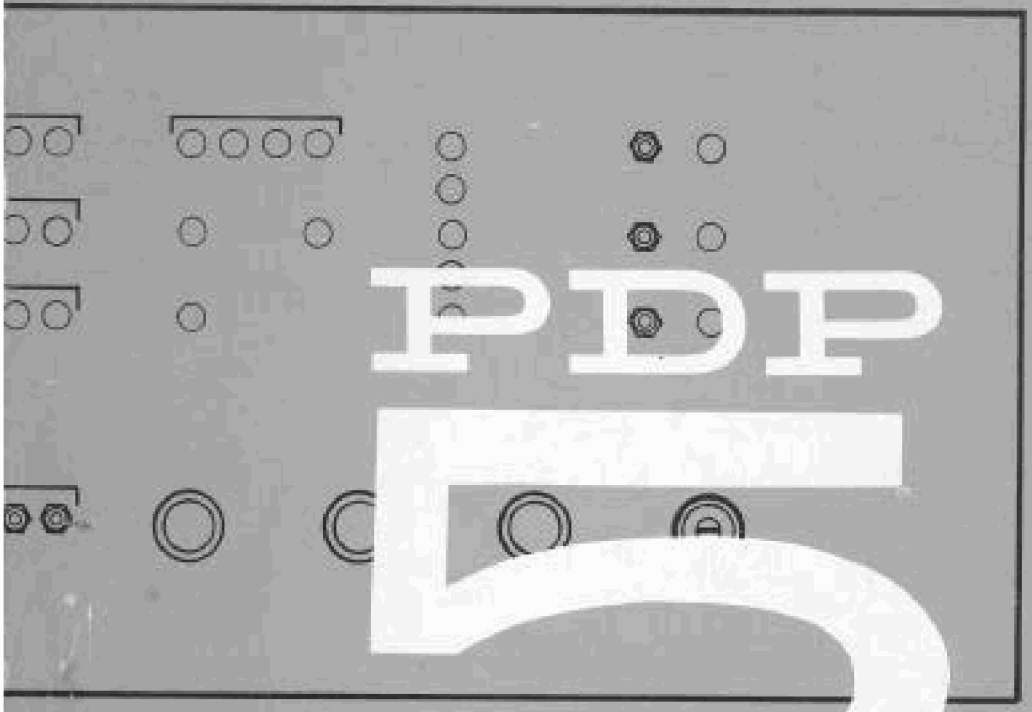
F-55
2/64



PDP 5

HANDBOOK

PROGRAMMED DATA PROCESSOR-5 HANDBOOK

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

FOREWORD

This handbook concerns programming and operating the Programmed Data Processor-5, a high-speed, stored program, digital computer manufactured by the Digital Equipment Corporation. Chapter 1 summarizes the electrical and logical features of the computing system and analyzes it into three major functional elements: arithmetic and control, input-output control, and input-output devices. Chapters 2, 3, and 4 present detailed information on the function, instructions, and programming of the three major system elements. Practical information for making electrical connections between any input-output device and the computing system at the input-output control is presented in Chapter 5. Appendixes provide 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 Program Assembly Language, available from the DEC Program Library.

Table of Contents

Table of Contents (continued)

List of Tables

List of Illustrations

Figure 1 Typical PDP-5 Computing System

CHAPTER 1
SYSTEM INTRODUCTION

The Digital Equipment Corporation Programmed Data Processor-5 (PDP-5) is designed for use as a small-scale general-purpose computer, an independent information handling facility in a larger computer system, or as the control element in a complex processing system. The PDP-5 is a one-address, fixed word length, parallel computer using 12 bit, twos complement arithmetic. Cycle time of the 1024- or 4096-word random address magnetic-core memory is 6 microseconds. Standard features of the system include indirect addressing and facilities for instruction skipping, program interruption, or program halting as functions of input-output device conditions.

The 6-microsecond cycle time of the machine provides a computation rate of 55,555 additions per second. Addition is performed in 18 microseconds (with one number in the accumulator) and subtraction is performed in 30 microseconds (with the subtrahend in the accumulator). Multiplication is performed in approximately 2.0 milliseconds by a subroutine that operates on two 12-bit numbers to produce a 24-bit product, leaving the 12 most significant bits in the accumulator. Division of two 12-bit numbers is performed in approximately 3.5 milliseconds by a subroutine that produces a 12-bit quotient in the accumulator and a 12-bit remainder in core memory.

Flexible, high-capacity, input-output capabilities of the computer allow it to operate a variety of peripheral equipment. In addition to standard Teletype and perforated-tape equipment, the system is capable of operating in conjunction with a number of optional devices such as high-speed perforated-tape readers and punches, card equipment, a line printer, analog-to-digital converters, cathode-ray tube displays, and magnetic-tape equipment. The system is easily adapted for connection to equipment of special design.

PDP-5 is completely self-contained, requiring no special power sources or environmental conditions. A single source of 115-volt, 60-cycle, single-phase power is required to operate the machine. Internal power supplies produce all of the operating voltages required. Solid-state system modules and built-in provisions for marginal checking insure reliable operation in ambient temperatures between 50 and 105 degrees Fahrenheit.

The primary functions of the PDP-5 system are performed by an arithmetic and control element, an input-output control element, and the input-output devices. Figure 2 shows the relationship of these elements.

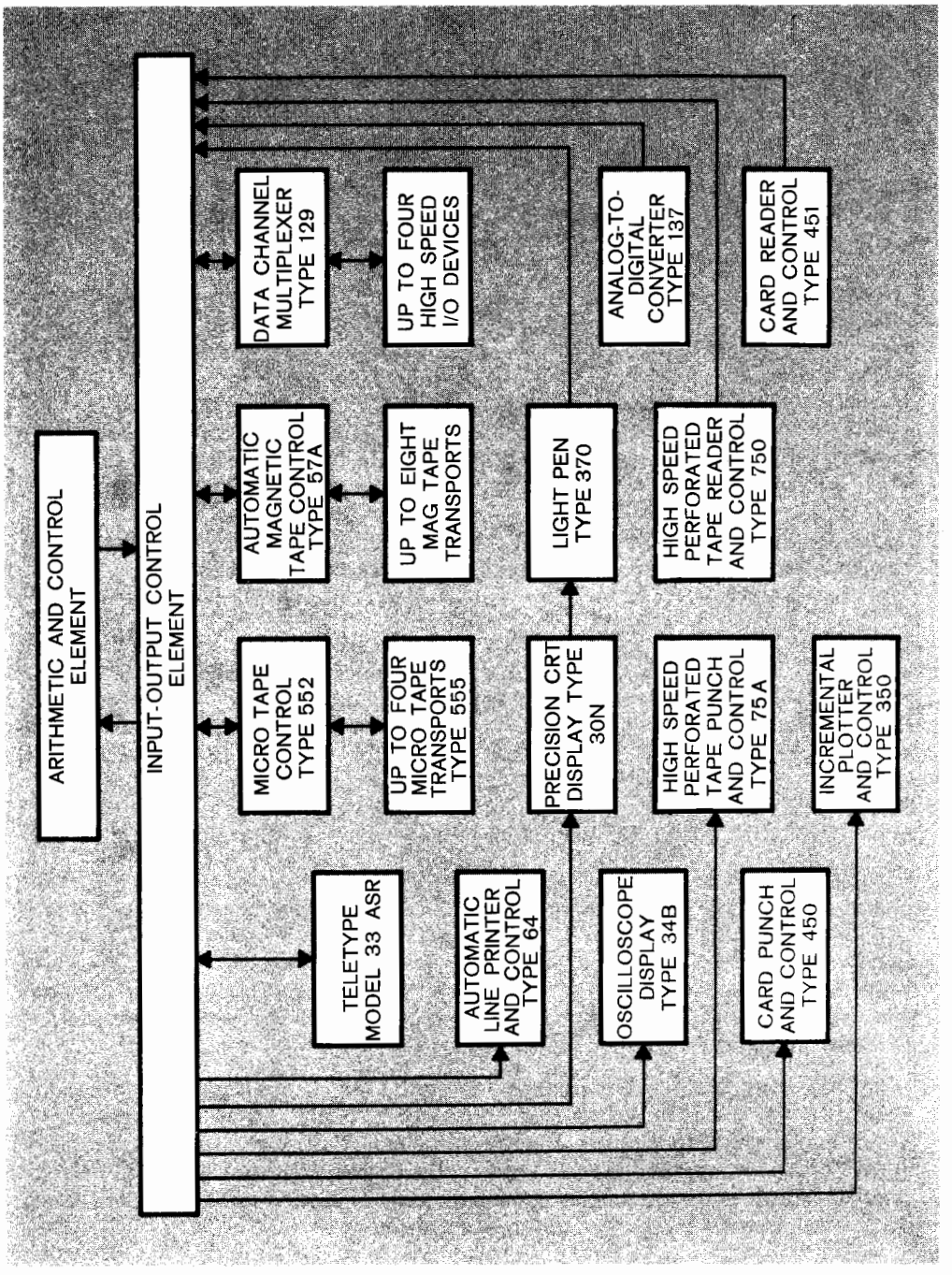


Figure 2 PDP-5 System Components

The arithmetic and control element contains all of the registers that perform arithmetic and logic operations, the core memory for storage and retrieval of data and instructions, and the operator console, which indicates the contents of registers and provides a means of modifying data. Chapter 2 of this handbook describes these functions in detail.

The input-output control element provides communications between the arithmetic and control element and standard, optional, or special input-output devices. Components of this control may be housed in the main computer cabinet or with the I/O device. I/O device selection, input-output skip, program interrupt, input-output halt, and data break control are features of this element and are discussed in Chapter 3.

All of the input-output devices are optional except the Teletype Model 33 ASR. Standard and optional I/O equipment programming information is presented in Chapter 4.

THE TELETYPE MODEL 33 ASR provides a means of supplying data to the computer from perforated tape or a keyboard, or of receiving output information from the computer in the form of perforated tape or typed copy. Maximum speed of these operations is ten characters per second.

THE HIGH SPEED PERFORATED TAPE PUNCH AND CONTROL TYPE 75A perforates 8-hole paper tape at a rate of 63.3 lines per second.

THE HIGH SPEED PERFORATED TAPE READER AND CONTROL TYPE 750 senses 8-hole perforated paper tape photoelectrically at the rate of 300 lines per second.

THE ANALOG-TO-DIGITAL CONVERTER TYPE 137 is wired into each system. Modules to activate this feature are optional. The converter operates in the normal successive approximation fashion, using existing computer registers as the shift register and buffer register. The converter provides a 12-bit word; however, the last bit is insignificant.

THE CARD READER AND CONTROL TYPE 451 operates at a rate of 200 or 800 cards per minute. Cards are read column by column. Column information may be read in alphanumeric or binary mode. The alphanumeric mode converts the 12-bit Hollerith Code of one column into the 6-bit binary-coded decimal code with code validity checking. The binary mode reads a 12-bit column directly into the PDP-5. Approximately one percent of a Card Reader program running time is required to read the 80 columns of information at the 200 cards per minute rate.

THE CARD PUNCH CONTROL TYPE 450 permits operation of a standard IBM Type 523 Summary Punch with the PDP-5. Punching can occur at a rate of 100 cards per minute. Cards are punched one row at a time at 40-millisecond intervals.

THE AUTOMATIC LINE PRINTER AND CONTROL TYPE 64 prints a selection of 63 characters at up to 300 lines of 120 characters per minute. Printing of one group of 120 characters can be carried out while the next 120 characters are being loaded into the printer. Loading, printing, and format are under program control. Format is program selected from a punched format tape in the printer.

THE INCREMENTAL PLOTTER AND CONTROL TYPE 350 provides high-speed plotting of points, continuous curves, points connected by curves, curve identification symbols, letters, and numerals under program control.

THE DATA CHANNEL MULTIPLEXER TYPE 129 automatically transfers data directly between the computer core memory and up to four I/O devices. The computer core memory address of each transfer is specified by the I/O device. Transfers are made through the normal data break facilities and breaks are performed in accordance with an assigned I/O device priority.

THE OSCILLOSCOPE DISPLAY TYPE 34B plots data point by point on a high resolution oscilloscope, such as the Tektronix Model RM 503. Each axis is determined by 10 binary bits.

THE PRECISION CRT DISPLAY TYPE 30N displays data on a 9¼ inch by 9¼ inch area. Information is plotted point by point to form either graphical or tabular data. The X and Y coordinates are each controlled by a separate 10-bit word.

THE LIGHT TYPE PEN 370 is a photoelectric device which signals the computer when it detects information displayed on the Type 30N Precision CRT Display. Upon signal from the light pen, the computer carries out previously programmed instructions.

THE DUAL MICRO TAPE SYSTEM TYPE 555-552 provides a fixed-address magnetic-tape facility for high-speed loading, readout, and program updating. A system consists of a Type 555 Micro Tape Transport and a Type 552 Micro Tape Control. Each transport contains two independent tape drivers. Up to four transports (8 drives) can be used with one control. Each reel, containing up to four-million bits of data, can be written or read under program control.

THE AUTOMATIC MAGNETIC TAPE CONTROL TYPE 57A reads and writes high and low density, IBM compatible magnetic tape at a transfer rate of 15,000 characters per second.

The following special terms are used throughout this handbook in the explanation of equipment functions and instructions:

Term	Explanation
C(A)	Contents of A
$A \Rightarrow B$	A replaces B or B is set to A
Y	Any core memory location
Y_j	Any given bit in Y
Y_{1-4}	Bits 1 through 4 of Y
Y_j^1	The 1 output of bit j of register Y
N_r	Number N to the radix r
$C(A)_{0-5} \Rightarrow C(Y)_{6-11}$	The contents of bits 6 through 11 of core memory location Y are set to correspond with the contents of bits 0 through 5 of register A
$\not\vee$	Exclusive OR
V	Inclusive OR
$\wedge$	AND
$\overline{A}$	Ones complement of A

CHAPTER 2

ARITHMETIC AND CONTROL

Functions

To perform the required arithmetic, logic, and data processing operations and to store, retrieve, control, and modify information the arithmetic and control element uses the logic components shown in Figure 3 and described in the following paragraphs.

ACCUMULATOR (AC)

Arithmetic operations are performed in this 12-bit register. The AC can be cleared or complemented. Its contents can be rotated right or left with the link. The contents of the memory buffer register can be added to the contents of the AC and the result left in the AC. The contents of both these registers may be combined by the logical operation AND, the result remaining in the AC. The memory buffer register and the AC also have gates which allow them to be used together as the shift register and buffer register of a successive approximation analog-to-digital converter. The inclusive OR may be formed between the AC and the switch register on the operator console and the result left in the AC.

The accumulator also acts as an input-output register. All programmed information transfers between core memory and an external device pass through the accumulator.

LINK (L)

This one-bit register is used to extend the arithmetic facilities of the accumulator. It is used as the carry register for twos complement arithmetic. This feature greatly simplifies multiple precision arithmetic. The link can be cleared and complemented, and it can be rotated as part of the accumulator.

