PDP 1

MAINTENANCE MANUAL

PROGRAMMED DATA PROCESSOR - 1 MAINTENANCE MANUAL

Technical Bulletin F-17

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

COPY NO.

TABLE OF CONTENTS

Paragraph | Page

CHAPTER 3
SYSTEM FUNCTION

CHAPTER 4
INSTALLATION

CHAPTER 7

ARITHMETIC UNIT

CHAPTER 10

CIRCUIT DESCRIPTION

CHAPTER 11
MAINTENANCE

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1-1 PURPOSE AND SCOPE.

The purpose of this instruction manual is to aid personnel in the installation, operation, and maintenance of the DEC Programmed Data Processor (PDP-1). The basic manual contains a complete description of all portions of the standard PDP-1 and of the optional additions to the central processor. The peripheral input-output equipment options are not treated in the basic manual, but are instead treated in separate supplements to the basic manual.

1-2 CHAPTER SUBJECTS.

A brief summary of system use and application is presented in Chapter 2, General Description. This chapter also contains a listing of system specifications and physical characteristics, and a brief description of presently available options.

Chapter 3, System Function, provides a full general description of all system operations. This chapter is written at a block-diagram level, and explains what the system does rather than how its functions are implemented in terms of hardware. Also included in Chapter 3 is an explanation of the flow diagrams which show the actual operations performed by the computer logic in executing the various program instructions.

Chapter 4, Installation, provides instructions for initial installation and set-up of the system.

Chapter 5, Operating Procedures, explains the use of all controls and indicators on the computer control panels. This chapter also outlines the basic operating procedures for normal computer operation.

System logical design is described in detail in four chapters, beginning with Chapter 6, Control. This chapter covers the general control functions of the computer, including the sequence break system and the high-speed channel control options.

Chapter 7, Arithmetic Unit, is a detailed explanation of the registers and other logic involved in computer arithmetic and logical operations. This chapter also includes a

description of the optional automatic multiply/divide logic.

Chapter 8, Memory, covers the operation of the computer core-memory system. In addition, it describes the logic included in the two memory field control options, type 13 and type 14.

Chapter 9, Input-Output System, explains the control of the standard input-output equipment furnished with the basic PDP-1. This equipment includes a photoelectric punched tape reader, a paper tape punch, and an automatic typewriter. (Separate maintenance manuals for these devices are furnished with the PDP-1 computer.)

Chapter 10, Circuit Analysis, describes the function, specifications, and theory of operation of the circuit modules used in the PDP-1 system.

Chapter 11, Maintenance, contains information useful for adjustment, calibration, trouble-shooting, and repair of the computer.

1-3 FIGURES.

This manual includes three general classes of figures: logic diagrams, circuit schematics, and miscellaneous figures such as photographs and block diagrams. For the convenience of maintenance personnel, the logic diagrams are collected in a separate D-size (22" x 34") package. All other photographs and drawings are assembled in numerical order at the back of the basic manual.

CHAPTER 2

GENERAL DESCRIPTION

2-1 PURPOSE OF SYSTEM

The DEC Programmed Data Processor-1 is a compact, solid-state, general purpose digital computer offering a combination of speed, flexibility and programming power unmatched by any other commercially available computer in its class. It is easy to install, operate and maintain, since it runs on ordinary 117-volt current, features simplified controls, and has built-in marginal checking to facilitate preventive maintenance.

a SPEED - PDP-1 has five-megacycle solid state logic circuits based on Digital's popular line of high-reliability circuit modules, a random-access magnetic core memory with a cycle time of five microseconds, and 18-bit fully parallel processing. These design features give PDP-1 a computation rate of 100,000 additions per second, including two calls on memory.

b FLEXIBILITY - PDP-1 is engineered to accomodate a wide variety of input-output equipment without internal machine changes. Standard equipment includes an alpha-numeric typewriter for on-line input and output operations, a photoelectric tape reader, a paper tape punch, and a one-channel automatic sequence break. Optional equipment includes extra memory modules, a 16-channel break system, 16-inch cathode-ray tube display, light pen, magnetic tape units and tape control units. In the standard machine, multiply and divide are performed by subroutines augumented by the special instructions Multiply Step and Divide Step. A fully automatic multiply and divide unit is available as an option.

c PROGRAMMING - PDP-1 is a single-address, single-instruction stored program machine operating on 18-bit 1's-complement binary numbers. Other equipment of the user's own design may be connected to the computer, either through the in-out register or a high-speed data channel which has direct access to memory. Numerous connections for inputs, outputs, data channel interrupts and similar devices are provided. Programming features include multiple-step indirect addressing, twelve variations of

arithmetic and logical shifting, ten conditional instructions and three Boolean operations.

d PHYSICAL FEATURES - The central processor is housed in three standard DEC equipment frames. All controls and standard input-output equipment are conveniently located in a fourth equipment frame directly attached to the central processor. No special wiring, subflooring or air conditioning is required.

2-2 STANDARD EQUIPMENT

The equipment included in the standard PDP-1 is shown between the two horizontal bars in figure 2-1. The central processor contains the control unit, arithmetic unit, in-out transfer control, type 12 memory module, and a one-channel sequence break system.

The type 12 memory module has a storage capacity of 4,096 18-bit words. Instructions are carried out by the control circuits in multiples of the memory cycle time of five microseconds. Add, subtract, deposit, and load, for example, are two-cycle instructions requiring ten microseconds. The control unit contains all of the registers and control circuits necessary to execute the various instructions in the program and handle the transfer of information between memory and the various registers within the central processor.

The arithmetic unit includes an accumulator, an in-out register which doubles as an accumulator extension, and the control circuits necessary to execute the various arithmetic and logical operations.

In-out transfer control handles the transfer of information between the in-out register and the various peripheral devices. The one-channel sequence break system allows a signal from a peripheral device to break the main program sequence In this case all information necessary for a subsequent return to the main program is stored in several fixed memory locations, program control then being transferred to a subroutine appropriate to the particular in-out device.

Attached to the central processor is a console through which the operator can manually control the computer. The contents of all registers in the central processor are displayed in the indicator lights on the console operator panel. This panel also includes the switches through which all computer operations are initiated, and through which the operator may transfer test words or test addresses into the central processor. Other control panel features

include: six program flags for automatic setting and computer sensing, and six sense switches for manual setting and computer sensing. The console also includes an indicator panel which displays the internal states of the control units for the standard in-out equipment and for the optional 16-channel sequence break system.

The standard PDP-1 also includes three in-out devices and the associated control units for these devices. The devices are a photoelectric tape reader, a paper tape punch and an automatic typewriter.

2-3 EQUIPMENT OPTIONS

The central processor options and input-output options which can be added to the standard PDP-1 are shown in the top and bottom sections of figure 2-1. The central processor options augment the control, arithmetic, and memory elements of the computer. The input-output options provide additional peripheral devices. Each of these devices can be added to the standard system simply by adding the necessary in-out transfer control circuits to the central processor.

a CENTRAL PROCESSOR OPTIONS - There are four options which may be added to the central processor. One of these four options is a memory extension control which allows additional type 12 memory modules to be added to the system. When additional modules are added, the original type 12 memory module (located in the central processor of the standard PDP-1) is then used as memory module 0 of the expanded system.

(1) Multiply/Divide Type 10 - When this option is added the instructions Multiply Step and Divide Step are replaced by the instructions Multiply and Divide. When either of these latter instructions is encountered in the program, the regular timing system of the computer is halted; execution of the Multiply or Divide instruction is controlled by a separate timing system. The order Multiply Y forms the double-length product of the contents of the accumulator and the contents of memory location Y. The instruction Divide Y forms the quotient of the double-length dividend stored in the accumulator and in-out register, and the divisor contained in memory location Y. Divide normally skips the following instruction. However, if the division is not possible, the skip does not occur. The type 10 multiply/

divide option performs multiplication in 14 to 25 microseconds and division in 30 to 40 microseconds (12 microseconds if division is not possible).

(2) Memory Extension Control Type 15 - The PDP-1 memory may be expanded to sixteen type 12 memory modules by installing a memory extension control. This expanded memory provides storage for 65,536 18-bit words. The 16-bit address format necessary to address 65,536 ($=2^{16}$) memory locations, is provided by extending both the program counter and the memory address register from 12 to 16 bits. The 4-bit address extension then selects the module while the regular 12-bit address specifies a single location in the selected module. In normal operation, instructions and operands are retrieved from a single module. However, the program may jump to another module, or retrieve and operand from another module, by performing an extend-mode cycle. In an extend-mode cycle, instead of being interpreted as a 12-bit address, a deferred address is interpreted as a 16-bit address.

(3) High Speed Channel Control Type 19 - This option transfers entire words directly between memory and a high-speed in-out device such as magnetic tape. To give a high-speed channel access to memory, the main program pauses for one memory cycle and then continues. Three channels are available in the option.

(4) Sequence Break System Type 20 - This automatic interrupt feature allows concurrent operation of several in-out devices and the main sequence. The system has 16 automatic interrupt channels arranged in a priority chain. An interrupt or break can be initiated by an in-out device at any time. When a break occurs, the computer stores in several fixed memory locations all information necessary for a later return to the main program. Program control is then transferred to a routine which serves the device causing the interrupt. When the Sequence Break System Type 20 is installed, it replaces the standard one-channel sequence break system.

b INPUT-OUTPUT OPTIONS - There are numerous regularly available input-output options (figure 2-1). Moreover, any other special equipment desired by the user can readily be added to the system.

(1) Visual CRT Display Type 30 - This is a 16-inch cathode-ray tube display mounted on a separate table. The Display command plots one point on the tube at the position indicated by the ten most significant bits of the accumulator and of the in-out register. Plotting a single point requires 50 microseconds.

(2) Precision CRT Display Type 31 - The operation of the 5-inch cathode ray tube display is similar to that of type 30. However, the over-all resolution of the type 31 is approximately four times as fine as that of the type 30. The type 31 option comes equipped with mounting bezel to accept a camera or a photomultiplier device.

(3) Light Pen Type 32 - This option allows information to be "written" on the cathode ray tube. The pen detects displayed information, and each time a pulse of light strikes the pen the pen output sets a program flag in the computer.

(4) Card Punch Control Type 40 - This control unit operates a standard IBM Type 523 card-punching machine. Each card row is punched from a 80-bit buffer which is loaded from the in-out register.

(5) Tape Transport Type 50 - The type 50 tape transport is compatible with IBM tape formats that have a recording density of 200 seven-bit characters per inch and an inter-record gap of three-fourths of an inch. The transfer rate is 15,000 characters per second at a tape speed of 75 inches per second. The method of recording is non-return-to-zero.

(6) Programmed Tape Control Type 51 - The programmed control transfers information between the computer and the tape one character at a time. All transfer operations, including error checking and assembly of characters into computer words, are performed by routines. Some choice of tape format is allowed, including the standard IBM format. The type 51 can control three tape units.

(7) Automatic Tape Control Type 52 - This high-speed tape control automatically transfers blocks of characters between the computer memory and the tape. By using the high-speed channels, it allows computation to continue while the transfer is in

process. Special features include scatter-read and gather-write; automatic, bit-by-bit read-compare with core memory; automatic lateral parity error detection while reading and writing; and rapid tape searching by means of skipping a pre-selected number of blocks. Tape format is standard IBM. The type 52 can control eight tape units.

(8) Automatic Line Printer and Control Type 62 - This is an on-line printing station capable of operating at up to 1000 lines per minute. A simple one-line buffer is used. The appropriate transfer instruction is repeated to fill the buffer, and the order to print is then given. Following the completion of the line print, the printer returns a completion pulse. Spacing of the paper is controlled by any one of eight format channels.

2-4 SYSTEM OPERATING SPECIFICATIONS

GENERAL SYSTEM	
Application	General purpose
Timing	Synchronous
Operation	Parallel processing
COMPUTER WORDS	
Word length	18 bits
Number length	Sign: 1 bit; Magnitude: 17 bits
Instruction length	
Memory reference	Operation code: 6 bits including an indirect address bit; Address: 12 bits
Augmented	Variable operation code; maximum length: 18 bits
Instruction type	Single address
ARITHMETIC UNIT	
Internal number system	Binary
Operation	Fixed point
Number range	$-(1 - 2^{-17}) \leq n \leq (1 - 2^{-17})$

Addition time	10 microseconds*
Multiplication by subroutine	325 microseconds
Division by subroutine	440 microseconds
Multiplication by option	14 to 25 microseconds*
Division by option	30 to 40 microseconds (if division is not possible, 12 microseconds)*

STORAGE

Media	Magnetic cores
Cycle time	5 microseconds
Capacity	4,096 words, expandable to 65,536 words

INPUT-OUTPUT SYSTEM

Operating Speeds

Photoelectric Tape Reader	400 lines/second
Paper Tape Punch	63 lines/second
Typewriter	9 characters/second
CRT Displays	20,000 points/second
Card Punch	100 cards/minute
Tape Transport	75 inches/second, 15,000 characters/second
Automatic Line Printer	600 lines/minute

Format

Reader and Punch	5- to 8-hole paper tape
Tape Transport	
Density	200 7-bit characters/inch
Inter-record gap	3/4 inch
Recording	NRZ
Line Printer	120 columns/line, 63 characters/column

2-5 PHYSICAL CHARACTERISTICS

CONSTRUCTION

* Including both instruction and operand access

The standard computer and all central processor options are housed in standard DEC bays (welded steel frames, steel-covered). Control panels are aluminum.

MODULES

Standard DEC system plug-in units, series 1000 and series 4000.

POWER EQUIPMENT

Power supplies series 700; power controls series 800.

LOGIC

Solid state. Transistors and crystal diodes utilizing static logic levels (0 vdc and -3 vdc).

DIMENSIONS

Standard PDP-1

- Height 69 1/2 inches
- Length 99 inches, including 17-inch console desk
- Width 27 inches
- Weight 1750 pounds

Single bay (when required for optional equipment)

- Height 69 1/2 inches
- Width 22 inches
- Depth 27 inches
- Weight 155 pounds

Additional type 12 memory modules

- Weight 150 pounds each

2-6 POWER REQUIREMENTS

LINE VOLTAGE INPUT

105 to 125 volts, 60 cycles, single phase.

CURRENT CONSUMPTION

Standard PDP-1

17 amperes. With reader, punch and typewriter in use and including all central processor options except additional memory modules, the computer requires less than 20 amperes.

However, because the punch draws a 9-ampere surge at turn-on, the computer should be connected to a 30-ampere line.

Additional memory modules

1.5 amperes each.

2-7 EQUIPMENT LISTING

The PDP-1 and most of its options are housed in standard DEC bays. The front of each bay can accommodate up to 12 horizontal logic panels. Each logic panel is a 19-inch mounting panel which can hold up to 25 of the standard DEC plug-in logic modules. Inside the double doors at the back of each bay is an inner plenum door. The required power supplies and power control panels are mounted on this door.

a BAYS - The Standard PDP-1 is shown in figure 2-2. The central frame containing the central processor and the console is made up of four DEC bays bolted together. As shown in the figure, the console (bay 11) is at the front of the computer. Extending behind the console are bays 1, 2, and 3 of the central processor. The rear of the bays is at the left side of the console. The plenum doors which hold the power equipment are inside the double doors shown in the figure.

On the console are the operator control panel and the in-out and sequence break indicator panel (under the metal cover at the top of the bay). The photoelectric tape reader, the paper tape punch, and the reader, punch and typewriter control logic are mounted inside the console. The typewriter is mounted on a table at the side of the console. Most central processor options are mounted in bays 1, 2, and 3 of the central frame. Additional memory modules and all in-out options must be housed in separate units.

b LOGIC PANELS AND POWER EQUIPMENT - The mounting panels and logic wiring of the central frame are shown in figure 2-3. Bay 11, containing the console, paper tape reader, and paper tape punch, is at the left. The control logic for the reader, punch and typewriter is at the bottom of this bay.

The central processor control unit and arithmetic unit fill the major portion of bays 1 and 2. The four logic panels in the upper part of bay 3 contain the memory module;

in-out transfer control is in the lower section. Above the in-out transfer section is a panel for six in-out plugs. A second in-out plug panel can be mounted above the one shown if necessary for optional in-out equipment. The logic panels at the tops of the bays contain the optional high-speed channel control and memory extension control. The space at the bottom of bay 2 is for the multiply/divide option.

Figure 2-4 shows the plenum doors on the backs of the bays. In the center of bay 3 are the large resistors for the memory power supply. At the top of bay 3 are the main power circuit breakers and the memory power switch.

The logic panels on the door of bay 1 are special 21-unit mounting panels. These four panels contain the plug-in units for the sequence break system type 20. This arrangement permits the type 20 option to be installed without adding an extra bay to the computer. If no sequence break system type 20 is included, blank panels are provided instead of the mounting panels.

At the bottom of each bay in the central processor (bays 1, 2, and 3) are two type 728 power supplies. These supplies provide the +10 and -15 volts dc required by the plug-in units. (Each supply provides power for half the mounting panels in a bay.) Bay 1 also contains an extra type 728 supply to provide power to the logic panels at the bottom of the console bay. Note: in some machines, type 729 power supplies are used in place of the 728 supplies.

Two power supplies, type 728 or 742, are mounted at the bottom of the console door. These two supplies furnish the -15 vdc required by the indicator lights on the console panels. The two supplies are also connected in series to provide the -30 vdc required by the solenoids in the punch and the typewriter.

Marginal check power supply controls are located at the top of bay 2. Behind these controls is a variable power supply type 734. The output of this unit can be varied from 0 to 20 vdc, and can be applied to either the +10 or -15 volt lines in any logic panel for marginal checking the plug-in unit components.

Figure 2-5 shows the backs of the logic panels in bay 3. Part of the type 15 memory extension control is at the top. The plug-in units in the four panels below the type 15 are the logic circuits for the memory module. The third memory panel also contains the 4,096-word core bank. Below the memory are the in-out plugs; at the bottom is

in-out transfer control.

The inside of the bay 3 plenum door is shown in figure 2-6. At the bottom of this door are the two type 728 power supplies that provide power for the logic in the bay. Above the 728 supplies, is the memory power supply type 735. At the top of the door is the type 813 power control panel, containing delays and isolating circuits for the power switches. A similar panel is mounted at the top of the console plenum door to control the punch motor.

The following table lists the mounting panel and power equipment requirements for the standard PDP-1 and the central processor options. All mounting panels except those used in the 16-channel sequence break system are type 1914 19-inch panels. These panels can each hold 25 system plug-in modules. The figure numbers in the table refer to logic panel and power equipment layout drawings.

(1) Standard PDP-1

Space requirement	4 bays
Logic panels (figure 2-7)	
Control unit and arithmetic unit	28 mounting panels
One channel sequence break system	1 mounting panel (1L)
Memory module type 12	4 mounting panels (3A to 3D)
In-out transfer control	2 mounting panels (3H, 3J)
Reader, punch, typewriter control	3 mounting panels (11A to 11C)
Power equipment (figure 2-8)	7 power supplies 728 or 729
	1 variable power supply 734
	1 power supply 735
	2 power supplies 728 or 742
	1 power control panel 813
	1 power control panel 812
	1 marginal check switch panel

(2) Central Processor Options (only those which can be mounted in the central frame and require no extra power equipment, figure 2-7.)

Multiply/Divide Type 10	2 mounting panels (2K, 2L)
Memory Extension Control Type 15	3 mounting panels (2Y, 2Z, 3Y)
High Speed Channel Control Type 19	2 mounting panels (1Y, 1Z)
Sequence Break System Type 20 (figure 2-8)	Four 21-unit mounting panels type 1916, in rear of bay 1 (R1A to R1D)

(3) Additional Memory Modules Type 12

Logic panels	4 mounting panels
Power equipment	1 power supply 728
	1 power supply 735 Note: three memory modules may be placed in one bay. In this case only one 728 power supply is needed for the entire bay.

c MODULE LIST - The following list includes all the plug-in modules required by the standard PDP-1 and central processor options. For convenience the requirements for the type 12 memory module are listed separately from the rest of the standard central processor requirements. Following the list for the standard PDP-1 the various options are listed in order by type number.

(1) Central Processor - (includes standard in-out transfer control, but not the memory module. Numbers in parentheses indicate requirements for the one-channel sequence break system. These numbers should be added to the requirements for the standard central processor, unless a sequence break system type 20 is included in the computer.)

Type	Quantity	Type	Quantity
Inverter 1103	65	Inverter 1105	83(1)
Inverter 1103R	8	Diode 1110	6
Inverter 1104	46(1)	Diode 1111	10(1)

Type	Quantity
Binary-to-octal decoder 1150	9
Flip-flop 1201	32
Dual flip-flop 1204	19
Dual flip-flop 1209	17
Delay 1304	1
Delay line 1310	9
Delay line 1311	3
Pulse generator 1410	1
Pulse amplifier 1607	24
Bus driver 1684	11
Bus driver 1685	7
Inverter 4105	2
Inverter 4106	2
Inverter 4106R	1

Type	Quantity
Diode 4110	(2)
Diode 4112	1
Diode 4113	1
Diode 4113R	8
Capacitor-diode 4129	10*
Dual flip-flop 4209	(2)
Quadruple flip-flop 4214	1
Delay 4301	1(1)
Clock 4401	1
Pulse generator 4410	5
Pulse amplifier 4603	26(1)

(2) 4096 Word Memory Module Type 12

Inverter 1103R	7
Inverter 1104	1
Quadruple flip-flop 1213	1
Delay line 1310	1
Sense amplifier 1540	18
Pulse amplifier 1607	1
Inhibit driver 1982	5

Power supply control 1701 (in power supply 735)	1
Read/write switch 1972	32
Memory driver 1973	2
Resistor board 1976	16
Resistor board 1978	3

(3) Reader, Punch, Typewriter Control

Indicator driver 1669	5
Switch filter 1703	2

Inverter 4105	7
Inverter 4106	2

* If required for optional input-output equipment, a second row of nine additional 4129 plug-in units may be installed in the in-out input mixer.

Type	Quantity	Type	Quantity
Diode 4110	5	Quadruple flip-flop 4214	9
Diode 4111	1	Delay 4301	7
Diode 4113R	3	Pulse generator 4410	5
Capacitor-diode 4126	1	Pulse amplifier 4603	5
Capacitor-diode 4128	9	Solenoid driver 4680	6
Flip-flop 4201	2	Solenoid driver 4681	3
Dual flip-flop 4209	3		

(4) Multiply / Divide Type 10

Inverter 1105	10	Delay line 1310	3
Inverter 1104	1	Delay line 1311	7
Diode 1111	4	Pulse amplifier 1607	12
Flip-flop 1201	5	Diode 4113	2
Dual flip-flop 1209	1		

(5) Memory Extension Control Type 15

Inverter 1103	10	Dual flip-flop 1209	4
Inverter 1104	1	Pulse amplifier 1607	10*
Diode 1110	10	Bus driver 1684	13
Binary-to-octal decoder 1151	1		

(6) High Speed Channel Control Type 19

Inverter 1103	2	Pulse amplifier 1607	6
Inverter 1103R	1	Bus driver 1685	5
Inverter 1104	1	Inverter 4105	3
Inverter 1105	3	Capacitor-diode 4129	17
Dual flip-flop 1209	2		
Pulse amplifier 4603	10		

(7) Sequence Break System Type 20

Inverter 1103R	8	Inverter 1105	1
Inverter 1104	4	Diode 1111	8

* Add one 1607 for each additional memory module.

Type	Quantity	Type	Quantity
Inverter 4106	2	Quadruple flip-flop 4214	16
Diode 4113	4	Delay 4301	1
Capacitor-diode 4126	4	Pulse amplifier 4603	7
Capacitor-diode 4128	16		

CHAPTER 3

SYSTEM FUNCTION

3-1 LOGICAL ORGANIZATION

The logical configuration of the standard PDP-1 is shown in figure 3-1. The computer logic is divided into four parts: control unit, arithmetic unit, memory and input-output system.

a CONTROL UNIT - The control unit of the computer governs the timing of all computer operations, the information transfers within the central processor, and the operation of the various registers. The control unit includes three internal registers, IR, PC and MA, and two console switch registers, TA and TW.

The operation code of each instruction is decoded from the five-bit instruction register into one of 32 alternate command levels. These command levels govern the execution of the instruction. Each instruction word is retrieved from the memory location specified by the contents of the 12-bit program counter, PC. During the execution of each instruction the program counter is advanced one position; consecutive instructions are thus taken from consecutive memory locations. Every memory access is made to the location specified by the contents of the 12-bit memory address register, MA. The MA register is loaded from the program counter for instruction retrieval, and from the address portion of the instruction word (in the memory buffer register) for deferred address retrieval and for memory reference. The operator can manually provide addresses and data words for use by the computer through two switch registers on the console control panel: the address register, TA, and the test word register, TW.

In addition to these registers the control unit also includes six program flags, the console switches, a one-channel sequence break system, and the control logic. The control logic contains the timing system of the computer and all of those control flip-flops and logic nets which govern computer cycles, information transfers, and the operation of computer registers. All central processor options, except additional memory modules, are expansions of the basic control unit.

Most elements of the control unit are described in paragraphs 3-2 and 3-3 below. However, those elements of control that are directly associated with the arithmetic unit, the memory, and the in-out equipment are described in conjunction with those units (paragraphs 3-4, 3-5 and 3-6, respectively).

b ARITHMETIC UNIT - The arithmetic unit includes three 18-bit registers, the accumulator, the in-out register, and the memory buffer register. Each of these registers holds an entire 18-bit computer word.

The memory buffer register, MB, serves two distinct functions: a memory function, and an arithmetic function. All transfers of information between memory and the other parts of the computer are made through MB. The memory buffer is used in the arithmetic unit as a passive register, that is, it serves only to hold the operand in arithmetic and logical instructions. (The MB register takes an active role only in the optional automatic Divide instruction.)

The accumulator, AC, is the major register in the arithmetic unit. The accumulator input logic includes transfer, shift, logical, and arithmetic gating. The accumulator is used in the execution of all logical and arithmetic instructions. Furthermore, the result of all such instructions always appears in the accumulator. The accumulator input gating allows the computer to perform the logic functions AND, inclusive OR, exclusive OR, and negation, and the arithmetic operation of addition. All other arithmetic operations are performed by using combinations of negation and addition. The individual bits of the accumulator can be shifted in either direction; the ends of the accumulator can also be joined to produce a cyclic shift or rotation.

The transfer of information to and from the peripheral devices is made through the in-out register, IO. Moreover, this register also serves as the multiplier-quotient register in the arithmetic unit. During multiplication and division the in-out register serves as an 18-bit right-hand extension of the accumulator for double-length products and dividends. The individual bits of IO can be shifted or rotated in either direction and the combination of AC and IO can also be shifted or rotated as a single 36-bit register.

c MEMORY - The memory of the standard PDP-1 consists of one memory module, type 12.

The type 12 module is a magnetic-core coincident-current memory containing 4,096 eighteen-bit words. The memory of PDP-1 can be readily expanded by adding more type 12 memory modules. Such expansion, however, requires addition of a memory extension control type 15 to the control unit.

During each five-microsecond memory cycle access is made to the memory location specified by the contents of the 12-bit memory address register ($4,096 = 2^{12}$). The word read from memory is transferred into the memory buffer. Since the read operation is destructive, the word contained in the memory buffer must be written back into the core memory during the same cycle. In preparation for the deposit of new information in memory, the memory buffer is cleared after the read operation. The new information is then transferred into MB and written into the addressed memory location by the write portion of the memory cycle.

d INPUT-OUTPUT SYSTEM - The input-output system of the standard computer includes in-out transfer control, three input-output devices and the control units for these devices.

The three standard devices are a paper tape punch for output, a photoelectric tape reader for input, and a typewriter for both input and output.

The command level for instructions in the in-out transfer group is applied to in-out transfer control from the instruction decoder. In-out transfer control decodes the secondary op code (bits 12 to 17) into command pulses for the appropriate control unit. All transfers of data between the computer and the control-unit buffers are made through the in-out register, IO. Data transfers between a device and its associated control unit are usually performed automatically after receipt of a command pulse from in-out transfer control.

When optional input-output equipment is added to the computer, in-out transfer control must be expanded to provide the additional command pulses which are necessary. Like the standard in-out devices, each optional device includes its own control unit. The method of data transfer and the manner in which command pulses are used depends on the device and the type of control unit. For example, with the automatic tape control unit type 52, control information goes through the in-out register, but data is transferred directly between memory and the control unit through a high-speed channel.

3-2 PROGRAM EXECUTION

This paragraph describes the instruction repertoire of PDP-1 and those elements of control which govern the execution of the program.

a NUMBER SYSTEM - PDP-1 is a fixed-point machine using binary arithmetic. Negative numbers are represented as the 1's complements of the positive numbers. Bit 0 is the sign bit, which is 0 for positive numbers. Bits 1 through 17 are magnitude bits, bit 17 being the least significant. The actual position of the binary point may be arbitrarily assigned to best suit the problem at hand. Two common conventions in the placement of the point are:

The binary point is placed to the right of the least significant bit, thus numbers represent integers.

The binary point is placed to the right of the sign bit, thus numbers represent fractions between -1 and +1.

The conversion of decimal numbers into the binary system for use in the computer, and the output conversion of binary numbers into the decimal system, may be performed automatically by subroutines. Operations for floating-point numbers are handled by interpretive programming or subroutines. The utility program system provides for automatic insertion of the routines required to perform floating-point operations and number-base conversion.

b INSTRUCTION FORMAT - There are two classes of PDP-1 instructions: memory reference instructions and augmented instructions. Memory reference instructions need access to memory for an operand; they therefore require two memory cycles for their execution. Augmented instructions have no operand, and for this reason are performed in one memory cycle. In the augmented instructions, the bits used to address the operand in the memory reference instructions are instead used to augment the control capability of the instructions. The augmented instructions are thus those instructions having augmented operation codes.

There are also three instructions which fall into neither of these classes. These three instructions use their own address portions as operands and hence are similar in execution to the memory reference instructions. Because these three instructions require no actual access to memory, they are executed in a single cycle.

The two major classes of instructions, memory reference instructions and augmented instructions, can each be further subdivided into four groups. Memory reference instructions include arithmetic instructions, logical instructions, data handling instructions, and program control instructions. The augmented instructions include shift instructions, skip instructions, operate instructions, and in-out transfer instructions.

The three instructions which are neither memory reference instructions nor augmented instructions include one data handling instruction (law) and two program control instructions (jmp and jsp).

The instruction words for memory reference instructions require both an operation code and a memory address. The op code (bits 0 through 5) specifies the particular operation to be performed. The location in memory to which reference must be made is specified by the memory address portion, Y (bits 6 through 17).

Bit 5 of the op code is the indirect address bit. This bit is normally 0. If bit 5 is 1, the original address Y of the instruction is not used, as it usually is, to locate the operand, jump location, deposit location, etc. of the instruction. Instead, the address portion of the instruction is then used to locate a memory register that contains a new address. This new address is used in place of the address portion of the original instruction. This indirect addressing technique frequently expedites the programming task.

An instruction which uses an indirect address is called a "deferred" instruction because the actual operation which the instruction performs is deferred until the new address is retrieved from memory. Thus in a deferred instruction, Y is not the location of the operand, but the location of the location of the operand. If the memory register containing the new address also has a 1 in bit 5, the indirect addressing procedure is repeated and a third address is located. There is no limit to the number of times this process can be repeated in normal operation. However, if the computer includes a memory extension control and is operating in the extend mode, indirect addressing is limited to a single level.

Note that a deferable instruction requires two of the 64 operation codes. The code as given in the instruction list is an even number, that is, with 0 in the indirect address bit. The following odd number is the op code for the same instruction with indirect addressing, that is, with 1 in bit 5.

Augmented instruction words use the entire word as an operation code. Thus the entire class of augmented instructions uses only eight of the available 64 primary op codes to perform a very large number of instructions. The instructions under each pair of primary op codes are referred to as an instruction group. There are, for example, 12 variations of shift and rotate instructions in the shift group. The instruction words for shift/rotate operations use bits 0 through 8 as the operation code; the number of 1's in bits 9 through 17 determine the number of shift or rotate steps to be performed by the instruction.

In the skip group, a 0 indirect address bit indicates that the skip shall take place if the condition specified by the address portion of the instruction is satisfied. However, if the indirect address bit is 1, the skip occurs if the condition is not satisfied. In the operate group, the indirect address bit is ignored and bits 6 through 17 specify the particular function to be performed by the operate instruction.

In the in-out transfer group, the primary op code is six bits. In these instructions a secondary op code (bits 12 through 17) specifies the particular in-out transfer instruction while bits 6 through 11 provide various types of control information necessary for the execution of the instruction. If the indirect address bit is 1 the computer waits for the completion of the transfer before continuing the program.

c COMPUTER CYCLES - All instructions except the optional automatic Multiply and Divide are performed in multiples of the basic five-microsecond memory cycle. During cycle zero of each instruction, the instruction word is retrieved from memory. Augmented instructions are completed during this single memory cycle. Note however, that during cycle zero of an in-out transfer instruction the only function performed by the central processor is the generation of the appropriate command pulses. These pulses are applied to the device control unit. The actual transfer of information may occur much later. The computer can either continue the program and return to the operation at a later time or perform an in-out wait until the transfer is complete.

In memory reference instructions cycle zero is followed by cycle one. During cycle one, the operand is retrieved from or deposited in memory and the operations required by the instruction are completed.

There are, in addition to cycle zero and cycle one, several special cycles. If an instruction

is deferred at the completion of cycle zero the computer goes into a defer cycle. During this cycle the address portion of the instruction is used to retrieve from memory a new address for the instruction operand. The defer cycle may be repeated as many times as is required by the instruction (b above). After the required number of defer cycles has been completed, the computer proceeds into cycle one and completes the instruction.

