DECIMAL

n	$n \log_{10} 2$	$n \log_2 10$	n	$n \log_{10} 2$	$n \log_2 10$
1	0.30102 99957	3.32192 80949	6	1.80617 99740	19.93156 85693
2	0.60205 99913	6.64385 61898	7	2.10720 99696	23.25349 66642
3	0.90308 99870	9.96578 42847	8	2.40823 99653	26.57542 47591
4	1.20411 99827	13.28771 23795	9	2.70926 99610	29.89735 28540
5	1.50514 99783	16.60964 04744	10	3.01029 99566	33.21928 09489

ADDITION AND MULTIPLICATION TABLES

Addition **Multiplication**

Binary Scale

$$0+1=\begin{matrix}0+0=0\\1+0=1\\1+1=10\end{matrix} \qquad 0\times 1=\begin{matrix}0\times 0=0\\1\times 0=0\\1\times 1=1\end{matrix}$$

Octal Scale

0	01	02	03	04	05	06	07
1	02	03	04	05	06	07	10
2	03	04	05	06	07	10	11
3	04	05	06	07	10	11	12
4	05	06	07	10	11	12	13
5	06	07	10	11	12	13	14
6	07	10	11	12	13	14	15
7	10	11	12	13	14	15	16

1	02	03	04	05	06	07
2	04	06	10	12	14	16
3	06	11	14	17	22	25
4	10	14	20	24	30	34
5	12	17	24	31	36	43
6	14	22	30	36	44	52
7	16	25	34	43	52	61

MATHEMATICAL CONSTANTS IN OCTAL SCALE

$\pi = 3.11037\ 552421_8$	$e = 2.55760\ 521305_8$	$\gamma = 0.44742\ 147707_8$
$\pi^{-1} = 0.24276\ 301556_8$	$e^{-1} = 0.27426\ 530661_8$	$\ln \gamma = -0.43127\ 233602_8$
$\sqrt{\pi} = 1.61337\ 611067_8$	$\sqrt{e} = 1.51411\ 230704_8$	$\log_2 \gamma = -0.62573\ 030645_8$
$\ln \pi = 1.11206\ 404435_8$	$\log_{10} e = 0.33626\ 754251_8$	$\sqrt{2} = 1.32404\ 746320_8$
$\log_2 \pi = 1.51544\ 163223_8$	$\log_2 e = 1.34252\ 166245_8$	$\ln 2 = 0.54271\ 027760_8$
$\sqrt{10} = 3.12305\ 407267_8$	$\log_2 10 = 3.24464\ 741136_8$	$\ln 10 = 2.23273\ 067355_8$

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

OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal)	0000 to 0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal)	0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

2000	1024
to	to
2777	1535
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

3000	1536
to	to
3777	2047
(Octal)	(Decimal)

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

4000 to 4777 (Octal)	2048 to 2559 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 to 5777 (Octal)	2560 to 3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

OCTAL-DECIMAL INTEGER CONVERSION TABLE (continued)

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000 to 6777 (Octal)	3072 to 3583 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	1022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000 to 7777 (Octal)	3584 to 4095 (Decimal)

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

OCTAL-DECIMAL FRACTION CONVERSION TABLE (continued)

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.0003..	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

OCTAL-DECIMAL FRACTION CONVERSION TABLE (continued)

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.0012	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

APPENDIX 4

INSTRUCTION SUMMARY

MEMORY REFERENCE INSTRUCTIONS

Mnemonic Symbol	Octal Code	Machine Cycles	Operation Executed
CAL	00	2	Call subroutine. The address portion of this instruction is ignored. The action is identical to JMS 20.
DAC Y	04	2	Deposit AC. The content of the AC is deposited in the memory cell at location Y.
JMS Y	10	2	Jump to subroutine. The content of the PC and the content of the L is deposited in memory cell Y. The next instruction is taken from cell Y + 1.
DZM Y	14	2	Deposit zero in memory. Zero is deposited in memory cell Y.
LAC Y	20	2	Load AC. The content of Y is loaded into the AC.
XOR Y	24	2	Exclusive OR. The exclusive OR is performed between the content of Y and the content of the AC, with the result left in the AC.
ADD Y	30	2	Add (1's complement). The content of Y is added to the content of the AC in 1's complement arithmetic and the result is left in the AC.
TAD Y	34	2	Two's complement add. The content of Y is added to the content of the AC in 2's complement arithmetic and the result is left in the AC.