MEMORY BUFFER REGISTER (MB)

All information transfers between the computer registers and the core memory are temporarily held in the MB. Information can be transferred into MB from the accumulator or memory address register. The MB can be cleared, incremented by one or two, or shifted right. Information can be set into the

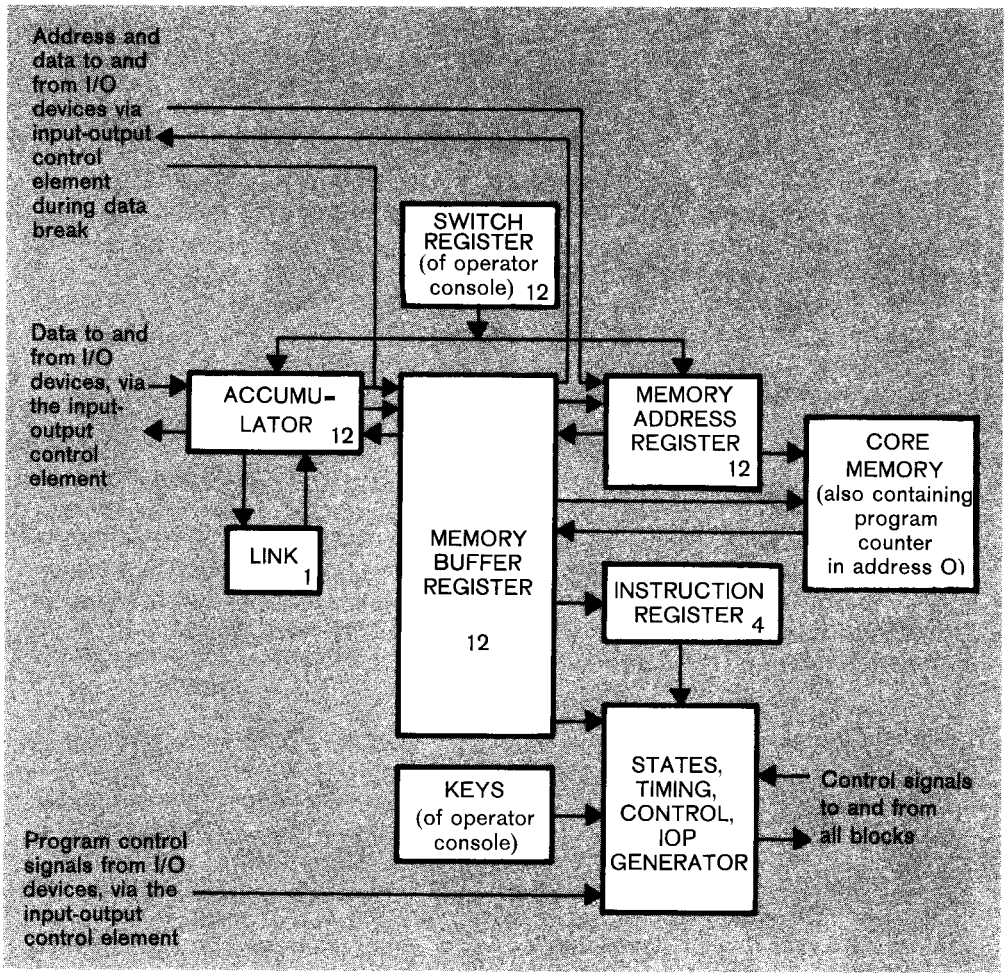


Figure 3 Arithmetic and Control Element

MB from an external device during a data break or from core memory, via the sense amplifiers. Information is read from a memory location in 2 microseconds and rewritten in the same location in another 2 microseconds of one 6-microsecond memory cycle.

MEMORY ADDRESS REGISTER (MA)

The address in core memory which is currently selected for reading or writing

is contained in this 12-bit register. Therefore, all 4096 words of core memory can be addressed directly by this register. The MA can be cleared or incremented by one. Data can be set into it from the memory buffer register, from the switch register, or from an I/O device. The output can be disabled (i.e. forced to indicate all binary zeros) without affecting the contents of the register.

INSTRUCTION REGISTER (IR)

This 4-bit register contains the operation code of the instruction currently being performed by the machine. Information is loaded into the IR from the memory buffer register during a Fetch cycle. The contents of the three most significant bits of the IR are decoded to produce the eight instructions, and affect the cycles and states entered at each step in the program. The least significant bit (the indirect bit) is used in addressing core memory to specify a defer cycle in addressable instructions and to differentiate the two groups operate instructions.

PROGRAM COUNTER (PC)

The program sequence, that is, the order in which instructions are performed, is determined by the PC. This 12-bit core memory register contains the address of the core memory location from which the last instruction was taken. Information enters the PC from the MB, since core memory address 0 is used as the PC. Because the PC is in core memory, it can be manipulated by the program in the same manner as any other core memory location.

CORE MEMORY

The core memory provides storage for instructions to be performed and information to be processed or distributed. This random addressable magnetic core memory holds either 1024 or 4096 12-bit words. Memory location 0 is used as the program counter, location 1 is used to store the contents of the PC following a program interrupt, and location 2 is used to store the first instruction to be executed following a program interrupt. (When a program interrupt occurs, the contents of the PC are stored in location 1; and program control is transferred to location 2 automatically.) Locations 10 through 17 are used for auto-indexing. All other locations can be used to store instructions or data.

STATES, TIMING, CONTROL, AND IOP GENERATOR

This logic component of the computer establishes the basic timing of all computer operations, controls the operation of all previously mentioned registers, and generates the three IOP pulses which are supplied to the device selectors in the input-output control element. It also establishes the cycles or primary control states entered to accomplish each instruction. The control state entered next is determined at the completion of the current one. All states except break are determined by the instruction.

PROGRAM COUNTER (P): This state reads the contents of the program counter from core memory location 0 into the MB, increments the contents of

the MB by 1 (or 2 for a skip instruction), and rewrites the contents of the MB back in location 0. The incremented contents of the PC remain in the MB as the address of the current instruction. During a jump or jump to subroutine instruction, the effective address specified by the jmp or jms is written into location 0 to transfer program control. Completion of a P cycle initiates a Fetch cycle.

FETCH (F): During this state an instruction word is read from the core memory location specified by the contents of the program counter.

EXECUTE 1 (E_1): This state occurs for all instructions requiring an operand from core memory. The contents of the core memory location specified by the least significant bits of the instruction are read into the memory buffer register and the operation specified by bits 0 through 2 of the instruction is performed.

EXECUTE 2 (E_2): When a jump to subroutine instruction is being executed, this state is entered to write the contents of the program counter into core memory location Y.

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

BREAK (B): When this state is established, the sequence of instructions is broken for a data interrupt. The break normally occurs at the completion of the current instruction. If the interrupt occurs during a jump or jump to subroutine instruction, the break begins only after two instructions have been completed (the instruction jumped to is executed). The data interrupt allows information to be transferred directly between core memory and an external device. When this transfer has been completed, the program sequence is resumed from the point of the break.

OPERATOR CONSOLE

Switches and keys on the operator console allow manual program and information insertion or modification. Indicator lamps display the status of the machine and the contents of major registers. Register indicators light to denote the presence of a 1 in a specific bit. While a program is running, the brightness of an indicator is related to the percentage of time that the related bit holds a 1.

Figure 4 shows the operator console and the following tables list the function of switches, keys, and indicators.

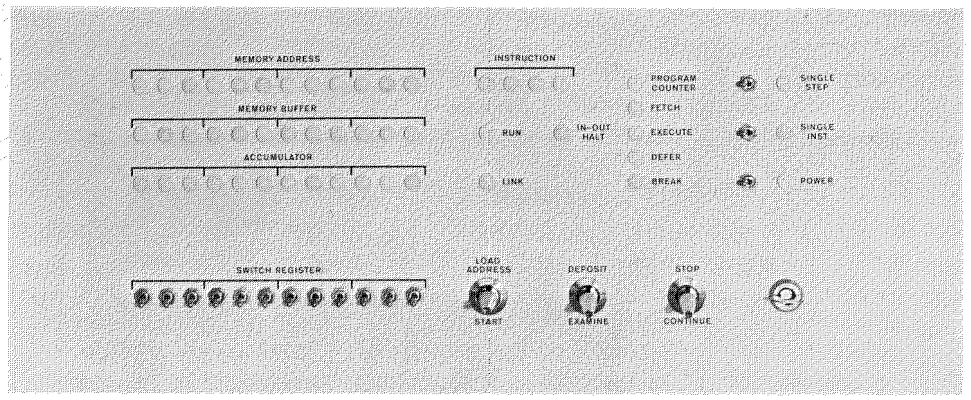


Figure 4 Operator Console

CONSOLE SWITCH FUNCTIONS

Switch	Function
SWITCH REGISTER	Provides a means of manually setting a 12-bit word into the machine. Switches in the up position correspond to ones, down, to zeros. Contents of this register are loaded into the memory address register by the LOAD ADDRESS key, or into the memory buffer register and core memory by the DEPOSIT key. The contents of the switch register (SR) can be set into the accumulator under program control.
SINGLE STEP	Causes the computer to halt at the completion of each memory cycle. Repeated operation of the CONTINUE key steps the program one cycle at a time so that the state of the machine can be examined at each step.
SINGLE INST	Causes the computer to stop at the completion of each instruction.
POWER	Controls primary power in the computer.

CONSOLE KEY FUNCTIONS

Key	Functions
LOAD ADDRESS	Deposits the contents of the switch register into the memory address register.

CONSOLE KEY FUNCTIONS (continued)

Key	Functions
START	Starts the computer after turning off the program interrupt system and clearing both the AC and L. The first instruction is taken from the core memory at the address presently in the memory address register.
DEPOSIT	Sets the word contained in the switch register into the core memory at the location specified by the memory address register. The results remain in the memory buffer register. The memory address register is then incremented by one, allowing rapid data deposits in sequential core memory locations.
EXAMINE	Sets the contents of the core memory location selected by the memory address register into the accumulator and the memory buffer register. The memory address register is then incremented by one, allowing rapid examination of data in sequential core memory locations.
STOP	Causes the computer to stop at the completion of the memory cycle in progress at the time of key operation.
CONTINUE	Causes the computer to resume execution of the instruction at the address held in the PC, from the program state indicated by the panel lamps.
LOCK SWITCH	Disables all console keys and switches except the SR to prevent inadvertent power turn-off or program interference while a program is in progress.

CONSOLE LAMP INDICATIONS

Lamp(s)	Indications
MEMORY ADDRESS	Indicate the contents of the memory address register.
MEMORY BUFFER	Indicate the contents of the memory buffer register.
ACCUMULATOR	Indicate the contents of the accumulator.
INSTRUCTION	Indicate the contents of the instruction register.
RUN	Indicates that the computer is executing instructions.

CONSOLE LAMP INDICATIONS (continued)

Lamp(s)	Indications
IN-OUT HALT	Indicates that the computer is waiting for an input-output device to complete its operation.
LINK	Indicates the contents of the carry link.
PROGRAM COUNTER, FETCH, EXECUTE, DEFER, BREAK	Indicate the primary control state of the machine and that the next memory cycle will be a program counter, fetch, execute, defer, or break cycle respectively.
SINGLE STEP and SINGLE INST	Indicate that the SINGLE STEP or SINGLE INST switch is on the ON position.
POWER	Indicates that power is turned on in the computer.

Instructions

Instruction words are of two types: memory reference and augmented. Memory reference instructions store or retrieve data from core memory, while augmented instructions do not. All instructions utilize bits 0 through 2 to specify the operation code. Operation codes of 0_8 through 5_8 specify memory reference instructions, and codes of 6_8 and 7_8 specify augmented instructions. Instruction execution times are multiples of the 6-microsecond computer cycle time. Memory reference instructions require 12, 18, or 24 microseconds for execution. Indirect addressing increases the execution time of a memory reference instruction by 6 microseconds. The augmented instructions, input-output transfer and operate, are performed in 12 microseconds.

MEMORY REFERENCE INSTRUCTIONS

Word format of memory reference instructions is shown in Figure 5, and the instructions are explained in the Memory Reference Instructions Table.

Since this system can contain a 4096-word memory, 12 bits are required to address all locations. To simplify addressing, the memory is divided into blocks, or pages, of 128 words (200_8 addresses). Pages are numbered 0_8 through 37_8, a 1024-word memory having pages 0_8 through 7_8, and a 4096-word memory using all 32 pages. The seven address bits (bits 5 through 11) of a memory reference instruction can address any location in the page on which the current instruction is located by placing a 1 in bit 4 of the instruction. By placing a 0 in bit 4 of the instruction, any location in page 0 can be addressed directly from any page of core memory. All other core memory locations must be addressed indirectly by placing a 1 in bit 3 and placing a

7-bit effective address in bits 5 through 11 of the instruction to specify the location in the current page or page 0, which contains the full 12-bit absolute address of the operand.

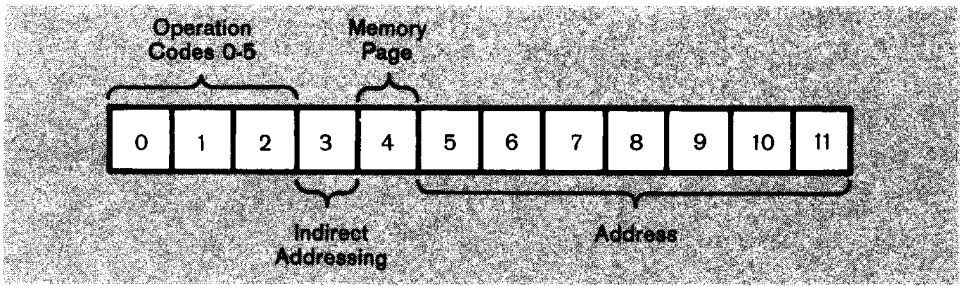


Figure 5 Memory Reference Instruction Bit Assignments

MEMORY REFERENCE INSTRUCTIONS

Mnemonic Symbol	Octal Code	Time (μsec)	Operation
and Y	0	18	Logical AND. The AND operation is performed between the C(Y) and the C(AC). The result is left in the AC, and the original C(AC) are lost. The C(Y) are unchanged. Corresponding bits are compared independently. This instruction, often called extract or mask, can be considered as a bit-by-bit multiplication. $C(Y)_i \wedge C(AC)_i \Rightarrow C(AC)_i$
tad Y	1	18	Twos complement add. The C(Y) are added to the C(AC) in twos complement arithmetic. The result is left in the AC and the original C(AC) are lost. The C(Y) are unchanged. If there is a carry from AC_0, the link is complemented. This feature is useful in multiple precision arithmetic. $C(Y) + C(AC) \Rightarrow C(AC)$.

Example

$C(AC)_i$ original	$C(Y)_i$	$C(AC)_i$ final
0	0	0
0	1	0
1	0	0
1	1	1

MEMORY REFERENCE INSTRUCTIONS (continued)

Mnemonic Symbol	Operation Code	Time (μsec)	Operation
isz Y	2	18	Index and skip if zero. The C(Y) are incremented by one in twos complement arithmetic. If the resultant C(Y) = 0, the next instruction is skipped. If the resultant C(Y) $\neq$ 0, the program proceeds to the next instruction. The C(AC) are unaffected. C(Y) + 1 => C(Y) . if result = 0, C(PC) + 1 => C(PC).
dca Y	3	18	Deposit and clear AC. The C(AC) are deposited in core memory location Y and the AC is then cleared. The previous C(Y) are lost. C(AC) => C(Y), then 0 => C(AC).
jms Y	4	24	Jump to subroutine. The C(PC) contained in core memory location 0 are deposited in core memory location Y. The next instruction is taken from location Y + 1. C(PC) + 1 => C(Y) Y + 1 => C(PC)
jmp Y	5	12	Jump to Y. The C(PC) contained in core memory location 0 are set to address Y. The next instruction is taken from core memory location Y. The original C(PC) are lost. Y => C(PC).

AUGMENTED INSTRUCTIONS

There are two augmented instructions or instructions which do not reference core memory. They are the input-output transfer, which has an operation code of 6, and the operate, which has an operation code of 7. Bits 3 through 11 within these instructions function as an extension of the operation code and can be microprogrammed to perform several operations with one instruction. Augmented instructions are two-cycle (P, F) instructions requiring 12 microseconds for execution. During the second cycle, three clock pulses are available to initiate operations as a function of bit microprogramming. These clock pulses are designated event times 1, 2 and 3 and are separated by 1 microsecond.

INPUT-OUTPUT TRANSFER INSTRUCTION: Microinstructions of the input-output transfer (iot) instruction effect information transfers between the arithmetic and control element and an input-output device via the input-output control element. Specifically, when operation code 6 is detected, the IOP

generator is enabled to produce IOP 1, 2, and 4 pulses as a function of the contents of bits 9 through 11. These pulses are gated in the device selector of the selected I/O device to produce the IOT pulses which enact a transfer.

The format of the iot instruction is shown in Figure 6. Bits 3 through 8 are used to select the I/O device; and bits 9 through 11 enable generation of IOP pulses during event times 3, 2, and 1, respectively. Operations performed by iot microinstructions are explained in Chapter 4.

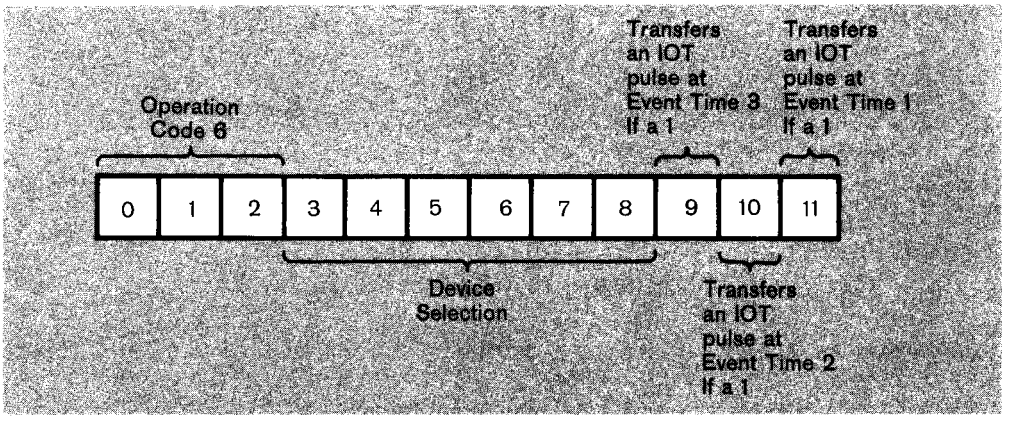


Figure 6 IOT Instruction Bit Assignments

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

Group 1 operate microinstruction format is shown in Figure 7, and the microinstructions are listed in the table below. Any logical combination of bits within this group can be combined into one microinstruction. For example, it is possible to assign ones to bits 5, 6, and 11; but it is not logical to assign ones to bits 8 and 9 simultaneously since they specify conflicting operations. If ral or rar is specified, neither cma or cml may be specified, and conversely. If rtl or rtr is specified, neither cma, cml, or iac may be specified, and conversely.