There are also several special cycles which allow interruption of the program by in-out devices through the sequence break system or a high-speed channel. When a sequence break is initiated, the computer enters break cycle one. During this cycle, the contents of the accumulator are stored in a memory location fixed by the number of the channel through which the break is made.

Upon the completion of break cycle one the computer performs, in succession, break cycles two and three. In these cycles the contents of the program counter and the in-out register are deposited in the second and third succeeding memory locations. At the end of break cycle three the computer returns to cycle zero and performs the instruction contained in the fourth memory location.

When a high-speed in-out device requires access to memory the computer switches into the high-speed channel cycle. This cycle merely interrupts the program for one memory cycle while access is made to memory through a high-speed channel. After a word has been deposited in memory or retrieved from memory through one of the channels, the interrupted program is resumed.

d INSTRUCTION CONTROL - During each cycle zero an instruction word is read from memory into the memory buffer register. Bits 0 through 4 of the instruction are then transferred to the instruction register. There they are decoded to determine the operation code. The decoder has 32 outputs numbered in accordance with the instruction op codes. (The indirect address bit is taken as 0.) Thus the 32 output command levels from the decoder are numbered 00, 02, 04, and so on by even numbers through 76 (octal). For each specific op code in IR, only one of these 32 command levels is asserted. The single asserted command level gates the central processor control circuits that carry out required operations for the selected instruction.

Although bits 0 through 4 must be transferred to IR, the other operation bits of the

instruction are utilized directly from the memory buffer register. If the command level indicates an augmented instruction, bits MB_{5-17} are decoded to determine the type of shift, the number of shift steps, the type of in-out transfer, and so forth.

If the instruction is deferable, a 1 in MB_5 causes the defer flip-flop to be set. This causes the computer to go into the defer cycle and retrieve a new operand address. After this new address is transferred into MB the defer bit is again checked and, if it is 1, another defer cycle is executed. This process is repeated until a 0 defer bit occurs. Then the computer goes into cycle one and the address contained in MB_{6-17} is used for the memory reference. Of course, if a memory reference instruction is not deferred, the computer goes directly into cycle one from cycle zero.

There are several op codes which are not currently used for any instruction. If one of these unused op codes is selected, the computer halts at the end of cycle zero.

e PROGRAM CONTROL - At the beginning of each cycle zero the contents of the 12-bit program counter are transferred into the memory address register. The current instruction in the program is then retrieved from the memory location addressed by the contents of MA. After the address transfer the contents of the program counter are incremented by 1. This causes the next instruction to be taken from the succeeding memory location during the following cycle zero. Besides counting main program locations, the program counter also counts the memory locations used by the break cycles.

During a skip instruction, if the skip condition is satisfied the program counter is advanced one extra position. This causes the program to skip the next instruction in normal sequence. The skip instructions can sense the states of various registers and flip-flops of the program-controlled program flags, and of the operator-controlled sense switches.

The program counter can be counted down as well as up. This allows the counter to return to the same position during every cycle of an in-out wait, i.e., while waiting for an in-out transfer to be completed. The program counter is counted down by 1 to return to the beginning of an instruction when a sequence break or a high-speed channel break interrupts the program in the middle of an instruction. The computer must start the instruction over again after the break has been completed.

Program control is transferred to a new location by loading a new address into the program counter. The counter is loaded from the memory buffer on most program transfers; it is loaded from the address switch register for console operations; and it is loaded from the memory address register for sequence breaks and certain subroutine-calling transfers.

f CONSOLE CONTROL - The states of all central processor registers and control flip-flops are shown in indicator lights on the console. The console control panel also includes the switches through which the operator exercises control over the computer. These switches allow the operator to start and stop computer operations, to control the mode of operation, to specify program control information that can be sensed by the computer, and to specify test words and addresses to be used by the computer. The initiation of any operation from the console is timed by a chain of special pulses, SP_1 through SP_4. After the completion of this special pulse chain the regular memory-cycle timing system of the computer begins.

The computer has six modes of operation. In the normal program-running mode one memory cycle follows another without interruption until the computer is halted by either the program or the operator. If flip-flop rim is set the computer enters the read-in mode. In this mode, the computer performs the console operation Read In by alternating the special pulse chain and the memory cycle.

There are two manual modes: the single-cycle mode and the single-instruction mode. These are controlled by the SINGLE STEP and SINGLE INST switches. In the manual modes, operations are begun from the console in the normal manner, but the computer halts at the end of the first memory cycle (single step) or the end of the first complete instruction (single instruction). During any operation the computer must be in one and only one of the above four modes. In addition, there are two other modes which are not exclusive. These are the sequence-break mode and the extend mode.

While in the normal, single-cycle, or single-instruction mode the computer may also be in the sequence-break mode, if desired by the programmer or operator. When in the sequence-break mode (flip-flop sbm is 1) the normal program sequence may be broken by external signals through the sequence break system. The computer cannot be in the sequence-break mode while in the read-in mode.

If the computer includes a memory extension control, there is also a sixth mode of operation, the extend mode. While the computer is in any of the other five modes it may also be in the extend mode (flip-flop EXD is 1). This mode is controlled by the EXTEND switch In this mode indirect addressing is limited to one level, but a deferred address is interpreted as a 16-bit address instead of the usual 12-bit address. This allows the program to jump to another memory module or to retrieve an operand from another memory module.

Console control of computer operations is exercised through six operating switches. Five of these console functions initiate computer operations, the sixth halts the computer. The console functions are as follows:

Start

The computer starts normal operation in cycle zero. The first instruction is taken from the memory location addressed by the address switches. The START switch has two on positions. If the switch is pushed up the computer enters the sequence-break mode before starting; if the switch is pushed down the computer leaves the sequence-break mode. When the switch is pushed either way the computer also enters the extend mode if the EXTEND switch is on. Start also initiates the first cycle of operation in the manual modes.

Continue

The computer resumes normal operation at the state indicated by the console lights. Continue also initiates each cycle or instruction after the first in the manual modes. Note: This console operation cannot be used when the computer is in the read-in mode.

Examine

The contents of the memory register addressed by the address switches are displayed in the accumulator and memory buffer lights on the console.

Deposit

The test word indicated by the console switches is deposited in the memory location addressed by the address switches.

Read In

The computer enters the read-in mode and reads data from paper tape. Of each pair of words read the computer deposits the second word in the memory register specified by the address portion of the first word. At the completion of Read In the computer either halts or begins normal operation at a memory location specified by the address portion of the final word read from the

tape. If the EXTEND switch is on the computer enters the extend mode before beginning normal operation.

Stop

If in the normal mode the computer halts at the end of the current memory cycle. If performing Read In the computer halts after reading the first word of the current pair of words from the tape.

g INSTRUCTION LIST - This list includes the title of the instruction, the normal execution time (i.e., without indirect addressing), the mnemonic code, the operation code, and a short description of the instruction. In the following list the contents of a register are indicated by C( ). Thus C(Y) means the contents of memory location Y; C(AC) means the contents of the accumulator. A specific bit of a register is indicated by a subscript number following the symbol for the register. Thus IO_{17} represents bit 17 of the in-out register.

Deferable instructions are indicated by an asterisk (*). The operation code for these instructions is given with a 0 in the indirect address bit. When the instruction is deferred, 1 must be added to the given operation code.

Arithmetic Instructions (Note: none of the following instructions can result in an answer of -0.)

Add (10 μsec) add Y Operation Code 40*

The final C(AC) are the sum of C(Y) and the original C(AC); C(Y) are unchanged. The addition is performed with 1's complement arithmetic. If the sum exceeds the capacity of the accumulator, the overflow flip-flop is set.

Subtract (10 μsec) sub Y Operation Code 42*

The final C(AC) are the original C(AC) minus C(Y); C(Y) are unchanged. The subtraction is performed with 1's complement arithmetic. If the difference exceeds the capacity of the accumulator, the overflow flip-flop is set.

Multiply Step (10 μsec) mus Y Operation Code 54*

This instruction is used in the multiply subroutine. If IO_{17} is 1, C(Y) are added to C(AC); if IO_{17} is 0, the addition does not occur. In either case, C(AC) and C(IO) are shifted right one place. This shift clears AC_0.

Multiply (14 to 25 μsec) mul Y Operation Code 54*

(This instruction replaces mus Y if machine includes the automatic Multiply/Divide Type 10)

The product of C(AC) and C(Y) is formed in AC and IO. The sign of the product is in both AC_0 and IO_{17}. The magnitude of the product is the 34-bit string from AC_1 through IO_{16}. The C(Y) are not affected by the instruction.

Divide Step (10 μsec) dis Y Operation Code 56*

This instruction is used in the divide subroutine. The C(AC) and C(IO) are rotated left one place, with the complement of AC_0 replacing $C(IO_{17})$. If IO_{17} is 1, C(Y) are subtracted from C(AC). If IO_{17} is 0, C(Y) + 1 are added to C(AC).

Divide (30 to 40 μsec if division possible, otherwise 12 μsec) div Y Operation Code 56*

(This instruction replaces dis Y if machine includes the automatic Multiply/Divide Type 10). The dividend must be in AC and IO with IO_{0-16} forming a 17-bit magnitude extension of the accumulator. IO_{17} is ignored. The divisor is C(Y). At the completion of the instruction, C(AC) are the quotient and C(IO) are the remainder. The sign of the remainder is the sign of the dividend. If the division is performed, the next instruction in sequence is skipped. If the division is not possible, C(AC) and C(IO) are unchanged and the computer performs the next instruction in sequence. The C(Y) are not affected by the instruction.

Index (10 μsec) idx Y Operation Code 44*

The C(Y) are replaced by C(Y) + 1. The C(Y) + 1 are left in the accumulator. The previous C(AC) are lost. Overflow is not indicated.

Index and Skip if Positive (10 μsec) isp Y Operation Code 46*

The C(Y) are replaced by C(Y) + 1. The C(Y) + 1 are left in the accumulator. The previous C(AC) are lost. If, after the addition, C(Y) + 1 are positive, the program counter is advanced one extra position and the next instruction in sequence is skipped. Overflow is not indicated.

Logical Instructions

Logical AND (10 μsec) and Y Operation Code 02*

The bits of C(Y) operate on the corresponding bits of C(AC) to form the logical AND. The result is left in the accumulator. The C(Y) are unaffected.

Inclusive OR (10 μsec) ior Y Operation Code 04*

The bits of C(Y) operate on the corresponding bits of C(AC) to form the inclusive OR. The result is left in the accumulator. The C(Y) are unaffected.

Exclusive OR (10 μsec) xor Y Operation Code 06*

The bits of C(Y) operate on the corresponding bits of C(AC) to form the exclusive OR. The result is left in the accumulator. The C(Y) are unaffected.

Data Handling Instructions

Deposit Accumulator (10 μsec) dac Y Operation Code 24*

The C(AC) replace C(Y) in memory. The C(AC) are unchanged; the original C(Y) are lost.

Deposit Address Part (10 μsec) dap Y Operation Code 26*

Bits 6 through 17 of C(AC) replace the corresponding bits of C(Y). The C(AC) and $C(Y_{0-5})$ are unchanged; the original $C(Y_{6-17})$ are lost.

Deposit Instruction Part (10 μsec) dip Y Operation Code 30*

Bits 0 through 5 of C(AC) replace the corresponding bits of C(Y). The C(AC) and $C(Y_{6-17})$ are unchanged; the original $C(Y_{0-5})$ are lost.

Deposit In-out Register (10 μsec) dio Y Operation Code 32*

The C(IO) replace C(Y) in memory. The C(IO) are unchanged; the original C(Y) are lost.

Deposit Zero in Memory (10 μsec) dzm Y Operation Code 34*

The contents of memory location Y are replaced by zero (i.e. memory location Y is cleared).

Load Accumulator (10 μsec) lac Y Operation Code 20*

The C(Y) are placed in the accumulator. The C(Y) are unchanged; the original C(AC) are lost.

Load In-out Register (10 μsec) lio Y Operation Code 22*

The C(Y) are placed in the in-out register. The C(Y) are unchanged; the original C(IO) are lost.

Load Accumulator with N (5 μsec) law N Operation Code 70 law -N Operation Code 71

The number in the address portion of the instruction word is placed in the accumulator. If the indirect address bit is 1 (operation code 71), the complement of N (-N) is put in the accumulator.

Program Control Instructions

Jump (5 μsec) jmp Y Operation Code 60*

The address Y replaces C(PC). The next instruction in the program is then taken from

memory location Y. The original C(PC) are lost.

Jump and Save Program Counter (5 μsec) jsp Y Operation Code 62*
The C(PC) are transferred to the accumulator. When the transfer takes place, the program counter holds the address of the instruction following the jsp in normal sequence. The address Y then replaces C(PC) and the next instruction in the program is taken from memory location Y. The original C(AC) are lost.

Jump and Deposit Accumulator (10 μsec) jda Y Operation Code 17
The C(AC) are deposited in memory location Y. The C(PC) are then transferred to the accumulator and the program next executes the instruction in memory location Y + 1. This instruction is equivalent to dac Y, followed by jsp Y + 1.

Call Subroutine (10 μsec) cal Operation Code 16
Call Subroutine is equivalent to the instruction jda 100. The address portion, Y, is ignored. This instruction may be used as part of a master routine to call subroutines.

Skip if Accumulator and Y Differ (10 μsec) sad Y Operation Code 50*
The C(Y) are compared with the C(AC). If the two numbers are different, the program counter is indexed one extra position and the next instruction in sequence is skipped. The C(AC) and the C(Y) are unchanged.

Skip if Accumulator and Y are the Same (10 μsec) sas Y Operation Code 52*
The C(Y) are compared with the C(AC). If the two numbers are identical, the program counter is indexed one extra position and the next instruction in sequence is skipped. The C(AC) and C(Y) are unchanged.

Execute (5 μsec plus time of instruction executed) xct Y Operation Code 10*
The instruction in memory location Y is executed. The program counter remains unchanged (unless a jump or skip were executed). Execute acts exactly as though the instruction being executed replaced the Execute instruction in the program. Execute may be indirectly addressed, and the instruction being executed may also use indirect addressing. An xct instruction may execute other xct commands.

Augmented Instructions

Shift Group (5 μsec) sft Operation Codes 66, 67
This group of instructions rotates or shifts the accumulator and/or the in-out register.

When the two registers are combined the in-out register is considered to be an 18-bit magnitude extension of the accumulator.

Shift is an arithmetic operation and is, in effect, multiplication of the number in the register by $2^{\pm N}$, where N is the number of shifts. Plus is left and minus is right. Shift may also be considered as an information transfer from bit to bit in a register. Transfer of $C(AC_n)$ into AC_{n-1} is left; $C(AC_n)$ into AC_{n+1} is right. The sign bit is not affected and information shifted out of either end of the register is lost.

Rotate is a nonarithmetic, cyclic shift. That is, the two ends of the register are logically joined together and information is rotated as though the register were a ring. The sign bit is included and no information is lost.

The full operation codes of these instructions are actually nine bits. If the defer bit is 0 (op code 66) the bits are shifted to the left; if the defer bit is 1 (op code 67) the bits are shifted right. Bits 6 through 8 of the instruction are decoded to determine the type of operation and the register operated upon. If bit 6 is 0, a shift is performed; if 1, a rotate is performed. If bit 7 is 1, IO is affected; if bit 8 is 1, AC is affected. If bits 7 and 8 are both 1, the instruction acts on the combination of the registers.

The number, N, of shift or rotate steps performed is determined by the number of 1's in bits 9 through 17 of the instruction word. Thus, Rotate Accumulator Right nine times is 671777. A shift or rotate of one place can be indicated nine different ways. The usual convention is to use the right end of the instruction word (rar 1 = 671001).

In the following list the titles of the instructions describe the operations performed. All instructions require five microseconds and the bits are shifted N positions. The full operation codes are given.

Rotate Accumulator Left, ral N Operation Code 661
Rotate In-out Register Left, ril N Operation Code 662
Rotate Combined AC and IO Left, rcl N Operation Code 663
Shift Accumulator Left, sal N Operation Code 665
Shift In-out Register Left, sil N Operation Code 666
Shift Combined AC and IO Left, scl N Operation Code 667
Rotate Accumulator Right, rar N Operation 671

Rotate In-out Register Right, rir N Operation Code 672

Rotate Combined AC and IO Right, rcr N Operation Code 673

Shift Accumulator Right, sar N Operation Code 675

Shift In-out Register Right, sir N Operation Code 676

Shift Combined AC and IO Right, scr N Operation Code 677

Skip Group (5 μsec) skp Operation Codes 64, 65

This group of instructions senses the states of various flip-flops and switches in the machine. The address portion of the instruction selects the particluar function to be sensed. A zero address is no selection. All instructions in the group have the same operation code.

The addresses in the skip group (except those which sense program flags or sense switches) may be combined to form the union of the separate skips. Thus, if address 3000 is selected, the skip would occur if the overflow flip-flop is 0 or if C(IO) are positive. The combined instruction still requires only five microseconds.

The skip instructions as listed below use operation code 64. The intent of any instruction can be reversed by using op code 65 (changing bit 5 to 1). For example, the instruction 640100 is Skip on Zero Accumulator; while 650100 is Skip on Nonzero Accumulator. This also reverses the intent of an instruction with zero address. No selection with op code 64 is a nop, but no selection with op code 65 is an absolute skip.

Skip on Zero Accumulator (5 μsec) sza Address 100

Skip if accumulator contains plus zero (i. e., all bits are 0).

Skip on Plus Accumulator (5 μsec) spa Address 200

Skip if AC_0 is 0.

Skip on Minus Accumulator (5 μsec) sma Address 400

Skip if AC_0 is 1.

Skip on Zero Overflow (5 μsec) szo Address 1000

Skip if overflow flip-flop (OV_1) is 0. This instruction also clears OV_1. Since OV_1 is not cleared by an arithmetic operation that does not cause an overflow, a whole series of additions and subtractions can be checked for correctness by a single szo.

Skip on Plus In-out Register (5 μsec) spi Address 2000

Skip if IO_0 is 0.

Skip on Zero Sense Switch (5 μsec) szs Addresses 10, 20, . . ., 70
Skip if selected sense switch is 0. Address 10 senses the position of sense switch 1, etc. If address 70 is selected all the switches are sensed and all six must be 0 to cause the skip.

Skip on Zero Program Flag (5 μsec) szf Address 1, 2, . . . , 7
Skip if selected program flag is 0. Address 1 senses the state of program flag 1, etc. If address 7 is selected all the flags are sensed and all six must be 0 to cause the skip.

Operate Group (5 μsec) opr Operation Code 76
This instruction group performs miscellaneous operations on various central processor registers and flip-flops. The address portion of the instruction specifies the operation to be performed.

The addresses of the operate instructions may be combined to form the union of the functions. The instruction opr 3200 clears AC, transfers TW to AC, and complements AC. If the number -0 (i.e., all 1's) is interpreted as an instruction, IO and AC are cleared, C(TW) and C(PC) are simultaneously transferred into AC (forming the inclusive OR of the two words), AC is complemented, all program flags are set and the computer halts.

The individual instructions are listed in the order in which the specific operations would occur in cycle zero.

No Operation (5 μsec) nop Address 0000
The state of the computer is unaffected by this operation, and the program counter continues in sequence.

Clear Accumulator (5 μsec) cla Address 200

Clear In-out Register (5 μsec) cli Address 4000

Load Accumulator from Test Word (5 μsec) lat Address 2200
The C(TW) are placed in the accumulator. The original C(AC) are lost. Note: this instruction results from the combination of addresses 200 (cla) and 2000. Programming opr 2000 alone, produces the inclusive OR of C(AC) and C(TW).

Load Accumulator from Program Counter (5 μsec) lap Address 300
The C(PC) are placed in the accumulator. The original C(AC) are lost. Note: this instruction results from the combination of addresses 200 (cla) and 100. Programming

opr 100 alone, produces the inclusive OR of C(AC) and C(PC).

Clear Selected Program Flag (5 μsec) clf Addresses 01 to 07 inclusive
Address 01 clears program flag 1, etc. Address 07 clears all program flags.

Set Selected Program Flag (5 μsec) stf Addresses 11 to 17 inclusive
Address 11 sets program flag 1, etc. Address 17 sets all program flags.

Complement Accumulator (5 μsec) cma Address 1000

Halt (5 μsec) hlt Address 400
Stops the computer.

In-out Transfer Group (5 μsec without in-out wait) iot Operation Codes 72, 73
The in-out transfer group instruction words include two separate operation codes. The primary operation code, either 72 or 73, is decoded at the instruction register in the usual manner. When the instruction decoder asserts the command level for the iot instruction group, bits 12 through 17 of the instruction word are also decoded. In-out transfer control decodes these six bits as a second operation code. This secondary op code is decoded into command pulses.

Bits 7 through 11 or, in some cases, 6 through 11 can be used to form subfunctions within the second op code, or even to form a third op code. E.g., in controlling magnetic tapes these bits may be used to indicate forward/backward, read/write, etc., or to specify one of several similar units.

In this way, a very large number of in-out transfer group instructions can be included under a single command level. These instructions govern all control functions and information transfers between the computer and in-out devices. Furthermore, some of the iot instructions govern central processor elements, such as the sequence break system and the memory extension control.

Completion of most in-out operations requires a definite minimum time. This minimum time is usually very long compared to a computer memory cycle. In those in-out instructions which require a minimum time longer than one memory cycle, the computer and the in-out device must be resynchronized. The resynchronization is provided by a completion pulse from the device. The control of the in-out wait and of the completion

pulse is exercised by bits 5 and 6 of the iot instruction word. If an in-out transfer instruction is given with a 0 in bit 5 (i.e. op code 72), the computer generates the appropriate command pulses in one memory cycle, and then continues with the normal sequence. If, however, bit 5 is a 1 (i.e. op code 73), then the computer goes into an in-out wait cycle and does not continue with the normal sequence until a completion pulse is received.

Bit 6 of the instruction word is used by the programmer to specify whether or not a completion pulse is necessary. If bit 6 is different from bit 5, then the completion pulse from the device control unit must be sent into the computer control logic. If bit 6 is the same as bit 5, then no completion pulse is necessary, and none is received.

All four combinations of requesting an in-out wait and requesting a completion pulse are used in in-out transfer instructions. This is because the known minimum time for completion of an in-out operation may be utilized for execution of other instructions in the program. Should input information be immediately available (as for example when a typewriter key has been struck), then the in-out transfer instruction requires neither a wait nor a completion pulse. In this case, bit 5 is 0 (op code 72) and bit 6 is also 0. Note that the same programming of bits 5 and 6 must also be used with any iot instruction that controls a central processor function. These iot instructions are always completed in a single memory cycle, and need neither a wait nor a completion pulse.

If the programmer does not wish to utilize the in-out wait time for executing additional instructions, he can program the iot instruction to start the in-out wait immediately. In this case the instruction is programmed with bit 5 a 1 (op code 73) and with bit 6 a 0. The 1 in bit 5 sets the in-out halt flip-flop, ioh. When ioh is 1, the computer repeatedly executes in-out wait cycles. Each wait cycle performs the iot instruction with no command pulses. The command pulses are produced only during the first cycle while the in-out commands flip-flop ioc is 1. At the end of the first cycle ioc is cleared so that no further commands are produced. The 0 in bit 6 indicates to the device control unit that a completion pulse is necessary. When the completion pulse arrives, the in-out synchronizer flip-flop, ios, is set. Setting this flip-flop clears ioh, causing the computer to return to the normal sequence.

If the programmer wishes to utilize the in-out wait time for executing additional instructions, he may either use the sequence break system or he can program the iot instruction with bit 5 a 0 (op code 72) and bit 6 a 1. This method of programming does not halt program operations, but the 1 in bit 6 does require in-out transfer control to provide a completion pulse. The computer can then continue with a normal sequence of instructions.

This sequence must, however, include an iot instruction 730000. This instruction performs nothing but the in-out wait. There must be one 730000 instruction for each iot that requests a completion pulse without initiating an in-out wait. If the device completion has occurred before the 730000 appears in the program, the 730000 is interpreted as a nop. If the 730000 occurs first, the computer enters the in-out wait until the completion pulse is received.

Because instruction 730000 has no secondary op code, no command pulses can be generated and the only effect of the instruction is to set flip-flop ioh. The state of bit 6 does not matter since there are no command pulses generated, and hence no need-a-completion-pulse signal is applied to any control unit. However, the programmer must never use 11 in bits 5 and 6 of any iot that includes a second op code. Such an instruction would begin an in-out wait without requesting a completion pulse, causing the computer to hang up.

In addition to control bits 5 and 6 of the instruction word, bits 7 through 11 may be used as control bits to extend the capabilities of the iot instructions. In central processor control instructions bits 6 through 11 may be used to address sequence break channels. For in-out instructions bits 7 through 11 may be used to vary the instruction (bits 6 through 11 are available if the device cannot utilize the completion pulse logic). For example, if there are several typewriters, the secondary op code may specify a Type In instruction, while the extra bits may address a specific typewriter.

The following list includes the central processor control instructions and the in-out instructions for the standard in-out equipment. In-out instructions for the optional equipment are included in the supplements to this manual that describe the optional

in-out devices (see also table 9-2). If a given instruction utilizes no in-out wait and no completion pulse then the entire instruction word is given. If the use of bits 5 and 6 may be varied at the discretion of the programmer, then only the secondary op code is given. Whenever an "X" appears in the instruction code the corresponding octal digit is ignored and is available for optional use.

Instructions Governing Standard In-Out Equipment

Read Punched Tape, Alphanumeric (2.5 ms) rpa Secondary Operation Code 01
All eight holes of a single line on the tape are read into the reader buffer. The completion pulse then transfers the information into IO_{10-17}.

Read Punched Tape, Binary (7.5 ms) rpb Secondary Operation Code 02
Three lines on the tape are read and assembled into a full computer word in the reader buffer. A line is recognized in the binary mode only if the eighth hole is punched; i.e. lines with no eighth hole are skipped. The seventh hole is ignored. The completion pulse transfers the word from the buffer to the in-out register.

Read Reader Buffer (5 μsec) rrb Secondary Operation Code 30
If no completion pulse has been requested in an rpa or rpb instruction, the automatic transfer of information from the buffer to the in-out register does not take place. To effect this transfer these instructions must be followed by an rrb instruction.

Read-In Mode
This is a special mode of operation initiated by the READ IN switch on the console. It provides a means of entering programs which require neither a stored program nor a plug board. When the READ IN switch is operated the computer enters the read-in mode and then starts the reader, which operates in the binary mode. The first group of three lines, and alternate groups of three lines, are interpreted as instructions.

Even-numbered groups of three lines are data. The instructions must be either Deposit In-out (dio Y) or Jump (jmp Y). If the instruction is dio Y, the next group of three lines are stored in memory location Y and the reader continues. If the instruction is jmp Y, the computer leaves the read-in mode and begins normal operation at memory location Y.

Punch Paper Tape, Alphanumeric (5.0 to 15.8 ms) ppa Secondary Operation Code 05
One line of tape is punched according to $C(IO_{10-17})$. If IO_{17} is 1, hole 1 is punched; if IO_{16} is 1, hole 2 is punched; and so on to IO_{10} which controls the punching of hole 8. The time required to punch is 5.0 milliseconds. The time between lines is 15.8 milliseconds. If a punch instruction follows immediately after a completion pulse, 15.8 milliseconds are available for the program.

Punch Paper Tape, Binary (5.0 to 15.8 ms) ppb Secondary Operation Code 06
One line of tape is punched according to $C(IO_{0-5})$. If IO_5 is 1, hole 1 is punched; if IO_4 is 1, hole 2 is punched; and so on to IO_0 which controls the punching of hole 6. The time required to punch is 5.0 milliseconds; the time between lines is 15.8 milliseconds. If a punch instruction follows immediately after a completion pulse, 15.8 milliseconds are available for the program.

Type Out (105 ms) tyo Secondary Operation Code 03
Types the single character specified by $C(IO_{12-17})$.

Type In (5 μsec) tyi Instruction Code 720X04
When a typewriter key is struck, the encoded character is loaded into the typewriter buffer. At the same time both program flag 1 and the typewriter buffer status bit are set. The program must check either the flag or the status bit to determine when a key has been struck. The instruction Type In clears both the in-out register and the status bit, and transfers the character into IO_{12-17}.

Instructions Governing Central Processor Elements

The first four of the following instructions are included in the standard computer; the others are added only if they are required for optional equipment. All instructions take five microseconds.

Enter Sequence Break Mode, esm Instruction Code 72XX55
Sets flip-flop sbm, allowing the main sequence to be interrupted through the sequence break system.

Leave Sequence Break Mode, lsm Instruction Code 72XX54
Clears flip-flop sbm, preventing interruptions of the main sequence through the sequence break system.

Clear Sequence Break System, cbs Instruction Code 72XX56
Clears control flip-flops in the sequence break system.

Check Status, cks Instruction Code 72XX33
This instruction loads the status bits of various in-out devices into specific bits of the in-out register. The instruction is expanded as required by optional equipment. In the standard machine cks checks five status bits as follows:

IO Bit	If Set
0	Displayed point sensed by light pen.
1	Tape reader buffer has been loaded but has not yet been read by an rrb.
2	Typewriter ready to Type Out.
3	Typewriter key struck and not yet read by a tyi.
4	Tape punch ready for output.

Deactivate Sequence Break Channel kn dsc Instruction Code 72kn50
Turns off channel kn in type 20 sequence break system.

Activate Sequence Break Channel kn asc Instruction Code 72kn51
Turns on channel kn in type 20 sequence break system.

Initiate Sequence Break Channel kn isb Instruction Code 72kn52
Initiates a sequence break on channel kn in type 20 sequence break system. (The break is initiated regardless of whether or not channel kn is on.)

Clear All Channels, cac Instruction Code 72XX53
Turns off all channels in type 20 sequence break system.

Enter Extend Mode, eem Instruction Code 724X74
Sets flip-flop EXD so that deferred addresses are interpreted as extended addresses.

Leave Extend Mode, lem Instruction Code 720X74
Clears flip-flop EXD, confining operation to a single memory module.

h TERMINOLOGY - Throughout this manual reference is often made to the various instructions by their instruction codes. When these codes are written in lower case (such as dac, dio, rpa, hlt) they always refer to the instructions as elements of a program. However, the same codes written in upper case frequently designate the names of levels and pulses in the computer logic.

For memory reference instructions, the upper-case codes refer to the op codes in the instruction register and to the command levels asserted from the instruction decoder. For example, when the computer performs the instruction dac, the op code DAC (010 10X) is in the instruction register and the instruction decoder asserts the command level DAC. The only exceptions to this convention are the instructions whose op codes differ in only the indirect address bit. There are no separate command levels JDA and CAL. The single command level JDA + CAL is asserted for either of the instructions jda or cal.

For the augmented instructions the situation is quite different. The instruction codes for the individual instructions in the shift, skip, and operate groups do not appear as control levels at all. The command levels for the groups sft, skp, and opr are SH/RO., SKP and OPR. The complete command level for any instruction in a group is a combination of the group command level and various other logical conditions. For example, the instruction Halt (hlt) corresponds to the computer logic level $OPR \cdot MB_9^1$. The instruction Rotate Accumulator Right (rar) corresponds to the shift/rotate pulses AC SH/RO R and the two levels AC ONLY and AC ROTATE.

For the in-out transfer instructions the group command level IOT causes in-out transfer control to decode the secondary op code (bits 12 through 17). This secondary op code is decoded into one or two command pulses. The name of the main command pulse is always the same as the code for the individual instruction. For example, in the instruction Type Out (tyo), the secondary op code is decoded into two successive command pulses, first a preparatory pulse, and then the main pulse. The first pulse, which clears the typewriter buffer, is labelled ^{0} TB. The main pulse , TYO, both loads a character from the in-out register into the buffer and also initiates the operation of the typewriter control unit.

3-3 CONTROL

This paragraph describes those elements of the control unit which are not discussed under program execution (paragraph 3-2). These control elements include the timing system, the cycle control, and the transfer logic. Also included are two central processor options, the sequence break system and the high-speed channel control. The symbols used in the drawings to represent the logic elements are also described.

a TIMING SYSTEM - The fundamental timing system of the computer is based upon a sequence of twelve timing pulses called the timing chain. The pulses are designated TP_0, TP_1, . . . , TP_9, TP_{9a}, and TP_{10}. These timing pulses occur at irregular intervals

chosen so as to optimize memory operations. There is no repetitive standard clock pulse providing a fundamental time unit for the computer. The twelve timing pulses follow each other in a chain covering one memory cycle of five microseconds. Each timing pulse is triggered through a delay from the previous timing pulse. The transition from one cycle to the next is controlled by flip-flop run. If run is 1, the final timing pulse in one cycle triggers the first timing pulse for the beginning of the next cycle. When run is cleared, the computer halts at the end of the current memory cycle because the final pulse in the cycle cannot trigger the first pulse of the following cycle.

The organization of a single memory cycle is shown in figure 3-2. The twelve irregularly spaced timing pulses are shown from left to right across the five-microsecond cycle. During each such cycle a single memory access is executed. The specific actions performed at each timing pulse depend upon the particular operation in which the memory cycle occurs.

The functions that control the actual memory access during each cycle are also shown in the figure. Each memory module contains four flip-flops which control a set of four functions. Three of these, the read, inhibit and write functions, are levels that control core driving. These functions are shown by the horizontal lines in the figure. The fourth function, the strobe, is a pulse that samples the output at the core-memory sense amplifiers approximately one microsecond after the initiation of the read function.

b CYCLE CONTROL - The primary cycle control element is the cycle flip-flop, cyc. When this flip-flop is 0, the computer is in cycle zero. For all other cycles the cycle flip-flop must be 1. If cyc is 1, and the high-speed channel flip-flop HSC_0 is also 1, then the computer executes a high-speed channel cycle. If cyc is 1 and the defer flip-flop df_1 is also 1, then the computer is in a defer cycle (which occurs between cycle zero and cycle one). During a defer cycle, if a second defer flip-flop df_2 is set, the computer executes another defer cycle.