MEMORY REFERENCE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Machine Cycles	Operation Executed
XCT Y	40	1+	Execute. The instruction in memory cell Y is executed.
ISZ Y	44	2	Increment and skip if zero. The content of Y is incremented by one in 2's complement arithmetic. If the result is zero, the next instruction is skipped.
AND Y	50	2	AND. The logical operation AND is performed between the content of Y and the content of the AC with the result left in the AC.
SAD Y	54	2	Skip if AC is different from Y. The content of Y is compared with the content of the AC. If the numbers are different, the next instruction is skipped.
JMP Y	60	1	Jump to Y. The next instruction to be executed is taken from memory cell Y.

EAE INSTRUCTION LIST

Mnemonic Symbol	Octal Code	Operation Executed
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).

EAE INSTRUCTION LIST (continued)

Mnemonic Symbol	Octal Code	Operation Executed
NORM	640444	Normalize, unsigned. Maximum shift is 44_8.
NORMS	660444	Normalize, signed (AC sign = L).
MUL	653122	Multiply, unsigned. The number in the AC is multiplied by the number in the next core memory address.
MULS	657122	Multiply, signed. The number in the AC is multiplied by the number in the next core memory address.
DIV	640323	Divide, unsigned. The 36-bit content of both the AC and MQ is divided by the number in the next core memory location.
DIVS	644323	Divide, signed. The content of both the AC and MQ as a 1's complement signed number is divided by the number in the next core memory location.
IDIV	653323	Integer divide, unsigned. Divide the number in the AC as an 18-bit unsigned integer by the number in the next core memory location.
IDIVS	657323	Integer divide, signed. Same as IDIV but the content of the AC is a 17-bit signed number.
FRDIV	650323	Fraction divide, unsigned. Divide the 18-bit fraction in the AC by the 18-bit fraction in the number in the next core memory location.
FRDIVS	654323	Fraction divide, signed. Same as FRDIV, but the content of the AC is a 17-bit signed number.
LACQ	641002	Replace the content of the AC with the content of the MQ.
LACS	641001	Replace the content of the AC with the content of the SC.
CLQ	650000	Clear MQ.

EAE INSTRUCTION LIST (continued)

Mnemonic Symbol	Octal Code	Operation Executed
ABS	644000	Place absolute value of AC in the AC.
GSM	664000	Get sign and magnitude. 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.
LMQ	652000	Load MQ

INPUT/OUTPUT TRANSFER INSTRUCTIONS

Mnemonic Symbol	Octal Code•	Operation Executed
		Program Interrupt
IOF	700002	Interrupt off. Disable the PIC.
ION	700042	Interrupt on. Enable the PIC.
ITON	700062	Interrupt and trap on. Enable PIC and trap mode.
		Real Time 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.
		Perforated Tape Reader
RSF	700101	Skip if reader flag is a 1.

INPUT/OUTPUT TRANSFER INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
		Perforated Tape Reader (continued)
RCF	700102	Clear reader flag, then inclusively OR the content of reader buffer into the AC.
RRB	700112	Read reader buffer. Clear reader flag and AC, and then transfer content of reader buffer into AC.
RSA	700104	Select reader in alphanumeric mode. One 8-bit character is read into the reader buffer.
RSB	700144	Select reader in binary mode. Three 6-bit characters are read into the reader buffer.
		Perforated Tape Punch
PSF	700201	Skip if the punch flag is set to 1.
PCF	700202	Clear the punch flag.
PSA or PLS	700204 700206	Punch a line of tape in alphanumeric mode.
PSB	700244	Punch a line of tape in binary mode.
		I/O Equipment
I/ORS	700314	Input/output read status. The content of given flags replace the content of the assigned AC bits.
TTS	703301	Test Teletype and skip if KSR 33 is connected to computer.
CAF	703302	Clear all flags.
SKP7	703341	Skip if processor is a PDP-7.