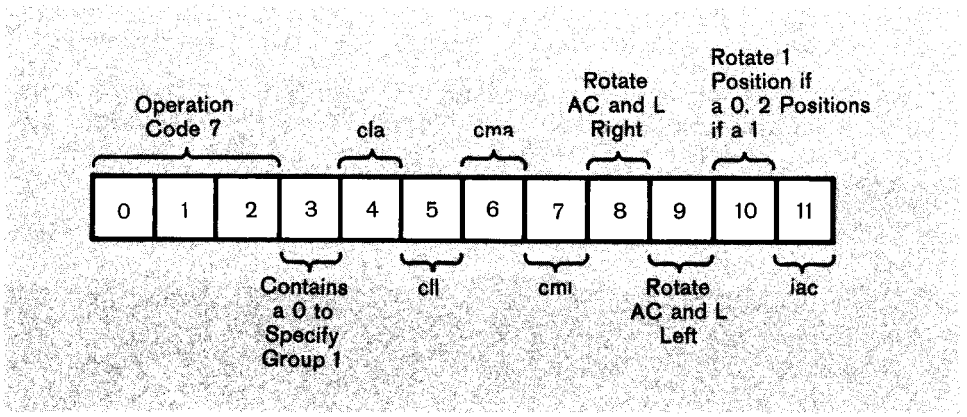


Figure 7 Group 1 Operate Instruction Bit Assignments

GROUP 1 OPERATE MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Event Time	Operation
cla	7200	1	Clear AC. To be used alone or in opr 1 combinations. $0 \Rightarrow C(AC)$.
cll	7100	1	Clear L. $0 \Rightarrow C(L)$.
cma	7040	2	Complement AC. The C(AC) are set to the ones complement of C(AC). $C(\overline{AC}) \Rightarrow C(AC)$.
cml	7020	2	Complement L. $C(\overline{L}) \Rightarrow C(L)$.
rar	7010	2	Rotate AC and L right. The C(AC) and the C(L) are rotated right one place. $C(AC)_i \Rightarrow C(AC)_{i+1}$ $C(AC)_{11} \Rightarrow C(L)$ $C(L) \Rightarrow C(AC)_0$
ral	7004	2	Rotate AC and L left. The C(AC) and the C(L) are rotated left one place. $C(AC)_i \Rightarrow C(AC)_{i-1}$ $C(AC)_0 \Rightarrow C(L)$ $C(L) \Rightarrow C(AC)_{11}$
rtr	7012	2,3	Rotate two places to the right. Equivalent to two successive rar operations.

GROUP 1 OPERATE MICROINSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Event Time	Operation
ral	7004	2	Rotate AC and L left. The C(AC) and the C(L) are rotated left one place. $C(AC)_i \Rightarrow C(AC)_{i-1}$ $C(AC)_0 \Rightarrow C(L)$ $C(L) \Rightarrow C(AC)_{11}$
rtr	7012	2,3	Rotate two places to the right. Equivalent to two successive rar operations.
rtl	7006	2,3	Rotate two places to the left. Equivalent to two successive ral operations.
iac	7001	3	Index AC. The C(AC) are incremented by one in twos complement arithmetic. $C(AC) + 1 \Rightarrow C(AC)$.
nop	7000	—	No operation. Causes a 12μsec program delay.

Group 2 operate microinstruction format is shown in Figure 8 and the microinstructions are listed in the table below. Any logical combination of bits within this group can be composed in one microinstruction.

If skips are combined in a single instruction, the inclusive OR of the conditions determines the skip. For example, if ones are designated in bits 6 and 7 (sza and snl), the next instruction is skipped if either C(AC) = 0, or C(L) = 1, or both. The cla microinstruction from group 1 can be combined with group 2 commands. This command occurs at event time 2 with respect to the event times listed in the following table.

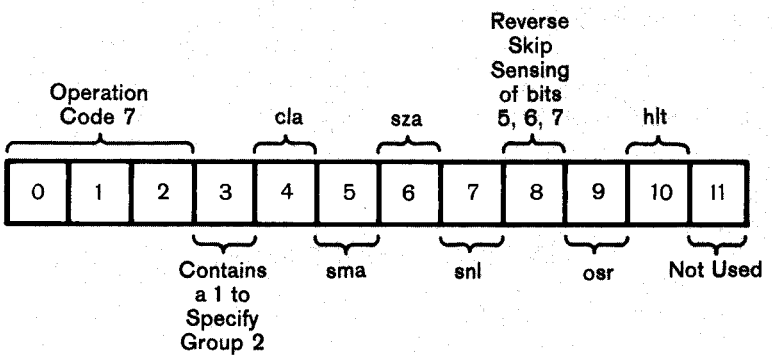


Figure 8 Group 2 Operate Instruction Bit Assignments

GROUP 2 OPERATE MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Event Time	Operation
cla	7600	2	Clear AC. To be used alone or in opr 2 combinations. 0 => C(AC)
spa	7510	1	Skip on positive AC. If the C(AC) is a positive number, the next instruction is skipped. If $C(AC)_0 = 0$, then C(PC) + 1 => C(PC)
sma	7500	1	Skip on minus AC. If the C(AC) is a negative number, the next instruction is skipped. If $C(AC)_0 = 1$, then C(PC) + 1 => C(PC)
sna	7450	1	Skip on non-zero AC. If C(AC) $\neq$ 0, then C(PC) + 1 => C(PC)
sza	7440	1	Skip on zero AC. If C(AC) = 0, the C(PC) +1 => C(PC)
szl	7430	1	Skip on zero L. If C(L) = 0, the next instruction is skipped. If C(L) = 0, the C(PC) + 1 => C(PC)
snl	7420	1	Skip on non-zero L. If C(L) = 1, then C(PC) + 1 => C(PC)
skp	7410	1	Skip, unconditional. The next instruction is skipped. C(PC) + 1 => C(PC)
osr	7404	3	OR with switch register. C(SR) V C(AC) => C(AC)
hlt	7402	3	Halt. Stops the program. If this instruction is combined with others in the opr 2 group, the computer stops immediately after completion of the cycle in process.

Programming

MEMORY ADDRESSING

The following terms are used in memory address programming:

Term	Definition
Page	A block of 128 core memory locations (200_8 addresses).

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

Organization of the core memory is summarized as follows:

Total locations (decimal)	1024	4096
Total addresses (octal)	0-1777	0-7777
Number of pages (decimal)	8	32
Page designations (octal)	0-7	0-37
Number of locations per page (decimal)	128	128
Addresses within a page (octal)	0-177	0-177

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

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

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

Page 0 Location	Contents	Page 1 Location	Contents	Page 2 Location	Contents	Remarks
				R	tad 00 A	Direct to data in page 0
				S	tad 01 B	Direct to data in same page
				T	tad 10 M	Indirect to address specified in page 0
				U	tad 11 N	Indirect to address specified in same page
					.	
					.	
					.	
					.	
A	xxxx	C	xxxx	B	xxxx	
M	C	D	xxxx	N	D	

Routines, using 128 instructions or less, can be written in one page using direct addresses for looping and using indirect addresses for data stored in other pages. 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	Is the program counter.
1_8	Stores the contents of the program counter following a program interrupt.
2_8	Stores the first instruction to be executed following a program interrupt.
10_8 through 17_8	Auto-indexing.

INDIRECT ADDRESSING: When indirect addressing is specified, the address part (bits 5-11) of a memory reference instruction is interpreted as the address

of a location containing not the operand, but the address of the operand. Consider the instruction tad A. Normally, A is interpreted as the address of the location containing the quantity to be added to the AC. Thus, if location 100 contains the number 5432, the instruction tad 100 causes the quantity 5432 to be added to the AC. Now suppose that location 5432 contains the number 6543. The instruction tad i 100 (where i signifies indirect addressing) causes the computer to take the number 5432, which is in location 100, as the effective address of the instruction and the number in location 5432 as the operand. Hence, this instruction results in the quantity 6543 being added to the contents of the AC.

AUTO-INDEXING: When a location between 10_8 and 17_8 in page 0 is specified as the address in an instruction, and bit 3 is a 1, the contents of that location are read, incremented by one rewritten in the same location, and then taken as the effective address of the instruction. This feature is called auto-indexing. If location 12_8 contains the number 5432 and the instruction dca i 12 is given, the contents of the accumulator are deposited in core memory location 5433, and the number 5433 is stored in location 12.

STORING AND LOADING

Data is stored in any core memory location by use of the dca Y instruction. This instruction clears the AC to simplify loading of the next datum. If the data deposited is required in the AC for the next program operation, the dca must be followed by a tad Y for the same address.

All loading of core memory information into the AC is accomplished by means of the tad Y instructions, preceded by an instruction that clears the AC such as cla or dca.

PROGRAM CONTROL

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

The jms Y is used to enter a subroutine which starts at location Y+1. The C(PC) + 1 => C(Y) and Y + 1 => C(PC). To exit a subroutine the last instruction is a jmp i Y, which returns a program control to C(Y).

Since the program counter is in core memory location 0, the program flow can be altered by depositing some number in location 0. If the number X is deposited in 0, the next instruction is taken from location X+1.

INDEXING OPERATIONS

The isz instruction is used to count repetitive program operations without disturbing the contents of the accumulator. Counting is performed by storing a twos complement negative number equal to the number of program loops to be counted. Each time the operation is performed, the isz instruction is used to

increment the contents of this stored number and check the result. When the stored number becomes zero, C(Y) = 0, the specified number of operations have occurred and the program skips out of the loop and back to the main sequence.

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

LOGIC OPERATIONS

The PDP-5 instruction list includes the logic instruction, and Y. From this instruction short routines can be written to perform the inclusive and exclusive OR operations.

LOGIC AND: The logic AND operation between the contents of the Accumulator and the contents of a core memory location Y is performed directly by means of the and Y instruction.

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

```
/calling sequence                                   jms ior
/                                                   (address of B)
/                                                   (return)
/enter with argument in AC; exit with logical result in AC
        ior,    0
                dca tem 1
                tad i ior
                dca tem2
                tad i tem2
                cma
                and tem1
                tad i tem2
                isz ior
                jmp i ior
        tem1,   0
        tem2,   0
```

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

```
/calling sequence                                   jms xor
/                                                   (address of B)
/                                                   (return)
/enter with argument in AC; exit with logical result in AC

        xor,    0
                dca tem1
                tad i xor
                dca tem2
                tad tem1
                and i tem2
                cma V iac
                cll V ral
                tad tem1
                tad i tem2
                isz xor
                jmp i xor
        tem1,   0
        tem2,   0
```

An xor subroutine can be written using fewer core memory locations by making use of the ior subroutine; however, such a subroutine requires longer to execute. A faster xor subroutine can be written by storing the value B in the second instruction of the calling sequence instead of the address of B; however, the resulting subroutine is not as utilitarian as the routine given here.

ARITHMETIC OPERATIONS

One arithmetic instruction is included in the PDP-5 order code, the twos complement add: tad Y. Using this instruction, routines can easily be written to perform addition, subtraction, multiplication, and division in twos complement arithmetic.

TWOS COMPLEMENT ARITHMETIC: In twos complement arithmetic, addition, subtraction, multiplication, and division of binary numbers is performed in accordance with the common rules of binary arithmetic. In PDP-5, 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 ones complement arithmetic is slightly different from that in twos complement arithmetic.

The ones 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 twos complement of a number is equal to the ones complement of the positive value plus one.

In ones complement arithmetic a carry from the sign bit (most significant bit) is added to the least significant bit in an end-around carry. In twos complement arithmetic a carry from the sign bit complements the link (a carry would set

the link to one if it were properly cleared before the operation), and there is no end-around carry.

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

Number	1s Complement	2s 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 twos complement there is only one representation for the number which has the value zero, while in ones complement there are two representations. Note also that complementation does not interfere with sign notation in either ones complement or twos complement arithmetic; bit 0 remains a 0 for positive numbers and a 1 for negative numbers.

To form the twos complement of any number in the PDP-5, the ones complement is formed, and the result is incremented by one. This is accomplished by the instruction cma combined with an iac instruction. Since both of these instructions are functions of the opr 1 instruction and the actions occur at different event times, they can be combined to form:

cia — 7041 — 2, 3 Complement and index AC

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 tad Y instruction, assuming that the binary point is in the same position and that both numbers are properly represented in twos 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 11 least significant bits will be correct.

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

```
cla
tad B          /Load subtrahend into AC
cia            /Complement and increment B (cma V iac)
tad A          /C(AC) = A - B
```

CHAPTER 3

INPUT-OUTPUT CONTROL

Functions

Selected input-output devices are controlled by iot (in-out transfer) instructions. The iot instruction is microprogrammed to allow one basic instruction to handle many devices (by changing the bits of the command). The command pulses occur at various times to allow flags to be sampled (and an instruction skipped), buffers to be cleared, and data to be transmitted to or from the accumulator. Operational circuits of the input-output control element are shown in Figure 9.

DEVICE SELECTOR (DS)

Input-output equipment connected into the system is controlled by various Device Selector pulses. These pulses:

a. Sample Device flag conditions which are fed into the input-output skip facility.
b. Reset external register.
c. write information into external registers from the AC output.
d. Read information from external register into the AC.
e. Control the I/O device.
f. Halt the computer until the external device has finished its operation.

The iot instruction causes the arithmetic and control element to produce IOP pulses based on the contents of bits 9 through 11 of the instruction. These pulses are designated IOP 4, 2, and 1, respectively, and occur at 1-microsecond intervals, which are identified as event times. Binary ones in the instruction word cause the IOP pulses to be generated as follows:

Instruction Bit	IOP Pulse	IOT Pulse	Event Time	Computer Cycle Time
11	IOP 1	IOT 1	1	T4
10	IOP 2	IOT 2	2	T5
9	IOP 4	IOT 4	3	T6

A device selector module exists for each I/O device or external register to be addressed separately. The DS is a gating element which receives both the 1 and 0 information from bits 3 through 8 of an instruction (MB_{3-8}) and the IOP

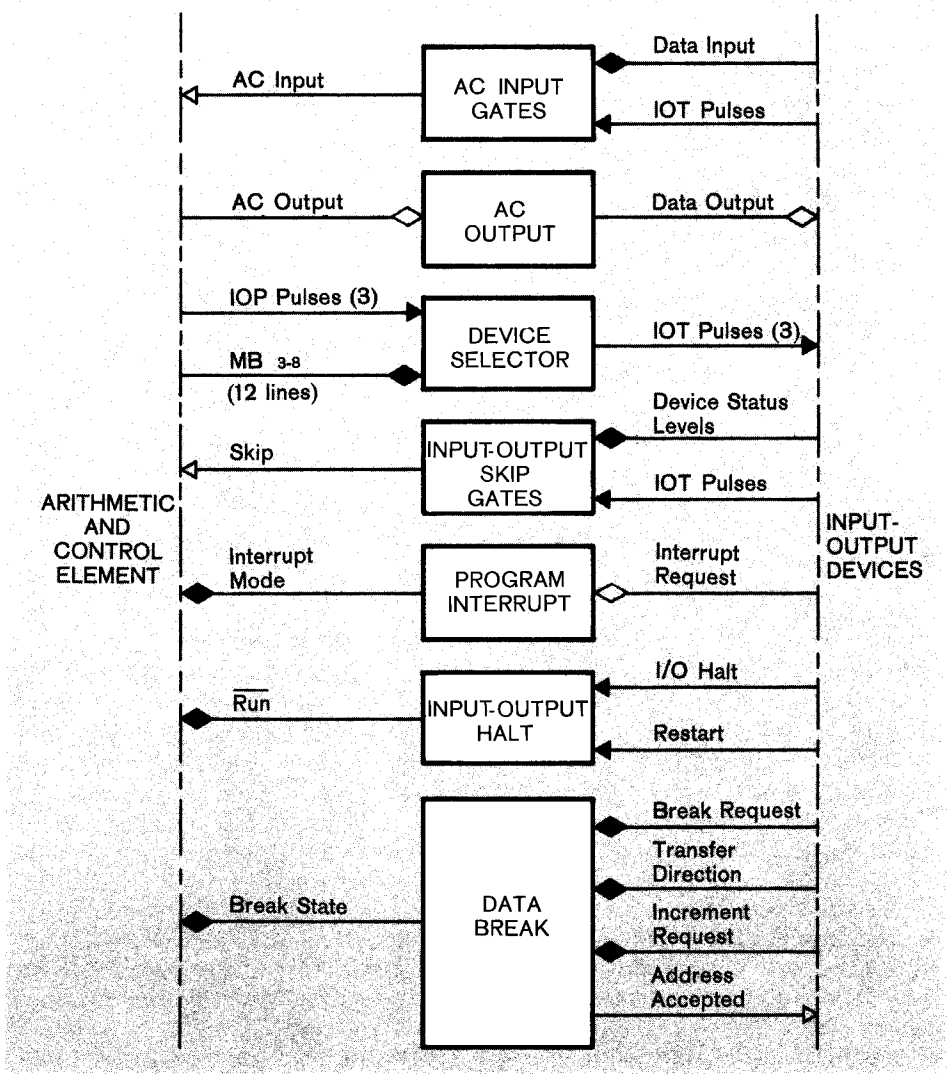


Figure 9 Input-Output Control Element

pulses. Each DS is wired to pass IOP pulses to a specific I/O device only when the I/O device selection bits are set to the code which specifies operation of the associated I/O device. The gated IOP pulses at the output of a DS are called IOT pulses and can be used to set or reset control flip-flops, gate information into the AC from external registers, gate information into external registers from the AC, or skip instructions.