If cyc is 1 and the break counter (flip-flops bc_1 and bc_2) contains any number other than 0, then the computer is in one of the break cycles. The particular break cycle depends upon the contents of the break counter. When the break counter contains 1, break cycle one is executed. Similarly when the break counter contains 2 or 3, break

cycle two or three, respectively, is executed. If, however, cyc is 1, and no special cycle (that is, no break cycle, high-speed channel cycle, or defer cycle) is being executed, then the computer is in cycle one. During cycle one the specific operations necessary for the second cycle of a memory reference instruction are executed.

c LOGIC SYMBOLS - The symbols used on the logic drawings are shown in figure 3-3. Note that in the rectangle which represents a flip-flop the 0-out terminal E and the 1-out terminal F are shown twice. Over the "0" the two terminals are shown with the polarities they have when the flip-flop is in the 0 state; over the "1" the two terminals are shown with the polarities they have when the flip-flop is in the 1 state. Therefore, the "0" and "1" in the rectangle represent both the output terminals and the contents of the flip-flop. In the normal convention the "0" is at the left of the rectangle and the 0-out terminal is represented by the left diamond in both pairs.

The two gatable inputs are shown at the bottom of the rectangle with the 0-in terminal at the left. Ungatable direct pulse inputs are always shown at the sides of a flip-flop. In the example in the figure a direct clear input is shown at the left. If the flip-flop had a direct set input, it would be shown at the right.

Some flip-flops also have complement inputs. Such inputs may be either direct negative-pulse inputs or gatable positive-pulse inputs and are always shown at the bottom center of the rectangle. If the complement pulse produces a carry pulse out of the flip-flop, the pulse output is shown at the top center of the rectangle.

If the flip-flop convention must be reversed the pin designations and the diamonds remain the same. However, in this case the "1" appears at the left of the rectangle and the terminals which were previously the 0-out, 0-in and direct-clear terminals become the 1-out, 1-in and direct-set terminals (i.e., all terminal names shown on the circuit schematic of the flip-flop are reversed). Similarly, the terminals previously designated as the 1 terminals become the 0 terminals. (In the standard computer this reversed convention is used only in the in-out control units.)

The principal advantage of having four logical outputs to represent two output terminals at two assertion levels is that there is never any need to invert a signal name which appears as an input to a logic net. Even though the computer uses inverter logic, all logical conditions appear in the drawings with correct truth values. When a flip-flop output is used

as the input to a logic net, the signal name indicates the correct state of the flip-flop that enables the net.

To determine the physical source of the signal (i.e. the output terminal to which the signal line is connected) one must consider both the signal name and the assertion level. For example, the signal A^1 at the negative assertion level actually originates at the 1-out terminal of flip-flop A; the signal A^1 at the ground assertion level actually originates at the 0-out terminal of flip-flop A. The signal designation A^1 can thus refer to the output signal generated at either terminal of flip-flop A, when that flip-flop is in the 1 state.

d TRANSFER LOGIC - All information transfers must take place between two information-storing devices. In most cases the transfer is made from one flip-flop register to another. The bit of information contained in the specific flip-flop of the source register is transferred to a corresponding flip-flop in the receiving register.

In memory access, information is transferred between the memory buffer flip-flops and the ferrite cores within the memory core-bank registers. In shift/rotate operations information is transferred from one flip-flop to another in the same register. Several bits of information may be provided to a register by setting a single source flip-flop if the source flip-flop is one of a set of flip-flops and represents one of a number of possibilities. In this case information from the source flip-flop must be applied to the receiving register through an encoder.

The operation of a 1 transfer is shown in figure 3-4A. In the example shown, the 1's in register K are transferred to the corresponding flip-flops in register M. A clear pulse, $\xrightarrow{0}$M, clears every flip-flop in M and then the transfer pulse, K$\xrightarrow{1}$M, sets each bit M_n if the corresponding bit K_n is 1. After both pulses have occurred C(M) = C(K). Note that if a 1 transfer is executed without a prior clear the final contents of M equal the inclusive OR function of the previous contents of M and the contents of K. That is, after the pulse K$\xrightarrow{1}$M a bit M_n is 1 if M_n was already 1 or if K_n is 1.

The transfer pulse is applied to all the bits in the receiving register except in special cases where less than a full-register transfer is desired. For example, the instruction part of a word may be transferred from K to M by the pulse K$\xrightarrow[0-5]{1}$M while the address

part may be transferred by the pulse $K \xrightarrow[6-17]{1} M$. The transfer pulse occurs at a specific timing pulse either in all memory cycles, in a specific type of cycle, or in a specific instruction. The 1 state of K_n is applied to the AND-input gate of M_n at the ground assertion level. This means that the origin of the flip-flop output level is the 0-out terminal of K_n. A jam transfer is a combination of a 1 transfer and a 0 transfer. A jam transfer is shown in figure 3-4B. No clear pulse is required in this case, because M_n is set if K_n is 1 and M_n is cleared if K_n is 0. The pulse $K \xrightarrow{i} M$ is logically equivalent to the two pulses $K \xrightarrow{1} M$ and $K \xrightarrow{0} M$ executed simultaneously. Note that a shift is a jam transfer from one bit to another of the same register. In a left shift information is transferred from M_n to M_{n-1}; in a right shift information is transferred from M_n to M_{n+1}.

If a register receives information from several different sources, a mixer may be used. A capacitor-diode gate mixer (figure 3-4C) is used whenever information can be transferred into a single central processor register from several different peripheral control units. Each bit of the mixer is composed of a pulse amplifier and a set of gates. Each bit of a source register is applied to one of the gates in the corresponding bit of the mixer. The transfer pulse for a given source register is applied to the corresponding gate in every bit of the mixer.

Each pulse output MM_n, of the register M mixer, sets the corresponding bit, M_n, of the receiving register. A transfer through a mixer is therefore a 1 transfer and the receiving register must be cleared before the transfer is made. No gating occurs at the receiving register. All gating is effected at the mixer. If a negative pulse output is used the emitter of the receiving register input transistor is grounded. If the flip-flops of the receiving register have direct set inputs, the mixer pulse amplifiers can be connected to produce a positive pulse. The receiving register then requires no input transistors.

If the output of the source is already in pulse form (e.g. the output of the sense amplifiers in memory) several sources can be mixed by a diode OR gate. A pulse from a specific bit of any source is then passed directly to the corresponding bit of the receiving register.

e SEQUENCE BREAK SYSTEM - Two sequence break systems are regularly available

with PDP-1. The standard computer includes a one-channel sequence break system. If the type 20 sixteen-channel break system is installed in the computer the one-channel system is removed. Both systems allow signals from in-out devices to interrupt the normal program sequence. Both systems are controlled by in-out transfer instructions in class 50. The standard computer includes three such instructions. For the optional system the number of instructions is expanded to seven. Two of the standard instructions control the sequence break mode; the third clears the sequence break system.

When a break is requested by either system, the request is usually granted for the next memory cycle. If the computer is in the sequence break mode (i.e. if flip-flop sbm is 1). The high-speed channels, however, have priority over the sequence break system. If a sequence break request is made at the same time as a high-speed channel break request, all high-speed channel interruptions are handled by the computer before the sequence break request is granted.

After a sequence break request is granted, the break counter controls the execution of the three break cycles. During these three cycles the contents of the accumulator, the program counter and the in-out register are deposited in three consecutive memory locations. During break cycle two the contents of OV_1 and (if present) EXD and EPC are saved along with PC. This stores all the information that the computer needs to return to the main sequence after execution of the break subroutine. After the computer completes break cycle three, it executes the instruction in the fourth consecutive memory location (the location following the deposit location of the in-out register). The instruction in this location is usually a jump to the subroutine appropriate to the break.

The break signal from a device to the one-channel break system may be either a pulse or a level. After the break signal is received, the break system is synchronized to the computer timing system and a break is requested. When the request is granted the computer breaks to memory location 0. The current program information is stored in memory locations 0, 1, and 2, and the computer then executes the instruction in memory location 3.

The type 20 sequence break system allows the same type of break in the program as the one-channel system, but this is done with 16 separate channels arranged in a priority

chain. The channels are numbered 0 through 17 octal. Each channel is controlled by four flip-flops. These are the Channel On, Synchronizer, Waiting Break, and Break Started flip-flops.

Program control over the 16-channel system is exercised by the three standard instructions plus four additional instructions. These additional instructions allow the programmer to turn any channel on or off, to initiate a break on any channel, or to turn off all channels.

The break signal from a device to the 16-channel system must be a pulse. If channel n is on when a break signal for channel n arrives, a break is initiated and synchronized to the computer timing system and channel n is then "waiting break". If channel n has priority (that is, if no higher-priority channel is waiting for or holding a break) then a break request is made for channel n. After the request is granted the break started flip-flop is set preventing any further break on channel n, and also preventing break requests on any lower-priority channel.

A sequence break for any channel is made to the memory location specified by the break encoder. Each channel uses four memory locations for a break. A break on channel 0 is made to memory location 0; a break on channel 1 is made to memory location 4. The memory location of a specific break is four times the number of the channel. Channel 2 breaks to memory location 10, channel 3 to memory location 14, and so on through channel 17 which breaks to memory location 74.

f HIGH-SPEED CHANNEL CONTROL - For high-speed in-out devices such as magnetic tape or data channel, a high-speed channel option is installed in the computer. This option provides three channels which allow a device control unit to gain access directly to the memory buffer without transferring information through the in-out register. There are three high-speed channels arranged in a priority chain. The channels are prewired to individual in-out device control units.

When a high-speed channel break request is made the break is allowed for the following memory cycle provided no break request is being made on any higher priority channel. The break is enabled by setting flip-flop HSC_0. When this flip-flop is 1 the computer goes into a high-speed channel cycle and the main program waits for

one cycle before resuming. During the high-speed channel cycle a memory address is transferred through the high-speed address mixer into the memory address register. If access is desired in order to transfer information into the computer, a word is then transferred through the high-speed buffer mixer into the memory buffer. If the access is being made to retrieve a word from memory, a word transfer is made to the device control unit after the regular read-out from memory into the memory buffer.

After the high-speed channel access has been completed, flip-flop HSC_0 is cleared and the computer enters cycle zero to retrieve the next instruction in the main program.

3-4 ARITHMETIC UNIT

The arithmetic unit includes three registers and their associated control circuits. The arithmetic unit may also include an automatic multiply/divide control as an option. This control allows the computer to perform multiplication and division as single instructions instead of subroutines.

The three registers in the arithmetic unit are the accumulator, AC, the in-out register, IO, and the memory buffer, MB. The memory buffer is a passive register in standard computer operations. That is, MB holds the operand in all two-term arithmetic and logical instructions. The outputs of the flip-flops in MB are applied to the input gating of the accumulator, but the contents of MB are not affected by the operation. Only in the automatic Multiply and Divide instructions can MB be complemented.

The accumulator is used in all arithmetic unit operations; the in-out register serves as the multiplier-quotient register. Both registers have transfer and shift gates; only the accumulator has arithmetic and indexing capability. The two registers can be shifted or rotated separately and the combination of the registers can also be shifted or rotated.

In addition, the contents of MB and AC are added together; in subtraction, the contents of MB are subtracted from the contents of AC. In either case the result appears in AC. In multiplication MB holds the multiplicand; the multiplier controls the operation from IO; and the double-length product appears in AC and IO. In division MB holds the divisor; AC and IO hold the double-length dividend; at the end of the operation the quotient is in IO and the remainder is in AC.

a ACCUMULATOR - The accumulator, AC, is the major register in the arithmetic unit. The computer can perform the logic functions AND, inclusive OR, and exclusive OR on the contents of AC and MB. The result appears in AC. For logical negation the contents of AC are complemented directly by an operate group instruction. In index instructions the number in the accumulator is indexed by an add-1-to-AC pulse; no other register is required.

Addition, the basic arithmetic operation in the computer, is performed in two steps, a partial add and a carry. The partial add is the logical function exclusive OR. All other arithmetic operations are executed by combining addition and logical negation.

In 1's complement arithmetic the logical complement of a number is equivalent to the arithmetic negative. Thus subtraction is performed by adding the negative. In the standard computer only the accumulator can be complemented so the complement pulse is required twice for subtraction. First AC is complemented, producing -C(AC). Then addition results in C(MB)-C(AC) in the accumulator. Finally a second complement pulse produces the desired result: C(AC)-C(MB).

Multiplication is performed by successive additions of the multiplier (MB) to form partial products in AC. The double-length product is produced by right-shifting AC into IO as the multiplier bits are dropped from IO.

Division is performed by successive subtractions of the divisor (MB) from the double-length dividend (AC and IO). As the dividend is shifted out of AC to the left, the quotient is constructed in IO from the right. At the end of the operation the remainder is left in AC.

In the standard computer, multiplication and division are performed by subroutines. The instructions Multiply Step and Divide Step perform only one step of the corresponding operation. The steps must be counted by indexing. If the automatic multiply/divide option is installed in the computer, hardware replaces programming, and the entire multiplication and division are performed by single instructions.

b IN-OUT REGISTER - All data transfers between the computer and low-speed or programmed input-output devices are made through the in-out register. Furthermore, control information is sent to the control units of high-speed devices through the in-out register.

The program can shift or rotate the contents of IO alone or in conjunction with AC by means of the instructions in the shift group.

However, as an element in the arithmetic unit, IO is required only for multiplication and division. In multiplication, each bit of the multiplier controls the formation of a partial product from IO_{17}. After a bit is used it is dropped from the IO register and a new bit is shifted into IO_{17} from IO_{16}. At the same time the less significant bits of the product are shifted into IO from the left, producing a double-length product.

In division, each time the divisor is subtracted from the dividend a bit of the quotient is placed in IO_{17}. At the same time the double-length dividend is shifted left, dropping the most significant bit. At the end of the operation the entire quotient is in IO while AC contains the remainder.

c ARITHMETIC UNIT CONTROL - The control circuits for the arithmetic unit include the overflow logic and the transfer and arithmetic logic for the accumulator and in-out register. The transfer logic generates the pulses that transfer information into AC or IO and the pulses that shift the contents of AC or IO to the left or the right. The arithmetic logic generates the pulses that perform the arithmetic and indexing operations.

d MULTIPLY/DIVIDE OPTION - This option includes a five-bit step counter, a timing system and various control circuits. The timing system is composed of a set of multiply/divide pulses, MDP-1 to MDP-13.

When this option is installed in the computer the instructions Multiply Step and Divide Step are replaced by the instructions Multiply and Divide. At the end of cycle one of either instruction the normal timing chain stops and the substitute multiply/divide timing system takes over control of the computer. Each step in the operation is executed by an automatic loop. Indexing is provided simultaneously by the step counter. When the appropriate number of steps has been executed, the timing chain is restarted and the computer enters cycle zero.

The result of the automatic Multiply is the same as the result of the standard multiplication subroutine (which utilizes Multiply Step). After completing Multiply the com-

puter continues with the next instruction in sequence. The automatic Divide, however, differs in two respects from the standard division subroutine (which utilizes Divide Step).

Divide includes an automatic provision that allows the program to compensate for an impossible division. If the first step in the division exceeds the capacity of the accumulator (i.e. if the divisor is smaller than the dividend), the computer returns the original dividend to AC and continues with the next instruction in sequence. However, if the division can be performed, a good-divide signal is generated. This signal advances the program counter one extra position. Then, when the division is completed the next instruction in sequence is skipped.

The second difference between automatic Divide and the standard division subroutine is that the result of the automatic operation is more accessible to the program than the result of the subroutine. At the end of Divide, the numbers in AC and IO are interchanged so that the quotient appears in the accumulator and the remainder is in the in-out register. Having the quotient in AC rather than in IO is convenient because it avoids the necessity of transferring the quotient back into AC for further program operations.

3-5 MEMORY

During every five-microsecond memory cycle, the memory address register addresses a single core register in the type 12 memory module. The memory cycle is divided into two portions, the read portion and the write portion. During the read portion of the cycle, a single 18-bit computer word is read from the addressed core register into the memory buffer. During the write portion of the cycle, the word contained in the memory buffer is written back into the addressed core register. For both the read and write portions of the memory cycle, the addressed core register is specified by the contents of the memory address register.

a MEMORY ADDRESS REGISTER - A standard 12-bit address is transferred into MA at the beginning of every memory cycle. This address controls memory access throughout the entire memory cycle. All operations occurring in the memory during the cycle affect only the single core register addressed by the contents of MA. At the end of each memory cycle, MA is cleared in preparation for use during the next cycle.

Included with the memory address register in the control unit are four binary-to-octal decoders. Each of these four memory address decoders decodes a three-bit section of MA. Each decoder asserts a single octal control level corresponding to the octal number contained in the associated three-bit section of the register. Therefore, the decoders provide a four-digit octal address to the memory module in place of the 12-digit binary address contained in MA.

b MEMORY BUFFER REGISTER - At the beginning of every memory cycle the memory buffer is cleared. During the read portion of the cycle a word is read out of the memory and transferred into MB. If new information is to be deposited in memory during the cycle then, after the read-out, MB is cleared. New information can then be transferred into it. During the write portion of the cycle the contents of MB (whether new or old) are written into the addressed memory location.

At the same time that a word is being written into memory the word is also available to the rest of the computer from the memory buffer. During logical or arithmetic instructions, the operand in MB is used by the accumulator input gating. During load instructions the word in MB is transferred to the accumulator or the in-out register.

If MB contains an augmented instruction word, bits 5 through 17 of the instruction word are decoded from MB. However, if MB contains a memory reference instruction word, the address portion of that word is transferred to MA at the beginning of the next memory cycle.

c MEMORY MODULE - The type 12 memory module contains a 4,096 word core bank and associated logic circuits for addressing memory locations, for reading information out of memory, and for writing information into memory. The 4,096 core registers in the core bank are arranged in a 64 by 64 matrix.

The outputs of the memory address decoders are applied to the memory module. At the module, these outputs select a single core register for use during the current memory cycle.

The memory control pulses from the control unit are applied to a four-bit shift register in the memory module. The memory timing functions (see figure 3-2) are generated

from the outputs of this shift register. The read function makes information from the addressed core register available to the 18 sense amplifiers. This information is sampled by the strobe pulse and transferred to the memory buffer.

Prior to the write function, the inhibit function applies inhibit current to all bits of the core register that correspond to 0's in the memory buffer. The write function then writes a 1 into each of the remaining (uninhibited) core bits. Information is thus written into the addressed core register by a 1 transfer. (The write function writes 1's into all bits of the core register except those bits which are kept in the 0 state by an inhibit current.)

d MEMORY EXTENSION CONTROL TYPE 15 - If extra type 12 memory modules are added to the computer, the memory extension control must be installed. The type 15 option allows expansion of the memory to 16 memory modules. In the expanded system, the outputs of the memory address decoders and the memory buffer register are applied to all memory modules through buffers in the memory extension control. Furthermore, information from any memory module is transferred to the memory buffer register through a mixer in the memory extension control.

Two versions of the type 15 control are available. The standard control allows expansion to eight memory modules; with a slight modification, control can be provided for 16 modules. The necessary address format is provided by extending the length of the memory address register and the program counter. To address the 2^{15} or 2^{16} memory registers contained in an eight-module or sixteen-module system requires a 15-bit or 16-bit address respectively.

The memory extension control provides this address format by utilizing a pair of three- or four-bit registers which function as the extensions of the memory address register and the program counter. These registers are EMA and EPC respectively. While MA and PC contain the address of a location in a single memory module, EMA and EPC contain the address of the module. To address a single location in the expanded memory, it is necessary to provide a module address to EMA at the same time that a register address within a specific module is provided to MA. During normal program cycles, the field address is provided to EMA from EPC, so that the program regularly operates within a single module.

However, the program may jump to another module or retrieve an operand from another module by performing an extend-mode cycle. Setting the extend flip-flop, EXD, puts the computer in the extend mode. The state of the extend flip-flop may be controlled by the operator from the console or by the programmer through the iot instructions Enter Extend Mode and Leave Extend Mode. While the computer is in the extend mode, the occurrence of a defer cycle causes the extend-mode cycle flip-flop, emc, to be set. The setting of emc limits indirect addressing to a single level, but causes the computer to interpret the deferred address as a 15-bit or 16-bit address instead of the usual 12-bit address. If the defer cycle is part of a jump instruction, a new module address is transferred to EPC at the same time that the usual 12-bit address is transferred to PC. If a memory reference instruction is deferred, a new module address is transferred to EMA at the same time that the usual 12-bit deferred address is transferred to MA.

3-6 STANDARD INPUT-OUTPUT SYSTEM

The standard in-ouf transfer control includes all logic necessary for decoding the secondary op codes of all the in-out transfer instructions. In the standard computer, however, this logic is utilized to produce only those command pulses necessary for control of the standard in-out system. The standard in-out transfer control also includes the completion pulse logic for the standard devices.

When the instruction register receives the op code for an iot instruction, the secondary op code (bits 12 through 17) is decoded by in-out transfer control into one or two command pulses. These command pulses can occur only at TP_7 or TP_{10}. For most iot instructions that control central processor functions (such as the sequence break system or the memory extension control) only one command pulse is necessary. This single command pulse always occurs at TP_7.

Two command pulses are usually necessary for the control of information transfers between the computer and the in-out devices. The preliminary command pulse at TP_7 generally clears either the buffer in the device control unit, or the in-out register. Then the main command pulse at TP_{10} either places the appropriate control unit into operation, or else transfers information between the buffer and the in-out register. Following the receipt of the main

command pulse, all control and information transfer functions between the device and the device control unit are performed automatically.

Closely associated with in-out transfer control is the input mixer, an element of the main control unit. Data from any input device is transferred to the in-out register through the input mixer. Also included in the logic for the standard input-output system are the control units for the three standard in-out devices. These devices are a photoelectric paper tape reader, a paper tape punch, and a typewriter.

a READER CONTROL - The control unit for the tape reader includes an 18-bit buffer register, RB. If the reader is operating in alphanumeric mode, a single line of eight holes on the tape is read. The eight data bits from this line are loaded into RB_{10-17}. If the reader is operating in binary mode, only holes 1 through 6 are read, but the reader reads three lines from the tape. The six data bits from the first line are loaded into RB_{12-17}. As each of the two subsequent lines is read the data in RB is shifted left six places and the six data bits from the new line are also read into RB_{12-17}. In binary mode a line of the tape is read only if hole 8 is punched. If hole 8 is not punched the reader skips the line. Therefore, to construct a full word in binary, the reader reads the first three lines in which hole 8 is punched.

Each line is moved past the reader photodiodes by engaging the reader clutch. When a signal is picked up from the feed hole, the output of the photodiodes is strobed and data is read into the reader buffer. In alphanumeric mode, the completion pulse is given after a single line is read. However, in binary mode a two-bit counter counts the lines read from the tape. The completion pulse is given only after three lines have been read. The completion pulse transfers information directly from the reader buffer to the in-out register if either a completion pulse has been requested by the program or the computer is performing Read In. Otherwise, the information is left in the reader buffer and must be retrieved later by the computer.

b PUNCH CONTROL - The punch control unit includes an eight-bit buffer, PB. The command pulse from in-out transfer control loads a single line of data into the punch buffer from the in-out register and puts the punch in operation.

In alphanumeric mode, an eight-bit character is transferred from IO_{10-17} to PB_{10-17}. In binary mode, a six-bit character is transferred from IO_{0-5} to PB_{12-17}, and furthermore PB_{10} is set automatically. Therefore when a line is punched in binary, hole 8 (which corresponds to PB_{10}) is always punched, and hole 7 (PB_{11}) is never punched. The six-bit binary character is punched in holes 1 to 6. For both alphanumeric mode and binary mode only one line is punched in the tape.

After receiving the command pulse, the punch control unit waits for a synchronizing signal from the punch motor. During the five-millisecond interval after receiving this synchronizing signal, the control unit punches a line. It does this by energizing appropriate solenoids corresponding to the contents of the punch buffer. At the same time the control unit advances the tape to the next position.

c TYPEWRITER CONTROL - The typewriter control unit contains a six-bit buffer, TB, which is used for both input and output operations. In a Type Out instruction, a single six-bit character is transferred from IO_{12-17} into TB. The outputs of the typewriter buffer are then applied to the mechanical decoder in the typewriter. This decoder determines which typewriter key is designated by the six-bit character in TB, and then strikes the appropriate typewriter key. Control characters such as carriage return, back space, shift, and so on, are sent to the typewriter in the same manner. After decoding these control characters, the typewriter carries out the indicated action

A similar sequence (but in reverse order) operates during input operations. When a typewriter key is struck, the mechanical encoders in the typewriter apply appropriate levels to the typewriter buffer logic. These levels correspond to the encoded representation of the character. The presence of this information at the TB gates causes two pulses. The first pulse clears the buffer; the second pulse strobes the information into the buffer. This second pulse also sets the typewriter buffer status bit and program flag 1, indicating to the computer that a typewriter key has been struck. The computer may then transfer the character from the typewriter buffer to the in-out register by executing a Type In instruction.

3-7 OPTIONAL INPUT-OUTPUT SYSTEM

The standard in-out transfer control is designed to permit any of the optional input-output devices to be added readily to the standard computer. All of the necessary command decoder levels and pulses are available at the taper pin panels in the in-out transfer control. Command pulses for optional equipment can be generated from the decoder levels and pulses merely by installing the necessary pulse amplifiers and logic nets in the bottom two mounting panels of bay 3. For brief descriptions of the in-out devices that are regularly available as options for the PDP-1 system see paragraph 2-3b.

Space is provided in the taper pin panels for the register outputs, mixer inputs, high-speed channel lines, and sequence break signal lines needed for the optional equipment. By adding the necessary bus drivers and input gates, information can be transferred into and out of the computer through these taper pins.

3-8 FLOW CHARTS

Paragraphs 3-9 through 3-15 describe the specific operations that can be executed by the computer. Each computer operation is a chronological sequence of events. Each individual event is a change in the state of the computer.

The flow charts show the operations as sequences of events. The main flow charts (figures 3-6 through 3-10) show the sequences that make up the various computer cycles. Each sequence begins at the top of a flow chart. Time is represented by horizontal bars on a nonlinear scale. Each horizontal bar represents the occurrence of a timing pulse. The time pulse numbers are written in the left hand column. The true time scale is shown in the drawing of the computer memory cycle (figure 3-2).

Each vertical path on a flow chart represents a sequence of events for a specific operation. The arrows indicate the direction of flow. At various timing pulses the line of flow is broken by a rectangle. The specific event that occurs at that timing pulse is written in the rectangle. All events that are written within a single horizontal bar along a single flow line occur at the same instant in time. For purposes of clarity, certain simultaneous events may be shown in separate rectangles.

Note that in many cases the state of a flip-flop (or register) is sensed by the same time pulse that changes the state of the flip-flop being sensed. This is possible because of the delay inherent in the flip-flop or in its input gating, i.e. the change in state of the flip-flop outputs lags behind the pulse applied to the input gating. Therefore the present state of a flip-flop can be sensed at the same time that the flip-flop is cleared, and the outputs of a register can be used at the same time that new information is transferred into the register.

If a specific event in a given line of flow depends only upon time, then that event is written alone in the rectangle. However, if other conditions which may or may not be fulfilled also govern the specific event, then these other conditions are also written in the rectangle. The conditions are written to the left of a colon; the specific event caused by the conditions is written to the right of the colon.

In some cases several sequences of events may begin with the same partial sequence. In this case the entire group of sequences is represented by a single flow line showing the common events. A branch point which distributes the flow into several separate sequences indicates the point at which the several sequences diverge. For example, in cycle zero all instructions are retrieved from memory by the same set of events. However, after the operation code is transferred to the instruction register, the sequence diverges depending upon whether the instruction is or is not a nondeferrable, one-cycle instruction. The line representing the deferrable instructions and the two-cycle instructions then has another branch point.

Movement along any specific branch must depend upon the fulfillment of some specific condition. The appropriate conditions are written on the individual branch lines. In all deferrable instructions the state of MB_5 is checked. If MB_5 is 1, the defer flip-flop, df_1 is set. Following the timing pulse at which this event occurs the line of flow branches into two possible sequences. One branch is followed if df_1 is 1; the other branch is followed if df_1 is 0.

In some cases separate branches may join, indicating that the events following the intersection point are the same for both sequences. Whenever a branch point or an intersection point occurs, arrows are drawn on all incoming lines.

A single path from top to bottom of any flow chart represents a single computer cycle. The path is entered at the top of the chart, according to the conditions listed. At the bottom of the chart each path is terminated by a reference to the cycle that follows the completed sequence.

In the main flow charts only those events that are peculiar to an individual sequence are shown in the line of flow through the chart. The events that are common to all cycles are listed in a column at the left of the chart. These common events include the events that make up the standard memory cycle. Also common to all cycles are those events that synchronize the high-speed channel control and the sequence break system to the main timing system.

The main flow charts show all of the operations that are executed within ordinary computer memory cycles. These include both the standard operations and the operations required for the three control options, the memory extension control, the sequence break system, and the high-speed channel control. Events required for optional operations are shown in parentheses. When appropriate, events involving extension registers are assumed to be included in events that affect the extended registers. That is, if the memory extension control is installed, a transfer between extended registers includes the corresponding transfer between the register extensions, and the transfer of PC to AC includes the transfer to AC of OV_1, EXD, and EPC.

Those computer operations that are not executed within memory cycles are shown in a different form. The events executed by special pulses and by in-out transfer command pulses are listed in tables 3-1 and 3-2, respectively. The special pulses include the power-clear pulse, the start-clear pulse, pushbutton pulses, and the special pulse chain, SP_1 through SP_4. The special pulse table includes all events except those contained in the operation Read In. Read In is shown in a separate flow chart (figure 3-5). The command pulse events for all standard in-out transfer instructions and all in-out transfer instructions that govern central processor options are included in table 3-2. This table provides the link between the cycle-zero in-out transfer flow line and the flow chart of the individual in-out transfer operations (figure 3-11).

There are four non-memory cycle flow charts. These flow charts include the sequences of events that make up the Read In operation, the in-out transfer operations and the two optional automatic instructions, Multiply and Divide. The Multiply and Divide flow charts are included in chapter 7 with the description of the multiply/divide logic. The non-memory cycle operations are not dependent upon the regular computer timing system. Read In utilizes the special pulse chain but Multiply and Divide utilize the chain of multiply/divide pulses. In the flow charts for these operations the appropriate pulses are listed in a column at the left. The sequences of events that make up the in-out transfer operations do not depend upon any sequence of timing pulses. Instead, they depend upon the signals from the in-out devices or delays included within

TABLE 3-1 TIMING CHART: SPECIAL PULSES

(Note: For Read In see figure 3-5)

	Start	Continue	Examine	Deposit
PB Pulse 500 μs	$\lvert 0 \rightarrow$ run $\lvert 0 \rightarrow$ rim	$\lvert 0 \rightarrow$ run	$\lvert 0 \rightarrow$ run $\lvert 0 \rightarrow$ rim	$\lvert 0 \rightarrow$ run $\lvert 0 \rightarrow$ rim
SP_1 25 μs	$\lvert 0 \rightarrow$ MA $\lvert 0 \rightarrow$ cyc SC	$\lvert 0 \rightarrow$ MA	$\lvert 0 \rightarrow$ MA SC	$\lvert 0 \rightarrow$ MA $\lvert 0 \rightarrow$ AC SC
SP_2 25 μs	TA $\xrightarrow{1}$ PC		TA $\xrightarrow{1}$ PC $\lvert 1 \rightarrow$ cyc $\lvert 1 \rightarrow IR_1$ (LAC)	TA $\xrightarrow{1}$ PC $\lvert 1 \rightarrow$ cyc $\lvert 1 \rightarrow IR_{1,3}$ (DAC) TW $\xrightarrow{1}$ AC
SP_3 1 μs	(EXTEND: $\lvert 1 \rightarrow$ EXD)		PC $\xrightarrow{1}$ MA	PC $\xrightarrow{1}$ MA
SP_4	TP_0 $\lvert 1 \rightarrow$ run	TP_0 $\lvert 1 \rightarrow$ run	TP_1	TP_1
SC	$\lvert 0 \rightarrow df_{1,2}$; $bc_{1,2}$; $OV_{1,2}$; ihs, ios, ioh; EXD, emc $\lvert 0 \rightarrow$ IR, PC; HSC_{0-3}; b2's, b3's b4's $\lvert 1 \rightarrow$ ioc SBM START: $\lvert 1 \rightarrow$ sbm; $\overline{\text{SBM}}$ START: $\lvert 0 \rightarrow$ sbm			
POWER CLEAR	$\lvert 0 \rightarrow$ R, RS, W, I $\lvert 0 \rightarrow rc_{1,2}$, rby, rcl, RBS $\lvert 0 \rightarrow$ pun $\lvert 0 \rightarrow$ TBS, TBB, tyo			
STOP PB PULSE: $\lvert 0 \rightarrow$ run				

the control-unit logic. In the non-memory cycle flow charts, delays between the various events in each sequence are shown by breaks in the line of flow. The length of the delay is written in the break.

3-9 SPECIAL PULSE OPERATIONS

The console-initiated operation Read In is shown in a flow chart, figure 3-5. All other events dependent upon special pulses are listed in table 3-1.

When power is first applied to the computer logic the initial state of each of the various computer flip-flops is indeterminate. As a result, it is possible for the initial states of flip-flops at power turn-on to cause information losses by generating unwanted information transfers. A power-clear pulse is used to prevent such information losses. Whenever the main power switch is operated the power-clear pulse clears the control flip-flops in the memory and in the in-out equipment control units.