Mnemonic Symbol	Octal Code	Operation Executed
		Teletype Keyboard
KSF	700301	Skip if the keyboard flag is set to 1.
KRB	700312	Read the keyboard buffer. The content of the buffer is placed in AC10-17 and the keyboard flag is cleared.
		Teletype Teleprinter
TSF	700401	Skip if the teleprinter flag is set.
TCF	700402	Clear the teleprinter flag.
TLS	700406	Load teleprinter buffer. The content of AC10-17 is placed in the buffer and printed. The flag is cleared before transmission takes place and is set when the character has been printed.
		Oscilloscope and Precision CRT Displays
DXC	700502	Clear the X-coordinate buffer.
DYC	700602	Clear the Y-coordinate buffer.
DXL	700506	Load the X-coordinate buffer from AC8-17.
DYL	700606	Load the Y-coordinate buffer from AC8-17.
DXS	700546	Load the X-coordinate buffer and display the point specified by the XB and YB.
DYS	700646	Load the Y-coordinate buffer and display the point specified by the XB and YB.
DSF	700701	Skip if display flag = 1.
DCF	700702	Clear display flag.
DLB	700706	Load the brightness register from AC15-17.

Mnemonic Symbol	Octal Code	Operation Executed
		Precision Incremental Display
IDVE	700501	Skip on vertical edge violation.
IDSI	700601	Skip on stop interrupt.
IDSP	700701	Skip if light pen flag is set.
IDHE	701001	Skip on horizontal edge violation.
IDRS	700504	Continue display. After a light pen interrupt, this command causes the display to resume at the point indicated by the content of the DAC.
IDRA	700512	Read display address. Transfers the address in the DAC to AC5-17.
IDLA	700606	Load address and select. The content of AC5-17 are placed in the DAC and the display is started.
IDRD	700614	Restart display. After a stop code interrupt, this command causes the display to resume at the point indicated by the content of the DAC.
IDCF	700704	Clear display control. All flags and interrupts are cleared.
IDRC	700712	Read X and Y coordinates. The content of bits XB0-8 is transferred into AC0-8 and the content of YB0-8 is transferred into AC9-17.
		Symbol Generator
GCL	700641	Clear done flag (also done by GPL or GPR).
GSF	701001	Skip on done.
GPL	701002	Generator plot left.
GLF	701004	Load format (bit 15 for space, bits 16 and 17 for size).

Mnemonic Symbol	Octal Code	Operation Executed
		Symbol Generator (continued)
GPR	701042	Generator plot right.
GSP	701084	Plot a space.
		General Purpose Multiplexer Control
ADSM	701103	Select MX channel. The content of AC12-17 is placed in the MAR.
ADIM	701201	Increment channel address. The content of the MAR is incremented by 1. Channel 0 follows channel 77_8.
		Analog-to-Digital Converters
ADSF	701301	Skip if converter flag is set.
ADSC	701304	Select and convert. The converter flag is cleared and a conversion is initiated.
ADRB	701312	Read converter buffer. Places the content of the buffer in the AC.
		Relay Buffer
ORC	702101	Clear output relay buffer flip-flop register.
ORS	702104	Set output relay buffer flip-flop register to correspond with the contents of the accumulator.
		Inter Processor Buffer
IPSI	702201	Skip on IPB information flag.
IPRB	702212	Read IPB buffer. The content of the IPB is read into the AC and the information flag is cleared. The buffer becomes available to one of the processors.

Mnemonic Symbol	Octal Code	Operation Executed
		Inter Processor Buffer (continued)
IPLB	702024	Load IPB buffer. The content of the AC is loaded into the IPB and the information flag of the other processor is set.
IPSA	702301	Skip on IPB available. The next instruction is skipped if the buffer is ready to accept data from the AC.
IPDA	702302	Disable IPB available. The available flag of the processor issuing the command is unconditionally cleared.
IPEA	702304	Enable IPB available.
		Incremental Plotter and Control
PLSF	702401	Skip if plotter flag is a 1.
PLCF	702402	Clear plotter flag.
PLPU	702404	Plotter pen up. Raise pen off of paper.
PLPR	702501	Plotter pen right.
PLDU	702502	Plotter drum (paper) upward.
PLDD	702504	Plotter drum (paper) downward.
PLPL	702601	Plotter pen left.
PLUD	702602	Plotter drum (paper) upward.
PLPD	702604	Plotter pen down. Lower pen on to paper.