ACCUMULATOR INPUT

Capacitor-diode gates are provided at the inputs to the accumulator to allow gated information to be written into the PDP-5 from several sources. Information levels from 12 separate bits of an external register can be simultaneously set into the AC by an IOT pulse. The AC must be clear at the time information is written in. Information pulses supplied to the AC input bus must drive it to ground potential to write a 1 in an accumulator bit.

ACCUMULATOR OUTPUT

A static level is available at an output bus from each bit of the accumulator. These static levels are ground potential for a binary 1 and −3 volts for a binary 0. Data supplied to an external register is strobed into it by means of IOT pulses.

INPUT-OUTPUT SKIP (IOS)

The IOS facility allows the program to skip (or branch) according to the condition of various external devices. An IOT pulse is used to strobe the external device, such as a flag, and sample its state. If the gating of the Device flag and IOT pulse drives the IOS bus to ground, the instruction following the iot instruction which issued the strobe is skipped. If the input is a −3 volt potential, the program sequence is not altered.

PROGRAM INTERRUPT

The program interrupt feature allows certain external conditions to interrupt the computer program. It is used to speed the information processing of input-output devices or to allow certain alarms to halt the program in progress and initiate another routine. When a program interrupt request is made, the computer completes execution of the instruction in progress before entering the interrupt mode. A program interrupt is similar to a jms to location 1; that is, the contents of the program counter are stored in location 1, and the program resumes operation in location 2. The interrupt program commencing in location 2 is responsible for finding the signal causing the interruption, for removing the condition, and for returning to the original program. Exit from the interrupt program, back to the original program, can be accomplished by a jmp i 1 instruction.

INPUT-OUTPUT HALT (IOH)

The input-output halt facility allows the computer to be halted during the time that external devices are operating and then restarted by a pulse from the device. The IOH state occurs when Type 137 Analog-to-Digital Converter is operating and may be wired to occur during the operation time of any other device.

A specific iot instruction initiates operation of an I/O device. The I/O device supplies an I/O-Halt pulse to the IOH that inhibits program advance. When the I/O device completes the programmed operation, it produces a Restart pulse

which is received by the IOH to clear the IOH mode and to allow program advance to the next instruction.

DATA BREAK

This facility allows transmission of data directly between an external device and core memory, via the memory buffer register. During a data break, the program is halted but the contents of the accumulator, instruction register, and program counter are not disturbed. Therefore, when a data transfer is complete, the program resumes from exactly the same condition which existed before the break.

Data breaks require receipt of three control signals: Break Request, Transfer Direction, and Increment Request. When a Break Request signal is received from an I/O device, the computer completes execution of the instruction in progress and then enters the data break mode. If a jmp or jms instruction is in progress when the request is received, the current instruction is completed, and the next instruction is performed before the break is instituted. The direction of transfer and the core memory address of each transferred word are specified by the I/O device when the break request is made. The Transfer Direction signal controls the read or write cycle of the computer, and the address is set directly into the memory address register. Data transfer then takes place between the memory buffer register and the I/O device. When the transfer is completed, the I/O device signals the computer to leave the break mode by removing the Break Request. If additional transfers are to occur, a new address must be specified to the memory address register or an Increment Request signal must be supplied to transfer data at sequential core memory location. Figure 10 indicates the timing of these signals. The levels of these signals are:

Signal	−3 Volts	0 Volts
Break Request	No request	Request Break
Transfer Direction	Data into PDP-5	Data out of PDP-5
Increment Request	Request increment	No request
Address	0	1
Data	0	1

Break Request, Transfer Direction, and Address Information signals should be supplied simultaneously for the first transfer. When the computer enters the data break mode, it supplies an Address Accepted pulse to the I/O device. When the direction of transfer is into the PDP-5 from the device, data must be supplied to the memory buffer register input no later than 1 microsecond after the Address Accepted pulse occurs and must be present for more than 2 microseconds. To discontinue the data break mode, the Break Request signal must be removed no later than 4 microseconds after the address accepted pulse occurs, or the computer will enter another cycle in the data break mode. The Transfer Direction signal must be present when the break request is made and cannot be changed until 4 microseconds after the Address Accepted pulse

occurs. Address information must also be present when the request is made, but can be changed any time after the address is accepted. To transfer data at sequential core memory locations the first transfer address must be supplied to the memory address register by the I/O device, and successive addresses can be specified by the Increment Request signal. This signal cannot occur before 1 microsecond after the address is accepted for the first transfer and must be present no later than 4 microseconds after the address is accepted. The maximum and minimum limits of this signal timing are indicated in Figure 10.

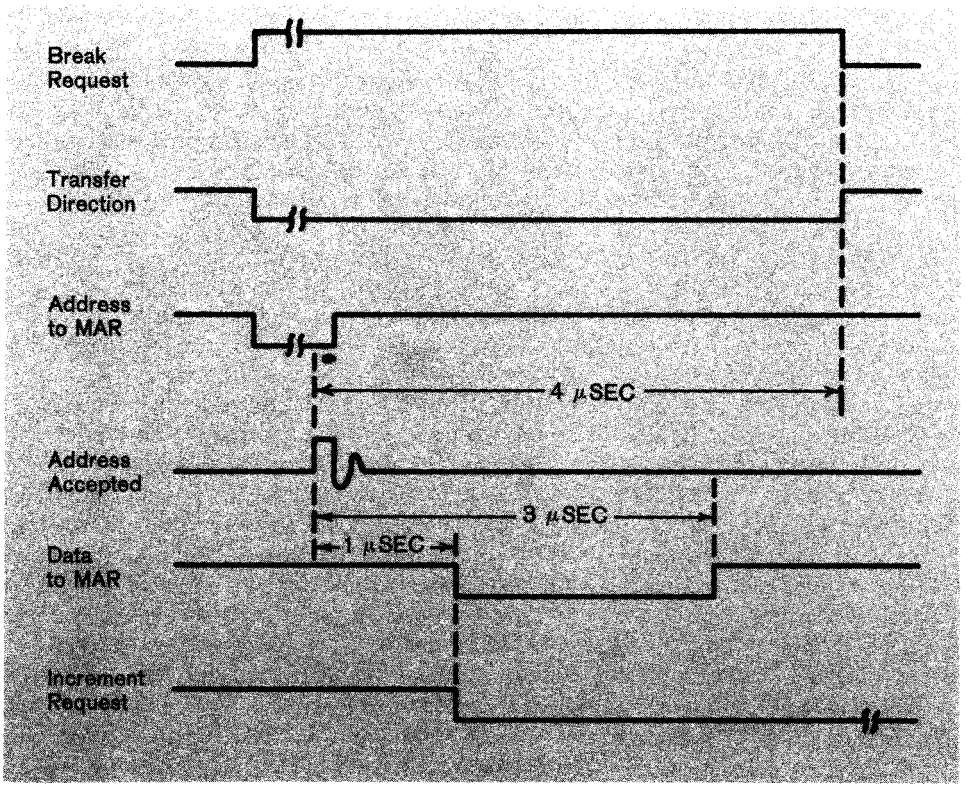


Figure 10 Data Break Timing

Instructions

Two types of instructions are associated directly with the input-output controls: those concerning the input-output skip and those concerning the program interrupt. The skip instructions are listed in Chapter 4 of this handbook with the instructions for the device whose status is checked. There are two instructions for the program interrupt. They are:

ion — 6001 — Turn interrupt on and enable the computer to respond to an interrupt request. When this instruction is given, the computer executes the next instruction, then enables the interrupt. The additional instruction allows exit from the interrupt subroutine before allowing another interrupt to occur.

iof — 6002 — Turn interrupt off i.e. disable the interrupt.

Programming

When an interrupt request is acknowledged, the interrupt is automatically disabled by the program interrupt circuits (not by instructions). The next instruction is taken from core memory location 2. Usually the instruction stored in location 2 is a jmp, which transfers program control to a subroutine, which services the interrupt. At some time during this subroutine an ion instruction must be given. The ion can be given at the end of the subroutine to allow other interrupts to be serviced after program control is transferred back to the original program. In this application, the ion instruction immediately precedes the last instruction in the routine. A delay of one instruction (regardless of length) is inherent in the ion instruction to allow transfer of program control back to the original program before enabling the interrupt. Usually exit from the subroutine is accomplished by a jmp i 1 instruction.

The ion can be given during the subroutine as soon as it has determined the I/O device causing the interrupt. This latter method allows the subroutine, which is handling a low priority interrupt, to be interrupted, possibly by a high priority device. Programming of an interrupt subroutine, which checks for priority and allows itself to be interrupted, must make provisions to relocate the contents of the program counter stored in location 1; so that if interrupted, the contents of the PC during the subroutine are stored in location 1, and the contents of the PC during the original program are not lost.

CHAPTER 4

INPUT-OUTPUT DEVICES

Use of the standard and optional equipment in a PDP-5 system is discussed in this section. The Teletype equipment is the only standard I/O device supplied with PDP-5. All other equipment is purchased at the option of the customer to compose a system tailored to his computing needs.

Teletype Model 33 ASR

The standard Teletype Model 33 ASR (automatic send-receive) can be used to type in or print out information at a rate of up to ten characters per second, or to read in or punch out perforated paper tape at a ten characters per second rate. Signals transferred between the 33 ASR 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 the stop mark which is two units. Appendix 2 lists the character code for the Teletype. Punched tape format is as follows:

	Tape Channel			
	87	654	S	321
Binary Code (Punch = 1)	10	110		100
Octal Code	2	6		4

Toggle switches on the right side of the Teletype console control primary power and allow the 33 ASR to communicate with the computer in on line operation or to prepare tapes, etc., without disturbing the computer program in local operations. Separate pushbutton switches on the punch are used to control power, release the mechanism to allow insertion and removal of tape, and allow backspacing to correct errors. A three position switch on the reader advances the tape, stops the tape, or allows free wheeling of the mechanism for tape insertion and removal.

KEYBOARD/READER

The keyboard and tape reader control contains an 8-bit buffer (LUI) which assembles and holds the code for the last character struck on the keyboard

or read from the tape. The Keyboard flag becomes a 1 to signify that a character has been assembled and is ready for transfer to the accumulator. When the flag is a 1, a relay contact opens to disable the reader. This flag is connected to the computer program interrupt and input-output skip facility. It is cleared by command. Instructions for use in supplying data to the computer from the Teletype are:

ksr — 6031 — Skip if Keyboard flag is a 1.
kcc — 6032 — Clear AC and clear Keyboard flag.
krs — 6034 — Read keyboard buffer static. (This is a static command in that neither the AC nor the Keyboard flag is cleared.)
$C(LUI)\ V\ C(AC)_{4-11} \Rightarrow C(AC)_{4-11}$
krb — 6036 — Clear AC, clear Keyboard flag, and read the contents of the keyboard buffer into $C(AC)_{4-11}$.

A program sequence loop to look for a Teletype keyboard or tape reader character can be written as follows:

```
200     6031     look,     ksf          /skip when LUI is full
201     5200               jmp look
202     6036               krb          /read LUI into AC
```

TELEPRINTER/PUNCH

The teleprinter and tape punch control contains an 8-bit buffer (LUO) which receives a character to be printed and/or punched from AC bits 4 through 11. The LUO receives the 8-bit code from the AC in parallel and transmits it to the teleprinter and tape punch 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 instruction list for printing or punching is:

tsf — 6041 — Skip if Teleprinter flag is a 1.
tcf — 6042 — Clear Teleprinter flag.
tpc — 6044 — Load the LUO from the $C(AC)_{4-11}$ and print and/or punch the character.
tls — 6046 — Load the LUO from the $C(AC)_{4-11}$, clear the Teleprinter flag, and print and/or punch the character.

A program sequence loop to print and/or punch a character when the LUO is free can be written as follows:

```
free,     tsf          /skip when free
          jmp free
          tls          /load LUO, print or punch
```

High Speed Perforated Tape Reader and Control Type 750

This device senses 8-hole perforated paper or Mylar tape photoelectrically at 300 characters per second. The reader control requests reader movement, transfers data from the reader into the reader buffer (RB), and signals the

computer when incoming data is present. Reader tape movement is started by a reader control request to simultaneously release the brake and engage the clutch. The 8-bit reader buffer sets the Reader flag to 1 when it has been filled from the reader and transfers data into bits 4 through 11 of the accumulator under computer command. The Reader flag is connected to the computer program interrupt and input-output skip facility. It is cleared by IOT pulses. Computer instructions for the reader are:

rsf — 6011 — Skip if Reader flag is a 1.
rrb — 6012 — Read the contents of the reader buffer and clear the Reader flag. (This instruction does not clear the AC.)
C(RB) V $C(AC)_{4-11}$ => $C(AC)_{4-11}$
rfc — 6014 — Clear Reader flag and reader buffer, fetch one character from tape and load it into the reader buffer, and set the Reader flag when done.

A program sequence loop to look for a reader character can be written as follows:

```
        rfc         /fetch character from tape
look,   rsf         /skip when RB full
        jmp look
        cla
        rrb         /load AC from RB
```

High Speed Perforated Tape Punch and Control Type 75A

The Teletype BRPE paper tape punch perforates 8-hole tape at 63.3 characters per second. Information to be punched on a line of tape is loaded on an 8-bit punch buffer (PB) from the AC bits 4 through 11. The Punch flag becomes a 1 at the completion of punching action, signaling that new information may be read into punch buffer (PB) (and punching initiated). The Punch flag is connected to the computer program interrupt and input-output skip facility. The punch instructions are:

psf — 6021 — Skip if Punch flag is a 1.
pcf — 6022 — Clear Punch flag and punch buffer.
ppc — 6024 — Load the punch buffer from bits 4 through 11 of the AC and punch the character. (This instruction does not clear the Punch flag or punch buffer.)
$C(AC)_{4-11}$ V C(PB) => C(PB)
pls — 6026 — Clear the Punch flag, clear the punch buffer, load the punch buffer from the contents of bits 4 through 11 of the accumulator, punch the character, and set the Punch flag to 1 when done.

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

```
free,   psf         /skip when free
        jmp free
        pls         /load PB from AC and punch character
```

Analog-To-Digital Converter Type 137

This converter operates in the conventional successive approximation manner, using the memory buffer register as a shift register and using the accumulator as the buffer register. An IOT pulse from the device selector starts the conversion and initiates an input-output halt. At the end of the conversion the converter produces a Restart pulse which is supplied to the input-output halt facility. At this time the digital equivalent of the Analog Input signal is contained in the accumulator as a 12-bit binary number. Insignificant magnitude bits can be rotated out of the AC by an instruction such as 7110 (rar and cll).