All other special pulses result from using the operating switches on the console. Whenever any operating switch is turned on, a pushbutton pulse is generated. For all console operations except Stop, the pushbutton pulse triggers the chain of special pulses, SP_1 through SP_4. This chain of special pulses times the execution of the appropriate operation. In all the special pulse chain operations except Continue, SP_1 also generates the start-clear pulse SC. The start-clear pulse prepares the computer for initial operations by clearing various registers and clearing or setting various control flip-flops throughout the system.

A flow chart of the special read-in mode of computer operation is shown in figure 3-5. This mode of operation utilizes both the special pulse chain and the regular timing chain to bring information into the computer without a stored program. In this operation the special pulse chain is not continuous.

Read In begins with a pushbutton pulse which triggers the special pulse chain. However, after SP_1 the chain is broken and the system waits until a single word in binary has been read from paper tape. The return of the reader completion pulse restarts the pulse chain, which then continues from SP_2 through SP_4. If the instruction read from the tape is Deposit In-Out the reader retrieves another word from the tape. This time the reader completion pulse triggers the regular timing chain so that the computer performs cycle one of dio. The final timing pulse of the instruction again triggers the special pulse chain.

Thus the Read In cycle is performed over and over again, each time retrieving a single word from the tape and storing it in memory. The process continues until the instruction Jump appears on the tape. When this happens SP_4 triggers the timing chain and places the computer into normal operation in cycle zero. The computer begins normal operation at the address specified by the Jump.

3-10 MEMORY CYCLE

Most of the events required for the basic memory cycle are common to all cycles and are shown in a column at the left in all of the main flow charts. The memory cycle begins with the transfer of an address to the memory address register at TP_0. This event is not shown as common to all cycles. Instead, the address transfer is shown in each individual cycle flow chart because the source of the address varies depending upon the type of cycle that is being performed.

At TP_2 the read level is enabled by setting flip-flop R. At TP_3 the memory buffer register is cleared and flip-flop RS is set. Three-tenths of a microsecond after RS is set the read strobe transfers a word from the addressed memory location to the memory buffer. The strobe also triggers TP_4. At this same timing pulse the sequence break system is synchronized to the computer timing system.

At TP_7 the read level is disabled by clearing flip-flop R. At the same time flip-flop W is set. The next timing pulse begins the assertion of the inhibit level by setting flip-flop I. At TP_9 flip-flop RS is cleared. The 0 state of RS in conjunction with the 1 state of W enables the write level. If the computer is operating in the single-cycle mode, TP_9 also clears flip-flop run. At TP_{9a}, the high-speed channels are synchronized to the computer timing system.

At TP_{10} both the inhibit and the write levels are disabled by clearing flip-flops I and W. If the computer is going to continue in normal operation, the memory address register is cleared in preparation for the next memory cycle. If flip-flop run is 0, the computer halts. The halt takes precedence over any other flow line shown leaving TP_{10} in the flow charts. The last timing pulse in the cycle also synchronizes the sequence break system reset function, freeing any channel on which a break has been requested.

3-11 CYCLE ZERO

The events that occur in cycle zero are shown in figure 3-6. Cycle zero events occur in two distinct groups. The standard program control and instruction retrieval operations are in the first half of the cycle. Operations required for individual instructions are executed in the second half of the cycle.

Those events that occur in the first half of every cycle zero are shown in the left flow line in the upper half of the figure. In each cycle zero an address is transferred from the program counter to the memory address register and the program counter is incremented. After the instruction is retrieved from memory the operation code is transferred from the memory buffer to the instruction register.

If the previously executed instruction was one of the shift group or the in-out transfer group the instruction is completed while a new instruction is being retrieved from memory. The additional events required for these instructions are shown in the flow lines to the right of the standard events.

After the op code is transferred to the instruction register, the flow branches into two main sequences. The flow line for the two-cycle instructions and the deferrable instructions goes to the left. The flow line for the nondeferrable one-cycle instructions goes to the right. If the indirect address bit of any deferrable instruction is 1, the defer flip-flop is set. The flow line for the deferred instructions goes on to the defer cycle. The flow line for the directly addressed two-cycle instructions goes on to cycle one. Flow lines for the directly addressed one-cycle jump instructions return to the one-cycle part of the flow chart.

The specific events required for the execution of the one-cycle instructions are shown at the right of the figure. In addition to the specific events shown in the flow lines, instructions of the skip, shift, and operate groups require the decoding of various bits of the memory buffer in order to determine certain characteristics of the instruction. The required decoding is shown in the upper right of figure 3-6.

The flow line for the in-out transfer instructions separates into four distinct branches depending upon the states of the indirect address bit and the in-out halt flip-flop. A nonwait in-out transfer (which may occur either in a break routine that interrupts an in-out wait or in a normal

program sequence) returns to cycle zero. If a normal iot begins an in-out wait, the return is made to the in-out wait cycle. The in-out wait cycle is cycle zero of an iot instruction in which the in-out halt flip-flop is 1. If the iot is performed as part of an in-out wait cycle, the return is made to a normal cycle zero if the completion pulse has been received. Otherwise the return is made to the in-out wait cycle.

In any non-wait iot instruction or in an iot instruction that begins an in-out wait, the second op code must be decoded into command pulses. This decoding is shown in table 3-2 and described in paragraph 3-15.

From the final timing pulse the flow lines continue to cycle zero, the defer cycle or cycle one, unless the normal program sequence is being interrupted. A break request is always granted at the end of cycle zero except in the case of a jump instruction that is deferred.

3-12 DEFER CYCLE

The two types of defer cycle are shown in figure 3-7. The standard defer cycle is shown in the left column. If a memory extension control type 15 is included in the system significant changes occur in the defer cycle. The altered defer cycle is shown in the right column of the figure.

The defer cycle may be entered either from cycle zero or from another defer cycle. In the standard cycle a 12-bit indirect address is transferred from the memory buffer to the memory address register. If the deferred instruction is Jump this address is checked to determine whether or not a return is being made from a sequence break routine. If the Jump is a break return, the appropriate sequence break system flip-flops are cleared to free the channel on which the break occurred. After the deferred address has been retrieved from memory the indirect address bit is again checked. If MB_5 is 1, df_2 is set and the defer cycle is repeated.

If the system includes additional memory modules, the 12-bit indirect address is taken from the memory buffer while the module address is taken from the extension of the program counter. Thus the indirect address is taken from the same module from which the instruction was retrieved during cycle zero. If the deferred instruction is a Jump that is extended the indirect address is checked to determine whether or not a sequence break return is being made. A sequence break return must be made by an extended deferred Jump from some location in field 0. If the Jump is a break return the appropriate sequence break system flip-flops are cleared to

free the channel on which the break occurred.

If the computer is operating in the extend mode, the extend-mode cycle flip-flop is set at TP_5. In an extend-mode cycle the computer interprets a deferred address as an extended 15-bit or 16-bit address instead of a normal 12-bit address. When the computer is operating in the extend mode, indirect addressing is limited to one level. After the deferred address is retrieved from memory, a second defer cycle is executed only if both the indirect address bit is 1 and the extend flip-flop is 0.

The terminating events for both types of defer cycle are the same. A break request is always granted at the end of a defer cycle. If a break is requested the computer continues to the interruption cycles. If there is no break the computer continues either to another defer cycle, to cycle one, or to cycle zero. If the defer flip-flop has not been cleared during the cycle the computer executes another defer cycle. If a two-cycle instruction is being performed, the computer continues to cycle one. The only instructions that can be completed in a defer cycle are the one-cycle jump instructions. If a deferred jump has been performed the computer returns to cycle zero.

3-13 INTERRUPTION CYCLES

The four interruption cycles are shown in figure 3-8. A high-speed channel interruption requires only one cycle. A sequence break system interruption requires three cycles.

Interruptions in the normal program sequence are granted only at the end of a cycle. When a break request is granted the computer must first determine whether the break is for high-speed channel access or a sequence break. The high-speed channels have priority over the sequence break channels. If a high-speed channel break is requested, flip-flop HSC_0 is set and the computer enters the high-speed channel cycle. If there is no high-speed channel request then the sequence break system gains priority. For a sequence break, the break counter is incremented to 1 and the computer enters break cycle one.

When a high-speed channel cycle is completed any further break requests are always granted. If the request is for another high-speed channel access, the high-speed channel cycle is repeated. Should no further high-speed channel break be requested, flip-flop HSC_0 is cleared and a sequence break is allowed. For a sequence break the computer continues to break cycle one.

If there is no break at all the computer returns to cycle zero.

The initial events in break cycle one vary depending upon whether the computer includes the standard one-channel sequence break system or the 16-channel sequence break system type 20. For the standard system a break is always made to memory location 0, so no address transfer is required. After the break is made, flip-flop b2 is cleared and flip-flop b4 is set. For a break on the 16-channel system, the break is made to the address specified by the break encoder. After the break is made, flip-flop bn3 is cleared while flip-flop bn4 is set.

No interruption of the normal sequence is ever allowed after a break cycle. When break cycle one is completed, the break counter is incremented and the computer automatically continues to break cycle two. After break cycle two the computer continues to break cycle three. After break cycle three is executed the break counter is incremented to 0 and the cycle flip-flop is cleared. This automatically returns the computer to cycle zero to perform the Jump to the desired break routine.

3-14 CYCLE ONE

The cycle one event sequences for the second cycle of the various memory reference instructions are shown in figures 3-9 and 3-10. In both figures, the events that are common to all memory reference cycles are shown in a column at the left of the figure. At the beginning of cycle one, the address for the memory reference is transferred from the memory buffer to the memory address register.

For computers that include the type 15 memory extension control, the origin of the address extension depends upon whether or not cycle one follows an extend mode defer cycle. If the extend mode cycle flip-flop is 1, an extended address is taken from the memory buffer. If the extend-mode-cycle flip-flop is zero, the module address is taken from the extension of the program counter. Consequently, the operand is retrieved from the same module from which the instruction was retrieved in cycle zero. Flip-flop emc is 0 either if cycle one follows directly from cycle zero, or if cycle one follows a defer cycle that was not extended.

The transfer of an address from the memory buffer occurs at the beginning of cycle one for every instruction except Call Subroutine. In this exceptional instruction, the usual transfer is inhibited and instead address 100 is transferred into MA. During cycle one, the computer

performs the actual operations required for the various memory reference instructions (computational operations, data-handling operations, or program transfer operations).

Between cycle zero and cycle one (or between a defer cycle and cycle one) the flow line representing the two-cycle instructions is distributed into 20 separate branches. Each branch represents the specific set of events required for the execution of a particular instruction. For all of the instructions except one, the flow line continues directly through cycle one. The single exception is the instruction Execute. In this instruction, the operand retrieved from memory is itself executed as an instruction. In order that the computer shall interpret the operand as an instruction, the computer is returned to cycle zero from TP_3 of cycle one.

An interruption in a normal program sequence is always allowed at the end of cycle one. If there is no interruption the computer automatically returns to cycle zero.

If the computer includes the multiply/divide option type 10, the standard instructions Multiply Step and Divide Step are replaced by the automatic instructions Multiply and Divide (figure 3-9). For these two automatic instructions the cycle-one events do not complete the instruction. Only certain initiating operations are carried out during cycle one, and then after cycle one is completed the computer initiates the appropriate automatic sequence. The flow charts of the automatic Multiply and Divide sequences are included with the description of the multiply/divide hardware in chapter 7 (figures 7-5 and 7-6, respectively).

Interruptions are allowed after the automatic instructions are completed. The required interruption-initiating events (such as the setting of HSC_0 or the incrementing of the break counter) are performed at TP_{10} of cycle one. The interruption cycle is, however, postponed until the automatic operation is completed. If a break has been granted during the previous cycle one, the computer continues to the interruption cycles after completion of the automatic sequence. If there has been no break, the computer returns to cycle zero after completing the automatic sequence.

If the computer is operating in either of the manual modes, it halts at the completion of cycle one. However, if either of the automatic instructions is being performed the computer continues from cycle one to the automatic sequence and halts after completing the instruction.

TABLE 3-2 TIMING CHART: COMMAND PULSE OPERATIONS FOR IN-OUT TRANSFER INSTRUCTIONS

(Note: All command pulses occur during the first cycle of an in-out transfer instruction, i. e., when the condition IOT · ioc[1] is satisfied).

2nd Op Code	Operations
00	No action
01	TP_7: RPA NAC: $\|_1 \rightarrow$ rcp $\overline{NAC}$: $\|_0 \rightarrow$ rcp $\|_0 \rightarrow RB_{0-11}$ $\|_1 \rightarrow RB_{12-17}$ $\|_0 \rightarrow$ rby $\|_1 \rightarrow$ rcl, $rc_{1,2}$ (see figure 3-11)
02	TP_7: RPB NAC: $\|_1 \rightarrow$ rcp $\overline{NAC}$: $\|_0 \rightarrow$ rcp $\|_0 \rightarrow RB_{0-11}$ $\|_1 \rightarrow RB_{12-17}$ $\|_1 \rightarrow$ rby, rcl, $\|_1 \rightarrow rc_1$ $\|_0 \rightarrow rc_2$ (see figure 3-11)
03	TP_7: $\|_0 \rightarrow$ TB TP_{10}: TYO NAC: $\|_1 \rightarrow$ tcp $\overline{NAC}$: $\|_0 \rightarrow$ tcp $IO \xrightarrow{1}{}_{12-17} TB$ $\|_1 \rightarrow$ tyo (see figure 3-11)
04	TP_7: $\|_0 \rightarrow IO$ TP_{10}: TYI $TB \xrightarrow{1}{}_{12-17} IO$ $\|_0 \rightarrow$ TBS
05	TP_7: $\|_0 \rightarrow$ PB $\|_1 \rightarrow$ pun TP_{10}: PPA NAC: $\|_1 \rightarrow$ pcp $\overline{NAC}$: $\|_0 \rightarrow$ pcp $IO \xrightarrow{1}{}_{10-17} PB$ (see figure 3-11)

TABLE 3-2 TIMING CHART: COMMAND PULSE OPERATIONS

FOR IN-OUT TRANSFER INSTRUCTIONS

(continued)

2nd Op Code	Operations
06	TP_7: $\lfloor_0 \rightarrow PB$ $\lfloor_1 \rightarrow pun$ TP_{10}: PPB NAC: $\lfloor_1 \rightarrow pcp$ $\overline{NAC}$: $\lfloor_0 \rightarrow pcp$ $IO_{0-5} \xrightarrow{1} PB_{12-17}$ $\lfloor_1 \rightarrow PB_{10}$ (see figure 3-11)
07	TP_7: $\xrightarrow{0} DB$ TP_{10}: DPY NAC: $\lfloor_1 \rightarrow dcp$ $\overline{NAC}$: $\lfloor_0 \rightarrow dcp$ (see Display supplement)
30	TP_7: $\lfloor_0 \rightarrow IO$ TP_{10}: RRB $RB \xrightarrow{1} IO$ $\lfloor_0 \rightarrow RBS$
33	TP_7: $\lfloor_0 \rightarrow IO$ TP_{10}: CKS Bits $\rightarrow$ IO Standard Bits: LP^1: $\lfloor_1 \rightarrow IO_0$ RBS^1: $\lfloor_1 \rightarrow IO_1$ tyo^0: $\lfloor_1 \rightarrow IO_2$ TBS^1: $\lfloor_1 \rightarrow IO_3$ pun^0: $\lfloor_1 \rightarrow IO_4$
50	TP_7: DSC $\lfloor_0 \rightarrow bnl$, $n=C(MB_{6-11})$
51	TP_7: ASC $\lfloor_1 \rightarrow bnl$, $n=C(MB_{6-11})$
52	TP_7: ISB $\lfloor_1 \rightarrow bn2$, $n=C(MB_{6-11})$

TABLE 3-2 TIMING CHART: COMMAND PULSE OPERATIONS
FOR IN-OUT TRANSFER INSTRUCTIONS
(continued)

2nd Op Code	Operations
53	TP_7: CAC $\vert_0 \rightarrow$ b1's
54	TP_7: LSM $\vert_0 \rightarrow$ sbm
55	TP_7: ESM $\vert_1 \rightarrow$ sbm
56	TP_7: CBS 1 Channel: $\vert_0 \rightarrow$ b3, b4 Type 20: $\vert_0 \rightarrow$ b2's $\vert_0 \rightarrow$ b3's $\vert_0 \rightarrow$ b4's
74	TP_{10}: EEM + LEM MB_6^0: $\vert_0 \rightarrow$ EXD MB_6^1: $\vert_1 \rightarrow$ EXD

3-15 IN-OUT TRANSFER OPERATIONS

In-out transfer instructions are executed during cycle zero. When the operation code for the in-out transfer group appears in the instruction register, the second operation code is decoded into command pulses by in-out transfer control. This decoding occurs only when the in-out commands flip-flop is 1; that is, during a nonwait iot or during the first cycle of an iot with an in-out wait. The command pulses are always generated either at TP_7 or at TP_{10}. The operations executed by the command pulses are shown in table 3-2.

Table 3-2 includes all of the iot instructions decoded by the standard machine as well as the iot instructions that govern the central processor options. Any iot instruction that performs some central processor function (such as checking status bits, governing sequence break channels, and so forth) is completed in cycle zero. The iot instructions that govern in-out information transfers usually serve only to initiate the operation of a specific device control unit. After operation is initiated the computer continues with other cycles while the device control unit proceeds independently.

The command pulses perform various operations in the control units, such as clearing a buffer register and setting or clearing various control flip-flops. The command pulses also control the states of the completion pulse flip-flops. The programmer must, by adjusting the states of bits 5 and 6 of the iot instruction word, determine whether or not the device and the computer shall be resynchronized by a completion pulse. If the need-a-completion-pulse level NAC is asserted, the appropriate completion pulse flip-flop is set. Level NAC is defined by the following equation:

$$NAC = MB_5^0 MB_6^1 + MB_5^1 MB_6^0 .$$

The input-output operations of the control units are shown in a flow chart, figure 3-11. All of these operations except typewriter input are initiated by the command pulses of in-out transfer instructions. The typewriter input sequence is initiated by the striking of a typewriter key. After the completion of the input sequence, the program must retrieve the typed character from the control unit by executing a Type In instruction. There is no sequence of timing pulses for the input-output operations. Each individual event in an in-out sequence either is triggered by a delay from some previous event, or else is triggered by a signal received from the in-out device.

3-16 USE OF DRAWINGS

Four types of illustrations are used in this manual: photographs, block diagrams, logic diagrams, and circuit schematics. The block diagrams and photographs illustrating the text (chapters 2, 3, 5, 7 and 11) are bound into the back of the manual. Figure references to these illustrations are of the form "figure 5-1" (i. e. the first figure in chapter 5).

The complete system logic is shown in D-size drawings furnished separately for more convenient use by maintenance personnel. Reference to these drawings is essential in understanding the detailed operation of the system. Because these drawings are the most frequently used source of troubleshooting information, it is important to be familiar with the symbols and conventions which they employ.

The figure numbers for the D-size logic drawings are always preceded by the prefix "D". Thus figure D8-1 is the first D-size logic drawing mentioned in chapter 8.

Figure D11-1 is a detailed layout drawing of the central-frame logic elements. This drawing shows the panel location of the principal computer logic networks and gives the number of the logic drawing on which each logic network may be found.

The standard DEC logic symbols used on the logic drawings are explained in the DEC Digital Logic Handbook. Additional symbols used in PDP-1 drawings are shown in figure 3-3. Each circuit included in the logic drawings is identified by type as well as by its physical location in the computer.

Circuit type is always shown as a four-digit number. This number is the same type number used to identify the circuit in the DEC catalog.

Examples:

4105 - - - - 5 inverters (500-kc series)

1105 - - - - 5 inverters (5-mc series)

1607 - - - - 3 pulse amplifiers (5-mc series)

All circuits other than logic nets are shown as blocks on the logic drawings. Besides the four-digit type number, these blocks usually include a two-letter mnemonic abbreviation of the circuit function.

Examples:

DE - - - - delay

PA - - - - pulse amplifier

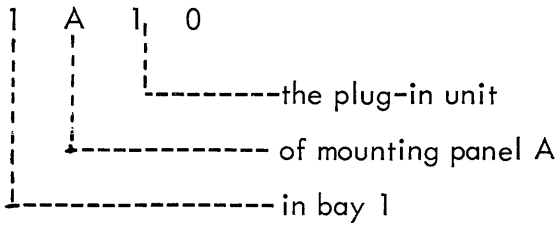
PG - - - - pulse generator

SD - - - - solenoid driver

BD - - - - bus driver

The circuit location code is lettered directly below the circuit type number. Circuit location code is shown as a single letter preceded by one digit and followed by one or two digits.

Example:



Terminal designations are formed by adding the pin letter to the plug-in unit location code described above.

Example:

1A10M ------- pin M of the connector in position 1A10

Since taper pin panels contain two or three rows of terminals, the number of the row is added after the pin letter in the taper pin designations.

Example:

3H25V3 ------- pin V in row 3 of the taper pin panel in location 3H25

Each logic drawing is laid out with rectangular map coordinates. The horizontal coordinates are 1 through 8 (from left to right), and the vertical coordinates are A through D (from top to bottom). Because a single drawing may contain a number of networks, coordinates are usually included in figure references to specific networks within a logic drawing. For example, a reference to the circuit "in figure D6-3B4" would mean that the circuit is located at coordinates B4 of the D-size logic diagram D6-3 (the third diagram referred to in chapter 6 of this manual).

Schematic diagrams for all computer circuits are bound into the back of the manual. These schematics are arranged in numerical order by circuit type designation. All of the circuits illustrated are described in chapter 10.

CHAPTER 4

INSTALLATION

4-1 GENERAL

This chapter provides the information needed to install the standard PDP-1 computer system. Installation and inspection procedures are described, together with general information on initial testing and use of the checkout programs.

Installation procedures for optional peripheral equipment are covered in supplements to this manual.

4-2 INSTALLATION

The standard PDP-1 consists of a four-bay central frame with three items of input-output equipment: a photoelectric paper tape reader, a paper tape punch, and an automatic typewriter. The tape reader and punch are mounted in one of the computer bays. The automatic typewriter is placed on a separate table, also included in the standard system. The central frame is shipped fully assembled except for installation of the in-out equipment. The in-out equipment items are packed separately and must be installed before the system is ready for use.

a SITE SELECTION - Before installing the PDP-1 system, a suitable location must be selected. Space requirements for the system depend upon the quantity of optional equipment to be used. The standard central processor occupies an area 97-1/4" by 27" and is 69-1/2" high. The typewriter table is approximately 3' x 3', and is usually placed beside the computer console desk. The computer is mounted on casters. A level floor is required, since these casters have no leveling adjustment. The floor should be capable of supporting 150 pounds per square foot. At least 3 feet clearance should be allowed on all sides of the central processor for ease of access during maintenance.

The system is designed to operate efficiently over an ambient temperature from 50° to 100° F. All necessary fans and blowers are installed at the factory. No additional

cooling equipment is required.

The user may elect to operate PDP-1 on either 110 or 220 vac. The internal power control connections for one type of line voltage or the other are made at the factory before shipment. Although the standard computer draws less than 20 amperes (at 110 vac) while in operation, turn-on surges in the in-out equipment (particularly the paper tape punch) may momentarily exceed this value. A 30-ampere line is therefore recommended for the standard computer. (Additional power must be provided for optional peripheral equipment).

b UNPACKING AND HANDLING - The central processor is shipped on a skid, and may be crated or not, depending on the mode of transportation. For truck shipment it may be left uncrated. A crate is furnished for air shipment. The crate containing the central processor is approximately 74 inches high, 3 feet wide, and 7 feet long. The typewriter table, typewriter, tape reader, and tape punch are separately crated for all types of shipment.

(1) If the central processor is crated, carefully remove all crating and strapping, and any packing material. If the computer is shipped uncrated, remove any protective padding.

(2) The plenum doors at the rear of the central-processor bays have spring catches. To reinforce these doors during shipment two screws are used to hold each door shut. Remove these screws and store them in the plastic loops provided.

(3) Remove any packing material, shipping blocks, etc. from the inside of the computer.

(4) The plug-in modules are taped into the logic panels to prevent damage in shipment. Remove the tape.

NOTE: If the user plans to reship the computer (or move it more than a short distance) in the near future, special packing materials should be saved for re-use. The containers for the tape reader, punch, and typewriter, in particular, have been designed especially to accommodate this equipment, and are the safest means of packing it for reshipment.

c INSTALLATION OF STANDARD IN-OUT EQUIPMENT - The paper tape reader, punch, and typewriter are packed in separate containers. To unpack, install and connect these devices, follow the procedures listed below.

(1) Tape Reader - The tape reader is shipped already fastened to its mounting frame. This frame is provided with rollers, and is designed to slide like a drawer into the end of bay 11 immediately above the console control panel.

(a) Carefully uncrate the tape reader and remove all packing material. Visually inspect the reader to make sure nothing has come loose during shipment. If reader appears to be undamaged, slide it carefully into place. When the back of the reader touches the stop bar, the mounting plate should fit flush with the other end panels of the console.

(b) With the reader in place, open the double doors and the plenum door at the rear of bay 11 (to the left of the console). The control cable for the tape reader is permanently connected to the left side of mounting panel 11A. The other end of this cable is equipped with a twist-lock connector. Connect it securely to the corresponding socket at the back of the reader.

(c) Power for the tape reader is taken from the ac distribution lines running along the top of the computer. The red and black twisted-pair reader power line is attached to one of the ac terminals at the top of bay 11. Connect the red wire to pin 6 and the black wire to pin 7 of the terminal strip on the tape reader.

(2) Tape Punch - The paper tape punch and cover are shipped in two separate containers. Carefully unpack and remove all packing material. Inspect for loose wires, screws, etc. Place cover on punch.

(a) Open the double doors and the plenum door at the rear of bay 11. Note the two aluminum seating studs on the shelf near the top of the bay. The two mounting plates fit over these studs. Place the punch on the seating studs, and determine that it is securely in position.

(b) The punch control cable is permanently connected to the left side of r ing panel 11B. Four wires from this cable are connected to terminals on type 812 power control panel at the top of the plenum door on this bay. loose end of the punch control cable is equipped with a 32-pin Amphenol Connect this plug to the corresponding socket at the back of the punch.

(c) The power cable for the punch is equipped with a twist-lock connector (The other end of this cable is permanently connected to the control panel the top of the bay 11 plenum door.) Twist the power cable connector firr into the corresponding socket at the back of the punch.

(3) Typewriter - The typewriter and typewriter table are shipped in separate co tainers. The typewriter table should be unpacked first and placed at either side of the console desk.

(a) Carefully uncrate the typewriter and remove all strapping and packing materials. Remove rubber stops at each side of the type basket. These stop prevent the type basket from shifting during shipment. Remove carriage locks at each end of the carriage guide rail. Each lock consists of three parts: a screw, a nylon piece, and a metal stop. Save for possible re-use all items that are taken off.

(b) Position the typewriter conveniently on the typewriter table. Rapid automatic typing may cause the typewriter to shift its position. To prevent this, four aluminum mounting cups are provided. Cement the mounting cup (using contact cement) to the table top in position to correspond to the type writer feet. Place the typewriter feet in these cups.

(c) The typewriter logic cable is equipped with a 50-pin Cannon connecto Plug this connector into the corresponding socket underneath the computer console desk.

(d) Plug the typewriter power cord into the power outlet under the desk beside the logic cable connector.

4-3 INSPECTION AND ADJUSTMENT

The PDP-1 system is thoroughly tested and checked before it leaves the factory. However, it should be inspected and checked again after installation to make sure that no damage has occurred during shipment.

a VISUAL INSPECTION - After the computer has been unpacked and the in-out equipment is in place, the system should be inspected visually.

Check the following:

(1) Have all shipping blocks, packing materials, tape, etc. been removed? If not, remove them.

(2) Are all plug-in units inserted firmly in position? Secure any that are loose.

(3) Are there any loose nuts or bolts? If so, tighten them.

(4) Are there any loose or broken wires? (Refer to Chapter 11 for repair of wiring).

(5) Are the tape reader and punch properly installed?

(6) Are the typewriter power and data cables plugged in?

(7) Make sure console POWER switch is off (right).

(8) Plug in system power cable. The cable is equipped with a Miller Electric Type 034-2 connector. Note that unless the system has been modified for use with 220 vac, the power cable must be plugged into a 110 vac outlet.

(9) Are the three MAIN POWER and the two PUNCH POWER circuit breakers on the type 813 power control panel on? If not, turn them on (up). Make sure the MEM POWER switch is off (down)!

b METER READINGS - Before starting to run the test program, all machine voltages should be checked with a meter.

(1) With the computer connected to its power source but with MEM POWER off, turn on the POWER switch on the console. The associated indicator should light.

(2) The type 728 power supplies each have three output lines: +10 vdc, ground, and -15 vdc. The output voltages should be checked for each type 728 supply before operating the computer. Measure these voltages at the top and bottom mounting panels of each bay. All A pins on each mounting panel are bussed together. Similarly, the B pins, the C pins, and the D pins are also bussed together. Pins A and B are at +10 vdc; pin C is at -15 vdc; pin D is at ground.

(3) Now turn on MEM POWER. This switch can now be permanently left on.

c PREOPERATIONAL CHECKOUT - The PDP-1 program library includes a set of test programs. These test programs are designed to check out different portions of the computer to ensure that they are functioning correctly. The test programs are a powerful aid in diagnosing computer malfunctions.

The test programs usually run at installation include Memory Checkerboard, Instruction Test, Reader Test, Punch Test, and Typewriter Test. Because all computer operations, including the running of test programs, depend on proper functioning of the memory, the Memory Checkerboard program usually should be run first. After the computer has passed the memory test, the Instruction Test program should be run, followed by the Reader Test. The Punch Test and Typewriter Test programs should be run last, in either order.

More detailed instructions concerning use and applications of the PDP-1 test programs are furnished in the maintenance chapter of this manual, Chapter 12.

CHAPTER 5

OPERATING PROCEDURES

5-1 GENERAL

The purpose of this chapter is to provide the operator with the information needed to operate the PDP-1 computer system. Descriptions of all controls and indicators are included, together with instructions covering the operation of the standard in-out equipment. In addition, this chapter provides general instructions for operating the computer under normal conditions. These general instructions supplement the special instructions included in each program write-up.

5-2 CONSOLE CONTROLS AND INDICATORS

Most of the controls and indicators associated with the central processor and the standard in-out equipment (reader, punch, typewriter) are contained in the two panels on the front of the console. The panel directly above the desk is the operator control panel. This panel is divided into two sections: a panel face which contains all of the indicators and some of the switches, and a panel shelf at the bottom which contains only switches. The in-out and sequence-break indicator panel is located at the top of the console.

There are two versions of the operator control panel. The panel shown in figure 5-1 contains the register extensions for use with the type 15 memory extension control. On this panel the reader and punch switches are located at the right end of the panel shelf. The panel shown in figure 5-2 does not have the register extensions. On this panel, the reader and punch switches are pushbuttons located on the right side of the panel. Figure 5-2 shows two views of the panel, one view from the front and one from the side. Any computer which includes the optional memory extension control has the control panel shown in figure 5-1. A computer which does not include the type 15 option may have either panel.

When any console indicator is lit, the associated flip-flop is in the 1 state or the associated function is true. Most toggle switches on the operator control panel are pushed up for on (or 1) and down for off (or 0). The power and mode switches are pushed to the left for on

and to the right for off. The operating switches on the panel shelf are two- or three-position momentary-contact switches with a center off position; the direction in which these switches should be operated is discussed with the individual switches.

The operator control panel is the main control panel of the computer. The operating switches used to initiate computer operations are all located on this panel, together with the indicator lights that monitor the contents of the central processor registers and the major control flip-flops. The face of the operator control panel is divided into three areas vertically; the left half of the panel face is devoted to the registers, the column just to the right of center is devoted to flip-flop indicators, and the right quarter of the panel face includes the power and mode switches and the program control elements.

The in-out and sequence-break panel, located at the top of the console, contains only indicator lights. If the computer includes a type 20 sequence break system, the panel is divided into two sections. The left half of the panel includes the in-out indicator lights; the right half of the panel includes the sequence-break indicator lights. This is the version of the panel shown in figure 5-3. If the optional sequence break system is not included, then the in-out indicator lights are located in the center of the panel.

In the remainder of this paragraph, the operator control panel is referred to merely as the "control panel", while the in-out and sequence-break indicator panel is referred to as the "indicator panel". All of the control switches are located on the control panel. Both panels include many indicator lights. Although the relative brightness of the lights gives some indication of their relative duty cycles while the computer is running, the primary purpose of these lights is to show the states of the functions being monitored after the computer has halted. Certain indicators are significant only for troubleshooting purposes.

In the following text, the switches and indicators are explained in terms of the figure 5-1 control panel. The figure 5-2 control panel is referred to explicitly only for those elements in which it differs from the figure 5-1 control panel. In reading the following text, the operator should refer to the phototgraph of the control panel that corresponds to his computer.

Functionally, the various switches and indicators on the two panels may be divided into the following eight groups:

1. Central processor registers (described in a below). Six sets of register indicators and two switch registers most of which are located on the left half of the control panel face. The instruction register is located in the lower right corner of the figure 5-1 control panel. However, in the figure 5-2 control panel this register is located with the other registers.

2. Central processor control flip-flop indicators (b below). Thirteen indicators (twelve in the figure 5-2 control panel) located in the center column of the panel face, and six indicators located at the right of the in-out indicators on the indicator panel.

3. Program control elements (c below). Six sense switches with associated indicators and six program flag indicators. Located in the right quarter of the panel face.