Mnemonic Symbol	Octal Code	Operation Executed
		Data Control
STC	704001	Skip on transfer complete. The next instruction is skipped if the done flag in the data control is set to 1.
LWC	704006	Load word count. The WC register is cleared then loaded from the content of AC3-17.
SEC	704101	Skip on error condition. If an error signal has been received by the data control from the selected device, the next instruction is skipped.
LAR	704106	Load address register. The AR is cleared then loaded by OR transfer from the content of AC3-17.
CDC	704201	Clear data control. All flags and registers of the data control are cleared, and the busy status is set.
LCW	704205	Load control word. The control word contained in the AC is transferred into the data control status register.
RWC	704212	Read word count. The AC is cleared, then the content of the WC is transferred into AC3-17.
		Automatic Priority Interrupt
CAC	705501	Clear all channels. Turn off all channels.
ASC	705502	Enable selected channel(s). AC2-17 are used to select the channel(s).
DSC	705604	Disable selected channel(s). AC2-17 are used to select the channel(s).
EPI	700044	Enable automatic priority interrupt system.
DPI	700004	Disable automatic priority interrupt system.

Mnemonic Symbol	Octal Code	Operation Executed
		Automatic Priority Interrupt (continued)
ISC	705504	Initiate break on selected channel (for maintenance purposes). AC2-17 are used to select the channel.
DBR	705601	Debreak. Returns highest priority channel to receptive state.
		Serial Drum
DRLR	706006	Load counter and read. Places the content of AC2-17 in the DCL and prepares the drum system for reading a block into core memory.
DRLW	706046	Load counter and write. Loads the DCL from AC2-17 and prepares the drum system for writing a block to be received from core memory.
DRSF	706101	Skip if drum transfer flag is set.
DRCF	706102	Clear drum transfer and error flags.
DRSS	706106	Load sector and select. Places the content of AC9-17 in the DTR, clears both drum flags, and initiates the block transfer.
DRSN	706201	Skip if drum error flag is not set.
DRCS	706204	Continue select. Clears the flags and iniates a transfer as specified by the content of the DCL and DTR.
		Card Punch
CPSF	706401	Skip if card reader flag is set.
CPLR	706406	Load the punch buffer, clear punch flag.
CPCF	706442	Clear the card row flag.
CPSE	706444	Select the card punch. A card starts moving from the hopper to the punch station and load the card punch buffer from the AC.

Mnemonic Symbol	Octal Code	Operation Executed
		Automatic Line Printer
LPSF	706501	Skip if printing done flag = 1.
LPCF	706502	Clear printing done flag.
LPL1	706562	Load one character into printing buffer.
LPL2	706522	Load two characters into printing buffer.
LPLD	706542	Load printing buffer (three characters).
LPSE	706506	Select printer and print.
LSSF	706601	Skip if spacing flag = 1.
LSCF	706602	Clear spacing flag.
LSLS	706604	Load spacing buffer and space.
		Card Readers
CRSF	706701	Skip if the card reader flag is set.
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.
CRRB	706712	Read the card reader buffer. The content of the CRB is placed in AC6-17.
CRSB	706744	Select and read a card in binary mode. A card is started through the reader and 80 columns are read as 12-bit numbers.
		Automatic Magnetic Tape Control
MSCR	707001	Skip if the tape control is ready (TCR=1).
MSUR	707101	Skip if the tape unit is ready (TTR).

Mnemonic Symbol	Octal Code	Operation Executed
		Automatic Magnetic Tape Control (continued)
*MCC	707401	Clear CA and WC.
MCA	707405	Clear CA and WC, and transfer the content of AC5-17 into the CA.
MWC	707402	Load WC. Transfers the content of AC5-17 into the WC.
MRCA	707414	Transfer the content of the CA into AC5-17.
MCD	707042	Disable TCR and clear CR. Clear WCO and EOR flags.
MTS	707006	Clear control register (CR) and clear job done, WCO, and EOR flags. Transmit unit, parity, and density to tape control.
MTC	707106	Transmit tape command and start. This command initiates the transfer.
MNC	707152	End continuous mode. Clears the AC; the operation terminates at the end of the current record.
MRD	707204	Switch mode from read to read/compare.
MRCR	707244	Switch from read/compare to read.
MSEF	707301	Skip if EOR flag is set.
*MDEF	707302	Disable EOR flag.
*MCEF	707322	Clear EOR flag.
*MEEF	707342	Enable EOR flag.
MIEF	707362	Initialize EOR flag. Clears and enables the flag.