To save program running time, the converter should be adjusted to provide only the accuracy required by the program application. Instructions for adjusting the accuracy are given in the maintenance manual covering Type 137. Maximum error of the converter is equal to the switching point error plus the quantization error. Maximum quantization error is equal to the least significant bit. Switching point error and total conversion time are functions of the adjusted accuracy of the converter.

Adjusted Bit Accuracy	Switching Point Error	Conversion Time per Bit (in μsec)	Total Conversion Time (in μsec)
6	$\pm$1.6%	3.5	24.5
7	$\pm$0.8%	4.0	32.0
8	$\pm$0.4%	4.5	40.5
9	$\pm$0.2%	5.0	50.0
10	$\pm$0.1%	6.0	66.0
11	$\pm$0.05%	11.0	132.0

There is only one instruction associated with the converter:

adc — 6004 — Convert the Analog Input signal to a digital value.

Card Reader and Control Type 451

The control of the card reader differs from the control of other input devices, in that the timing of the read-in sequence is dictated by the device. Once the command to fetch a card is given, the card reader reads all 80 columns of information in sequence. To read a column, the program must respond to a flag set as each new column is started. The instruction to read the column must come within 2.3 milliseconds. The commands for the card reader are:

crsf — 6632 — Skip if Card Reader flag is a 1. If a card column is present for reading, the next instruction is skipped.

cers — 6634 — Card equipment read status. Reads the status of the Card Reader flag and status levels into bits 6 through 9 of the AC.

crrb — 6671 — Read the card column buffer information into the AC and clear the Card Reader flag. One crrb reads alphanumeric information. Two crrb instructions read the upper and lower column binary information.

crsa — 6672 — Select a card in alphanumeric mode. Select the card reader and start a card moving. Information appears in alphanumeric form.

crsb — 6674 — Select a card in binary mode. Select the card reader and start a card moving. Information appears in binary form.

Upon instruction to read the card reader buffer, 6 information bits are placed into AC bits 6 through 11. Alphanumeric (or Hollerith) information on the card is encoded or represented with these six bits. The binary mode enables the 12 bits (or rows) of each column to be obtained. The first read buffer instruction transfers the upper six rows (Y, X, 0, 1, 2, and 3); the second instruction transfers the lower six rows (4, 5, 6, 7, 8, and 9). The mode is specified with the card read select instruction. The mode can be changed while the card is being read.

Card Punch Control Type 450

The card punch dictates the timing of a read-out sequence, much as the card reader controls the read-in timing. Once a card leaves the hopper, all 12 rows are punched at intervals of 40 milliseconds. Punching time for each row is 24 milliseconds, leaving 16 milliseconds to load the buffer for the next row. A flag indicates that the buffer is ready to be loaded. The commands for the card punch control are:

cpsf — 6631 — Skip if Card Punch flag is a 1. The Card Punch flag indicates the punch buffer is available, and should be loaded.

cers — 6634 — Card equipment read status. Reads the status of the Card Punch flag and the Card Punch error level into the contents of bits 10 and 11 of the AC, respectively.

cpcf — 6641 — Clear Card Punch flag.

cpse — 6642 — Select the card punch. Transmit a card to the 80-column punch die from the hopper.

cplb — 6644 — Load the card punch buffer from the C(AC). Seven load instructions must be given to fill the buffer.

Since 12 bits are transmitted with each iot instruction, 7 iot instructions must be issued to load the 80-bit row buffer. The first six loading instructions fill the first 72 bits (or columns); the seventh loads the remaining 8 bits of the buffer from AC bits 4 through 11.

After the last row of punching is complete, 28 milliseconds are available to select the next card for continuous punching. If the next card is not requested in this interval, the card punch will stop. The maximum rate of the punch is 100 cards per minute in continuous operation. A delay of 1308 milliseconds follows the command to select the first card; a delay of 108 milliseconds separates the punching of cards in continuous operation.

The Card Punch flag is connected to the program interrupt and to bit 10 of the cers instruction. Faults occurring in the punch are detected by status bit 11 of the cers and signify the punch is disabled, the stacker is full, or the hopper is empty.

A program sequence to punch 12 rows of data on a card can be written as follows, assuming the data to be punched in each row is stored in seven consecutive core memory locations beginning in location 100.The program begins in register pnch.

```
pnch,   cpse        /select the card
        cla
        tad loc     /initialize the card image
        dca 10
        tad rcnt
        dca tem1    /initialize the row counts, 12
lp1,    cla
        tad gpct    /initialize the 7 groups per row
        dca tem2
        cpsf        /sense punch load availability
        jmp .-1
lp2,    cla
        tad i 10    /7 groups of 12 bits per row
        cplr        /load buffer command
        isz tem2
        jmp lp2
        isz tem1    /test for 12 rows
        jmp lp1
        hlt         /end punching 1 card
loc,    77          /location of card image
rcnt,   -14         /12 rows per card
gpct,   -7          /7 groups per row
tem1,   0           /row counter
tem2,   0           /group counter
```

Automatic Line Printer And Control Type 64

The line printer can print 300 lines of 120 characters per minute. Each character is selected, from a set of 63 available, by means of a 6-bit binary code (Appendix 2 lists the character specified by each code). Each 6-bit code is loaded separately into a printing buffer from bits 6 through 11 of the AC. The printing buffer is divided into two sections; each section can hold 120 codes. Therefore, 120 load instructions can be given to fill one section of the printing buffer. A print command causes the characters specified by the contents of the print buffer section last loaded to be printed on one line. As printing is in progress, the alternative section of the printing buffer can be reloaded. After the last character in a line is printed, the section of the printing buffer from which characters were just printed is cleared automatically. The section of the printing buffer that is loaded and printed is alternated automatically within the printer and is not program specified.

A 3-bit format register within the printer is loaded from bits 9 through 11 of the AC during a print command. This register selects one of eight channels of a perforated tape to control spacing of the paper. The tape moves in synchronism with the paper until a hole is sensed in the selected channel to halt paper advance. A recommended tape has the following characteristics:

Channel	Spacing
0	1 line
1	2 lines
2	3 lines
3	¼ page
4	½ page
5	¾ page
6 or 7	top of form

Loading of a 6-bit code into the printing buffer requires approximately 1.6 milliseconds. When the transfer of a code is completed, the Line Printed flag rises to indicate that the printer is ready to receive another code. The Line Printer flag is connected to the program interrupt facility.

The iot microinstructions which command the line printer are:

lcf — 6652 — Clear Line Printer flag.

lpr — 6655 — Clear the format register, load the format register from the $C(AC)_{9-11}$, print the line contained in the section of the printer buffer loaded last, and advance the paper in accordance with the selected channel of the format tape if the $C(AC)_8 = 1$. If the $C(AC)^8 = 0$, the line is printed and paper advance is inhibited.

lsf — 6661 — Skip if Line Printer flag is a 1.

lcb — 6662 — Clear both sections of the printing buffer.

lld — 6664 — Load printing buffer from the $C(AC)_{6-11}$.

The following routine demonstrates the use of these commands in a sequence which prints an unspecified number of 120-character lines. This sequence assumes that the printer is not in operation, that the paper is manually positioned for the first line of print, and that one-character words are stored in sequential core memory locations beginning at 2000. The "print" location starts the routine.

```
print,    lcb          /initialize printing buffer
          cla
          tad loc      /load initial character address
          dca 10       /store in auto-index register
lrpt,     tad cnt      /initialize character counter
          dca temp
loop,     lsf          /wait until printing buffer ready
          jmp loop
          tad i 10     /load AC from current character address
          lld          /load printing buffer
          cla
          isz temp     /test for 120 characters loaded
          jmp loop
          tad frm      /load spacing control and
          lpr          /print a line
          cla          /ready for next line
          jmp lrpt     /jump to print another line
loc,      1777         /initial character address -1
cnt,      -170         /character counter
```

```
temp,    0        /current character address
frm,     10       /spacing control and format
```

Oscilloscope Display Type 34B

Type 34B 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 2 through 11 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 25-kilocycle rate. The instructions for this display are:

dcx — 6051 — Clear X coordinate buffer.
dxl — 6053 — Clear and load X coordinate buffer.
$C(AC)_{2-11} => C(XB)$
dcy — 6061 — Clear Y coordinate buffer.
dyl — 6063 — Clear and load Y coordinate buffer.
$C(AC)_{2-11} => C(YB)$
dix — 6054 — Intensify the point defined by the contents of the X and Y coordinate buffers.
diy — 6064 — Intensify the point defined by the contents of the X and Y coordinate buffers.
dxs — 6057 — Executes the combined functions of dxl followed by dix.
dys — 6067 — Executes the combined functions of dyl followed by diy.

The following program sequence to display a point begins at location 200, and assumes that the X and Y coordinate data is stored in absolute addresses 176 and 177.

```
176              X,
177              Y,
200     7200     beg,    cla
201     1176             tad X     /load AC with X
202     6053             dxl       /clear and load XB
203     7200             cla
204     1177             tad Y     /load AC with Y
205     6067             dys       /clear and load YB, and display point
```

Precision CRT Display Type 30N

Type 30N functions are similar to those of the Type 34B Oscilloscope Display in plotting points on a self-contained 16-inch cathode-ray tube. A 3-bit brightness register is contained in Type 30N to control the amplitude of the Intensify signal supplied to the CRT. This register is loaded by jam transfer (transfer ones and zeros so that clearing is not required) from the AC by the instruction:

dlb — 6074 — Load brightness register (BR) from bits 9 through 11 of the AC.
$C(AC)_{9-11} => C(BR)$

All other instructions and the instruction sequence are similar to those used in the Type 34B.

Light Pen Type 370

The light pen is a photosensitive device which detects the presence of information displayed on a CRT. If the light pen is held in front of the CRT at a point displayed, the Display flag will be set to a 1. The commands are:

dsf — 6071 — Skip if Display flag is a 1.
dcf — 6072 — Clear the Display flag to a 0.

The Display flag is connected to the input-output skip facility, and to the program interrupt.

Incremental Plotter and Control Type 350

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

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 drum and 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 input-output skip facility.

Instructions for the recorder and control are:

plsf — 6501 — Skip if Plotter flag is a 1.
plcf — 6502 — Clear Plotter flag.
plpu — 6504 — Plotter pen up. Raise pen off of paper.
plpr — 6511 — Plotter pen right.
pldu — 6512 — Plotter drum (paper) upward.

pldd — 6514 — Plotter drum (paper) downward.
plpl — 6521 — Plotter pen left.
pldu — 6522 — Plotter drum (paper) upward. (Same as 6512.)
plpd — 6524 — 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 PDP-5 can track the location of the pen on the paper by counting the instructions that increment the pen and the drum.

Automatic Magnetic Tape Control Type 57A

This control, operating through interface logic, such as Type 520, 521, or 522, transfers information between PDP-5 and up to eight tape transports. Data transmission format is compatible with IBM high and low densities (800-556 and 200 characters per inch, respectively) in either BCD or binary parity modes. Transports can be DEC Type 50 or Type 570, or IBM Types 729 II, IV, V, VI, or (with certain restrictions) the 7330. The transports are capable of operating at the following densities: 200 cpi only, Type 50; 200 and 556 cpi only, Type 570 or IBM Types 729 II and 7330; all three densities, IBM Type 729 V.

The following functions are controlled by various combinations of iot instructions:

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

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 as a function of synchronizing flags if status conditions appear.

All data transfers are under control of the PDP-5 data break facility; and commands issued during a transfer control, operate, and monitor Type 57A functions by means of the PDP-5 program interrupt facility. Assembled, 12-bit, PDP-5, data words pass between the computer MBR and the control final data buffer register. The core memory address of each word transferred is specified to the computer MAR by the control current address register. Use of the program interrupt facility allows the main computer program to continue during long tape operations without running in a loop which waits for Tape flags. The

program interrupt subroutine for Type 57A loads the AC with numbers, then issues iot instructions to the control. Specific tape control modes are interpreted from the contents of the AC during some iot instructions. In addition, the current address (CA) register and the word count (WC) registers of the control are loaded from the AC.

Tape functions can be monitored by the program either during or at the end of an operation. They can be altered during operation to a limited degree. The control senses for several types of possible error condition throughout an operation.The results of this sensing can be interrogated by the subroutine at any time.

Two crystal clocks are used to generate one of three character writing rates, depending on the density (200, 556, 800) specified by the programmer. In writing or reading, a composite 12-bit binary word passes between the computer and the control; that is, bits 0 through 5 constitute one tape character, and bits 6 through 11 constitute a second tape character.

In normal operation, six iot commands initiate reading or writing of one record. When the word count exceeds the number stored in the WC, the transport is stopped and the control is free for another command. In continuous operation, any number of records is written or read without the need for further transport commands except stop.

The following automatic safeguards are inherent in the design of Type 57A:

END POINT: If the end point is reached during reading or writing, the control ignores the end point and finishes the operation (ample tape is allowed).
Beyond the end point, tape commands specifying forward direction are illegal, and the tape will not respond to such commands. If the end point is passed during spacing, the transport is shut down regardless of word count.

LOAD POINT: If the load point is reached during back spacing, the transport is stopped regardless of word count. At load point, a space back command is legal, and the tape may be unloaded. When the write command is given at load point, the tape is erased 3 inches beyond the load point before writing the first record. After giving a read command at the load point, the read logic is disabled until the load point marker is past the read head before the read logic is turned on.

WRITE LOCK RING: Without the write lock ring in the tape reel, writing is illegal and the transport will not respond to a write command.

FORMAT CONTROL: If the PDP-5 halt command is given during normal reading or read comparing, the tape proceeds to the end of record, and the control shuts down the transport. If a halt is given in continuous reading or read comparing, the transport will proceed to end of tape and shut down. If a halt command is given in normal spacing, the transport will proceed to EOR and shut down. If halt is given during continuous spacing, the transport will proceed until WC overflows or until it senses a file marker, load point, or end point, then shut down.

If halt is given during writing in the normal mode, the last word to be transferred is written, the rest of the record is written as zeros, and the transport is shut down. If halt is given during writing in the continuous mode, the record is completed; then zeros are written to the end of the tape. If a WC overflow occurs during a normal read or read compare, the transport proceeds to EOR before shutting down.

The functions of Type 57A Automatic Magnetic Tape Control are controlled by combinations of the following iot instructions:

mscr — 6701 — Skip if the tape control Ready (TCR) level is 1. A 1 is added to the contents of the program counter if the tape control is free to accept a command. The TCR flag is connected to the program interrupt.

mcd — 6702 — Disable the TCR flag from the program interrupt and clear command register. Clear Word Count Overflow (WCO) flag. Clear End of Record (EOR) flag. This instruction should be immediately preceded by the two instructions cla and tad (4000) to obtain the operation indicated.

mts — 6706 — Disable the TCR flag from the program interrupt, turn off the WCO flag and EOR flag and select the unit, the mode of parity, and the density from the contents of the AC. The AC bit assignments are:

AC_1
(Type 521 and 522 interface only)
0=high sense level
1=low sense level

AC_2
0=200 or 556 density
1=800 or 556 density

AC_8
0=200 density
1=556 density

AC_2	AC_8	Density
0	0	200
0	1	556
1	0	800
1	1	556

AC_7
0 = even parity (BCD)
1 = odd parity (binary)

AC_{9-11}
These three bits select one of eight tape units, addresses 0

msur — 6711 — Skip if the tape transport is ready (TTR). The selected tape unit is checked, using this command, and must be free before the following mtc command is given.

mnc — 6712 — Terminate the continuous mode. This instruction clears the AC at completion. It should be immediately preceded by the

two instructions cla and tad (4000) to obtain the operation indicated.

mtc — 6716 — Place $C(AC)_{3-6}$ in the tape control command register and start tape motion. Bit 6 selects motion mode.