4. Power and mode switches (d below). Three switches located in the upper right corner of the panel face. However, the figure 5-1 control panel has an additional mode switch located to the left of the address switch register.

5. Operating switches (e below). Six switches located at the left of the panel shelf.

6. Reader and punch switches (f below). Two switches located at the right of the panel shelf. In the figure 5-2 control panel, these switches are replaced by three pushbuttons located on the right side of the panel.

7. In-out indicators (g below). Indicators for the buffer registers and flip-flops in the control units for the reader, punch and typewriter. These are located on the left half of the indicator panel (figure 5-3) if the computer also includes the type 20 sequence break system. If the type 20 option is not included, the lights are in the center of the panel.

8. Sequence break indicators (h below). Lights that indicate the states of the 64 flip-flops in the sequence break system priority chain. Located in the right half of the indicator panel.

a CENTRAL PROCESSOR REGISTERS (figures 5-1, 5-2)

Register Indicators

INSTRUCTION (IR)

Five-bit register which contains the instruction code of the instruction being performed or just performed.

PROGRAM COUNTER (PC)

This 12-bit register contains the address of the next instruction in the program.

EXTENSION (EPC) Four-bit extension of the PROGRAM COUNTER. The contents

displayed indicate the memory module from which the next instruction is to be obtained. The next operand will also be obtained from this module unless extended indirect addressing is employed.

MEMORY ADDRESS (MA)

This 12-bit register contains the address of the previous memory access.

EXTENSION (EMA) Four-bit register which contains the address of the memory module used during the previous memory access.

When the computer has stopped, these lights display the last address used, because the normal end-of-cycle clearing of MA and EMA is inhibited immediately prior to a halt.

MEMORY BUFFER (MB)

All transfers into or out of core memory take place through this 18-bit register. Furthermore this register holds the operand for all computational instructions. When the computer has halted, MEMORY BUFFER indicates the word contained in the memory location addressed by MEMORY ADDRESS.

ACCUMULATOR (AC)

This 18-bit register is the major arithmetic and operating register in the computer, and is involved in most computer operations. In computational instructions the operand from MB operates on the contents of AC. The results of computations always appear in AC.

IN-OUT (IO)

Eighteen-bit in-out and multiplier-quotient register. During in-out operations, IO is used as a buffer for transferring data between the computer and the in-out device control units. During multiplication and division, it serves as a magnitude extension of the accumulator.

Switch Registers

ADDRESS (TA)

A 12-bit toggle switch register through which the operator provides the memory address for the console functions Start, Examine and Deposit.

EXTENSION (ETA) Four-bit toggle switch register which provides a module address for the location specified by ADDRESS. EXTENSION also specifies the memory module for the operation Read In.

CAUTION

After beginning Read In do not operate the ETA switches until the computer has left the read-in mode.

TEST WORD (TW)

Lifting DEPOSIT transfers the contents of this 18-bit toggle switch register into the memory location specified by the ADDRESS switch register. During Load Accumulator from Test Word (lat) the TEST WORD is transferred into the accumulator.

b CENTRAL PROCESSOR CONTROL FLIP-FLOP INDICATORS -

Control panel (figures 5-1, 5-2)

RUN (run)

Lit while the computer is running in normal mode. Whenever the run flip-flop is cleared, the computer stops at the end of the current memory cycle.

CYCLE (cyc)

If this light is not on, the next cycle to be performed is cycle zero. When the light is on, the computer is about to perform cycle one or a special cycle. Before a defer cycle, high-speed channel cycle or one of the break cycles, CYCLE is lit together with the appropriate one of the following: DEFER, H.S. CYCLE, or BRK. CTR. 1 and/or 2.

DEFER (df_1)

When lit, indicates that the next cycle to be performed is a defer cycle (i.e. an indirect address cycle).

H.S. CYCLE (HSC_0)

When lit, indicates that the next cycle to be performed is a high-speed channel cycle.

BRK. CTR. 1, 2 (bc_1, bc_2)

Two-bit break counter. This counter contains the number of the break cycle to be performed.

OVERFLOW (OV_1)

When this light is on an overflow condition has occurred in the accumulator since START was pressed and since the last szo instruction.

READ IN (rim)

Lit while the computer is in the read-in mode.

SEQ. BREAK (sbm)

Lit while the computer is in the sequence-break mode.

EXTEND (EXD)

Lit while the computer is in the extend mode. Note: this light is not included on the figure 5-2 control panel.

I-O HALT (ioh)

Lit during an in-out wait.

I-O COM'DS (ioc)

Lit at all times except during an in-out wait.

I-O SYNC (ios)

This flip-flop is set by the completion pulse to end an in-out wait. It is cleared at the beginning of the following cycle provided that the computer is in the in-out wait. Otherwise it is cleared when the in-out wait begins.

Indicator Panel (figure 5-3)

MEMORY: READ, STROBE, INHIBIT, WRITE (R, RS, I, W)

These four flip-flops generate the memory core-drive pulses in memory module 0.

DEFER 2 (df_2)

This flip-flop is set whenever a defer cycle is followed by a defer cycle, but it is cleared too quickly to turn on the indicator. Used only for troubleshooting to indicate a malfunction.

OV_2 (OV_2)

This flip-flop is set during cycle one of any add or sub in which an overflow can occur, but it is cleared too quickly to have any effect on the indicator. Used only for troubleshooting, to indicate a malfunction.

c PROGRAM CONTROL ELEMENTS (figures 5-1, 5-2)

SENSE SWITCHES, 1 to 6 (SS_{1-6})

These switches may be set for manual intervention in a program. The levels produced

by these switches can be sensed by the program. When a switch is on, the associated indicator is lit.

PROGRAM FLAGS, 1 to 6 (pf_{1-6})

These flip-flops can be sensed by the program for control purposes. They can be set or cleared by the program or by an in-out device, but not from the console.

d POWER AND MODE SWITCHES (figures 5-1, 5-2).

POWER

Pushing this toggle switch to the left turns on computer power and lights the associated indicator. After turning POWER on, the operator must wait five seconds to allow the memory power supply to turn on before starting computer operations. When POWER is switched off, memory power turn-off is immediate but computer power remains on for five seconds.

This switch normally applies power to the entire system unless a piece of optional in-out equipment is turned off individually.

SINGLE STEP

If this switch is pushed to the left, the computer enters the single-cycle mode, lighting the associated indicator. In this mode, the computer executes a single memory cycle when START is operated. Subsequent cycles in this mode are executed one at a time each time CONTINUE is operated.

SINGLE INST.

If this switch is pushed to the left, the computer enters the single-instruction mode, lighting the associated indicator. In this mode, the computer executes a single instruction when START is operated. Subsequent instructions in this mode are executed one at a time each time CONTINUE is operated.

In the single-instruction mode, the computer treats the set of three break cycles together with the transfer instruction to the break routine as a single instruction. Note that if both SINGLE STEP and SINGLE INST. are on, the single-cycle mode has preference.

EXTEND

When this switch is on (up) the computer enters the extend mode whenever either

START or READ IN is operated. EXTEND is located to the left of the ADDRESS switches on the figure 5-1 control panel.

e OPERATING SWITCHES - All the operating switches are spring loaded to return to the off (center) position as soon as they are released.

START

This is a three-position switch; it can be operated upwards or downwards. When START is lifted up, the computer starts operating in the sequence-break mode; i.e. the start level is generated and flip-flop sbm is set. Pushing START downwards also generates the start level, but sbm is cleared. In either case the computer starts operating in normal mode; the first instruction executed is taken from the location specified by the ADDRESS switches. If EXTEND is on when START is operated, the computer also enters the extend mode.

STOP

When the computer is running in normal mode, pushing this switch down causes the computer to halt by clearing the run flip-flop. If the computer is operating in read-in mode, this switch stops Read In following the reading of an odd-numbered word from the tape (one of the dio instructions).

CONTINUE

When this switch is pushed down, the computer resumes normal operation, starting at the point indicated by the console lights. This switch must never be operated while the computer is in the read-in mode.

EXAMINE

When this switch is pushed down, the contents of the memory location specified by the ADDRESS switches are displayed in both the ACCUMULATOR and MEMORY BUFFER lights.

DEPOSIT

Lifting this switch up stores the contents of the TEST WORD switch register in the memory location specified by the ADDRESS switch register.

READ IN

When this switch is pushed down, the computer enters the read-in mode and begins

the Read In operation. This operation reads information from paper tape into memory without a program. Upon completion of Read In, if the last word read from the tape is a Jump instruction, the computer automatically begins normal operation, starting at the location specified by the Jump. If EXTEND is on, the computer enters the extend mode before beginning normal operation. For additional instructions on the use of this switch, refer to paragraph 5-5a.

f READER AND PUNCH SWITCHES (figures 5-1, 5-2)

READER

Lifting this switch up turns on the reader drive motor and energizes the brake. Note: the reader must be turned on before running any program in which the reader is used. Pushing this switch down turns off the reader, releasing the brake. Note: the reader must be off to load or unload tape. In the figure 5-2 control panel, this switch is replaced by a pair of READER pushbuttons START and STOP, located on the side of the panel.

TAPE FEED

While this switch is held down the paper tape punch runs, punching the tape with feed holes but no data holes. In the figure 5-2 control panel, this switch is replaced by a pushbutton located on the side of the panel.

g IN-OUT INDICATORS (figure 5-3)

TAPE READER

0 through 17 (RB)

Eighteen-bit tape reader buffer.

BINARY (rby)

This light goes on whenever the reader begins reading tape in binary mode. It goes off whenever the reader begins reading tape in alphanumeric mode.

NAC (rcp)

Need-a-completion-pulse flip-flop. Lit if the last reader iot requested a completion pulse.

RC1, RC2 (rc_1, rc_2)

Two-bit reader counter which counts the number of lines read from tape by a single iot. This counter also controls reader buffer shifting. For alphanumeric mode, tape is read one line at a time. In binary mode, tape is read three lines at a time; these three lines are assembled in the buffer into a complete 18-bit computer word.

CLUTCH (rcl)

Lit when the reader clutch is engaged.

TYPEWRITER

STATUS (TBS)

Lit when a typewriter key is struck; cleared by tyi.

BLACK (TBB)

Lit while the typewriter ribbon is positioned for black typing. The light is off while the typewriter is printing on the red portion of the ribbon. Note that the ribbon may be controlled only by the computer, not from the typewriter keyboard.

TYO (tyo)

Lit during a Type Out iot.

NAC (tcp)

Need-a-completion-pulse flip-flop. Lit if the last tyo requested a completion pulse.

12 through 17 (TB)

Single characters are transferred between the typewriter and IO_{12-17} through this buffer.

TAPE PUNCH

Punch buffer lights (PB)

These eight unlabelled lights are the punch buffer. The relative spacing of the lights is the same as the position of the corresponding holes punched on the tape. The gap between the first five and the last three lights corresponds to the space left for the feed holes.

ON (pun)

Lit by a punch iot; cleared by the completion pulse.

NAC (pcp)

Need-a-completion-pulse flip-flop. Lit if the last punch iot requested a completion pulse.

h SEQUENCE BREAK INDICATORS (figure 5-3). - The indicator panel includes these four sets of lights only if the type 20 sequence break system is installed in the computer. Each set includes 16 lights (numbered 0 to 17 octal), one for each of the 16 channels. The descriptions given below are for a single light in each set because all the lights in a set have the same meaning, each for its corresponding channel.

CHANNEL ON, 0 to 17 (b001 to b171)

Lit when sequence breaks are being allowed on the corresponding channel.

SYNCHRONIZER, 0 to 17 (b002 to b172)

This flip-flop is set when the corresponding channel receives a break signal from an in-out device, provided that channel is on. However, the program can set the flip-flop (and initiate a break) regardless of the state of the CHANNEL ON flip-flop. In either case, synchronization to the computer timimg system is provided during the same cycle, so the flip-flop is cleared too quickly to light the indicator.

WAITING BREAK, 0 to 17 (b003 to b173)

Lit while the channel is waiting for a break. The light goes on when the break signal is synchronized to the computer timing system and goes off when the break request is granted.

BREAK STARTED, 0 to 17 (b004 to b174)

Lit while the computer is performing the break routine for the corresponding channel. The light goes on when the break request is granted, and goes off when the computer returns to the interrupted program.

5-3 CENTRAL PROCESSOR POWER CONTROLS

This paragraph describes the power controls for the central processor. Alternating line voltage is distributed to the various computer power supplies through the type 813 power control panel located at the top of the bay 3 plenum door. In some machines, a type 810 power control panel is used instead of the type 813 panel.

A separate panel is provided for control of marginal check voltage. This panel includes switches for applying marginal check voltage to specific portions of the computer. In addition, individual marginal check toggle switches are located on the front of each logic panel.

a POWER CONTROL PANEL TYPE 813 - Central processor power is controlled by the POWER switch on the operator control panel (paragraph 5-2d). Turning this switch on activates the power control panel type 813 at the top of the bay 3 plenum door.

The type 813 power control panel (figure 5-4) contains an elapsed-time meter, three MAIN POWER circuit breakers, two PUNCH POWER circuit breakers, and the MEM POWER toggle switch. The elapsed-time meter counts the number of hours main computer power is on. The circuit breakers and toggle switch are normally left on at all times.

The three MAIN POWER circuit breakers provide overload protection to the computer power supplies. (There are only two MAIN POWER circuit breakers on the type 810 power control panel.) Line voltage for the entire computer, except for the paper tape punch motor, goes through these three circuit breakers. A separate pair of PUNCH POWER circuit breakers is provided for the tape punch motor. This is necessary because the punch draws a 9-amp surge at turn-on. Note that the punch-power line makes line voltage available to the type 812 punch control panel, but that the punch motor is not actually turned on until the punch is needed. This prevents excessive wear of the punch.

The 813 control delays memory power turn-on. This delay turns on memory power five seconds after main power is turned on, to ensure that turn-on transients in the computer do not affect the memory. The MEM POWER switch on the type 813 panel permits turning off this delayed line-voltage input to the memory power supply.

Memory power can thus be turned off separately from the rest of the computer for maintenance or troubleshooting purposes. If the system includes more than one memory module, delayed line voltage to all the memory power supplies goes through the MEM POWER switch.

For computer turn-off another set of delays is included in the power input to the computer. While memory power turn-off is immediate, the turn-off delay keeps main computer power on for five seconds after the console POWER switch is turned off.

b MARGINAL CHECK CONTROLS - The variable power supply type 734 furnishes marginal check voltages to the computer. It is located at the top of the bay 2 plenum door. This power supply provides voltages which can vary from 0 to -20 or +20 vdc, depending on the setting of the associated polarity switch. Output values between 0 and 20 volts are controlled by a variac and monitored on the MARGINAL CHECK voltage meter (figure 5-5). Line voltage for the type 734 power supply is supplied directly from the power control panel with no intervening switch. Therefore the marginal check power supply is one whenever the rest of the computer is on.

The plug-in unit pins to which marginal check voltage is applied are selected by three toggle switches (at the left of each logic panel on the front of the bays, figure 5-6) and an associated three-position polarity switch (on the marginal check switch panel, figure 5-5). To make positive marginal check voltage available to the computer, the polarity switch is set to +10 MC. Marginal voltage can then be applied to the A lines of any panel by pushing up the top toggle switch on that panel, and to the B lines by pushing up the center toggle switch. For marginal check of the -15 vdc lines, the polarity switch is set to -15MC and the bottom toggle switch of each panel being tested is pushed up. Note that all lines not being marginal checked automatically receive their normal voltages.

Although no marginal voltage can be applied to the computer if all three toggle switches on every panel are off (down), it is also possible to disconnect all marginal-voltage inputs by turning the polarity switch to the OFF position. This applies normal voltages throughout the computer regardless of the settings of the toggle switches.

There are five toggle switches to the right of the three-position polarity switch on the marginal check switch panel (figure 5-5). For all these switches, the up position is on; down is off. Only the first three switches on the left are used; the two on the right are spares.

The SENSE AMP switch applies marginal voltage to pin A of the memory module sense amplifiers. These sense amplifiers take up only a portion of panel 3D. For ease in troubleshooting, the sense amplifiers are isolated from the rest of the panel and are checked independently by the SENSE AMP switch. The rest of the plug-in units in panel 3D are marginal checked in the usual way by the switches on the mounting panel.

Marginal checking of the sensing circuits in the photoelectric tape reader is done with the FEED HOLE and INFO HOLES switches. To facilitate troubleshooting, separate switches are provided for feed-hole and information-hole sensing circuits. When one of these switches is pushed up, marginal voltage is applied to the +10 vdc lines in the corresponding circuit if the setting of the polarity switch is +10MC.

5-4 OPERATION OF STANDARD IN-OUT EQUIPMENT

The in-out devices furnished as standard equipment with PDP-1 are a photoelectric punched tape reader, a paper tape punch, and an automatic typewriter. Manufacturer's manuals for these devices are provided with the PDP-1 computer. In addition, some special instructions and precautions are included below regarding the use of these devices as part of the PDP-1 system.

a PHOTOELECTRIC PUNCHED TAPE READER - The tape reader, used by the computer as an input device, is mounted on the computer console, directly above the operator control panel (figure 5-7). Reader operation is as follows.

(1) Loading - Before loading or unloading the tape reader, the reader motor must be turned off by pushing the READER switch down. This releases the brake and prevents damage to the tape.

When loading the tape reader, the tape must be oriented so that it unfolds from the top of the fan-folded stack, and with the edge nearer the feed holes away

from the operator. The fan-folded stack is placed in the right-hand tape bin.

(2) Operation - Once the tape is properly loaded into the tape reader, the reader is turned on by lifting the READER switch. This energizes the brake and starts the reader motor. A reader iot (or the console operation Read In) can then make use of the tape reader by sending signals to the reader clutch. When the clutch is engaged, the tape moves past the sensing photocells.

CAUTION

Before running a program including any tape-reader iot's, or before executing a Read In, the tape reader must be turned on. Failure to do this will cause the computer to halt at the point where the tape reader is requested. When this occurs, the program must be run again from the beginning. To avoid such loss of computer time, always lift the READER switch before beginning any computer operation which includes the tape reader.

(3) Unloading - After the reader has finished reading a tape, the tape may be removed from the left storage bin. To prevent damage to the tape, always push the READER switch down before attempting to remove the tape.

(4) Coding - When reading tape in binary mode, the reader reads only the six least significant bits of each character that has the eighth hole punched, and assembles three such characters into an 18-bit computer word. When reading tape in alphanumeric mode, the reader reads all eight bits in each character. These eight bits may be in any code. The FIO-DEC Code and Concise Code are listed in table 5-1. The FIO-DEC Code is used on tapes prepared with the off-line typewriter sold by Friden for use with the PDP-1. This code includes a parity bit (bit 8). Characters are actually defined by the six least significant bits, which are the same in both FIO-DEC and Concise Code. Tape prepared by the computer on the paper tape punch may be in either code, depending on the program. The automatic typewriter uses only Concise Code.

b PAPER TAPE PUNCH - The paper tape punch, used by the computer as an output device, is mounted on a shelf inside the upper portion of the console, and is accessible from either the front or rear of the console bay. The punch mechanism faces the

double doors on the front of the bay (figure 5-8). Fan-folded tape is fed to the punch from a container. After punching, the tape is fed into a storage bin. A slot on the console (above the tape reader, see figure 5-7) allows access to this storage bin without opening the double doors. Punch operation is as follows.

(1) Loading - Load the paper tape punch as shown in figure 5-8. After tape has been properly positioned through the device, hold the TAPE FEED switch down long enough to feed approximately 18 inches of leader. Make sure the tape is feeding and folding properly in the storage bin.

(2) Unloading - To remove a length of punched tape from the storage bin, first hold TAPE FEED down long enough to provide an adequate leader at the end of the tape (and also at the beginning of the next length of tape).

Reach into the tape storage bin slot and remove the fan-folded tape. Tear off the tape at a point within the leader area (that portion of the tape with only feed holes punched).

After removal from the storage bin, the stack of folded tape should be turned over so that the beginning of the tape is on top, and then labeled.

Make sure enough leader is left in the punch storage bin to make at least three folds, with the first fold towards the bin opening. This ensures that the tape will stack properly inside the bin. If necessary, hold down TAPE FEED to provide additional leader.

c AUTOMATIC TYPEWRITER - The automatic typewriter is used by PDP-1 as an input-output device. In most cases, the operator communicates with the computer through the typewriter, especially when debugging a program. The typewriter is mounted on a separate table, usually placed beside the console, and connected to the computer by a cable plugged into a connector underneath the console table. An electric outlet, also under the console table, provides 110 vac to run the typewriter motor. Instructions for use of the typewriter as part of the PDP-1 system are as follows.

Figure 5-9 shows the typewriter keyboard. Special symbols replace certain of the standard punctuation marks on this keyboard. The typewriter uses Concise Code (see table 5-1).

Color shifting (from black to red and vice-versa) can be done only by signal from the computer, not from the keyboard. The BLACK light on the indicator panel is lit when the typewriter is using the black portion of a two-color ribbon.

Operating the SHIFT ↑ key on the right locks the typewriter in upper case. All typing will be done in upper case until the left-hand SHIFT key is operated. This key locks the typewriter in lower case.

The typewriter is turned on by pushing back a switch under the right side of the keyboard. When the typewriter is on, a window directly over the switch will show white. To turn the typewriter off, move the switch forward.

Special instructions on the use of the typewriter with a particular program (such as tab settings, etc.) may be found in the written description of that program. Typewriter use is determined by the programmer when writing a program.

CAUTION

Before running a program that calls for use of the automatic typewriter, make sure the typewriter is on. Failure to turn on the typewriter will cause the computer to halt at the point in the program where the typewriter is required. When this occurs, turn on the typewriter and hit the space bar. The first Type Out character is lost but the computer continues.

5-5 COMPUTER OPERATING MODES

The computer may operate in the following four modes: read-in mode, normal mode, or either of the two manual modes, single cycle or single instruction. In normal or manual mode, the computer may simultaneously be in sequence-break mode. In this mode the computer grants any break requests made by the in-out equipment. No sequence breaks may occur in read-in mode.

In addition to the operating modes, the computer performs two independent console operations, Examine and Deposit. These operations are initiated by the EXAMINE and DEPOSIT operating switches. The computer leaves the mode it is in before executing either operation.

a NORMAL MODE - The START switch initiates normal-mode operation. It is also used, in conjunction with the manual-mode toggle switches, to initiate operations in

manual mode. When START is operated, the computer starts in cycle zero; the first instruction access is made to the location specified by the ADDRESS switches. For concurrent sequence-break mode operation, START is lifted up. Normal-mode operation without sequence-break mode is initiated by pushing START down.

When the computer is operating in normal mode, it can be halted by pushing down the STOP switch. Always press STOP before operating any other initiating switch. This prevents any accidental loss of information in the computer.

If the computer has been halted while in normal mode, it can be restarted by pressing CONTINUE. This causes the program to continue where it left off. Note, however, that whenever a halt is followed by any operation that changes the state of the computer (such as Examine or Deposit), the computer must be started as at the beginning of normal-mode operation. The address of the next instruction must be set into the ADDRESS switches before START is operated.

b READ-IN MODE - The read-in mode reads from paper tape without a program. This operation is used to load programs or data into the computer, and is initiated by the READ IN switch.

Each tape word of information used in Read In alternates with a dio instruction. The memory location where a word is stored is specified by the address portion of the dio instruction immediately preceding it. In systems including more than one memory module, this address specifies a location in the particular module selected by the operator. Selection is made by setting the appropriate module address into the ADDRESS EXTENSION switches. This address is transferred into the module selection registers at each cycle of Read In. Therefore, once set, the switches should not be disturbed until Read In is completed.

At the completion of Read In, if the last word on tape is a Jump instruction, the computer automatically leaves the read-in mode and begins normal operation at the location specified by the jump. When no Jump instruction is included at the end of the tape, the computer halts at the completion of Read In. Normal operation can then be initiated by START at the location specified by the ADDRESS switches. The START switch causes the computer to leave the read-in mode and begin operation in normal mode.

The two console functions Examine and Deposit also cause the computer to leave the read-in mode before either function is performed. These two functions are sometimes used at the end of a Read In to examine the contents of certain memory locations or to deposit additional information into memory.

The STOP switch may be used to halt the computer in the middle of Read In. Note, however, that if for any reason Read In is interrupted, the computer cannot resume the operation at the place where it stopped. Instead, the operator must take out the tape and begin the Read In over again form the beginning.

CAUTION

Never operate the CONTINUE switch while in the read-in mode. This causes the computer to attempt to run in both read-in mode and normal mode simultaneously.
Never operate the READ IN switch twice in succession. Pushing READ IN while the computer is already running in the read-in mode may cause the reader to begin reading the wrong lines on the tape.

5-6 OPERATOR'S CHECKLISTS

The following checklists are provided for the operator's convenience. Checklists are included for operating the computer in read-in mode, in normal mode, and in either manual mode. Special instructions for running a particular program may be found in the write-up of that program.

a READ-IN MODE - To operate the computer in the read-in mode, follow the steps below in the order given.

1) Turn off all SENSE SWITCHES, ADDRESS switches, and TEST WORD switches. Make sure the two manual-mode switches (SINGLE STEP and SINGLE INST.) are both off.
2) Push READER switch down to release reader brake.
3) Load punched tape into reader.
4) Lift READER switch up to energize brake and turn on reader motor.
5) If the computer includes the optional memory extension control type 15, set the ADDRESS EXTENSION switches to the address of the memory module to be read into during Read In.

6) Press the READ IN switch.

7) To stop the computer during Read In, press the STOP switch. Do not operate any initiating switch without first halting the computer. Never operate CONTINUE during Read In. If Read In is interrupted, the entire operation must be started again from the beginning; repeat from step 2 above.

8) If the tape includes a Jump instruction at the end of the tape, the computer automatically starts normal operation at the end of Read In.

9) If the tape does not include the automatic start feature (step 8) normal operation must be initiated manually. Wait until the tape reader has finished reading the tape. Then proceed as for normal-mode starting, from step 2 in b below.

b NORMAL MODE - To operate start the computer in normal mode, follow the steps below in the order given. This checklist assumes the computer already contains the program to be run (see a above for Read In).

1) Turn off all SENSE SWITCHES, ADDRESS switches, and TEST WORD switches. Make sure the two manual-mode switches (SINGLE STEP and SINGLE INST.) are both off.

2) Check program write-up for in-out equipment needed for the current program run. Where needed, load the equipment with the required tapes, etc.

3) Turn on all in-out equipment to be used during program run. Failure to do this will cause the computer to halt. The entire program must then be repeated from the beginning.

4) Set address of first instruction into the ADDRESS switches.

5) To start in sequence-break mode, lift START switch. To start without sequence-break mode, push START switch down.

6) Check program write-up for any special instructions to be followed during the program run.

7) To halt the computer, press STOP. To continue with the program, press CONTINUE However, if the state of the computer has been changed after the halt (such as by an Examine or Deposit operation), proceed as for starting, from step 4 above.

c MANUAL MODES - To operate the computer in either of the manual modes, follow the steps below in the order given. This list assumes that the computer is already loaded with

an appropriate program.

1) Turn off all SENSE SWITCHES, ADDRESS switches, and TEST WORD switches. Make sure the two manual-mode switches are off.

2) Turn on the appropriate manual-mode switch (SINGLE STEP for single-cycle mode, or SINGLE INST. for single-instruction mode). Note that if both switches are on simultaneously, the single-cycle mode takes preference.

3) Check program write-up for in-out equipment needed for the current program run. Where needed, load the equipment with the required tapes, etc.

4) Turn on all in-out equipment to be used during program run. Failure to do this will cause the computer to halt. The entire program must then be repeated from the beginning.

5) Set address of first instruction into the ADDRESS switches.

6) To run in the sequence-break mode, lift START. To run without sequence-break mode, push START down. The computer performs a single memory cycle if SINGLE STEP is on, or a single instruction if SINGLE INST. is on. Note that the single-instruction mode, with sequence breaks, treats all the break cycles and the following instruction as a single instruction.

7) For each subsequent cycle or instruction, press CONTINUE. The computer stops after every cycle or instruction.

8) To leave the manual mode, turn off the manual-mode switch that is on. To complete the program in normal mode, press CONTINUE. However, if the state of the computer has been changed after leaving the manual mode, proceed as for normal-mode starting (b above, step 4).

TABLE 5-1 ALPHANUMERIC CODES

Character		FIO - DEC Code	Concise Code
a	A	61	61
b	B	62	62
c	C	263	63
d	D	64	64
e	E	265	65
f	F	266	66
g	G	67	67
h	H	70	70
i	I	271	71
j	J	241	41
k	K	242	42
l	L	43	43
m	M	244	44
n	N	45	45
o	O	46	46
p	P	247	47
q	Q	250	50
r	R	51	51
s	S	222	22
t	T	23	23
u	U	224	24
v	V	25	25
w	W	26	26
x	X	227	27
y	Y	230	30
z	Z	31	31

TABLE 5-1 ALPHANUMERIC CODES

(Continued)

Character	FIO-DEC Code	Concise Code
0 →	20	20
1 "	01	01
2 !	02	02
3 ~	203	03
4 ⊃	04	04
5 ∨	205	05
6 ∧	206	06
7 <	07	07
8 >	10	10
9 ↑	211	11
([	57	57
)]	255	55
¯ | *	256	56
- +	54	54
. _ *	40	40
, =	233	33
. ×	73	73
/ ?	221	21
Lower Case	272	72
Upper Case	274	74
Space	200	00
Backspace	75	75
Tab	236	36
Carriage Return	277	77
Tape Feed	00	00
Red **	--	35
Black **	--	34
Stop Code	13	--
Delete	100	--

*Nonspacing characters (dead keys)
**Used on typewriter Type Out only (not on keyboard)

CHAPTER 6

CONTROL

6-1 GENERAL

The control unit of the computer includes all the logic which governs the timing of operations within the computer, the transfer of information within the central processor, the execution of the program and the individual instructions within the program, the operation of the various registers, and the storage and retrieval of information from memory. This chapter describes the general control functions, including console control, timing, and cycle control; instruction control; program control; the shift/rotate logic; memory address and memory buffer transfer logic; the one-channel and 16-channel sequence break systems; and the high-speed channel control.

Certain portions of central processor control are discussed in other chapters. The arithmetic unit control circuits and the multiply/divide option are included in the discussion of the arithmetic unit (chapter 7). The memory address register and memory extension control option are included in the memory system (chapter 8). Control of information transfers between the computer and the control units of the individual in-out devices is discussed under in-out transfer control (chapter 9).

The control elements described in this chapter are shown in ten logic drawings, figures D6-1 through D6-10. For information on the use and organization of these drawings see paragraph 3-16.

6-2 GENERAL CONTROL FUNCTIONS

The general control functions of PDP-1 are shown in figure D6-1. These functions control the initiation, timing, and completion of all central processor operations. They also control the cycles within which the various types of computer operations occur.

a CONSOLE CONTROL SWITCHES - There are two types of console control switches: operating switches and mode switches. The operating switches start and stop the specific

computer operations while the mode switches control the operating modes of the computer.

(1) Operating Switches - The six operating switches are located across the bottom of the operator control panel (figure 5-1). The inputs to the control logic from these switches are shown in figure D6-1, fields B1 and C1. The outputs of the STOP, CONTINUE, EXAMINE, DEPOSIT and READ IN switches are applied directly to five of the six pulse generators shown in the figure. The levels produced by both on positions of the START switch are applied to the OR net in C1. The up position of this switch causes the computer to enter the sequence break mode before starting normal operation. The down position causes the computer to leave the sequence break mode before starting. The start level is generated when START is pushed to either position. This level in turn is applied to the sixth pulse generator in B1. The -3vdc level from any switch is used in the control logic directly. Each switch also produces a corresponding initial pulse through one of the pulse generators at the instant the switch is turned on.

Five of the operating switches are initiating switches, that is, they begin various operations within the computer. These operations are Start, Continue, Examine, Deposit and Read In. The initial pulses produced by all five of these switches begin the chain of special pulses that controls the console operations. The sixth switch, STOP, halts the computer. Therefore, the initial pulse produced by STOP does not initiate the special pulse chain.

(2) Manual Mode Switches - The computer performs programs in normal mode. Read In is performed in the special read-in mode. In addition, two manual modes are available: single cycle and single instruction. These modes are controlled, respectively, by the SINGLE STEP and SINGLE INST switches located in the upper right corner of the operator control panel (figure 5-1).

Both of these switches generate the manual run level (figure D6-1C3). When MANUAL RUN is true the computer halts at the end of the current memory cycle. MANUAL RUN is asserted continuously when SINGLE STEP is on so that the computer halts at the end of each memory cycle. When SINGLE INST is on, however, manual run is asserted only when the instruction-done level is also true. Because this level

is true during the final cycle of every instruction (f below), the computer halts at the end of each complete instruction when SINGLE INST is on. In either mode the first cycle or instruction is initiated by START and succeeding cycles or instructions by CONTINUE.

Grouped with the manual-mode switches on the control panel is the POWER switch. Operation of this switch applies power to the system. Each time the POWER switch is turned on, the computer is cleared by the power-clear pulses (figure D6-5D6). When power is applied to the system, terminal T of plug-in unit 1B2 is temporarily grounded. During the time when pin T is grounded, the clock in 1B1 produces power-clear pulses through pulse amplifier 1B2. The power-clear pulses clear various flip-flops and registers in the machine, especially in the in-out control equipment, in order to prevent accidental information transfers.

b SPECIAL PULSES - The networks that generate the special pulses are shown in figure D6-1, B1 through B4. The special pulses include the start-clear pulse, SC, and the chain of special pulses SP_1 through SP_4. This pulse chain is the timing system for initiation of console operations.