*These instructions must be combined with other commands to be recognized by the PDP-7 Symbolic Assembler.

INPUT/OUTPUT TRANSFER INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation Executed
		Automatic Magnetic Tape Control (continued)
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.
MTRS	707314	Read tape status.
		DECtape System
MMRD	707512	Read. Clears the AC and transfers one word from the data buffer in the control into AC0-17.
MMWR	707504	Write. Transfers one word from AC0-17 to the data buffer in the control.
MMSE	707644	Select. Connects the unit designated in AC2-5 to the DECtape control.
MMLC	707604	Load control. Sets the DECtape control to the proper mode and direction from AC12-17.
MMRS	707612	Read status. Clear the AC and transfer the DECtape status conditions into AC0-8.
MMDF	707501	Skip on DECtape data flag.
MMBF	707601	Skip on DECtape block end flag.
MMEF	707541	Skip on DECtape error flag.

*These instructions must be combined with other commands to be recognized by the PDP-7 Symbolic Assembler.

INPUT/OUTPUT TRANSFER INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Event Time	Operation Executed
		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.

OPERATE INSTRUCTIONS

Mnemonic Symbol	Octal Code	Event Time	Operation Executed
OPR or NOP	740000	---	Operate group or no operation. Causes a 1-cycle program delay.
CMA	740001	3	Complement accumulator. Each bit of the AC is complemented.
CML	740002	3	Complement link.
OAS	740004	3	Inclusive OR ACCUMULATOR switches. The word set into the ACCUMULATOR switches is OR combined with the content of the AC, the result remains in the AC.
RAL	740010	3	Rotate accumulator left. The content of the AC and L are rotated one position to the left.
RLR	740020	2	Rotate accumulator right. The content of the AC and L are rotated one position to the right.
HLT	740040	---	Halt. The program is stopped at the conclusion of the cycle.
SMA	740100	1	Skip on minus accumulator. If the content of the AC is negative (2's complement) number the next instruction is skipped.

Mnemonic Symbol	Octal Code	Event Time	Operation Executed
SZA	740200	1	Skip on zero accumulator. If the content of the AC equals zero (2's complement), the next instruction is skipped.
SNL	740400	1	Skip on non-zero link. If the L contains a 1, the next instruction is skipped.
SKP	741000	1	Skip. The next instruction is unconditionally skipped.
SPA	741100	1	Skip on positive accumulator. If the content of the AC is zero (2's complement) or a positive number, the next instruction is skipped.
SNA	741200	1	Skip on non-zero accumulator. If the content of the AC is not zero (2's complement), the next instruction is skipped.
SZL	741400	1	Skip on zero link. If the L contains a 0, the next instruction is skipped.
RTL	742010	2,3	Rotate two left. The content of the AC and the L are rotated two positions to the left.
RTR	742020	2,3	Rotate two right. The content of the AC and the L are rotated two positions to the right.
CLL	744000	2	Clear link. The L is cleared.
STL	744002	2,3	Set link. The L is set to 1.
RCL	744010	2,3	Clear link, then rotate left. The L is cleared, then the L and AC are rotated one position left.
RCR	744020	2,3	Clear link, then rotate right. The L is cleared, then the L and AC are rotated one position right.
CLA	750000	2	Clear accumulator. Each bit of the AC is cleared.

OPERATE INSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Event Time	Operation Executes
CLC	750001	2,3	Clear and complement accumulator. Each bit of the AC is set to contain a 1.
LAS	750004	2,3	Load accumulator from switches. The word set into the ACCUMULATOR switches is loaded into the AC.
GLK	750010	2,3	Get link. The content of L is set into AC17.
LAW N	76XXXX	---	Load the AC with LAW N.

PRINTED IN U.S.A.

20-6/65