AC_6

0 = Normal
1 = Continuous

AC_{3-5} are decoded as follows:

0 = no operation
1 = rewind
2 = write
3 = write end of file (EOF)
4 = read compare
5 = read
6 = space forward
7 = space backward

mswf — 6721 — Skip if the WCO flag is a 1. The flag is connected to the program interrupt.

mdwf — 6722 — Disable WCO flag.

mcwf — 6722 — Clear WCO flag. This instruction should be immediately preceded by the two instructions cla and tad (2000) to obtain the operation indicated.

mewf — 6722 — Enable WCO flag. This instruction should be immediately preceded by the two instructions cla and tad (4000) to obtain the operation indicated.

miwf — 6722 — Initialize WCO flag. This instruction should be immediately preceded by the two instructions cla and tad (6000) to obtain the operation indicated.

msef — 6731 — Skip if the EOR flag is a 1. This flag is connected to the program interrupt.

mdef — 6732 — Disable ERF.

mced — 6732 — Clear ERF. This instruction should be immediately preceded by the two instructions cla and tad (2000) to obtain the operation indicated.

meef — 6732 — Enable ERF. This instruction should be immediately preceded by the two instructions cla and tad (4000) to obtain the operation indicated.

mief — 6732 — Initialize ERF, clear and enable. This instruction should be immediately preceded by the two instructions cla and tad (6000) to obtain the operation indicated.

mtrs — 6734 — Read tape status bits into the contents of the AC. This instruction should be immediately preceded by a cla instruction to obtain the operation indicated. The bit assignments are:

0 = data request late
1 = tape parity error
2 = read compare error
3 = end of File flag set
4 = write lock ring out
5 = tape at load point

6 = tape at end point
7 = tape near end point (Type 520)
7 = last operation write (Type 521 and 522 interfaces)
8 = tape near load point (Type 520)
8 = write echo (Type 522 interface)
8 = B control using transporting (Type 521 interface with multiplex transport)
9 = transport rewinding
10 = tape miss character

mcc — 6741 — Clear CA and WC.

mrwc — 6742 — Transfer $C(AC)_{0-11}$ to $C(WC)_{0-11}$

mrca — 6744 — Transfer $C(CA)_{0-11}$ to $C(AC)_{0-11}$. This instruction should be immediately preceded by a cla instruction to obtain the operation indicated.

mca — 6745 — Clear CA and WC, and transfer $C(AC)_{0-11}$ to $C(CA)_{0-11}$.

All operations begin with the program events indicated in the following basic program sequence. When the main program branches to this sequence (having received, for example, a high priority data break request from the tape control), the control and transport are interrogated for availability (mscr, msur) and if ready are instructed to carry out the specified task (mts, mtc). If the task is one of the eight listed in the instruction list under mtc, the mscr instruction completes the program sequence; if not, the program branches at "begin" to another routine (write, read, etc.), returning afterwards to "wait" in the basic program.

```
begin,      mscr          /skip if tape control free
            jmp.-1        /tape control not free, jump back to mscr
                          /instruction
            cla
            tad ia-1      /load AC with initial address minus one
            mca           /transfer AC to CA
            cla
            tad-n+1       /load AC with complement of number of
                          /words to be transferred plus one
            mrwc          /transfer AC to WC
            cla
            tad (*)       /load AC with selected information*
            mts           /transfer AC to control with parity density
                          /and unit number
            msur          /skip if tape transport ready
            jmp.-1        /transport not ready, jump back to msur
                          /instruction
            mtc           /transfer AC to control with command
                          /and tape motion mode
wait,       mscr          /wait for tape function to complete
            jmp.-1        /tape function not complete, jump back
                          /to mscr
            hlt           /operation completion
```

*A set of mnemonics that specifies all tape operations is furnished with the Type 57A.

When programming in the interrupt mode, the TCR flag causes an interrupt in the operating program and the flag may be tested by using the mscr instruction. The TCR flag must be cleared with the mcd command before dismissing the interrupt. WCO and ERF flags must be disabled before dismissing the interrupt, with the option of clearing or not clearing the flags.

CHAPTER 5

INTERFACE ELECTRICAL CHARACTERISTICS

One of the strong features of the PDP-5 is the relative ease of input-output device connection. Input-output devices can be connected into the system up to the limits specified in this section. Refer to the Digital Modules catalog A-705 for an explanation of standard DEC signals and loading definitions used in this section.

A coordinate system is used to locate cabinets, racks, modules and cable connectors, and terminals in the PDP-5. Cabinets are numbered beginning with the cabinet containing the operator console. Each position on the front of the cabinet is assigned a capital letter, beginning with A at the top, as indicated on Figure 11. Modules are numbered from 1 through 25 from left to right in a rack, as viewed from the wiring side. Connectors are numbered from 1 through 6, from left to right as viewed from the front of the machine. Blank module and connector locations are numbered. Terminals on a module connector are designated by capital letters from top to bottom, omitting G, I, O, and Q. Therefore, 1D05F is in cabinet 1, the fourth location from the top (D), the fifth module from the left (05), and the six (F) terminal from the top of the module.

Two 50-terminal cable connectors are available on the connector panel (1J01 and 1J02) for connection to I/O devices. Additional connector locations (1J03-1J05) are available for installation of connectors, as needed. Corresponding terminals of 1J01 and 1J02 are connected together and routed to signal origins or destinations in the machine logic. In the following discussions, origins of output signals and destinations of input signals are given with the terminal connection at 1J02. In this manner, the connections of both 1J01 and 1J02 are explained, and wiring to a new signal connector can be planned for bus connection to 1J02 or direct connection to the logic. Connections to 1J01 and 1J02 are summarized in Appendix B.

Device Selector

The device selector function is performed by a Type 4605 Pulse Amplifier for each I/O device or external register, which is individually selected. Each I/O device added to the system must contain a Type 4605 module, which has been

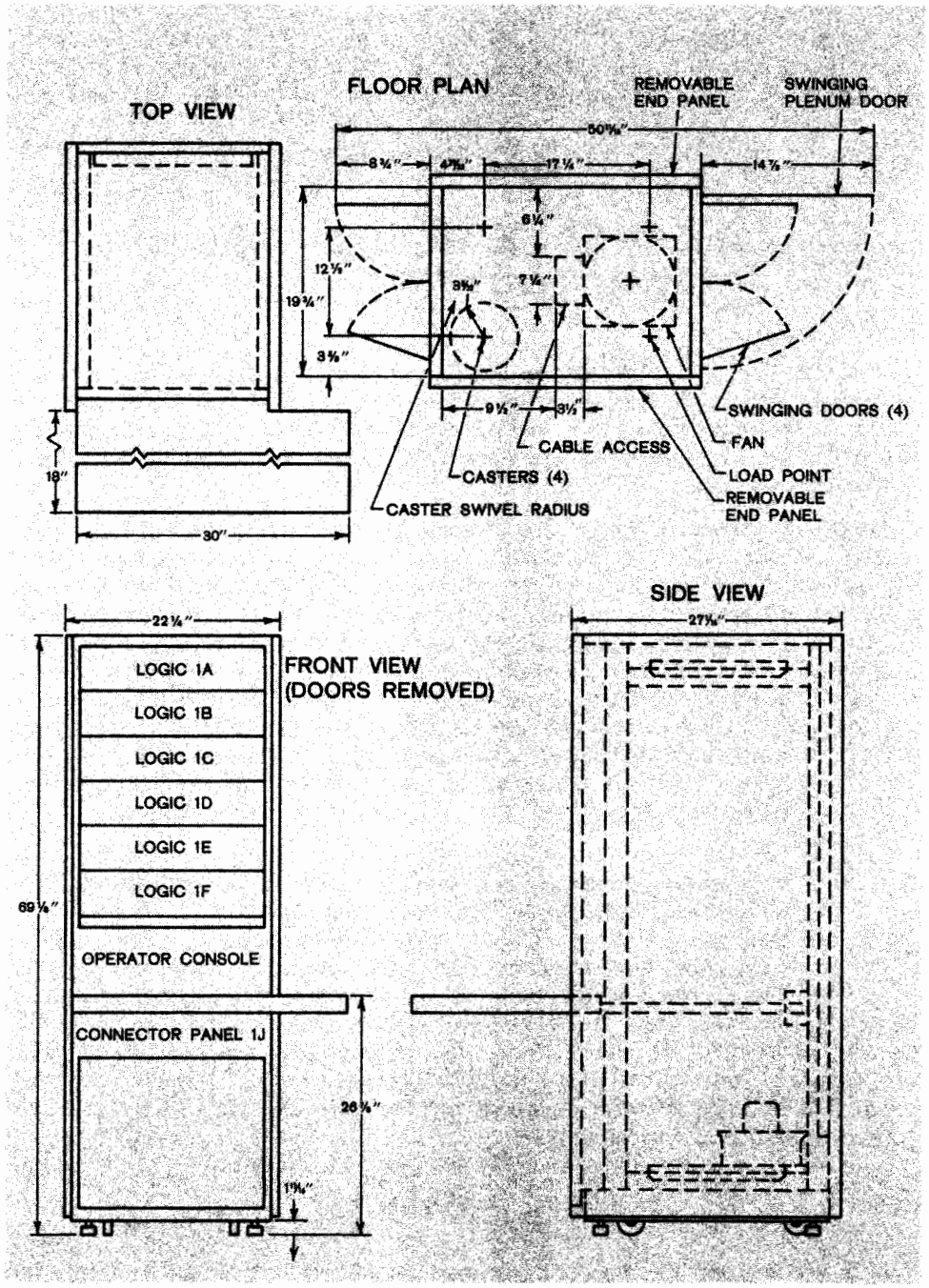


Figure 11 Component Location and Installation Diagram

prepared to select the device for a given combination of bits 3 through 8 of an iot instruction. When selected in this manner, Type 4605 produces IOT pulses related to the IOP pulses which are generated in accordance with the presence of ones in bits 9, 10, and 11 of the iot instruction. These IOT pulses, in turn, must be wired to initiate operation of the I/O device.

Therefore, cable connections must supply inputs to each Type 4605 from both the 1 and 0 output of memory buffer register bits 3 through 8 (12 lines) and from the three IOP generator outputs (6 lines or 3 twisted pairs). Connections are then made directly from the three output terminals of Type 4605 directly to the logic circuits of the I/O device. The input and output terminals of Type 4605 module are indicated in the logic diagram shown in Figure 12.

Type 4605 Pulse Amplifier modules are delivered with a jumper wire from both complementary inputs of each MB bit connected to one of the six inputs of the -AND diode gate. (Jumpers are indicated as dotted lines in Figure 12). The user must remove one jumper from each -AND gate input to establish the appropriate select code. (Both jumpers may be removed if the selection code requires it.) This system allows select codes to be changed in the module and not in cable connections. As delivered, these modules are also wired to produce negative IOT pulses. Positive IOT pulses can be obtained by reversing both jumper wire connections of a pulse transformer secondary winding.

Note that the input connections to Type 4605 must be as specified in Figure 12 and cannot be modified to operate more than one pulse amplifier (per module) at the same time. Should an I/O device require coincident positive and negative IOT pulses, two separate Type 4605 modules must be used, or an IOT pulse can be used to trigger external positive and negative pulse amplifiers. Note also that positive IOT pulses cannot be inverted to produce negative IOT pulses but can be used to trigger a pulse amplifier, such as Type 4604 or 4606 modules.

Output pulses from a Type 4605 Pulse Amplifier are standard for the DEC 4000 Series systems modules (2.5 volts, 0.4 microsecond). Each output is capable of driving 16 units of pulse load.

Memory Buffer Register

Bits 3 through 8 of an iot instruction are used to select the I/O device addressed by the instruction. During the F cycle, the instruction word is read from memory and placed in the memory buffer register. Complementary outputs from flip-flop bits MB_{3-8} are wired to input terminals of each device selector module connector. When the device selector is located within the I/O device, these MB lines must be connected to a cable connector.

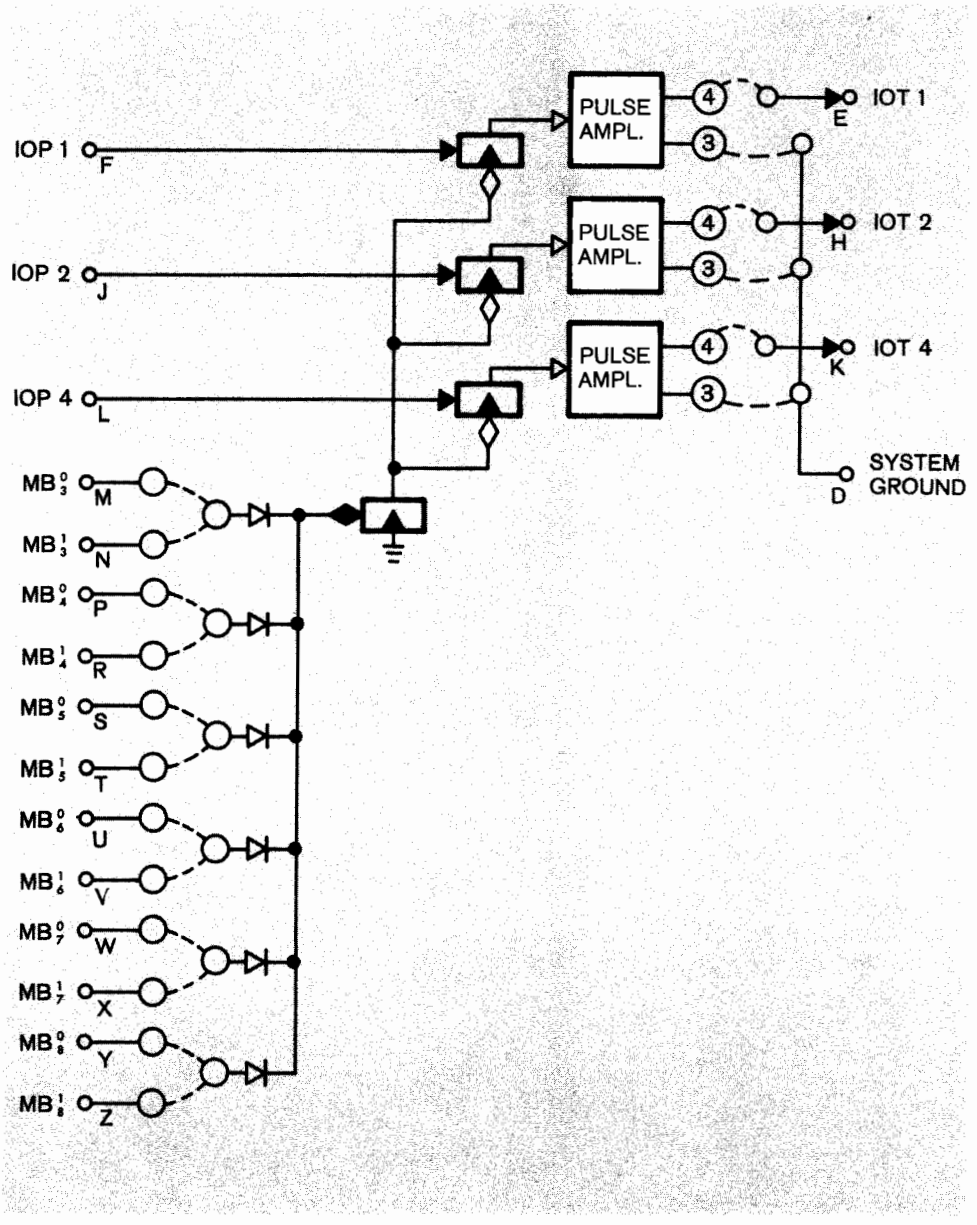


Figure 12 Type 4605 Pulse Amplifier Logic Diagram

The terminal locations for this connection are:

Signal	Origin	Bus Driver Output	Connection	Signal	Origin	Bus Driver Output	Connection
$MB\,^{0}_{3}$	1B05L	1F09L	1J02-27	$MB\,^{1}_{3}$	1B05K	1F09N	1J02-28
$MB\,^{0}_{4}$	1B06L	1F09T	1J02-29	$MB\,^{1}_{4}$	1B06K	1F09R	1J02-30
$MB\,^{0}_{5}$	1B07L	1F10L	1J02-31	$MB\,^{1}_{5}$	1B07K	1F10N	1J02-32
$MB\,^{0}_{6}$	1B08L	1F10T	1J02-33	$MB\,^{1}_{6}$	1B08K	1F10R	1J02-34
$MB\,^{0}_{7}$	1B09L	1F11L	1J02-35	$MB\,^{1}_{7}$	1B09K	1F11N	1J02-36
$MB\,^{0}_{8}$	1B10L	1F11T	1J02-37	$MB\,^{1}_{8}$	1B10K	1F11R	1J02-33

Memory buffer register outputs are wired from their point of origin in a Type 4206 Triple Flip-Flop module at locations 1B05 through 1B10 to connectors at 1F09 through 1F11. Normally, locations 1F09 through 1F11 contain dummy plugs which jumper terminals corresponding to the input and output of a Type 1684 Bus Driver. Therefore, when sufficient device selectors are added to the system to overload the normal driving capabilities of the Type 4206 modules, these dummy plugs can be removed and replaced by Type 1684 Bus Driver Modules. Each Type 4206 output can drive four Type 4605 Pulse Amplifier modules in the device selector. When the bus drivers are inserted in the system, each MB signal can drive at least 12 Type 4605 Pulse Amplifier modules, since Type 1684 can supply $\pm$15 milliamperes, and each Type 4605 requires 1.25 milliamperes shared among the grounded inputs. Under most circumstances, a single Type 1684 output can drive more than 12 Type 4605 modules because the load presented by a Type 4605 is shared by Type 1684 modules that drive it. To determine the maximum number of Type 4605 modules which can be driven by Type 1684 modules look for the condition where the minimum number of bus drivers is holding the maximum number of outputs at ground level. Under these conditions, the current delivered by each driver in a Type 1684 is equal to 1.25 milliamperes times the number of loads, divided by the number of bus drivers. This current must not exceed 15 milliamperes per driver circuit.