(1) Start Clear - To ready the computer for operation, most control flip-flops and registers in the computer are cleared by the SC pulse. This pulse is produced by SP_1 in Start, Examine, Deposit, and Read In (figure D6-1B1). Because Continue requires that operations begin according to the current state of the control unit, a start-clear pulse is never produced by this console function.

The SC-4 pulse generated by the pulse amplifiers in A6 is a 0.4-microsecond pulse triggered by the 70-nanosecond SC pulse. The SC pulse and the SC-4 pulse are logically identical.

(2) Special Pulse Chain - The initial pulse produced when any operating switch is turned on halts normal computer operation and triggers the special pulse chain. The 500-microsecond delay (figure D6-1B2) between the initial pulse and SP_1 allows plenty of time for computer operations to cease before the chain of special pulses begins. The special pulses following SP_1 are produced by the chain of delays and

pulse amplifiers in B3 and B4.

This chain of special pulses (SP_1 through SP_4) times computer operations until the regular memory cycle timing chain begins. In Examine and Deposit the computer executes only one memory cycle following the special pulse chain; in Start and Continue the computer enters the normal operating mode. However, in Read In the computer enters the normal operating mode only when the entire read-in is completed ((3) below).

(3) Read-in Mode Timing - In Read In the initial pulse puts the computer into the special read-in mode by setting flip-flop rim (paragraph 6-3a). When SP_1 is triggered the condition rim^1 breaks the special pulse chain and pulses RPB rim^1 (figure D6-1A5). This pulse causes the computer to read one word in binary from paper tape into the in-out register. Because the instruction register is clear at this time, the reader-return signal (delayed two microseconds by the 4301 delay in A4) restarts the chain at SP_2, provided STOP is off. Note that STOP halts the computer whether it is in the normal operating mode or in the read-in mode.

During the rest of the special pulse chain the op code portion of the word read from tape is loaded into IR. If the op code is DIO, RPB rim^1 is pulsed again, causing the computer to read another word from the tape. When the reader-return-delayed signal is received with DIO in IR, the computer goes into cycle one and deposits the word in memory. At the end of the cycle (B3) SP_1 is triggered, beginning the entire process again.

Read In ends when the op code JMP appears in IR instead of DIO. Then at SP_4 the computer leaves the read-in mode and begins normal operation at the location specified by the address portion of the Jump instruction.

c TIMING CHAIN - The main timing system of the computer is a chain of timing pulses, TP_0 to TP_9, TP_{9a} and TP_{10}. These timing pulses occur at irregular intervals throughout the five-microsecond memory cycle. The series of delays and pulse amplifiers shown across the top of figure D6-1 produces the timing pulses. The time of each pulse is

written above the name of the pulse. This timing system is also shown in the diagram of the memory cycle, figure 3-2.

The pulse amplifiers shown in figure D6-1A7 generate 0.4-microsecond pulses from the 70-nanosecond timing pulses. These are logically equivalent to the standard timing pulses and are indicated by "-4" following the subscript number of the pulse (e. g. TP_{7-4}).

Like the special pulses (b above) each timing pulse is produced by a pulse amplifier triggered through a delay from the previous timing pulse in the chain. Unlike the special pulses, however, not all pulses are triggered by the preceeding pulse: TP_3 triggers TP_7 after a 1.5-microsecond delay. During this interval, TP_4 is triggered by the memory strobe rather than by TP_3. Pulses TP_5 and TP_6 follow from TP_4. This aligns the set of three pulses, TP_4 to TP_6, with the retrieval of information from memory. The interval between TP_6 and TP_7 is great enough so that the main timing chain can continue without a race problem.

In normal operation the memory cycle is repeated over and over again until the computer is halted. Pulse TP_{10} triggers TP_0 through a delay, causing the cycle to start over again, but only if flip-flop run is 1. The computer is halted by clearing run. This prevents repetition of the memory cycle. Note, however, that the timing chain can be broken only at TP_{10}. Therefore, no matter when run is cleared, the computer halts only at the end of a full cycle.

When operations are initiated from the console the special pulses control the various functions which must precede the first memory cycle. Once these initial operations are complete, the memory cycle is started by SP_4. The first timing pulse may be TP_0 or TP_1, depending on the operation.

The cycle is started on TP_0 in Start and Continue. It is also started on TP_0 following an automatic Multiply or Divide. In these optional instructions the timing chain stops and a substitute timing system takes control of the computer. At the end of the instruction the MD-restart pulse triggers TP_0, restarting the normal timing chain.

In Examine and Deposit the cycle is started on TP_1. These console operations are performed as cycle one of the instructions lac and dac respectively, so TP_0 is skipped to prevent the

usual cycle-one transfer of MB into MA.

The other conditions that begin the timing chain apply only to the read-in mode of the computer. In this mode the computer alternately uses the special pulse chain and the timing chain to bring information into the computer without a program (b(3) above). After each SP_1 in Read In the computer reads an instruction from paper tape. If the op code is DIO the computer reads a data word from the tape and the delayed reader-return signal starts the memory cycle on TP_0. The cycle then deposits the data word in memory. If the op code is JMP, however, the computer leaves the read-in mode and begins normal program operation at the location specified by the Jump. In this case the first normal-mode memory cycle starts with TP_1 to skip the usual cycle-zero transfer of PC into MA.

d RUN CONTROL - The computer operates in the normal mode, with one memory cycle following another, while flip-flop run is 1 (figure D6-1C3). Each time the memory cycle ends, TP_{10} triggers TP_0 of the next cycle, causing the computer to continue. Whenever run is cleared the current cycle is completed and the timing chain ends at TP_{10}.

Flip-flop run is cleared at TP_9, causing the computer to halt at the end of the current memory cycle, on the instruction Halt (OPR MB_9^1); on an incorrect op code selection; and on the assertion of MANUAL RUN. (The last condition can occur only when the computer is in the single-cycle or single-instruction mode.) The pulse $MD \xrightarrow{0}$ run also clears run and ends the timing chain, but the computer does not stop. Instead the substitute multiply/divide timing system takes over until the end of the arithmetic operation. Control is then returned to the timing chain by MD RESTART, which sets run again.

The initial pulse produced when any console operating switch is turned on also clears run. The computer is halted in this way when STOP is operated; this is the only function of the stop pulse. In the same way, operation of any initiating switch halts current normal operation prior to the beginning of the desired console function. Because Examine and Deposit do not use the normal mode, run remains cleared for the single memory cycle of these two operations. For Start and Continue, run is cleared while the special pulses perform the initial operations, and is set again by SP_4 to return the computer to the normal mode. In Read In, the timing chain is initiated separately for each memory cycle,

and alternates with the special pulses. Run is initially cleared and remains 0 until JMP occurs at the end of the read in; the computer then leaves the read-in mode to begin normal operation.

e MEMORY CONTROL PULSES - The memory control pulses are generated by three pulse amplifiers (figure D6-1A8) which apply certain of the timing pulses to the memory module control circuits. The pulse $\lfloor 0 \rightarrow$ MEM, which clears the memory control flip-flops in preparation for the next memory cycle, is produced initially by the power-clear pulse and subsequently by the final pulse of every memory cycle. The memory operate pulses

$MOP_{2, 3, 7, 9}$ time memory operations by shifting control information through the four control flip-flops in the memory module. These pulses are all applied to the same line because flip-flops R, RS, W and I are a single shift-register plug-in unit with all pulses applied to the same input pin. The pulses $MOP_{2, 3, 7, 9}$ are equivalent to TP_2, TP_3, TP_7 and TP_9. The other pulse applied to the memory control flip-flops is INHIBIT (equivalent to TP_8). A separate pulse is required for this function because flip-flop I is set out of sequence in the memory module timing shift register.

If the computer contains only one memory module the pulses produced by the pulse amplifiers in 1F25 are applied directly to the shift register in that memory module. However, if a memory extension control is installed, then $MOP_{2, 3, 7, 9}$ and INHIBIT are applied to the module selection logic, which in turn applies these pulses to the selected memory module (see paragraph 8-5d). The pulse $\lfloor 0 \rightarrow$ MEM is applied to all modules. Similarly, the strobe from memory is applied directly to TP_4 in the timing chain if there is only one memory module. If more than one memory module is used, however, the strobe pulses from the various modules are applied to an OR net in the memory extension control; the output of this net is then applied to TP_4.

f CYCLE CONTROL - The cycle flip-flop, cyc, is shown in figure D6-1C2. When this flip-flop is in the 0 state, the computer is in cycle zero, and the current memory cycle is utilized to retrieve an instruction word from memory. When cyc is set, the computer goes into either cycle one or a special cycle. The control function for cycle one is generated by the net shown in C2 and C3. This net asserts the level CY1 (indicating that the com-

puter is in cycle one) if cyc is 1 and the computer is not in any special cycle (i.e. a defer cycle, a high-speed channel cycle or a break cycle).

Flip-flop cyc is set by SP_2 in the console operations Deposit, Examine and Read In because the memory cycles of these operations are used as cycle one of the instructions dac, lac and dio, respectively. In normal operation cyc is set at the end of cycle zero if the computer must go into a special cycle or into cycle one of a memory reference instruction. Such instructions are indicated by an op code less than 60, i.e. either IR_0 or IR_1 is 0.

Normal operation of the computer begins in cycle zero on the console operation Start or at the completion of Read In. Flip-flop cyc is also cleared at TP_{10} of either the final cycle of an instruction or of a high-speed channel cycle, provided no HSC or SBS break is next required. After BC3 the computer always returns to cycle zero, and performs the transfer to the break routine without interruption. Flip-flop cyc is also cleared at TP_3 during cycle one of Execute. The computer then performs (in cycle zero) the instruction retrieved from memory as the operand of the Execute instruction .

The instruction-done level (figure D6-1C8) indicates the computer is in the final cycle of an instruction. This level is asserted when the computer is not in a break cycle and any of the following conditions is satisfied:

(1) The computer is in the final cycle of a memory reference instruction (cycle one);

(2) The computer is in cycle zero of a one-cycle instruction (op code ≥ 60) and no defer cycle is required; or

(3) The computer is in the final defer cycle ($DF \cdot df_2^0$) of a one-cycle instruction.

g DEFER CYCLE LOGIC - The defer cycle logic is shown in figure D6-1, D5 to D7. The execution of defer cycles is controlled by a pair of flip-flops, df_1 and df_2.

A defer cycle can follow cycle zero of any instruction which can use indirect addressing. If df_1 is set in cycle zero the computer enters the defer cycle after cycle zero is completed. Flip-flop df_1 is set if the indirect address bit is 1 and the computer is not executing a nondeferrable instruction. (The nondeferrable instructions include all the aug-

mented instructions and those memory reference instructions in which MB_5 is used for other control purposes.)

During a defer cycle the defer level DF is true. This level is equivalant to the condition that both df_1 and cyc are 1. If df_2 is set while the computer is in a defer cycle, the defer cycle is repeated and another address is retrieved from memory. Flip-flop df_2 is set at TP_6 of a defer cycle if MB_5 is 1. In this case, no instruction conditions are required, because the computer cannot be in a defer cycle unless it is already executing a deferrable instruction. However, if a memory extension control option is included in the computer, another condition is added to the set gating of df_2. The second defer flip-flop can be set only if the extend-mode flip-flop, EXD, is 0. When the computer is in the extend mode, indirect addressing is limited to a single level.

At TP_{10} of a defer cycle, df_1 is cleared allowing the computer to continue with the program only if df_2 is 0. If df_2 has been set during the cycle then df_1 is not cleared and another defer cycle is executed. In any case df_2 is cleared by TP_{10}. Both defer flip-flops are cleared by SC and also by any condition that requires the program counter to be counted down. That is, the pulses $\vert\!\!\rightarrow^{0}$ df_1 and $\vert\!\!\rightarrow^{0}$ df_2 are logically equivalent to $\vert\!\!\rightarrow^{-1}$ PC. This prevents the computer from attempting to perform a defer cycle simultaneously with a high-speed channel cycle or a break cycle.

h IN-OUT HALT CONTROL - The in-out halt control is a system of four flip-flops and associated gating circuits (figure D6-1, B4 to B6). These four flip-flops are: in-out commands, ioc; in-out halt store, ihs; in-out synchronizer, ios; and in-out halt, ioh.

Two levels used by in-out halt control are generated by the net shown in C4. The in-out-halt-done level is true if both ioh and ihs are 0 during an in-out transfer. Note that this means that IO HALT DONE is true not only after a halt is completed, but also at the beginning of an IOT, before the halt has even started. The level $\overline{\text{IO HALT DONE}}$ is true whenever IO HALT DONE is false during an IOT. When the computer is not performing an in-out transfer, both IO HALT DONE and $\overline{\text{IO HALT DONE}}$ are false.

An in-out wait is included in an iot only if desired by the programmer. If information is immediately available, as for example after a typewriter key has been struck, then no in-out wait can be used. If a long period of time is required between the initiation of the in-out transfer and the actual availability of information, then the programmer can skip the in-out wait and use the time for computation. In this case, unless the sequence break system is being used, another iot that does nothing but begin the wait must occur some time before the information transfer is ready.

The start-clear pulse sets ioc and clears the other three in-out halt control flip-flops. This is the initial configuration of the flip-flops. Note that ioc is normally 1. During an in-out transfer instruction the command pulses are generated by in-out transfer control only while ioc is 1 (B6). If there is no in-out wait the iot is performed in one cycle and there is no need to clear ioc. The commands flip-flop is therefore ready for the next occurrence of an iot in the program.

If an in-out wait is required, bit 5 of the iot instruction word must be 1. When the in-out transfer appears in the program, ioh is set at TP_7 if MB_5 is 1 and IO HALT DONE is true. Since the in-out wait has not started yet, IO HALT DONE is true at this time. While ioh is 1 the timing chain continues but no operations are performed for any further instructions. The computer merely repeats cycle zero over and over, decrementing the program counter by 1 every time it is advanced during each cycle zero. In this way the same iot is retrieved from memory each time.

At TP_2 of the cycle following the setting of ioh (that is, after $\overline{\text{IO HALT DONE}}$ becomes true) ioc is cleared. Thus, even if an in-out wait covers many computer cycles, no in-out command pulses are generated after the first cycle of the waiting in-out transfer instruction.

When the in-out transfer is completed, IOT DONE (paragraph 9-2) sets the in-out synchronizer flip-flop, ios. This resynchronizes the in-out operation and the internal operation of the central processor. After ios is set, if ihs is 0, ioh is cleared at TP_9 and the computer continues with the next instruction. After ioh is cleared IO HALT DONE becomes true, clearing ios at TP_2 of the following memory cycle. At the same

time ioc is set in preparation for future in-out transfers.

The in-out halt store flip-flop, ihs, is needed for sequence break operations. If the in-out wait is interrupted by a sequence break and the break routine itself includes an iot without an in-out wait, then that iot must be executed immediately. After the completion of the non-wait iot, the computer is allowed to return to the previous in-out wait. This is effected by the circuit in B3.

If IOT occurs with MB_5^0 while ioh is 1 (that is, during an in-out wait), TP_7 sets ioc and ihs and clears ioh. This ends the previous halt and, while ioc is 1, the commands for the new iot are provided. At TP_{10}, since ihs is 1, ioh is again set putting the computer back into the in-out wait which was interrupted by the sequence break. Because ioh is now 1 IO HALT DONE is true, so in the following cycle ioc and ihs are both cleared by TP_2. In this manner the in-out halt control flip-flops are returned to the in-out wait configuration after the non-wait iot is performed.

i BREAK COUNTER - The break cycles are controlled by the two-bit counter shown in figure D6-1 C7. A sequence break requires three special memory cycles before the computer returns to cycle zero.

The counter initially contains 00. It is advanced to 01 at TP_1 when an SBS break occurs. The source of the signal SBS BREAK depends upon whether or not the computer includes a high-speed channel control. If there is no HSC control, SBS BREAK is generated by the circuit shown in B8. When an SBS break request is made (by either the single-channel or 16-channel sequence break system) the signal HSC + SBS BREAK becomes true if either a mid-instruction break is permitted (paragraph 6-4c) or if the computer is currently in the final cycle of an instruction (f above). Although the level produced by this logic net is labeled HSC + SBS BREAK, in this case it is equivalent to SBS BREAK alone because there is no HSC control. The level is used as SBS BREAK by the break counter, and its complement is used as $\overline{\text{HSC + SBS BREAK}}$ by cycle control (f above).

If high-speed channels are included in the computer the request for an SBS break is made to HSC control, because the high-speed channels have priority over the sequence break system. An SBS break can occur only if no HSC break is requested.

When SBS BREAK is true, bc_1 is set at TP_{10}. The computer then goes into a set of three break cycles controlled by the states of the break counter. The counter control is shown in C7. The counter is advanced at each TP_{10} while at least one bit of the counter is 1. Note that this OR net of bc_1^1 and bc_2^1 also generates the control levels which indicate whether or not the computer is in some break cycle.

The control levels for the individual break cycles are produced by the decoding net in B7. When the counter contains the numbers 01, 10, and 11, the levels BC1, BC2 and BC3, respectively, are asserted. These levels control computer operations in break cycles one, two and three. At the end of break cycle three, when the counter contains 11, TP_{10} counts the counter back to 00 so that no further counting can occur. Then since no BC level is true, the computer enters cycle zero to perform the program transfer to the break routine.

j SEQUENCE-BREAK MODE CONTROL - The sequence break system, whether one-channel or 16-channel, is allowed to make break requests only when the computer is in the sequence-break mode, i.e. while flip-flop sbm is 1. The state of this flip-flop can be controlled both by the programmer and by the operator. Operator control is exercised through the START switch. If this switch is pushed up (Start, in sequence-break mode) SC sets sbm. If START is pushed down (Start, not in sequence-break mode), SC clears sbm. Programmer control of sbm is exercised through two of the class 50 in-out transfer instructions. Command pulse ESM (IOT 55, Enter Sequence Break Mode) sets sbm; the flip-flop is cleared by command pulse LSM (IOT 54, Leave Sequence Break Mode).

The sequence-break mode can occur concurrently with normal operation or one of the test modes, but not with the read-in mode. Sequence breaks are not allowed when information is being read into the computer on Read In; therefore the initial read-in pulse automatically clears sbm.

k HIGH-SPEED CHANNEL CYCLE CONTROL - Flip-flop HSC_0 controls the high-speed channel cycle. This is a general control which determines whether or not the computer is in a high-speed channel cycle. The individual channels are controlled by

HSC control flip-flops HSC_{1-3} (paragraph 6-10). When a high-speed channel break is required, the signal HSC BREAK is true, and TP_{10} sets HSC_0. This puts the computer into the HSC cycle, which allows memory access through one of the high-speed channels. As soon as all HSC requests have been satisfied, HSC BREAK becomes false. Flip-flop HSC_0 is cleared by TP_{10}; the computer then continues with cycle zero.

6-3 INSTRUCTION CONTROL

The instruction control logic is shown in figure D6-2. This includes the five-bit instruction register, the read-in mode flip-flop, the instruction decoders and a group of miscellaneous order control levels. The control of augmented instructions also includes the memory buffer decoders (figure D6-4).

a INSTRUCTION REGISTER - The instruction register and associated control circuits are shown at the lower left of figure D6-2. The instruction register is cleared initially by SC. It is then cleared at TP_4 of every cycle zero, break cycle one and HSC cycle. After IR is cleared in each cycle zero the transfer $MB \xrightarrow{1} IR$ is performed at TP_5, bringing the op code from MB_{0-4} into IR. Note that the instruction register is left clear throughout the break cycles and the high-speed channel cycle.

In addition to the standard instruction control during normal operation there are also several special circuits which control IR during console functions. For any operation initiated from the console, IR is cleared by SC. In Examine, IR_1 is set by SP_2 loading op code 20 (LAC) into IR. In Deposit, SP_2 sets both IR_1 and IR_3 loading op code 22 (DAC) into IR.

Control of IR during the console operation Read In is as follows. The read-in mode flip-flop rim, is set by the initial pulse from the READ IN switch. While this flip-flop is 1 the computer is in a special read-in mode and may alternately use the special pulse chain and the timing chain to transfer information from paper tape to memory without a stored program. The start-clear pulse occurs at SP_1 each time the computer goes through the special pulse chain. Then at each SP_3 the transfer $MB \xrightarrow{1} IR$ is produced. Because every other word on the tape is a dio instruction, this transfer loads the op

code for Deposit In-Out into IR. This dio instruction is executed during the single memory cycle which follows the chain of special pulses. In the final cycle of Read In the op code for Jump is loaded into IR instead. This clears rim and ends the read-in mode. Flip-flop rim is also cleared by the initial pulse on the console operations, Start, Deposit and Examine, so that the computer leaves the read-in mode before attempting to perform any of these operations.

b INSTRUCTION DECODER - The outputs of the IR flip-flops are applied to two types of decoder. Flip-flops IR_0 and IR_1 are decoded by the nets in figure D6-2, B1 and B2. These circuits decode the first two bits of the register into four outputs corresponding to the numbers 00, 01, 10 and 11 in the two flip-flops. These levels are then used by the main decoder to generate the specific command levels. The instructions which require only a single memory cycle are those with op codes in the 60's and 70's. These numbers require that IR_0 and IR_1 both be 1. Thus the decoder output $IR_{0,1} = 11$ is needed for cycle control (paragraph 6-2f).

The other three bits of IR are decoded by a standard binary-to-octal decoder type 1150. This produces eight outputs corresponding to the numbers 000 through 111 in IR_{2-4}. These eight outputs and the four outputs of the IR_{0-1} decoder are applied to the AND gates in the upper right of the figure. The AND function of one of the set of eight IR_{2-4} decoder levels with one of the set of four IR_{0-1} decoder levels asserts one out of the 32 specific command levels. These levels are designated both by the octal op code and by the name of the signal. The latter is, in most cases, the same as the three-letter mnemonic code for the instruction or instruction group.

Note that the 32 op codes are the even numbers less than 77. The complete op code includes bits 0 to 4 in IR and the indirect address bit. The command levels are numbered for the full op code with the indirect address bit taken as 0.

c MEMORY BUFFER DECODERS - Only the first five bits of the op code are decoded by the instruction decoder. Indirect addressing is controlled directly by the sixth bit from the memory buffer (MB_5). In the nondeferrable instructions, operations are frequently controlled by the state of a specific bit of MB. However, in various augmented

instructions a number of operations depend not on the state of a single bit of the instruction word but rather on the octal number contained in a specific three-bit section of the word.

To control such operations the address portion of MB is divided into four sections of three bits each. The outputs of the three flip-flops in each section are applied to a binary-to-octal decoder (figure D6-4, B1 and B2). Each of these decoders asserts one out of eight output levels corresponding to the octal number contained in the three bits. The three least significant bits (MB_{15-17}) are decoded by MBD_A. Similarly, MB_{12-14}, MB_{9-11} and MB_{6-8} are decoded by MBD_B, MBD_C and MBD_D respectively.

d MISCELLANEOUS ORDER LEVELS - The various additional order levels required for instruction control are shown in the lower right of figure D6-2. The net which detects an incorrect op code selection is in C5 and C6. Whenever an undefined op code is loaded into IR, the incorrect-op-code-selection level is enabled. This signal halts the computer by clearing flip-flop run (paragraph 6-2d).

The operation codes not currently assigned are 00, 12, 14, 36 and 74. Note that the incorrect-op-code-selection level cannot be generated during a high-speed channel cycle or a break cycle. This is necessary because the instruction register is cleared (op code 00) during such cycles without a subsequent transfer in.

The wiring for the multiplication and division instructions is shown in D8. Note that in the instruction decoder operation codes 54 and 56 are labeled MUS + MUL and DIS + DIV respectively. This is done because these instructions are not unique and depend on whether or not the computer includes the automatic multiply/divide option. If the option is not included the command levels MUS + MUL or DIS + DIV are wired to generate the specific command levels, MUS and DIV, respectively. If the multiply/divide option is installed in the machine the wiring is reversed and the command levels MUL and DIV are produced instead.

The other logic nets shown in the lower right of figure D6-2 are needed for special operations or for operations common to more than one instruction. The levels generated include the following:

Level	Significance
JMP + JSP	The two single-cycle jump instructions.
JMP. rim^1	Final instruction in Read In
df_2^0 (JMP + JSP)	Final defer cycle of JMP or JSP
$df_2^0 \cdot$ JSP	Final defer cycle of JSP
BC1 + CY1 (JDA + CAL)	The only operations which require both the transfer PC→AC and the transfer MA→PC.

The remaining circuits do not generate any new logic functions; they buffer certain command levels to satisfy the electrical loading requirements of the computer logic nets.

6-4 PROGRAM CONTROL

The program control elements comprise the program counter, six program flags, and six sense switches. Each instruction in the program is retrieved from the memory location addressed by the contents of the program counter. The program counter is stepped one position during each cycle zero. This causes instructions to be taken from consecutive memory locations. The program flags are flip-flops which can be sensed by the computer. Program flags can be set individually either by the program or by external signals. The sense switches can be set by the operator only, and can be sensed by the computer.

The programmer controls the program sequence by means of the skip instructions and the jump instructions. The skip instructions cause the computer to skip one instruction in the normal sequence, if a specific condition is satisfied. The skip operation is implemented by advancing the program counter one extra position. The jump instructions can transfer program control to any chosen location. This transfer is accomplished by loading a new address into the program counter.

a PROGRAM COUNTER, PC - The 12-bit program counter is shown in figure D6-3. The program counter contains the standard 12-bit address required by the type 12 memory module. In machines including additional memory modules, the extra address bits are provided by the extension of the program counter in the memory extension control (paragraph 8-5b). Each flip-flop in the program counter has two complement inputs; both of these inputs produce carry pulses to the next more significant stage of the register. These carry pulses are gated by opposite outputs from each flip-flop so that one carry

chain causes the register to function as an up counter, while the other carry chain causes it to function as a down counter.

The pulse $\xrightarrow{|-1}$ PC decrements the contents of the program counter by 1. The carry chain associated with $\xrightarrow{|-1}$ PC complements PC_n if PC_{n+1} changes from 0 to 1. This is the logical condition required to subtract 1 from the contents of the program counter.

Conversely, the pulse $\xrightarrow{|+1}$PC increments the contents of the program counter by 1. The carry chain associated with $\xrightarrow{|+1}$PC complements PC_n if PC_{n+1} changes from 1 to 0. This is the logical condition required to add 1 to the contents of the program counter. The pulse $\xrightarrow{|+1}$PC counts memory locations in the program, and in sequence breaks. The pulse $\xrightarrow{|-1}$PC backs the counter one location when a mid-instruction break occurs, or when the computer completes each in-out wait cycle. A detailed description of the program count logic is included in c below.

The program counter is cleared by the pulse $\xrightarrow{|0}$PC. This pulse is generated initially by SC, and is also generated prior to any transfer of program control. Addresses may be transferred into the program counter from the memory buffer register, the ADDRESS switch register on the console, or the memory address register. In each of these three transfers, bits 6 through 17 of MB, TA, or MA, respectively, are transferred into PC_{6-17}. The program transfer logic is described in d below.

The outputs of the program counter bits are applied to the memory address register input gating, and to the accumulator input gating. The contents of the program counter are transferred to MA for memory access. The contents of the program counter are transferred to the accumulator to save the current program-location address.

b PROGRAM FLAG LOGIC - The program flags are shown in the upper right of figure D6-4. These six program flags can be sensed by the computer to control the program (c below). The state of each program flag can be independently controlled by the program. This program control is exercised through the input gating of the program flag. Each program flag also has a direct set input which permits the flag to be used for detecting external signals.

Program control of the flags is exercised through the instructions in the operate group. The input gating of the flags is enabled according to the three-bit address in bits 15 through 17 of the instruction word. These bits are decoded in the usual manner by MBD_A. Address 1 enables the input gating to program flag 1, address 2 enables the input gating to program flag 2, and so on for the other addresses up to address 6. Address 7 enables the input gating to all six program flags.

The addressed flag is set or cleared depending on the contents of bit 14 of the instruction word. If MB_{14} is 0 in an operate instruction, the addressed program flag is cleared at TP_8. However, if MB_{14} is 1, the addressed flag is set at TP_8.

c PROGRAM COUNT LOGIC - The program count logic is shown in the lower left of figure D6-4. The nets shown at coordinates B3 and B4 generate $\xrightarrow{|-1|}$ PC in order to back up the counter for a break or for an in-out wait. The other nets generate $\xrightarrow{|+1|}$ PC. This pulse advances the counter for ordinary counting and for skipping.

(1) Decrementing PC - The pulse $\xrightarrow{|-1|}$ PC decrements the program counter and clears both defer flip-flops. The program counter must be backed up during the following two operations.

First: At the end of each cycle during an in-out wait ($IOT \cdot ioh^1$), the program counter is returned to the address it held at the beginning of the cycle. This prevents the computer from going on to the next instruction until the in-out wait is completed.

Second: Whenever a break occurs before the final cycle of an instruction PC is backed up one position. In this way, after the break operations are completed, the computer again begins the interrupted instruction. The program counter is backed up to the beginning of the interrupted instruction by pulsing $\xrightarrow{|-1|}$ PC. This pulse is generated at the end of any cycle in which an HSC or SBS break request is granted before the current instruction is completed.

The control level that permits mid-instruction breaks is generated by the circuit shown at B3. This control level also governs the granting of break requests (paragraphs 6-2i and 6-10a). A mid-instruction break is permitted during certain portions of all mem-

ory reference instructions. During these instructions breaks are permitted while the computer is in either cycle zero or in a defer cycle (i.e., not in cycle one). Mid-instruction breaks are also permitted during a one-cycle jump instruction that is deferred more than once.

Note that a break is always allowed at the completion of an instruction. Therefore, if the computer is performing any instruction which uses the regular memory-cycle timing system, the entire set of break conditions can be summarized as follows. A break is always allowed at the end of the current memory cycle, except in the case of a one-cycle jump that is deferred only once.

There are, however, two other situations that do not satisfy the above criterion. These are the break cycles and the automatic multiply/divide operations. If the computer is in any break cycle, no new break request is granted until the completion of the transfer following break cycle three. Thus a sequence break cannot be interrupted before the program counter contains the address of the first instruction in the break routine.

In the automatic Multiply and Divide instructions a substitute timing system takes control of the computer after the completion of cycle one. If a break request is made during cycle one of Multiply or Divide the request is granted for the next memory cycle. However, the next memory cycle does not occur until after the automatic Multiply or Divide instruction is completed.

(2) Counting - The program counter is incremented by $^{+1}$ PC to count memory locations in the program, during sequence breaks, and during certain subroutine-calling transfers.

The standard program counting occurs at TP_2 of every cycle zero. The address of the first location in a break is loaded directly into MA. Succeeding locations are then counted by PC at the end of each break cycle (B4). The program counter is also advanced at TP_{9a} in cycle one of the instructions CAL and JDA. This transfers program control to the location following the deposit location of the accumulator.

(3) Skipping - All advances of the program counter, other than those listed in

(2) above, cause the computer to skip an instruction. There are two signals that pulse $\xrightarrow{|+1}$ PC directly. One of these signals is the good-divide signal from the automatic multiply/divide logic. The other signal is the external $\xrightarrow{|+1}$PC signal from in-out transfer control. All other skips are caused either by two-cycle skip instructions or by one-cycle skip-group instructions. For all skips in these categories $\xrightarrow{|+1}$ PC is pulsed at TP_8.

The logic nets governing the two-cycle skip instructions are shown in C3 and D3. In all three of these instructions the skip is made in cycle one. The skip occurs on ISP if the sign bit of the accumulator is plus; on SAD if AC contains the number +0; and on SAS if AC contains the number -0.

All other circuits in the lower left portion of figure D6-4 govern skipping for instructions in the skip group. In these instructions both the condition on which the skip is made and the significance of the instruction can be varied by the programmer. The control level ENABLE is asserted if the addressed condition is fulfilled. If the addressed condition is not fulfilled, the control level $\overline{\text{ENABLE}}$ is asserted. The skip may be made on either control level; the choice depending on whether the indirect address bit is 0 or 1. The circuit governing the pulsing of $\xrightarrow{|+1}$PC is shown in C4. When the command level SKP is true, the skip occurs on either of the conditions ENABLE$\cdot MB_5^0$ or $\overline{\text{ENABLE}}\cdot MB_5^1$.

The generating circuits for the enabling functions are shown at the left of figure D6-4. The level ENABLE is true if the addressed condition is fulfilled; otherwise $\overline{\text{ENABLE}}$ is true. At the extreme left of the figure are the sensing circuits for the six program flags and the six sense switches. The flags are addressed through MBD_A (MB_{15-17}); the switches are addressed through MBD_B (MB_{12-14}). Addresses 1 to 6 address the individual flags or switches numbered 1 to 6. Address 7 addresses all of the flags or all of the switches. All flags and switches are sensed for the 0 state. The circuits shown at C3 check the remaining conditions which may be addressed by skip-group instructions. For these conditions no binary-to-octal decoding is necessary. The individual conditions are each addressed by a 1 located in a specific bit of MB. The conditions that may be sensed are addressed as follows:

MB_7: Sign bit of IO is positive (= 0)

MB_8: $OV_1 = 0$

MB_9: Sign bit of AC is negative (= 1)

MB_{10}: Sign bit of AC is positive (= 0)

MB_{11}: C(AC) = +0

d PROGRAM TRANSFER LOGIC - All transfers of program control are effected by clearing the program counter and then transferring a new address into it. Following the transfer, PC counts successive memory locations in the usual manner. The transfer logic utilizes the following four pulses (lower right, figure D6-5):

$\overline{|0}\rightarrow PC$

The program counter is cleared by SC prior to any console initiated transfer, and at TP_8 prior to any programmed transfer or sequence break.

$MA \xrightarrow{1} PC$

Successive locations in a sequence break are counted by the program counter. Since the break address is transferred directly in MA from the break encoder, the subsequent transfer $MA \xrightarrow{1} PC$ is required at TP_9 of break cycle one. This transfer is also required at TP_9 in CY1 of JDA or CAL because in these instructions program control is transferred to the address following the storage location of the accumulator.

$MB \xrightarrow{1} PC$

Programmed transfers and the starting location of the program following Read In are specified by the address portion of the instruction word in MB. The transfer $MB \xrightarrow{1} PC$ therefore occurs at SP_4 when JMP appears in the read-in mode, and at TP_9 in the final cycle of any programmed jump instruction. The latter includes cycle zero of a directly addressed JMP or JSP, and the final defer cycle ($DF \cdot df_2^0$) of an indirectly addressed JMP or JSP.

$TA \xrightarrow{1} PC$

This transfer occurs at SP_2 of the console operations Start, Deposit and Examine. In Start, the program begins at the location specified by the ADDRESS switches. In Deposit and Examine information is transferred to or from the memory location specified by the ADDRESS switches. Thus the transfer $TA \xrightarrow{1} PC$ is made merely to utilize the subsequent transfer $PC \xrightarrow{1} MA$.

6-5 SHIFT/ROTATE LOGIC

The shift/rotate logic includes shift/rotate pulses and control levels. A single step in a shift or a rotation is performed each time a shift/rotate pulse is generated. The application of the shift/rotate pulses to the AC and IO registers is governed by arithmetic unit control (paragraph 7-2a, e). The shift/rotate control levels control the type and direction of the shift and the linkage of the registers.

a SHIFT/ROTATE PULSES - The circuit which generates the shift/rotate pulses is shown in the upper left of figure D6-4. Shift/rotate pulses occur during three different operations. These are the instructions Divide Step and Multiply Step, and any instruction in the shift group (SH/RO). In DIS and MUS only one shift/rotate pulse is generated. This pulse occurs during cycle one at TP_1 and TP_8 respectively.

In the shift instructions the number of shift/rotate pulses depends on the number of 1's in MB_{9-17}. Each bit of MB is sensed at a specific timing pulse. Thus when the command level SH/RO is true the first SH/RO pulse is generated at TP_7 if MB_{17} is 1. Then at each consecutive timing pulse through the rest of cycle zero the next more significant bit of MB is sensed. The last pulse, TP_{10}, senses MB_{13}. However, during the next cycle zero (in which a new instruction is retrieved from memory) the command level SH/RO remains true until IR is cleared at TP_4. Thus the last three possible SH/RO pulses in a shift group instruction actually occur during cycle zero of the following instruction. The first three timing pulses in the cycle sense MB_{12}, MB_{11} and MB_{10} respectively.

b SHIFT/ROTATE CONTROL LEVELS - The levels which determine the effect of a SH/RO pulse are generated by the logic nets shown in figure D6-4, C5 to C8. These levels control the direction of shifting and perform the correct linkages in the registers for cyclic shifts and two-register shifts. For example, the level $\overline{IO \cdot SHIFT}$ links IO_1 to IO_0 on all left IO shifts except an arithmetic shift that affects only IO. On all other IO left shift/rotate operations information is shifted from IO_1 into IO_0.

The various shift/rotate control levels are generated as follows:

$$SH/RO\ L = SH/RO \cdot MB^0_5$$

$$SH/RO\ R = SH/RO \cdot MB^1_5$$

$$ROTATE = SH/RO \cdot MB^0_6$$

AC ONLY = SH/RO · MB_7^0 · MB_8^1 (links AC_0 to AC_{17} on AC SH/RO L)

IO ONLY = SH/RO · MB_7^1 · MB_8^0 (links IO_0 to IO_{17} on IO SH/RO L)

$\overline{\text{IO SHIFT}}$ = SH/RO ($MB_6^0 + MB_8^1$) + DIS + DIV (links IO_1 to IO_0 on IO SH/RO L)

AI = SH/RO · MB_7^1 · MB_8^1 (links AC_0 to IO_{17} on IO SH/RO L)

The following AND and OR functions are also generated:

AC · ROTATE (links AC_{17} to AC_0 on AC SH/RO R)

IO · ROTATE (links IO_{17} to IO_0 on IO SH/RO R)

AI · ROTATE (links IO_{17} to AC_0 on AC SH/RO R)

ROTATE + DIV + DIS (links AC_1 to AC_0 on AC SH/RO L)

AI + MUS + MUL + DIS + DIV (links IO_0 to AC_{17} on AC SH/RO L, and AC_{17} to IO_0 on IO SH/RO R)

6-6 MEMORY ADDRESS TRANSFER LOGIC

At the beginning of each memory cycle an address is transferred into the memory address register. Access is then made to the location specified by the address in MA during the cycle (paragraph 8-2). Since all transfers into MA are 1 transfers, the register must be cleared prior to the transfer. Addresses for high-speed channel access are transferred into MA from the HSC address mixer under HSC control. Transfers generated within the main control unit include the following four pulses (upper left, figure D6-5):

$\xrightarrow{|0}$ MA

The memory address register is cleared by SP_1 prior to any console-initiated memory cycle, and at TP_{10} prior to all succeeding cycles while the computer is in the normal operating mode (run^1). Note that SP_1 also clears MA prior to every memory cycle while the computer is in the read-in mode.

PC $\xrightarrow{1}$ MA

Since both program and sequence-break locations are counted by the program counter, this transfer occurs at TP_0 of cycle zero and break cycles two and three (bc_2^1). The transfer PC $\xrightarrow{1}$ MA is also necessary at SP_3 of Examine and Deposit because transfers from TA to MA must be made through PC (paragraph 6-4d).

MB $\xrightarrow{1}$ MA

This transfer occurs at TP_0 for address retrieval in every defer cycle and for the standard memory reference in cycle one. Note that the cycle-one transfer is inhibited in the instruction CAL (see below). The transfer MB $\xrightarrow{1}$ MA is also executed at SP_3 in the read-in mode for each deposit of a data word from paper tape into memory and, in the final cycle, for the starting location of the program.

100 $\xrightarrow{1}$ MA

In CAL the address portion of the instruction word is ignored and address 100 is used instead. Thus the transfer 100 $\xrightarrow{1}$ MA replaces MB $\xrightarrow{1}$ MA at TP_0 in cycle one of Call Subroutine.

BE $\xrightarrow{1}$ MA

Each sequence break is made to a specific initial location corresponding to the channel through which the break occurs. When the break begins the initial address is transferred directly from the break encoder to MA. Succeeding locations in the break are then counted on the program counter.

6-7 MEMORY BUFFER TRANSFER LOGIC

All gated transfers into MB are made from the accumulator or the in-out register. Most of these transfers are for the purpose of depositing information in memory. However, all transfers between AC and IO must be made via MB even though no information is deposited in memory. In the read-in mode only every second word read from paper tape is deposited. The other words are executed by the computer as instructions.

Information retrieved from memory is not gated by a transfer pulse. The pulse outputs of the sense amplifiers are applied directly to the individual flip-flops in MB. If there is more than one memory module, the sense amplifier outputs are applied to the individual flip-flops in MB through the memory buffer mixer. Information deposited in memory through a high-speed channel must also be applied to individual MB bits through the memory buffer mixer.

The main control unit generates five control pulses for MB. One of these, the complement pulse, can be generated only if the computer includes the automatic multiply/divide option. The transfer of IO to MB is a 1 transfer requiring a prior clear-MB pulse. Because the transfer of AC to MB is a jam transfer, no prior clear is necessary. This transfer is, however, performed by two simultaneous pulses, each affecting a portion of MB.

$\overset{|0}{\longrightarrow}$ MB

The memory buffer is cleared prior to all transfers from IO to MB and prior to all in-transfers through a high-speed channel (HSC $\overset{|0}{\longrightarrow}$ MB, paragraph 6-10a). The MB register is also cleared at TP_3 of every cycle to ready the register for retrieval of information from memory and at TP_5 in cycle one of DZM so that the regular read-write memory cycle clears (i.e., writes zero into) the addressed memory register.

IO $\overset{1}{\longrightarrow}$ MB

This pulse transfers information to MB for subsequent deposit in memory in BC3 and CY1 of DIO. This transfer also occurs at every SP_3 during Read In to transfer the words read from tape to MB, and at MDP-12 during automatic Divide to transfer the quotient from IO to MB for subsequent transfer to AC (see below).

AC $\overset{i}{\underset{0-5}{\longrightarrow}}$ MB, AC $\overset{i}{\underset{6-17}{\longrightarrow}}$ MB

These two lines are pulsed simultaneously whenever an entire word must be transferred from AC to MB. This occurs in BC1 and BC2 to deposit the current contents of AC and the current program location, respectively, during a sequence break. (During BC1, after C(AC) are deposited, C(PC) are transferred to AC. Then the location of the interrupted program is saved during BC2 by transferring C(AC) to MB a second time). The transfer also occurs in cycle one of any instruction that deposits a full word into memory from AC(CAL, JDA, DAC, IDX, and ISP).

The pulse MD:AC $\longrightarrow$ MB from the multiply/divide option causes a full word transfer at two different times. At the beginning of an automatic multiplication (MUL · CY1 · TP_2) the multiplier is transferred from AC to MB for subsequent transfer to IO. The multiplier controls the formation of partial products from IO. At the end of automatic division (MDP-13) the remainder is jam transferred from AC to MB at the same time that the quotient (previously held in IO) is jam transferred from MB to AC. The remainder is then transferred to IO.

The two AC-to-MB jam transfer lines are pulsed independently if only part of a word is being deposited in memory. The pulse AC $\overset{i}{\underset{0-5}{\longrightarrow}}$ MB transfers the op code (bits 0 through 5) in Deposit Instruction Part; the pulse AC $\overset{i}{\underset{6-17}{\longrightarrow}}$ MB transfers the address (bits 6 to 17) in Deposit Address Part.

$\overset{|C}{\longrightarrow}$ MB

In the optional automatic Multiply, MB holds the multiplicand. Since multiplication is performed only on positive numbers, MB may have to be complimented before

the execution of the multiplication algorithm begins (paragraph 7-5a).

In the optional automatic Divide the divisor in MB is repeatedly subtracted from the dividend in AC. Since the divisor may not always "go into" the dividend at each step, MB may have to be complemented many times during the execution of the division algorithm (paragraph 7-5c). To complement MB, the line $\overset{C}{\rightarrow}$ MB is pulsed whenever the complement signal arrives from the multiply/divide logic.

6-8 STANDARD SEQUENCE BREAK SYSTEM

PDP-1 includes a one-channel sequence break system as standard equipment. This one-channel system is removed if the 16-channel sequence break system type 20 is installed. The control circuits for the one-channel break are shown in the lower right of figure D6-4.

Program control over the one-channel sequence break system is exercised through three in-out transfer instructions. Two of these three instructions, esm and lsm, control the sequence break mode (paragraph 6-2j) by setting and clearing flip-flop sbm. The third instruction, cbs, clears the sequence break system. These three instructions are used in both the one-channel and the 16-channel sequence break system and for this reason are included in all machines. The decoding of these three iot's is shown with the in-out transfer control for standard equipment (figure D9-1).

A break through the single channel is controlled by three flip-flops, b2, b3, and b4. Either a pulse or a level from the taper pins in iot control can provide the external signal that initiates the break. If the break is initiated by a pulse, b2 is set directly. If b2 is set, or if the break is initiated by a level, then b3 is set by TP_4. Flip-flop b3 thus synchronizes the break system to the computer timing system.

If sbm is 1 and b4 is 0 when b3 is set, then an SBS break request is made. This request is applied directly to the break counter logic if there are no high-speed channels in the computer (paragraph 6-2i). However, if the computer includes a high-speed channel control type 19, the SBS break request is applied to HSC control (paragraph 6-10a). This is necessary because the high-speed channels have priority over the sequence break system.

If the break request is granted the break counter is advanced one position and the computer enters break cycle one. At TP_4 of BC1, flip-flop b4 is set, thereby preventing any further

break request from interrupting the present break. At the same time, flip-flop b2 is cleared.

Since there is only one channel, the break is made to memory location 0. During break cycle two the present contents of the program counter are stored in memory location 1. After the break routine is completed, a deferred Jump to memory location 1 effects the transfer back to the main program. Detection of the return is gated by sbm^1, however, to ensure that the memory location is being used for sequence break purposes. When this Jump instruction occurs in the break routine, flip-flops b3 and b4 are both cleared at TP_2. Clearing these two flip-flops, clears the break system, making it available for another break request. After a 3.5-microsecond delay, the SBS restoring pulse restores the original state of the overflow flip-flop according to the contents of bit 0 in memory register 1.

If more than one memory module is included in the machine, the transfer back to the main program must be made by a deferred Jump to memory location 1 in module 0. Then at the same time that the break flip-flops are cleared, DEBREAK sets the extend-mode flip-flop so that the return to the main program may be made to any memory module. After a 3.5-microsecond delay, the SBS restoring pulse restores the original states of OV_1 and EXD according to the contents of bits 0 and 1 respectively in memory register 1.

In addition to the automatic clear of the break system at the end of a break routine, the programmer may clear flip-flops b3 and b4 by CBS (IOT 56). All three flip-flops in the chain are also cleared by SC.

6-9 SEQUENCE BREAK SYSTEM TYPE 20

The type 20 sequence break system allows the main program to be broken by an external signal on any one of 16 channels arranged in a priority chain. The priority sequence is prewired. It is determined by the connections which are made between the break system and the various input-output devices. If a break is initiated on any channel either by the external signal or by the program a break request is made for that channel provided there is no break on any higher priority channel.

If the break request is granted, the break is made to a fixed memory location corresponding to the break channel. Information necessary for a subsequent return to the interrupted program is stored in three consecutive memory locations. Program control for the break routine is then

transferred to f fourth memory location. Any break routine may itself be broken by a request on a higher priority channel. At the end of any break routine, program control is transferred back to the most recently interrupted routine. This may be either some lower priority sequence break routine or else the main program.

a BREAK SYSTEM PRIORITY CHAIN - The 64 flip-flops in the break system priority chain are shown in figure D6-6. Each of the 16 break channels is controlled by a set of four flip-flops, bn1 to bn4 (where n is the number of the channel; n = 00, . . . , 17). The setting of each flip-flop represents a stage in the break process as follows:

bn1: Channel n is on

bn2: Initiate break on channel n and synchronize by setting bn3

bn3: Await break on channel n

bn4: Break for channel n has been started.

Program control over the system is exercised through the seven iot instructions in class 50 (see chart in figure D6-7A6). Some of these instructions control the sequence-break mode or the break system as a whole while others control only a single channel. When an iot controls a single channel, the number of the channel is specified by bits 6 to 11 of the instruction word. These bits are decoded in the usual manner by MBD_D and MBD_C. The decoded control information is applied to the input gating of the channel flip-flops in two stages. Bits MB_{6-8} are decoded to produce a control pulse for one set of eight channels. If MBD_D is 0 the appropriate pulse is applied to the capacitor-diode gating of the flip-flops in channels 0 to 7; if MBD_D is 1 the pulse is applied to channels 10 to 17. The specific channel out of the set of eight is selected by the asserted level output of MBD_C (MB_{9-11}). For example, the programmer turns channel 13 on with the iot instruction 721351 (Activate Sequence Break Channel 13). With MBD_D^1, ASC produces the pulse $\overset{1}{\rightarrow}$b1's $_{10-17}$. This pulse in combination with the level output MBD_C^3 sets flip-flop b131, turning on channel 13 (figure D6-6D4). Generation of all break system control pulses is explained in b below.

The control flip-flops for each of the 16 sequence break channels function in the same manner. The following description treats the channel 2 flip-flops, b021 to b024, but applies equally to the control flip-flops for the other 15 channels. The channel 2

flip-flops are shown in figure D6-6, A3 and B3

The programmer may turn channel 2 on or off respectively by setting or clearing flip-flop b021. This flip-flop is set by $MBD^2_C \cdot \overset{1}{\rightarrow} b1's_{0-7}$; it is cleared by $MBD^2_C \cdot \overset{0}{\rightarrow} b1's_{0-7}$. If the channel is on, a break can be initiated by applying a break signal from an external device to pin V of R1A3. (The break signals are applied to each set of four channels through the following pin connections: pins F, N, V and Y, respectively, of the plug-in units type 4126 located at R1A3, R1C13, R1B3 and R1C18). The externally-generated break signal at pin R1A3V initiates the break by setting b022, provided that b021 is 1. Note, however, that the program can initiate a break whether the channel is on or off, because condition $MBD^2_C \cdot \overset{1}{\rightarrow} b2's_{0-7}$ can also set b022.

After b022 is set, the pulse SYNC (TP_4) synchronizes the break system to the computer timing system by setting b023. Then RESET SYNC (TP_{10}) clears b022. The setting of b023 also indicates to the system that channel 2 is waiting for a break. The outputs of b023 are applied to the channel-to-channel priority chain which crosses figure D6 from left to right between the b3 and b4 flip-flops. The priority of a break on channel 2 is determined by the state of this priority chain to the left of b023.

When b023 is set, the signal SEQ 2 is asserted if -3 vdc appears on pins K and J of R1A10. Pin K is at -3 vdc if b024 is 0, that is, if no break is currently being held for channel 2. Pin J is at -3 vdc if channel 1 is not waiting for a break ($b103^0$) and if, furthermore, pins N and M of R1A10 are both at -3 vdc; and so forth up the chain. Thus a channel 2 break request is made by asserting SEQ 2 if the following two conditions apply: first, if no break is currently being held for channel 2; and second, if no higher-priority (i.e., lower-numbered) channel is either waiting for a break or holding a break. Note that the chain to the right of channel 2 has no effect on channel 2. This permits a higher-priority channel to interrupt a break on a lower priority channel.

The signal SEQ 2 is applied to the break system control (b below). If the break request is granted, the pulse HOLD BREAK (equivalent to $TP_4 \cdot BC1$) sets B024, thereby indicating that a break is being held for channel 2 and disabling SEQ 2. HOLD BREAK also clears b023. Since b024 remains 1 throughout the break routine, no additional break can be requested on either channel 2 or on any lower-priority channel until the channel 2 break routine is completed. However, because both b022 and b023 are cleared during the break

process, it is possible to initiate and synchronize another break on channel 2. If a new break is synchronized on channel 2, then channel 2 is again waiting for a break while the original channel 2 break routine is being performed. Then, when b024 is cleared at the end of the break routine, a new break request is made for channel 2 (provided, of course, that no higher-priority channel is also waiting for a break).

When the break routine is completed the channel is freed by clearing the corresponding b4 flip-flop. The control pulses applied to the input gating of the b4 flip-flops each govern eight stages of the chain. However, these pulses do not function in the same manner as the pulses for the other flip-flops in the break system.

In the case of the b4 flip-flops, one pulse is applied to the even-numbered channels, and the other pulse is applied to the odd-numbered channels.

The specific channel affected by the b4 flip-flop control pulses is determined by the output of MBD_B. The eight outputs of this decoder each govern two adjacent channels. If the level MBD_B^0 is asserted either b004 or b014 is cleared depending on whether $\overset{0}{\longrightarrow}$ EVEN b4's or $\overset{0}{\longrightarrow}$ ODD b4's is pulsed. Similarly MBD_B^1 is applied to the input gating of b024 and b034; and so on to MBD_B^7 which governs channels 16 and 17. Selection is made in this way because of the method of return to an interrupted program after a break (b (3) below).

The break process through the set of four flip-flops governing any channel is the same as that described above. However, the priority-determining chain is not continuous from channel to channel. The chain is not fast enough to assure proper functioning of the lowest-priority channels. To remedy this the chain is continuous only for sets of four adjacent channels. The chain runs from channel 0 to channel 3, but the output of stage 3 is not applied to stage 4. Instead the diode net in A5 asserts the level NO $BREAK_{0-3}$ if all b3 and b4 flip-flops in channels 0 to 3 are 0. Thus the priority of a break on channel 4 is indicated by the level NO $BREAK_{0-3}$ instead of by the output of the priority chain from stage 3. Similarly, the diode net in B5 generates NO $BREAK_{0-7}$ for channel 10, while the net in C5 indicates to channel 13 that no higher-priority channel is waiting for or holding a break.

In addition to the control pulses already mentioned break-system control also generates

eight direct-clear pulses, each of which clears an entire row of flip-flops.

b BREAK SYSTEM CONTROL - The generating circuits for the control levels and pulses of the break system are shown in figure D6-7. Functionally these control signals may be divided into three groups: (1) the pulses that control the flip-flops during the break process; (2) the break-request level and break encoder; and (3) the pulses that free a channel after the completion of a break routine.

(1) Program control over the break system is exercised through the seven iot instructions in class 50. These seven instructions include both the three iot instructions that control the one-channel break system in the standard machine, and the four additional instructions necessary for the 16-channel system. Each of these seven iot instructions generates only a single command pulse. That pulse performs a single operation at TP_7. The decoding of the standard instructions is shown with the standard in-out transfer control (figure D9-1). The decoding of the extra instructions for the type 20 option is shown in figure D6-7, B2 and B3 (for an explanation of iot decoding see paragraph 9-2).

The standard command pulses ESM and LSM govern the sequence-break mode (paragraph 6-2j). The third standard pulse, CBS, clears the break system by clearing all b2, b3 and b4 flip-flops (upper right, figure D6-7). Initially, the system is cleared by SC. One of the additional instructions also affects all channels. This instruction is CAC, which turns off all channels by clearing all b1 flip-flops (C7, C8).

The other three additional instructions affect only the single channel addressed by bits 6 to 11 of the instruction word. Each command pulse generates one of two alternate pulses. Each of these pulses is applied to the input gating of an entire row of eight flip-flops in the break system priority chain (figure D6-6). If MBD_D (MB_{6-8}) is 0, a pulse is applied to channels 0 to 7; if instead MBD_D is 1, a pulse is applied to channels 10 to 17 (figure D6-7, B6 and C6). The final selection of a single channel from the eight partially selected channels is then determined by the number in MB_{9-11}. This number is decoded by MBD_C (a above). If MB_{6-11} contains the number n, command pulse ASC activates channel n by setting bn1; DSC deactivates channel n by clearing bn1; and ISB initiates a break on channel n by setting bn2.

When a break is initiated on channel n, the channel flip-flops are controlled by the pulses shown in the lower right of figure D6-7. After bn2 is set, SYNC sets bn3 at TP_4; then RESET SYNC clears bn2 at TP_{10}. If channel n has priority and the break request is granted, the computer goes into break cycle one. The pulse HOLD BREAK then sets bn4 and clears bn3 at TP_4.

(2) The priority chain output levels from all channels are applied to the diode nets in the lower left of figure D6-7. If any SEQ signal is asserted, then the SBS break request level is also asserted provided that flip-flop sbm is 1 (that is, provided that the computer is in the sequence-break mode). As in the case of the single-channel system, the break request is applied to HSC control if the computer includes high-speed channels (paragraph 6-10a), but is applied directly to the break counter logic if no HSC option is installed (paragraph 6-2i).

Above the break request net is the break encoder. This is a sexadecimal-to-binary encoding matrix. If the level SEQ n is true the number n, encoded in binary, appears at the four outputs BE_{12-15}. If n = 0, all four outputs are at ground; if n = 17, all outputs are at -3 vdc. At the beginning of break cycle one, n, the number of the channel, is loaded from the break encoder into MA_{12-15}. Since MA_{16} and MA_{17} both contain 0, this transfer is equivalent to loading the number 4n into MA. Each sequence break requires four consecutive memory registers. The break is therefore made to memory location 4n for a break on channel n.

(3) During break cycle two of a break on channel n the current program address of the interrupted sequence is deposited in memory register 4n + 1. For example, the address for channel 0 is stored in location 1; the address for channel 1, in location 5; the address for channel 2, in location 11; and so on through channel 17 in location 75. Therefore, when the break routine is completed, the return to the interrupted sequence must be made by a deferred Jump to a memory location which has an address that is less than 100 and that ends in either "1" or "5".

On a return from a break for channel n, flip-flop bn4 must be cleared to indicate to the break system that the break is finished. This flip-flop is cleared by the net in figure D6-7, A4 to A6. If the Y portion of a deferred Jump instruction is an address less than 100 (that is, if MBD^0_C and MBD^0_D are both true, indicating that bits 6

through 11 of the address are all 0) then $\overset{0}{\longrightarrow}$ EVEN b4's is pulsed by TP_2 on MBD_A^1 and $\overset{0}{\longrightarrow}$ ODD b4's is pulsed on MBD_A^5. These pulses are applied to the input gating of the b4 flip-flops in the even- and odd-numbered channels respectively (figure D6-6).

Each of the eight MBD_B outputs selects a pair of adjacent channels. These eight outputs are each applied to the input gating of the b4 flip-flops in the two selected channels. For example, if bits 12 through 14 of the Jump address contain 0 the output MBD_B selects channels 0 and 1. This output is applied to the input gating of flip-flops b004 and b014, the b4 flip-flops of channels 0 and 1 respectively. One of these two flip-flops is then cleared. The choice of the flip-flop to be cleared is determined by the contents of bits 15 through 17. If these three bits contain 1, all of the even numbered b4 input gates are pulsed. This clears the channel 0 b4 flip-flop, b004. Conversely, if bits 15 through 17 contain 5, all of the odd numbered b4 input gates are pulsed. This clears the channel 1 b4 flip-flop, b014.

If the MBD_B^1 output is asserted instead of the MBD_B^0 output, the return is from channel 2 or 3 instead of from 0 or 1. The choice between channels 2 and 3 again depends upon whether MBD_A (MB_{15-17}) is 1 or 5. A similar system is followed for each pair of channels: 4 and 5, 6 and 7, and so on through the eighth pair of channels, 16 and 17. For sequence breaks on channels 16 and 17 the return to the interrupted sequence is made to memory locations 71 and 75, respectively.

Detection of the return from the break routine is gated by sbm^1 to ensure that the Jump location is being used for sequence break purposes. The same pulse that clears bn4 also triggers a 3.5-microsecond delay which produces SBS RESTORE. This pulse restores the original state of the overflow flip-flop according to the contents of bit 0 in memory register 4n + 1.

If more than one memory module is included in the machine, the transfer back to the interrupted program must be made by a deferred Jump to memory location 1 in module 0. Then at the same time that bn4 is cleared, DEBREAK sets the extend-mode flip-flop so that the return to the interrupted program may be made to any module. After a 3.5-microsecond delay, the SBS restoring pulse restores the original states of OV_1 and EXD according to the contents of bits 0 and 1, respectively, in memory register 4n + 1.

6-10 HIGH SPEED CHANNEL CONTROL TYPE 19

This optional control unit governs three high-speed channels arranged in a priority chain. The channels are prewired to the individual in-out devices. Through these channels a high-speed device such as magnetic tape or data channel may gain direct access to memory for the transfer of information to or from the computer. When a device requests access on a high-speed channel the computer program pauses for one memory cycle while access is made and then continues with the program.

The HSC control unit includes the control circuits which govern the priority chain and the pulse logic for the information transfers. The unit also includes two input mixers, one for the transfer of addresses into the memory address register, and another for the transfer of data into the memory buffer register.

a REQUEST AND TRANSFER LOGIC - The control circuits for the high-speed channels are shown in figure D6-8. In the upper left of the figure are the channel request flip-flops. Whenever a request is made on any channel, the corresponding channel request flip-flop is set. The outputs of these flip-flops are applied to a logic net whose stages form a priority chain. The initial input to the chain is at the left (A1). This input requires that no break can be made through the chain unless the computer is either in the final cycle of an instruction or a mid-instruction break is being permitted. If this initial condition is fulfilled, then a break can be made through the chain by the highest priority flip-flop which is in the 1 state.

If HSC_1 has been set, then a break is made for channel 1. For each channel the break can be made only if the initial condition is fulfilled and no higher priority channel is requesting a break. Note that the SBS break request signal is applied to the right hand end of this chain. Thus the high-speed channels have priority over the sequence break system. If the initial condition is fulfilled and no HSC request is made, then an SBS break request is granted.

If a break is made on any high-speed channel, the level HSC BREAK becomes true, setting flip-flop HSC_0. This flip-flop continues to enable the priority chain (A1) even if the other initial conditions become false. This is necessary so that the previously granted channel break signal will remain true throughout the HSC cycle. If either an HSC break or a sequence

break is granted the signal HSC + SBS BREAK becomes true. This signal is used in cycle control and in the program count logic (paragraphs 6-2f and 6-4c).

The pulse control for high-speed channel information transfers is shown at the right of the figure. All pulses occur only for the channel on which the break is granted while the computer is in a high-speed channel cycle (i. e., while HSC_0 is 1). The pulses for all channels are identical. Therefore the following description of channel 1 (A5) applies equally to all channels.

At the beginning of the cycle (TP_0) a memory address is transferred into MA through the HSC address lines. Besides transferring an address into MA, the pulse ADD→MA generates the pulse WORD XFER. This word-transfer pulse signals the in-out device that access to memory has been granted. If the HSC access is made for retrieval of information from memory, the in-out device must generate a transfer-in pulse. This pulse must occur after the data is made available to the HSC data-out lines from MB (i. e., after the TP_4 memory read-out which occurs approximately 1.5 microseconds after WORD XFER). In any event the device must drop the channel request before TP_{9a} (4 microseconds) or the computer will perform a second HSC cycle for the same request.

If the signal CHAN #1 IN is true, the device does not accept data from the computer, and at TP_5 the memory buffer is cleared. Then at TP_7 a word is transferred into MB from the HSC data-in lines.

After the data transfer is completed TP_9 clears the corresponding channel request flip-flop (B2 and B3). This once more makes the channel available and allows the computer to handle any postponed request on any lower priority channels. If there are no requests HSC BREAK becomes false. This clears HSC_0, causing the computer to return to the program.

b HIGH-SPEED CHANNEL MIXERS - The high-speed channel control unit includes two capacitor-diode-gate mixers. At the beginning of each HSC cycle an address is transferred into the memory address register through the HSC address mixer HSAM. The length of the address loaded through the mixer may be 12 or 15 bits, depending upon the number of memory modules included in the computer. If the high-speed channel access is utilized for the transfer of information into the computer, the memory buffer is cleared at TP_5 and at TP_7 an entire word is transferred into MB through the 18-bit HSC buffer mixer HSBM.

(1) Address Mixer - The high-speed channel address mixer is shown in figure D6-9. One HSC address line for each channel is applied to the level input gate of the corresponding bit of the mixer. The address lines are prewired and fixed in the same order as the request lines to the priority chain. The transfer pulses ADD→MA are applied to the input gates across all the bits of the mixer. Thus on the pulse ADD→MA for a given channel, output pulses occur from those bits of the mixer which correspond to asserted input levels from the HSCA lines of the corresponding channel.

If only one memory module is in use the address mixer contains 12 bits $HSAM_{6-17}$. If the computer contains a type 15 memory extension control, a 15-bit address is required. The standard 12-bit address, specifying a location within a single memory module, is transferred through $HSAM_{6-17}$ to MA. The three-bit module address is transferred through $HSAM_{3-5}$ to the extension of MA in the type 15 control (paragraph 8-5b). If the memory is expanded beyond eight modules, another bit ($HSAM_3$) must be added to the mixer.

(2) Buffer Mixer - The high-speed channel buffer mixer is shown in figure D6-10. This mixer contains 18 bits so that an entire word can be transferred into the memory buffer from the HSC data lines. As in the case of the address mixer, one HSCD line from each channel is applied as the level gate input to the corresponding bit of the mixer. Data transfers through the mixer from a given channel are made on the WORD→MB transfer pulse for that channel. The pulse outputs of the high-speed channel buffer mixer are applied through the memory buffer mixer to the memory buffer register (paragraph 8-5a).

However, if the computer does not include a memory extension control, there is no memory buffer mixer. The HSBM pulse outputs are then applied directly to the MBM input gates of MB (paragraph 8-3). In this case HSBM acts as a mixer, both for the high-speed channels, and also for data transfers from the single memory module to MB. The pulse outputs of the memory sense amplifiers are applied to the transistor input gates mounted on the 1607 pulse amplifier units in HSBM.

6-11 DATA CHANNEL TYPE 123 - This optional unit acts as a control for high speed information transfers between PDP-1 and an external device. The data channel, DC, includes its own data buffer DB, word counter WC, and location counter LC. Through two iot instructions the program initiates data channel operations by specifying the number of words to be transferred and the initial address for high speed channel memory access. Data channel control then responds only to signals from the external device. At each word transfer, in either direction, the location counter is incremented by 1 so that each subsequent HSC cycle makes access to the next memory location. After processing the specified number of words, DC control signals the computer that the block transfer is complete. This completion signal may initiate a sequence break if desired by the programmer.

In a single block, all words are transferred in only one direction. However, DC may control a single device for both input and output, or two devices, one for input, the other for output. For output operations DC makes immediate HSC access to memory in order to make the first word available to the device. It then requests subsequent access only upon signal from the device (i.e. after the device retrieves the current word). For input the external device controls the loading of information into the DC data buffer. On signal from the device, DC control requests HSC access to transfer the contents of the data buffer to memory.

a EQUIPMENT LAYOUT - The data channel requires five panels in the standard DEC bay (Figure 6-11). In the logic drawings, letters A to E designate panel locations, although panels may be mounted in any position. The equipment includes an indicator panel (A), three standard 25-module logic panels (B, C, D) and an in-out plug panel (E).

The unit requires two 50-connector cables for connection to the computer. Computer connections are made through standard 50-pin Amphenol connectors from the bay 3 in-out plug panel to sockets EA and EB at the data channel. Connections to the DC logic are made directly from EA and EB. Signal connections for the external device are available at 22-pin Amphenol connectors mounted on the logic panels. Plugs specified by the user may be mounted in any of the other panel E sockets. The actual connections made from the logic panel connectors to the panel E plugs depend upon the mode of information transfer between the external device and the DC buffer.