IOP Generator

The IOP pulses trigger the selected pulse amplifiers in the device selector located in the I/O device. These pulses are produced in a Type 4606 Pulse Amplifier module in location 1D25 and are routed as twisted-wire pairs to the appropriate input terminals of all Type 4605 Pulse Amplifier module connectors. Each IOP pulse can drive 16 Type 4605 modules.

Specific connection points for IOP pulses are:

Signal	Origin	Connection
IOP 1	1D25H	1J02-39*, 40
IOP 2	1D25P	1J02-41*, 42
IOP 4	1D25W	1J02-43*, 44

*Ground side of pulse amplifier transformer secondary winding to be connected to terminal D of the Type 4605 module in the device selector.

Accumulator Outputs

Data contained in the AC is available as static levels to supply information to I/O devices. These static levels can be strobed into an I/O device register by IOT pulses from the associated DS. Binary designation for the static output levels of the AC is:

−3 volts when AC bit contains a 0
0 volts when AC bit contains a 1

Connection points for these outputs are:

Signal	Origin	Bus Driver Output	Connection	Signal	Origin	Bus Driver Output	Connection
AC $\frac{1}{0}$	1B02E	1F06L	1J02-1	AC $\frac{1}{6}$	1B08E	1F07T	1J02-7
AC $\frac{1}{1}$	1B03E	1F06N	1J02-2	AC $\frac{1}{7}$	1B09E	1F07R	1J02-8
AC $\frac{1}{2}$	1B04E	1F06T	1J02-3	AC $\frac{1}{8}$	1B10E	1F08L	1J02-9
AC $\frac{1}{3}$	1B05E	1F06R	1J02-4	AC $\frac{1}{9}$	1B11E	1F08N	1J02-10
AC $\frac{1}{4}$	1B06E	1F07L	1J02-5	AC $\frac{1}{10}$	1B12E	1F08T	1J02-11
AC $\frac{1}{5}$	1B07E	1F07N	1J02-6	AC $\frac{1}{11}$	1B13E	1F08R	1J02-12

Accumulator outputs are wired from their point of origin in a Type 4206 Triple Flip-Flop to module connectors at locations 1F06, 07, and 08. Normally these locations contain dummy plugs which jumper terminals corresponding to the input and output of a Type 1685 Bus Driver. When sufficient I/O devices are connected to the AC to overload Type 4206 modules, these dummy plugs can be removed and replaced by Type 1685 Bus Driver modules.

With the dummy plugs in the system each AC output signal is capable of driving:

six 1500-ohm capacitor-diode gate level inputs or
ten units of 5MC base load or
six units of 500KC base load or
two units of DC emitter load.

With the dummy plugs replaced by bus drivers each AC output signal is capable of driving:

one hundred 1500-ohm capacitor-diode gate level inputs or
fifteen units of base load or
twelve negative OR diode gates.

Each output can supply ±15 milliamperes. The rise and fall times of the output signals are approximately 1 microsecond. For more than a 5000-picofarad output load, the maximum rise or fall time in microseconds is equal to the capacitance in picofarads divided by 5000. Maximum rise or fall time of a bus driver output should be limited to 10 microseconds.

Accumulator Inputs

Transfer of data from an I/O device to the PDP-5 is normally received at the AC input. The AC input is accessible only through a pulse input to Type 4130 Capacitor-Diode Gate modules at locations 1E10 through 1E15. The level input to these gates is permanently connected to system ground and the pulse input is clamped at −3 volts by the Type 1000 Clamped Load Resistor module at location 1E16. Therefore, gated register outputs from many I/O devices can be connected to the AC input, so that IOT pulses set the information into the PDP-5. The input terminals are:

Signal	Connection	Load	Destination	Signal	Connection	Load	Destination
AC $\frac{1}{0}$	1J02-13	1E16E	1E10M	AC $\frac{1}{6}$	1J02-19	1E16M	1E13M
AC $\frac{1}{1}$	1J02-14	1E16F	1E10Y	AC $\frac{1}{7}$	1J02-20	1E16N	1E13Y
AC $\frac{1}{2}$	1J02-15	1E16H	1E11M	AC $\frac{1}{8}$	1J02-21	1E16P	1E14M
AC $\frac{1}{3}$	1J02-16	1E16J	1E12M	AC $\frac{1}{9}$	1J02-22	1E16R	1E14Y
AC $\frac{1}{4}$	1J02-17	1E16K	1E12M	AC $\frac{1}{10}$	1J02-23	1E16S	1E15M
AC $\frac{1}{5}$	1J02-18	1E16L	1E12Y	AC $\frac{1}{11}$	1J02-24	1E16T	1E15Y

Driving any AC input connection point to ground potential sets a 1 into the corresponding AC flip-flop. The input change should be a maximum of 0.5 volts to avoid setting a flip-flop to a 1, and must be at least 2 volts with a rise time of less than 0.3 microseconds to reliably set a 1 into the AC. Each input presents a load of one standard clamped load resistor in parallel with 330 picofarads to ground.

Input-Output Skip

A skip bus is available for input connections to the PDP-5 from gated Skip pulses generated in I/O equipment. Input Skip pulses are usually produced by a flag or device status level which is strobed or sampled by an IOT pulse. The IOT pulse from the DS strobes the flag; and if it is in the preselected binary condition, the instruction following the iot is skipped.

Connection points for IOS are:

Signal	Connection	Load	Destination
IOS	1J02-25	1C04R	1D03E

To cause an instruction to be skipped, the IOS bus must be driven to ground potential for 0.4 microseconds by a pulse with a rise time of less than 0.2 microseconds. This pulse must originate in a high-impedance source, such as a transistor in a standard DEC inverter, diode gate, or capacitor-diode gate. The source of the IOS pulse cannot exhibit more than 1000 picofarads for the driving transistor.

These input pulses provide the complement input to the Type 4215 Four-Bit Counter module at location 1D03. Within the equipment this point is clamped at −3 volts by the collector load resistor of a Type 4129 Negative Capacitor-Diode Gate at location 1C04.

Program Interrupt

Signals from I/O devices, which interrupt the program in progress, are connected to a bus on the PDP-5. Connections to this bus must be in the form of static levels: ground potential to interrupt, −3 volts for no effect. The PI connection points are:

Signal	Connection	Destination
PI	1J02-26	1E04Y

The PI signal level is clamped at −3 volts by the collector load of the Type 4114 Diode NOR at location 1D04, is inverted and isolated by the Type 4102 Inverter at location 1E04, and is supplied to one input of the Type 4115 Diode AND at location 1D05 as the primary condition for initiating the internal interrupt gate. Connection to the PI bus represents 1 unit of dc emitter load. The maximum total leakage current from all sources connected to the PI bus must not exceed 6 milliamperes.

Input-Output Halt

The IOH facility provides a means of halting the advance of the program for an undetermined length of time while an I/O device executes a programmed operation. A specific iot instruction is decoded in the I/O device DS to produce IOT pulses which initiate device operation and return to the PDP-5 as an I/O Halt pulse. The I/O Halt pulse sets the I/O Halt flip-flop to 1, which in turn sets the run flip-flop to 0, so that the program stops. When the I/O device completes the operation specified by the iot instruction, it supplies a Restart pulse to the PDP-5 which returns the run flip-flop to the one state to continue the program and sets the I/O Halt flip-flop to 0.

These connections are:

Signal	Connection	Destination
I/O Hlt	1J02-46	1D12Y
Restart	1J02-48	1E02Y

I/O Halt pulses must be Standard DEC Negative Pulses (−2.5 volts, 0.4 microsecond) or equivalent. The dc load presented to the signal by the input is ⅛ unit of dc emitter load. This load is shared by those inputs which are at ground. The transient load presented to a pulse input is 1 unit pulse load. I/O Halt pulses are received by a Type 4116 Diode module at location 1D12 which functions as a negative OR gate. The inverted output of this gate sets the I/O halt flip-flop when it is at ground potential. This flip-flop is contained in the Type 4215 module at location 1D01. The 1 output at the I/O halt flip-flop sets the run flip-flop to 0. The run flip-flop is also contained in the module at location 1D01.

The Restart pulse is received at the pulse input of a Type 4129 (negative) Capacitor-Diode Gate at location 1E02. The conditioning level input to this gate is provided by the one status of the I/O halt flip-flop. The Restart pulse may be driven from a Standard DEC 0.4 microsecond −2.5 volt Negative Pulse, or it may be driven from a negative-going level change. The level change should be 2.5 to 3.3 volts, with a maximum fall time of 0.4 microseconds. The input represents 3 units of pulse load.

Cabling

Power and signal cables enter the computer cabinet through a port in the bottom. The power cable is permanently wired to the equipment and signal cables mate with connectors, which are mounted on the front of the cabinet, facing the center of the machine.

Power cables for the computer and for most peripheral equipment are supplied with twist-lock connectors, rated at 30 amperes. To mate with the power cables, power sources should be provided with Hubbell 7310B, or equivalent twist-lock, flush receptacles rated at 30 amperes, 115 volts alternating current. Note that the receptacle terminal stamped GR or marked with green paint must be grounded.

Signal cables are 50-wire, shielded, with Amphenol 115-114P male connectors and 1391 shells on both ends. To mate with a signal cable, special equipment in the system must be provided with Amphenol 115-114S female connectors. Unless otherwise specified by the user, power cables are supplied in 20 foot lengths; signal cables, in 25 foot lengths. Power cables are 11/16 inch in diameter; signal cables are 13/16 inch in diameter.

APPENDIX 1

INSTRUCTIONS

MEMORY REFERENCE INSTRUCTIONS

Mnemonic Symbol	Operation Code	Time (μsec)	Operation
and Y	0	18	Logical AND. The AND operation is performed between the C(Y) and the C(AC). $C(Y)_i \wedge C(AC)_i \Rightarrow C(AC)_i$.
tad Y	1	18	Twos complement add. The C(Y) are added to the C(AC) in twos complement arithmetic. $C(Y) + C(AC) \Rightarrow C(AC)$.
isz Y	2	18	Index and skip if zero. The C(Y) are incremented by one in twos complement arithmetic. If the resultant C(Y) = 0, the next instruction is skipped. $C(Y) + 1 \Rightarrow C(Y)$. If result = 0, $C(PC) + 1 \Rightarrow C(PC)$.
dca Y	3	18	Deposit and clear AC. The C(AC) are deposited in core memory location Y and the AC is cleared. $C(AC) \Rightarrow C(Y)$, then $0 \Rightarrow C(AC)$.
jms Y	4	24	Jump to subroutine. The C(PC) are deposited in core memory location Y. The next instruction is taken from location Y+1. $C(PC) + 1 \Rightarrow C(Y)$ $Y + 1 \Rightarrow C(PC)$
jmp Y	5	12	Jump to Y. The C(PC) are set to address Y. The next instruction is taken from core memory location Y. $Y \Rightarrow C(PC)$.

BASIC IOT MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation
		PROGRAM INTERRUPT
ion	6001	Turn interrupt on
iof	6002	Turn interrupt off
		ANALOG-TO-DIGITAL CONVERTER
adc	6004	Convert analog to digital
		HIGH SPEED PERFORATED TAPE READER
rsf	6011	Skip if Photoreader flag = 1
rrb	6012	Read the contents of the photoreader buffer into $C(AC)_{4-11}$ and clear the Photoreader flag
rfc	6014	Clear Photoreader flag and buffer, fetch one character from tape and load it into the photoreader buffer, and set the Photoreader flag when done.
		HIGH SPEED PERFORATED TAPE PUNCH
psf	6021	Skip if High Speed Punch flag = 1
pcf	6022	Clear the Punch flag and buffer
ppc	6024	Load the punch buffer from $C(AC)_{4-11}$ and punch the character (this instruction does not clear the High Speed Punch flag or buffer).
pls	6026	Clear the Punch flag and buffer, load the punch buffer from $C(AC)_{4-11}$, punch the character, and set the Punch flag when done.
		TELETYPE KEYBOARD/READER
ksf	6031	Skip if Keyboard flag = 1
kcc	6032	Clear AC and Keyboard flag.
krs	6034	Read the contents of the keyboard buffer into $C(AC)_{4-11}$ (does not clear AC or flag.)
krb	6036	Clear AC, read keyboard buffer into AC, clear Keyboard flag.
		TELETYPE TELEPRINTER/PUNCH
tsf	6041	Skip if Teleprinter flag = 1
tcf	6042	Clear Teleprinter flag
tls	6046	Load the LUO from the $C(AC)_{4-11}$, clear Teleprinter flag, and print and/or punch the character.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation
		OSCILLOSCOPE DISPLAY AND PRECISION CRT DISPLAY
dcx	6051	Clear X buffer
dxl	6053	Clear and load the X buffer $C(AC)_{2-11} => C(YB)$.
dcy	6061	Clear Y buffer
dyl	6063	Clear and load the Y buffer $C(AC)_{2-11} => C(YB)$.
dix	6054	Intensify the point defined by C(XB) and C(YB)
diy	6064	Intensify the point defined by C(XB) and C(YB)
dxs	6057	Executes the combined functions of dxl followed by dix
dys	6067	Executes the combined functions of dyl followed by diy.
dsf	6071	Skip if Display flag = 1
dcf	6072	Clear Display flag
dlb	6074	Load brightness register. $C(AC)_{8-11} => C(BR)$
		INCREMENTAL PLOTTER
plsf	6501	Skip if Plotter flag = 1
plcf	6502	Clear Plotter flag
plpu	6504	Plotter pen up
plpr	6511	Plotter pen right
pldu	6512	Plotter drum upward
pldd	6514	Plotter drum downward
plpl	6521	Plotter pen left
pldu	6522	Plotter drum upward
plpd	6524	Plotter pen down
		LINE PRINTER
lcf	6652	Clear Line Printer flag.
lpr	6655	Clear the format register. Load the format register from $C(AC)_{9-11}$, print the line contained in the last half of the printing buffer, and advance the paper according to the contents of the format register if $C(AC)_8 = 1$.
lsf	6661	Skip if Line Printer flag = 1.

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation
		LINE PRINTER (continued)
lcb	6662	Clear printing buffer.
lld	6664	Load printing buffer from $C(AC)_{6-11}$.
		CARD READER AND CONTROL
crsf	6632	Skip if Card Reader flag = 1.
cers	6634	Card equipment read status. Reads the status of the card reader into $C(AC)_{6-9}$.
crrb	6671	Read the contents of the card column buffer into the C(AC) and clear the Card Reader flag.
crsa	6672	Select a card in alphanumeric mode.
crsb	6674	Select a card in binary mode.
		CARD PUNCH CONTROL
cpsf	6631	Skip if Card Punch flag = 1.
cers	6634	Card equipment read status. Reads the status of the Card Punch flag into bit 10 and the card punch error level into bit 11 of the AC.
cpcf	6641	Clear Card Punch flag.
cpse	6642	Select the card punch and transmit a card from the hopper to the 80-column punch die.
cplb	6644	Load the card punch buffer from C(AC).
		AUTOMATIC MAGNETIC TAPE CONTROL
mscr	6701	Skip if tape control unit is ready. If TCR = 1, then C(PC) + 1 => C(PC)
mcd	6702	Disable the TCR flag from the program interrupt; clear command register, WCO, and EOR. Used when C(AC) = 4000.
mts	6706	Disable the TCR flag from the program interrupt, clear WCO and EOR. Select unit, parity mode, and density.
msur	6711	Skip if tape transport unit is ready. If TTR = 1, then C(PC) + 1 => C(PC)
mnc	6712	Terminate continuous mode. Used when C(AC) = 4000.
mtc	6716	Load tape control unit command register, start tape motion,

BASIC IOT MICROINSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Operation
		AUTOMATIC MAGNETIC TAPE CONTROL (continued)
mswf	6721	Skip if WCO flag = 1 and clear AC.
mdwf	6722	Disable WCO flag. Used when C(AC) = 2000.
mewf	6722	Enable WCO flag. Used when C(AC) = 4000.
miwf	6722	Initialize WCO flag (clear, enable). Used when C(AC) = 600.
msef	6731	Skip if EOR flag = 1.
mdef	6732	Disable ERF.
mced	6732	Clear ERF. Used when C(AC) = 2000.
meef	6732	Enable ERF. Used when C(AC) = 4000.
mief	6732	Initialize ERF (clear, enable). Used when C(AC) = 6000.
mtrs	6734	Read tape status bits into C(AC). Used when C(AC) = 0000.
mcc	6741	Clear CA and WC.
mrwc	6742	Read word counter. C(WC) => $C(AC)_{0-11}$
mrca	6744	Read current address. Used when C(AC) = 0000. C(CA) => $C(AC)_{0-11}$
mca	6745	Read current address, and clear CA and WC. Executes the combined functions of mcc with mrca.

GROUP 1 OPERATE MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Event Time	Operation
nop	7000	—	No operation. Causes a 12 μsec program delay.
iac	7001	3	Index AC. C(AC) +1=> C(AC)
ral	7004	2	Rotate the C(AC) and the C(L) left one place. $C(AC)_i$ => $C(AC)_{i-1}$ $C(AC)_{11}$ => C(L) C(L) => $C(AC)_0$
rtl	7006	2, 3	Rotate two left.

GROUP 1 OPERATE MICROINSTRUCTIONS (continued)

Mnemonic Symbol	Octal Code	Event Time	Operation
rar	7010	2	Rotate the C(AC) and the C(L) right one place. $C(AC)_i \Rightarrow C(AC)_{i+1}$ $C(AC)_{11} \Rightarrow C(L)$ $C(L) \Rightarrow C(AC)_0$
rtr	7012	2, 3	Rotate two right.
cml	7020	2	Complement L. $C(\overline{L}) \Rightarrow C(L)$
cma	7040	2	Complement AC. $C(\overline{AC}) \Rightarrow C(AC)$
cll	7100	1	Clear L. $0 \Rightarrow C(L)$
cla	7200	1	Clear AC. $0 \Rightarrow C(AC)$

GROUP 2 OPERATE MICROINSTRUCTIONS

Mnemonic Symbol	Octal Code	Event Time	Operation
hlt	7402	3	Halt. Stops the program.
osr	7404	3	OR with Switch Register C(SR) V C(AC) $\Rightarrow$ (CAC)
skp	7410	1	Skip, unconditional. C(PC) + 1 $\Rightarrow$ C(PC)
snl	7420	1	Skip on non-zero L. If C(L) = 1, then C(PC) + 1 $\Rightarrow$ C(PC)
szl	7430	1	Skip on zero L. If C(L) = 0, then C(PC) + 1 $\Rightarrow$ C(PC)
sza	7440	1	Skip on zero AC. If C(AC) = 0, then C(PC) + 1 $\Rightarrow$ C(PC)
sna	7450	1	Skip on non-zero AC. If C(AC) $\neq$ 0, then C(PC) +1 $\Rightarrow$ C(PC)
sma	7500	1	Skip on minus AC. If $C(AC)_0 = 1$, then C(PC) + 1 $\Rightarrow$ C(PC)
spa	7510	1	Skip on positive AC. If $C(AC)_0 = 0$, then C(PC) + 1 $\Rightarrow$ C(PC)
cla	7600	2	Clear AC 0 $\Rightarrow$ C(AC)

APPENDIX 2

CODES

TELETYPE CODE

Character	8-Bit Code (in octal)	6-Bit Trimmed Code (in octal)	Character	8-Bit Code (in octal)	6-Bit Trimmed Code (in octal)
A	301	01	!	241	41
B	302	02	"	242	42
C	303	03	#	243	43
D	304	04	$	244	44
E	305	05	%	245	45
F	306	06	&	246	46
G	307	07	'	247	47
H	310	10	(	250	50
I	311	11	)	251	51
J	312	12	*	252	52
K	313	13	+	253	53
L	314	14	,	254	54
M	315	15	-	255	55
N	316	16	.	256	56
O	317	17	/	257	57
P	320	20	:	272	72
Q	321	21	;	273	73
R	322	22	<	274	74
S	323	23	=	275	75
T	324	24	>	276	76
U	325	25	?	277	77
V	326	26	@	300	00
W	327	27	[	333	33
X	330	30	\	334	34
Y	331	31	]	335	35
Z	332	32	↑	336	36
0	260	60	←	337	37
1	261	61	EOT	204	—
2	262	62	W RU	205	—
3	263	63	RU	206	—
4	264	64	BELL	207	—
5	265	65	Line Feed	212	—
6	266	66	Return	215	—
7	267	67	Space	240	40
8	270	70	ACK	374	—
9	271	71	ALT MODE	375	—
			Rub Out	377	—

CARD READER AND LINE PRINTER OCTAL CODES

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

CARD READER AND LINE PRINTER BINARY CODES

Low Order Bits	High Order Bits 00		01		10		11	
	Card Reader Character	Line Printer Character	Card Reader Character	Line Reader Character	Card Reader Character	Line Reader Character	Card Reader Character	Line Reader Character
0000	....	space		0	—	°	+[&]	—
0001	1	1	/	/	J	J	A	A
0010	2	2	S	S	K	K	B	B
0011	3	3	T	T	L	L	C	C
0100	4	4	U	U	M	M	D	D
0101	5	5	V	V	N	N	E	E
0110	6	6	W	W	O	O	F	F
0111	7	7	X	X	P	P	G	G
1000	8	8	Y	Y	Q	Q	H	H
1001	9	9	Z	Z	R	R	I	I
1010	0	′	....	″	....	....	....	×
1011	=[#]	~	′	′	$	=	.	.
1100	′[@]	⊃	([%]	>	*	—	)[□]	+
1101	....	V	....	↑	....	)	....	]
1110	....	∧	....	→	....	—	....	I
1111	....	<	....	?	....	(	....	[

HOLLERITH CARD CODE

Digit	Zone No Zone	12	11	0
no punch	blank	+[&]	—	O
1	1	A	J	/
2	2	B	K	S
3	3	C	L	T
4	4	D	M	U
5	5	E	N	V
6	6	F	O	W
7	7	G	P	X
8	8	H	Q	Y
9	9	I	R	Z
8-3	=[#]	.	$	,
8-4	′[@]	)[□]	*	([%]

APPENDIX 3

PERFORATED-TAPE LOADER SEQUENCES

READIN MODE LOADER

The readin mode (RIM) loader is a minimum length, basic, perforated-tape reader for the PDP-5. It is initially stored in memory by manual use of the operator console keys and switches. The loader is permanently stored in 17 locations of the highest numbered page.

A perforated tape to be read by the RIM loader must be in RIM format:

Tape Channel 8 7 6 5 4 S 3 2 1	Format
1 0 0 0 0 . 0 0 0	Leader-trailer code
0 1 A1 . A2 0 0 A3 . A4	Absolute address to contain next 4 digits
0 0 X1 . X2 0 0 X3 . X4	Contents of previous 4-digit address
0 1 A1 . A2 0 0 A3 . A4	Address
0 0 X1 . X2 0 0 X3 . X4	Contents
(Etc.)	(Etc.)
1 0 0 0 0 . 0 0 0	Leader-trailer code

A tape in RIM format is generally concluded with address = 0000 and content = SA-1, where SA indicates starting address. In this way, the SA of the routine just loaded is stored in the program counter of the PDP-5. The next instruction to be executed will then be taken from the SA, (i.e., the program counter is incremented, then used as the address of the instruction). Therefore, the loaded routine is self-starting. It is suggested that this procedure always be used. If it is not desirable for the routine to be self-starting, simply store a halt instruction in the SA. Pressing the CONTINUE key then starts the routine.

The RIM loader can only be used in conjunction with the 33 ASR reader (not the high-speed perforated-tape reader). Because a tape in RIM format is, in effect, twice as long as it need be, it is suggested that the RIM loader be used only to read the binary loader when using the 33 ASR.

The complete PDP-5 RIM loader (SA = 1700 in systems with IK memory or 7700 in systems with 4K memory) is as follows:

Addr.	Octal Contents	Tag	Inst'n I Z	Comments
700,	6032	beg,	kcc	/clear AC and flag
701,	6031		rsf	/skip if flag = 1
702,	5301		jmp .−1	/looking for char
703,	6036		krb	/read buffer
704,	7106		cll rtl	
705,	7006		rtl	/ch 8 in AC_0
706,	7510		spa	/checking for leader
707,	5301		jmp beg +1	/found leader
710,	7006		rtl	/OK, ch 7 in link
711,	6031		ksf	
712,	5311		jmp .−1	/read, do not clear
713,	6034		krs	/checking for address
714,	7420		snl	
715,	3720		dca i temp	/store contents
716,	3320		dca temp	/store address
717,	5300		jmp beg	/next word
720,		temp,		/temp storage

Placing the RIM loader in core memory by way of the operator console keys and switches is accomplished as follows:

1. Set the appropriate starting address in the switch register (SR).
2. Press LOAD ADDRESS key.
3. Set the first instruction in the SR.
4. Press the DEPOSIT key.
5. Set the next instruction in the SR.
6. Press DEPOSIT key.
7. Repeat steps 5 and 6 until all 16 instructions have been deposited.

To load a tape in RIM format, place the tape in the reader, set the SR to the appropriate starting address, press the LOAD ADDRESS key, press the START key, and start the Teletype reader.

BINARY LOADER

The binary loader (BIN) is used to read machine language tapes (in binary format) produced by the program assembly language (PAL). A tape in binary format is about one half the length of the comparable RIM format tape. It can, therefore, be read about twice as fast as a RIM tape and is, for this reason, the more desirable format to use with the 10 cps 33 ASR reader.

The format of a binary tape is as follows:

LEADER: about 2 feet of leader-trailer codes.

BODY: characters representing the absolute, machine language program in easy-to-read binary (or octal) form. The section of tape may contain characters representing instructions (channels 8 and 7 not punched) or origin resettings (channel 8 not punched, channel 7 punched) and is concluded by 2 characters (channels 8 and 7 not punched) that represent a check-sum for the entire section.

TRAILER: same as leader

Example of the format of a binary tape:

Tape Channel 8 7 6 5 4 S 3 2 1	Memory Location	Contents
1 0 0 0 0 . 0 0 0	leader-trailer code	
0 1 0 0 0 . 0 1 0		
0 0 0 0 0 . 0 0 0		
0 0 1 1 1 . 0 1 0		
0 0 0 0 0 . 0 0 0	0200	cla
0 0 0 0 1 . 0 1 0		
0 0 1 1 1 . 1 1 1	0201	tad 277
0 0 0 1 1 . 0 1 0		
0 0 1 1 1 . 1 1 0	0202	dca 276
0 0 1 1 1 . 1 0 0		
0 0 0 0 0 . 0 1 0	0203	hlt
0 1 0 0 0 . 0 1 0		
0 0 1 1 1 . 1 1 1	original setting at 0277	
0 0 0 0 0 . 0 0 0		
0 0 1 0 1 . 0 1 1	0277	0053
0 0 0 0 1 . 0 0 0		
0 0 0 0 0 . 1 1 1	sum check 1007	
1 0 0 0 0 . 0 0 0	leader-trailer code	

After a BIN tape has been read in, one of the two following conditions exists:

a. No check-sum error: halt with AC = 0
b. Check-sum error: halt with AC = (computed check-sum) − (tape check-sum)

The BIN loader in no way depends upon or uses the RIM leader. To load a tape in BIN format place the tape in the reader, set the SR to 1777, press the LOAD ADDRESS key, press the START key, and start the tape reader.

APPENDIX 4

SOFTWARE

A programming parcel is supplied to each user of the PDP-5. Each parcel consists of program descriptions and perforated-paper tapes applicable to a particular system, selected from the DEC Program Library. The following programs are included in each package:

a. Program Assembly Language (PAL)
b. Readin Mode and Binary Tape Loaders
c. Symbolic Tape Editor
d. Mnemonic-Octal Debugging Routine
e. Multiply and Divide Subroutines, single and double precision
f. Square Root, Sine, and Cosine Subroutines
g. Binary-to-Decimal and Decimal-to-Binary Conversion Subroutines
h. Interpretive Floating Point Package
i. Floating Point I/O Package
j. Teletype Output Package
k. Maintenance Programs

New techniques, routines, and programs are constantly being developed, field-tested, and documented in the DEC Program Library for incorporation in users' systems.

APPENDIX 5

TABLE OF 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

APPENDIX 6

INTERFACE CONNECTIONS

Connection points for interface signals at the I/O connectors 1J01 and 1J02 are listed in the following table.

PDP-5 I/O CONNECTIONS

Signal	Symbol	Terminal	Signal	Symbol	Terminal
AC $\frac{1}{0}$ Out		1	AC $\frac{1}{4}$ In		17
AC $\frac{1}{1}$ Out		2	AC $\frac{1}{5}$ In		18
AC $\frac{1}{2}$ Out		3	AC $\frac{1}{6}$ In		19
AC $\frac{1}{3}$ Out		4	AC $\frac{1}{7}$ In		20
AC $\frac{1}{4}$ Out		5	AC $\frac{1}{8}$ In		21
AC $\frac{1}{5}$ Out		6	AC $\frac{1}{9}$ In		22
AC $\frac{1}{6}$ Out		7	AC $\frac{1}{10}$ In		23
AC $\frac{1}{7}$ Out		8	AC $\frac{1}{11}$ In		24
AC $\frac{1}{8}$ Out		9	IOS		25
AC $\frac{1}{9}$ Out		10	PI		26
AC $\frac{1}{10}$ Out		11	MB $\frac{0}{3}$		27
AC $\frac{1}{11}$ Out		12	MB $\frac{1}{3}$		28
AC $\frac{1}{0}$ In		13	MB $\frac{0}{4}$		29
AC $\frac{1}{1}$ In		14	MB $\frac{1}{4}$		30
AC $\frac{1}{2}$ In		15	MB $\frac{0}{5}$		31
AC $\frac{1}{3}$ In		16	MB $\frac{1}{5}$		32

PDP-5 I/O CONNECTIONS (continued)

Signal	Symbol	Terminal	Signal	Symbol	Terminal
MB $\frac{0}{6}$		33	IOP 4*		43
MB $\frac{1}{6}$		34	IOP 4		44
MB $\frac{0}{7}$		35	1 MC clock		45
MB $\frac{1}{7}$		36	I/O Hlt		46
MB $\frac{0}{8}$		37	AC Clear		47
MB $\frac{1}{8}$		38	Restart		48
IOP 1*		39	Power Clear		49
IOP 1		40	Ground		50
IOP 2*		41			
IOP 2		42			

*Ground side of pulse transformer secondary winding.

Connection points for data break signals at connector 1J03 are presented in the following table.

PDP-5 DATA BREAK CONNECTIONS

Signal	Symbol	Terminal	Signal	Symbol	Terminal
MB $\frac{1}{0}$ Out		1	MB $\frac{1}{11}$ In		24
MB $\frac{1}{1}$ Out		2	Data Addr. Bit 0		26
MB $\frac{1}{2}$ Out		3	Data Addr. Bit 1		27
MB $\frac{1}{3}$ Out		4	Data Addr. Bit 2		28
MB $\frac{1}{4}$ Out		5	Data Addr. Bit 3		29
MB $\frac{1}{5}$ Out		6	Data Addr. Bit 4		30
MB $\frac{1}{6}$ Out		7	Data Addr. Bit 5		31
MB $\frac{1}{7}$ Out		8	Data Addr. Bit 6		32
MB $\frac{1}{8}$ Out		9	Data Addr. Bit 7		33
MB $\frac{1}{9}$ Out		10	Data Addr. Bit 8		34
MB $\frac{1}{10}$ Out		11	Data Addr. Bit 9		35
MB $\frac{1}{11}$ Out		12	Data Addr. Bit 10		36
MB $\frac{1}{0}$ In		13	Data Addr. Bit 11		37
MB $\frac{1}{1}$ In		14	Break^1 State		41
MB $\frac{1}{2}$ In		15	Run^1 State		42
MB $\frac{1}{3}$ In		16	Break Request		43
MB $\frac{1}{4}$ In		17	Transfer Direction (Into PDP-5)		44
MB $\frac{1}{5}$ In		18	Increment Request		45
MB $\frac{1}{6}$ In		19	SP 0		46
MB $\frac{1}{7}$ In		20	Power Clear		47
MB $\frac{1}{8}$ In		21	Data $=>$ MB		48
MB $\frac{1}{9}$ In		22	Address Accepted		49
MB $\frac{1}{10}$ In		23	Ground		50

digital

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

3565 PRINTED IN U.S.A. 25-4/64