The three logic panels include DC control in panel B, word counter and location counter in C, and data buffer in D. The respective logic drawings are Figures D6-15, -16 and -17.

If the data channel is mounted in a bay separate from the rest of the computer it requires both a 728 power supply and an 813 power control. The operator may turn the data channel on or off independently of the computer by means of the power switch on the indicator panel. If the data channel is mounted in a bay bolted to the rest of the computer it requires only the power supply. Data channel power is switched by the computer power control, and the DC power switch is not connected.

The following table lists the complete module requirements for the data channel (Figure 6-11 shows the module layout).

Type	Quantity
Bus driver 1685 or 1690	9
Inverter 4105	1
Inverter 4106R	4
Diode 4112R	1
Negative capacitor-diode 4127	5
Negative capacitor-diode 4127R	2
Negative capacitor-diode 4129	9
Quadruple flip-flop 4213	7
4-bit counter 4215	8
Pulse amplifier 4603 added to iot control	2
Pulse amplifier 4604	9

b LOGICAL ORGANIZATION AND TIMING - Figure 6-12 includes a detailed block diagram of the data channel and a timing chart for data channel operations. The block diagram shows the control section and the registers as well as connections from the data channel to the computer and to an external device. If a line in the figure represents more than one physical connection, the plurality is indicated by a number in parentheses. For example, transfer of incoming data from DB through HSBM to MB requires 18 lines. Diamonds and arrowheads on the lines indicate signal polarities.

The data channel logic is shown between the vertical dashed lines in the block diagram. At the left are connections to the various sections of the computer. The figure lists both signal names and pin connections on in-out plugs. Listed at the right are signals provided to the external device and signals that must be provided by the device. Some choice is available in both data and control signals, depending upon the mode of information transfer. The figure lists only mounting panel connectors because actual connections made to in-out plugs depend upon the type of signals required.

The program initiates data channel operations by specifying the number of words to be transferred, an address for the initial HSC access to memory, and up to eight additional bits of control information. Both word counter and location counter are 14 bits in length, so the maximum number of words that may be transferred in any single block is 2^{14} and these words must be stored in or retrieved from the lower numbered 2^{14} memory locations. Besides loading LC and WC the program may also load up to eight bits of control information into two 4-bit registers, A and B. The state of A_0 determines whether HSC access shall transfer data into the computer or out to the device (A_0^1 = IN). The other seven bits may be used for any control purposes as required by the external device.

The program initiates data channel operations through the following pair of iot instructions:

Set Channel Word Counter .....

.... scw Instruction Code 72X057 C(IO_{0-3}) replace C(A); the complement of C(IO_{4-17}) replace C(WC). This instruction also clears two of the three DC control flip-flops: TRANSFER DONE and WCOV (see below). The bit transferred into A_0 determines the direction of information transfers. All four A bits are available to the external device for control purposes. WC receives the complement of the number contained in IO_{4-17} because, in PDP-1 binary arithmetic, the complement of a number is equivalent to its negative. Then as each word is processed WC can count up normally, through negative numbers, instead of counting down. When the counter reaches 0 (i.e. WC overflows, WCOV) DC control terminates the block transfer. Since the maximum, when interpreted as a negative number, is -0 (all 1s) the counter must count through -0 in order to reach +0 (all 0s). Thus in order to cause the data channel to process a block of n words the program must specify the number n-1 in IO_{4-17}.

Set Channel Initial Location .....

.... sci Instruction Code 72X157 C(IO_{7-3}) replace C(B); C(IO_{4-17}) replace C(LC). If the preceding scw specified an output operation (A_0 = 0), sci also requests initial HSC access to memory to make the first word available to the device. For input operations sci signals the device that DB is ready to receive data. LC receives a memory address for the first HSC access from IO_{4-17}. As each word is processed the counter is incremented by 1 so that each HSC access is made to the next consecutive memory location.

The iot command pulse inputs to the data channel are shown at the lower left of the block diagram. SCW_7 clears both A and WC; then SCW_{10} transfers IO_{0-3} into A and the complement of IO_{4-17} into WC. Similarly, SCI_7 clears B and LC, while SCI_{10} transfers C(IO)

into them. SCI_{10} also triggers operations in DC control; all further events then occur only on signal from the external device.

For output operations the 0 state of A_0 negates the HSC channel-in signal and SWC_{10} sets the HSC request flip-flop causing an initial request for HSC access to memory. The location counter provides the address for each such access to HSAM. As soon as an HSC request is granted, the WORD XFER pulse from HSC control (paragraph 6-10a) triggers a chain of time pulses within DC control.

Two of these time pulses control the transfer of outgoing data from computer memory to DB. The first time pulse clears the buffer; the second transfers data into it after the computer memory cycle has made the data available at the memory buffer. At the same time DB is cleared, DC control signals the device that outgoing data is almost ready. Then, after allowing time for DB to settle, control signals the device that data is available. The data channel then waits until the device signals it to load the next data word by again requesting HSC access to memory.

For input operations, the 1 state of A_0 asserts the channel-in signal and causes SCW_{10} to trigger the chain of time pulses. No HSC access is yet requested. DC control then clears the data buffer and signals the device that DB is almost ready to receive incoming data. After allowing time for DB to settle down, DC control signals the device that the buffer is available. The device must then assemble a complete 18-bit word in the data buffer. This may be done in any one of four modes.

The A and B modes both require 18 data lines. In the A mode the device must provide two transfer pulses to assemble a word from two 9-bit characters. The B mode requires three transfer pulses to assemble a complete word from three 6-bit characters. The C mode requires only one data line and one control line, and the data buffer functions as a shift register. The device must provide 18 transfer pulses on the single control line, each one of which transfers a single bit of information into DB_{17} and shifts the contents of DB one place to the left. The device may also transfer a full 18-bit word by providing both A-mode transfers pulses or all three B-mode pulses simultaneously.

After assembling an entire word in DB, the device signals DC control to deposit the data word by requesting HSC access. When the request is granted WORD XFER again triggers the time chain. LC provides the memory address to HSAM while DB provides the incoming data to HSBM. After HSC control has transferred the data to MB, DC control again clears DB and signals the device that it is once more available for input.

When each HSC request is granted, WORD XFER triggers the DC timing chain and increments both the location counter and the word counter. Incrementing LC in every HSC cycle assures that access will be made to consecutive memory locations. As each word in the block is processed, the WC is also incremented. Finally, the transistion of WC from all 1s to all 0s sets the overflow flip-flop WCOV preventing further HSC requests. The transition also produces a completion pulse, indicating that the block transfer is finished, and sets the transfer done status flip-flop. These signals allow the computer to initiate a sequence break if the sequence break system is on. WCOV and TRANSFER DONE both remain in the 1 state until the program again initiates DC operations with an scw instruction.

Complete timing for data channel operations is shown at the right in Figure 6-12. The DC timing chain includes six pulses spaced 1 microsecond apart. The upper two portions of the chart show the standard memory cycle and HSC timing as related to the DC timing chain. In order to gain access to a particular computer cycle DC must establish both the memory address and the HSC request 1 microsecond before the cycle begins. The lower portions of the chart show the timing of data channel transfers and DC signals to the device. Note that in order to operate at the maximum 200-kilocycle word transfer rate (one transfer every 5 microseconds) the device must signal DC to request HSC access by the 2.7 microsecond time in every computer memory cycle.

The data channel equipment also includes an indicator panel (Figure 6-13). The register indicators on this panel show the contents of A, B, WORD COUNTER, LOCATION COUNTER and DATA BUFFER. Note that the WORD COUNTER lights are connected to the 1 outputs of the flip-flops but the register receives the complement of the number of word transfers specified by IO. Thus, to determine the number of transfers remaining in the block, read the counter indicators in reverse - i.e. an off light represents a 1 while an on light represents a 0. The panel also includes indicators for the three control flip-flops: REQUEST, TRA DONE (transfer done) and WC3. The last indicator represents the state of WCOV - that is, the overflow from WC_4, the most significant bit of the word counter.

If the equipment is mounted in a bay separated from the rest of the computer (and thus includes an 813 power control), power is applied to the system by pushing to the left the POWER switch located in the lower left corner of the panel. Located above the control flip-flop indicators is the POWER light.

c DATA CHANNEL LOGIC - Figure 6-14 shows the two 4603 pulse amplifiers that must be added to iot control when a data channel is installed in the computer (for module locations see Figure D9-7). The two iot instructions scw and sci use secondary operation codes 057 and 157 respectively. Neither instruction requires an in-out wait; therefore, both must use primary op code 72. Each instruction generates two command pulses at TP_7 and TP_{10} (for decoding of iot instructions see paragraph 9-2).

The complete data channel logic is shown in three D-size block schematics, Figures D6-15, -16 and -17. Figure D6-15 shows the DC control logic. Control inputs from the computer are the four iot command pulses at the upper left. The two TP_7 pulses merely clear DC counters and control registers. In addition to loading A and WC, SCW_{10} clears TRANSFER DONE and WCOV (A2, D2). SCI_{10} loads B and LC and also performs other functions depending upon the direction of data flow.

Whether DC is to control transfers of data into or out of the computer is determined by the state of A_0 (the buffered outputs of this flip-flop are labeled BA_0). For outgoing transfers (A_0^0), SCI_{10} sets the REQUEST flip-flop at the same time that it loads the registers (C3, C2). When access is granted, HSC control returns WORD XFER (paragraph 6-10a) which directly increments both the word counter and the location counter (B1). Through a PA (B1), WORD XFER also clears REQUEST and, if WCOV is 0, triggers the timing chain (A3). The chain is is produced by a series of six 4604 pulse amplifiers. External connections are made on these PAs so that each produces a 1-microsecond negative output pulse. The leading edge of a given output pulse is the negative-going time pulse while the trailing positive-going edge 1 microsecond later triggers the next PA to produce the next pulse in the chain.

The first time pulses (T0, T1, T2) are not used within the DC logic but are available to an external device if required. DC waits for the completion of the read portion of the HSC cycle. Then T3 clears DB and indicates to the device that outgoing data is almost ready (B3). Next, T4 transfers the contents of MB into DB and, after allowing 1 microsecond for the buffer to settle, T5 signals the device that data is ready. After the device has retrieved data from DB it signals DC control to load the next data word by setting REQUEST (C3).

For input operations, A_0^1 asserts CHANNEL IN (C1) and causes SCI_{10} to trigger the timing chain when it loads B and LC. DC then waits until HSC control has transferred incoming

data to MB. At T3, DC clears DB and signals the device that the buffer is almost ready to receive incoming data (B4). Finally, after waiting 1 microsecond for DB to settle, DC signals the device that the buffer is available at T4.

The device must then transfer data into DB by making data available to the DB input gates and generating the appropriate transfer pulses. Each pulse generated by the device is applied to a pulse amplifier to generate a corresponding local transfer pulse (A6, B6). In the A mode the device must generate two transfer pulses to assemble a complete 18-bit word from two 9-bit characters. The B mode requires three pulses to assemble a word from three 6-bit characters; the C mode requires 18 pulses on a single line to shift information into the buffer from the right, one bit at a time. After assembling the entire word, the device signals DC to deposit the word by setting REQUEST (C3).

Transfers continue with one HSC access after another until the word counter is incremented to all 1s and then recycles to all 0s. Since the most significant bit WC_4 can change from 1 to 0 only when the entire counter changes from 1s to 0s the overflow flip-flop WCOV is set on the transition of WC_4 (D3). The negative-going transition of the WC_4 0 output is used as a pulse in a capacitor-diode gate. The negative gating level is provided by the WC_4 1 output. The delay inherent in a capacitor-diode gate prevents the gating level from falling during the state change of the flip-flop. $WCOV^1$ holds REQUEST in the 0 state preventing any further requests for HSC access to memory. To prevent any spurious HSC requests, WCOV is also set by the start clear pulse when computer operations are begun. The same pulse that sets WCOV also generates the completion pulse BTD (B2). This pulse initiates a sequence break if such breaks are being allowed by the program, and also sets the status flip-flop TRANSFER DONE.

Figure D6-16 shows the word counter, the location counter, and the two control registers A and B. Instruction sci transfers a full 18-bit word from IO into B and LC (IO_{0-3} into B, IO_{4-17} into LC). All IO bits are available with the 1 state at ground assertion. However, bits 4 to 17 are inverted to provide the necessary input polarity to LC.

Instruction scw transfers the contents of IO into A and WC (IO_{0-3} into A, IO_{4-17} into WC). However, the transfer into WC is not a normal 1 transfer. The IO bits are available with the 1 state asserted at ground but bits 4 to 17 are <u>not</u> inverted at the WC inputs. Thus, the IO_{4-17} input bits are actually 0s asserted negative; and a given WC bit is set

if the corresponding IO bit is 0. This means that the transfer into WC produces the complement of the number originally contained in IO. Since in PDP-1 binary arithmetic the complement is used to represent the negative, WC is counted from a negative number up to 0.

The 1 outputs of all LC bits are available through bus drivers to HSAM to provide a memory address for HSC access. Both outputs of A_0 are buffered for use in DC control to specify the direction of data flow. Both outputs of all eight A and B bits are available to the external device for general control purposes. The transition of WC_4 from 1 to 0 signals the transition of WC from -0 to +0. No other WC outputs are used except in counter internal connections.

Both LC and WC are composed of type 4215 4-bit counter modules. At the beginning of every HSC access to memory WORD XFER increments both counters by complementing their least significant bits (LC_{17} and WC_{17}). Connections from one 4215 flip-flop to the next cause any given flip-flop to be complemented on the ground-going transition of the preceding flip-flop 1 output. Thus, whenever a given bit changes from 1 to 0, a carry is generated which complements the next more significant bit.

Besides counters and control registers the data channel also includes an 18-bit data buffer DB (Figure D6-17). The buffer gating includes both 1 transfer gates for normal parallel transfers and left shift gates for shifting data into the register one bit at a time from the right. Before any transfer in, DC control clears the buffer at T3 (in the middle of every HSC cycle). For output, DC control loads DB from the memory buffer by means of an 18-bit parallel transfer. For input both the data gating levels and the transfer pulses must be provided by the external device. The device may assemble a full word in DB either by two transfer pulses for 9-bit characters (A mode) or three transfer pulses for 6-bit characters (B mode). For the C mode the device must assemble a full word from 18 single bits by pulsing the shift left line 18 times. At each shift a new bit of data must be made available by a gating level at DB_{17} (A8). The bottom set of gates below the buffer may be used for special applications (such as sense amplifier outputs) in which single bits of data are provided to each buffer bit individually. This type of transfer requires a separate pulse and gating level for every bit.

The 1 outputs of all DB bits are made available through bus drivers for data transfers to the external device or through the HSC buffer mixer to memory.

CHAPTER 7

ARITHMETIC UNIT

7-1 GENERAL

The standard arithmetic unit includes three full-word (18 bit) registers (two active and one passive) and associated control circuits. The one passive register, the memory buffer register, is described in detail as part of the memory system (paragraph 8-3). The two active registers, the accumulator and the in-out register, are described in this chapter.

The control elements associated with the arithmetic unit are the overflow logic and the logic nets that generate the various control pulses for the accumulator and the in-out register. In addition to describing the standard arithmetic-unit equipment, this chapter also describes one central processor option. This option, the automatic multiply/divide logic, is a control system that allows multiplication and division to be performed as single instructions instead of as subroutines.

The arithmetic elements described in this chapter are shown in four logic drawings, figures D7-1 through D7-4. For information on the use and organization of these drawings see paragraph 3-16.

7-2 ACCUMULATOR

The accumulator is made up of 18 type 1201 flip-flops (figure D7-1). Each of these flip-flops has two gated complement inputs, in addition to the usual gated 0 and 1 inputs and direct clear input. A complement output terminal is associated with each complement input. When a positive-going pulse is applied to one of the complement inputs, an output pulse is available at the corresponding complement output terminal. However, only one of the complement outputs is actually used. No output connection is made to the complement output terminal corresponding to the complement input at the left (pin J). This input receives both the regular ungated complement pulse for the entire register and the gated partial-add complement pulse.

The complement input on the right (pin L) produces complement output pulses at pin M when it receives gated complement pulses. These gated pulses are generated by either the main carry function or the bit-to-bit carry from the next less significant stage of the register. The associated

pulse output produces the carry into the next more significant stage of the register.

The bit-to-bit or "ripple" carry is an addition carry. The pulse output from AC_n is applied to the complement input of AC_{n-1} only if AC_n changes state from 1 to 0. The initial input to the ripple carry chain is the add-1-to-AC pulse applied to the pin L complement input of AC_{17}. The effect of the standard add-1 pulse ($\xrightarrow{+1}$ AC) is duplicated by the pulse CARRY. This pulse is the end-around carry produced from AC_0 when the sign bit changes from 1 to 0 (from plus to minus). The end-around carry is a necessary characteristic of 1's complement arithmetic (see e below).

All but two of the accumulator control pulses are applied to the input gates of all bits in the register. The two exceptions are the ripple carry and the 1 transfer from MB to AC. The ripple carry output for each bit is applied only to the next more significant bit of the register. The 1 transfer from MB to AC is divided into two control pulses, $MB_{0-5} \xrightarrow{1}$ AC and $MB_{6-17} \xrightarrow{1}$ AC.

The level gates for transfers from other registers are applied to each bit of the accumulator from the corresponding bit of the source register. Each bit of the accumulator also receives the outputs of the stages at either side as level gates for shift/rotate operations. The shift/rotate control levels (paragraph 6-5b) are applied only to the sign bit and the least significant bit of the register (i.e. bits AC_0 and AC_{17}) to control register linkage in shift/rotate operations.

The outputs of the accumulator flip-flops are used by the accumulator input gating for both shift/rotate and arithmetic operations. These outputs are also applied to the jam transfer input gating of the memory buffer to deposit the contents of the accumulator in memory. If either the visual display option or the precision display option is included with the computer, the 0 outputs of certain bits of the accumulator are made available, through type 4113R buffers, at taper pins in in-out transfer control. For the visual display, bits AC_{0-9} are available; for the precision display bits AC_{0-11} are available. If the type 52 tape control is included with the computer, bits AC_{15-17} are also made available, through type 1685 bus drivers, for addressing the tape units.

The four types of accumulator input gating are descirbed in a through d below. These four types of gating are the transfer, shift/rotate, logic, and arithmetic gating, respectively. The addition algorithm used in the accumulator logic is described in e below.

a TRANSFER GATES - Transfers to the accumulator are made from the program counter,

from the console TEST WORD switch register, and from the memory buffer. The first two of these transfers are 1 transfers requiring a prior clear. The transfer from the memory buffer, however, is a jam transfer; hence no clear is necessary.

The contents of the program counter are transferred into the accumulator through the bottom row of input gates (figure D7-1). The 12-bit address in the program counter is transferred into AC_{6-17} during a sequence break and during any jump instruction that saves the contents of the program counter. Although only bits 6 to 17 are required for PC, the pulse $PC \xrightarrow{1} AC$ is applied to the input gates of all bits in the accumulator.

At the same time that the program address is transferred into AC, the state of the overflow flip-flop OV_1 is saved in AC_0. If the computer includes extra memory modules, the contents of the extend flip-flop EXD and of the extension of PC (which contains the module address) are transferred into AC_1 and AC_{3-5}, respectively. Bit AC_2 is available as a spare extension bit in case the memory is expanded beyond eight modules.

For sequence break operations the states of OV_1 and EXD are saved because these two flip-flops may be affected during execution of a break routine. This permits these flip-flops to be returned to their original states when the computer resumes the interrupted sequence.

The pulse $TW \xrightarrow{1} AC$ is applied to all bits of the accumulator. This pulse transfers a full 18-bit word from the console TEST WORD switch register into the accumulator. A test-word transfer may be initiated either from the console or by the program.

Transfers from the memory buffer to the accumulator are controlled by three transfer pulses. One of these is a 0 transfer which is applied to the 0 input gates of all bits in the register. The other two pulses are 1 transfers, which are applied to the 1 input gates of AC_{0-5} and AC_{6-17}, respectively. Whenever a full word is transferred from the memory buffer to the accumulator, all three transfer lines are pulsed. This results in a jam transfer of all bits from MB into AC; no prior clear is necessary.

The 0 transfer and full 1 transfers are used alone only in certain logical operations (c below). The 1 transfer into AC_{6-17} is used alone in the instuction Load Accumulator with N (where N is the address portion of the instruction word). After this transfer is completed, bits AC_{0-5} always contain 0, because the accumulator must be cleared prior to the transfer.

The MB $\xrightarrow[0-5]{1}$ AC line is never pulsed alone.

b SHIFT/ROTATE GATES - Two shift/rotate pulses are applied to the accumulator input gating. One of these pulses shifts the contents of the accumulator one place to the left; the other shifts them one place to the right. A one-place shift occurs each time one of these lines is pulsed.

The effect of the shift/rotate pulses on bits AC_{1-16} is always the same. On a left shift the contents of AC_n are transferred into AC_{n-1}. Similarly, on a right shift the contents of AC_n are transferred into AC_{n+1}. The effect on the ends of the register (AC_0 and AC_{17}), however, depends on the operation in which the shift/rotate pulses occur.

Two types of instructions generate shift/rotate pulses: the shift group instructions and the instructions for multiplication and division. Within the shift group there are two classes of shift operations: the cyclic shift and the arithmetic shift. The cyclic shift, or rotation, is a nonarithmetic shift which includes the sign bit. The arithmetic shift multiplies or divides the number in the accumulator by 2, but does not affect the sign bit. The division shift and the multiplication shift are variations of the cyclic and arithmetic shifts, respectively.

In either case the effect on bits AC_0 and AC_{17} depends on whether the accumulator is shifted alone or together with the in-out register. When both registers are shifted together, the in-out register is treated as a less significant extension of the accumulator. For arithmetic shifts of both registers, AC_{17} is linked to IO_0. For double-length cyclic shifts, AC_0 and IO_{17} are linked together, as well as AC_{17} and IO_0. The linking of these two registers is governed by the shift/rotate control levels (paragraph 6-5b).

When the accumulator is rotated alone, the sign bit AC_0 is linked with AC_{17}. A left rotation transfers the contents of AC_0 into AC_{17}, while a right rotation transfers the contents of AC_{17} into AC_0.

In an arithmetic shift the order of magnitude of the number in the accumulator is changed. However, except for bits that are lost by being shifted out of the register, the significant bits of the number remain the same. If the number is positive, 0's are shifted into bits vacated by the significant bits of the number. For negative numbers, 1's complement arithmetic requires that the vacated bits of the shifted number be replaced by 1's. Because

the state of the sign bit is 0 for positive numbers and 1 for negative numbers, the sign bit itself is shifted into the vacated positions in the register. The state of AC_0 remains unchanged throughout the entire arithmetic shift. The multiplication shift varies from the standard arithmetic shift in that the sign bit, AC_0, is automatically cleared.

The shift/rotate input gating to AC_0 is shown in figure D7-1A2. Information is transferred into AC_0 from AC_1 on a left shift only if the computer is performing a rotate instruction or if the shift occurs as part of a division. On a right shift, information may be transferred into AC_0 from either of two sources: from AC_{17} in Rotate Accumulator Right, or from IO_{17} in Rotate AC and IO Right. If the right shift occurs as part of the automatic Divide instruction or as part of either the Multiply Step or the automatic Multiply instruction, then AC_0 is automatically cleared.

The shift/rotate input gating to AC_{17} is shown in figure D7-1C8. On all right shifts, the contents of AC_{16} are automatically transferred into AC_{17}. In any left shift operation involving only the accumulator, bits are shifted into AC_{17} from AC_0. Left shift/rotate pulses which occur as part of multiplication or division, or in the instructions rcl or scl, transfer the contents of IO_0 into AC_{17}.

c LOGIC GATES - The computer can perform four logic functions. These are: logical negation, AND, inclusive OR, and exclusive OR.

The logical negation of a word in the accumulator is performed by the complement pulse. This pulse directly complements each bit of the accumulator by pulsing the noncarry complement input. In 1's complement arithmetic, logical negation is equivalent to the arithmetic negative. If the contents of the accumulator are interpreted as a number, the complement pulse produces the negative of that number.

The other three logic pulses produce the logic function of two words, one from the memory buffer and the other from the accumulator. The result appears in the accumulator. Two of these logic functions are produced by utilizing a transfer pulse without a prior clear.

If 1's are transferred from MB to AC without first clearing AC, the result is the inclusive OR function of the contents of MB and the original contents of AC. On the other hand, if AC already contains a word and the 0's from MB are transferred into AC, the accumulator then contains the AND function of the contents of MB and the original contents of AC.

The third two-term logic function is produced by the partial-add pulse. This pulse is the first of the pair of pulses which produce addition in the accumulator. Partial addition is equivalent to the exclusive OR function. The partial add complements a bit of the accumulator if the corresponding bit of the memory buffer contains 1. Thus the final state of a bit of AC depends upon the initial state of that bit and the state of the corresponding bit of MB as follows:

MB	$AC_{original}$	AC_{final}
0	0	0
0	1	1
1	0	1
1	1	0

The final state of a given AC bit is thus the exclusive OR of the corresponding MB bit and the original state of the given AC bit. The final state is 1 if either (but not both) of the original bits were 1; i.e. if the original bits were not alike.

d ARITHMETIC GATES - The accumulator includes two addition gating systems. One of these is the standard gated bit-to-bit carry chain that allows the accumulator to function as a counter. The chain begins with the add-1-to-AC control pulse applied to the complement input of AC_{17}. This pulse ripples through the accumulator because a pulse applied to the complement input produces a pulse output which is in turn applied to the carry chain gate of the next more significant bit of the register. Each bit is complemented if the next less significant bit changes from 1 to 0. Thus, when the beginning of the carry chain is pulsed, the carry ripples as far through the register as is required to add 1 to the integer contained in the register.

The other arithmetic gating system allows full-word addition to the contents of the accumulator. This system includes two sets of gates which receive pulses applied to all bits in the register, and it also utilizes the regular bit-to-bit carry chain. The ripple chain, however, is not necessarily initiated at the least significant bit of the register. Instead the ripple carry may be initiated at any point in the register, adding 1 to the least significant bit in the section of the register to which the pulse is applied.

The addition operation is carried out in two stages. The first is a partial addition; the

second is a carry function. The partial addition is performed by the partial-add pulse which produces the exclusive OR function of the contents of MB and the contents of AC (c above). After the partial sum has been formed the full-register carry pulse changes the exclusive OR into the true arithmetic sum. At the end of the operation the number represented by the state of the accumulator is the sum of the original contents of the accumulator plus the contents of the memory buffer.

The partial sum (that is, the result of the partial addition) is equal to the true arithmetic sum for any bit which does not receive a carry. If the sum of two binary numbers is considered on a bit-by-bit basis, the exclusive OR function of the two numbers is actually the correct sum in a given bit if there is no carry into that bit. For example the sum of two 0's is 0. The sum of 1 and 0 is 1. The partial sum, i.e. the exclusive OR function, of two 1's is 0. (The last example, however, requires that there be a carry into the next more significant bit).

A given bit of the partial sum is valid as a bit of the true arithmetic sum provided that no carry into the bit is present. But if there is a carry into the bit, then that bit of the partial sum is the opposite of the correct bit of the arithmetic sum. After the partial sum is produced, the full-register carry function changes the states of all those bits of the partial sum which are incorrect as bits of the true arithmetic sum.

The full-register carry pulse $\overset{\text{CARRY}}{\longrightarrow}$ AC produces the correct carry function both by complementing certain bits in the accumulator and by initiating the ripple carry at those bits that it complements. The full-register carry complements (i.e. carries into) a bit of the partial sum if the next less significant bits of the summands were both 1.

The computer cannot sense the previous state of a flip-flop, so instead it senses the corresponding configuration of the partial sum. This produces the same result because of the following reason. The partial add changes the state of AC_n only if MB_n is 1. If after the partial addition there is a 0 in AC_n and a 1 in MB_n then both bits must originally have been 1. The carry therefore complements AC_{n-1} if MB_n is 1 and AC_n is 0.

At each stage, the complement pulse produces a pulse output which complements the next more significant stage of the accumulator if the complemented bit changes from 1 to 0. Thus the carry initiated by the partial addition of two 1's ripples up the register until a 0-bit is complemented. A complemented pulse from either type of carry function complements

a bit whether it is 0 or 1, but a 0 inhibits the ripple carry from propagating to the next bit.

The arithmetic gating therefore allows addition to be performed by a pair of control pulses in the following way. The partial add produces in the accumulator a partial sum which is the exclusive OR function of the contents of MB and AC. The full-register carry then initiates the ripple carry at each place in the partial sum where the next less significant bit is a 0 resulting from the partial addition of two 1's. Each ripple carry propagates as far as is necessary to produce the correct sum of the contents of MB and the original contents of AC. That this algorithm does in fact produce the correct sum of two binary numbers is proved in e below.

e ADDITION ALGORITHM - Let A be the original contents of the accumulator; B the contents of the memory buffer; PS the partial sum produced in AC by the partial addition; and S the arithmetic sum of A and B. For convenience let A and B be positive binary fractions whose sum is less than 1, i.e. there is no overflow.

A bit of the partial sum PS_n is equal to a bit of the sum S_n if and only if there is no carry into S_n. If there is a carry into S_n then PS_n is equal to the complement of S_n. Since both signs are plus and there is no sign change, there is no carry into PS_{17}; therefore $PS_{17} = S_{17}$.

Divide PS into sections from the right so that the first section starts with PS_{17} and ends at the first bit PS_k which satisfies the conditions $PS_k = 0$, $A_k = B_k = 1$. The second section starts with PS_{k-1} and extends to the next bit that satisfies the same conditions as PS_k. Proceed in this way through the entire partial sum.

The least significant bit of the partial sum must be correct since there can be no carry into it. If this bit is 1, or if it is a 0 resulting from the partial addition of two 0's, then there is no carry out. If no carry exists, the next bit of the partial sum is also correct. Proceed with each more significant bit of the partial sum until a bit PS_k is reached which is 0 resulting from the partial addition of two 1's. Bit PS_k is also correct; therefore all the bits of the partial sum in the first section are correct.

Because the partial sum in PS_k generates a carry, PS_{k-1} is not correct. A 1 from the first section is carried into PS_{k-1} by the full-register carry $\xrightarrow{\text{CARRY}}$ AC. If PS_{k-1} is 1 (resulting from the partial addition of 0 and 1), then there must be a carry into PS_{k-2}. This

carry is provided by the ripple carry which complements PS_{k-2} when PS_{k-1} changes from 1 to 0.

The ripple carry thus propagates up the register until a 0 bit is encountered. If this 0 is the result of the partial addition of two 0's, then no further carry is generated. All further bits are correct up to the next 0 that results from the partial addition of two 1's; i. e. up to the end of the section.

If the 0 that terminates the ripple carry results from the partial addition of two 1's, then there must be a carry into the next bit. However, the partial addition of two 1's is the condition that ends the section. The full-register carry therefore begins a new ripple carry in the next section. Consequently, the carry operation complements all incorrect bits of the partial sum. At the completion of the carry operation the result S is the correct sum of A and B.

The preceding example shows that the addition algorithm works for the special case of two positive numbers. Before proving the algorithm for the remaining cases (including negative operands) four further facts that are necessary for the discussion must be mentioned:

1) The sign bits are included in the partial addition; that is, the partial sum of two minus signs (1's) is a plus sign (0).
2) Both carry functions apply to the accumulator sign bit.
3) The full-register carry complements AC_{17} if two negative numbers are added.
4) If the sign changes from minus to plus in the carry operation, the ripple carry propagates into AC_{17}. This is the end-around carry.

Assume that the binary point is to the left of the most significant bit, that is, that all computer numbers are 17-bit fractions. The computer representation of the positive number x is therefore

$$+ .[x]$$

where the brackets enclose the number contained in AC_{1-17}. The sign of this number is contained in AC_0.

In 1's complement arithmetic the negative of a number is produced by subtracting the number from a number that is all 1's. This is done by changing the sign and subtracting the magnitude from $(1 - 2^{-17})$. The computer representation of the number -x is therefore

$$-. [1 - x - 2^{-17}]$$

Consider two positive 17-bit fractions, x and y. There are four possible cases of addition:

1) $x + y$

2) $(-x) + (-y)$

3) $x + (-y)$; $y \leq x$

4) $x + (-y)$; $y > x$

(1) This is the case discussed in the basic description of the addition algorithm. Addition is as follows:

$$+ .[x]$$
$$+ .[y]$$

$$+ .[x+y]$$

If $(x + y) \leq 1$, the overflow changes the sign during the carry operation. If the addition of two positive numbers results in a negative answer, the sum has exceeded the capacity of the accumulator. When such an overflow occurs, the contents of AC represent the number

$$-.[x + y - 1]$$

(2) The partial addition and all carry operations not involving the signs result in the following:

$$-.[1 - x - 2^{-17}]$$
$$-.[1 - y - 2^{-17}]$$

$$+.[1 + 1 - x - y - 2^{-17} - 2^{-17}]$$

The partial addition of two 1's in the sign bit causes the full-register carry to complement AC_{17}, adding 2^{-17} to the contents of the accumulator. If $(x + y) < 1$, the carry overflows into the sign bit. The complete result is

$$-.[1 - (x + y) - 2^{-17}]$$

which is the computer representation of $-(x + y)$.

If $(x + y) \geq 1$, there is no carry into the sign bit. In this case (two negative operands), the absence of a carry into the sign bit indicates that the result of the addition overflows, that is, that the sum exceeds the capacity of the accumulator. Consequently, if the addition of two negative numbers results in a positive answer, it is evident that the result has overflowed. The accumulator then contains the number:

$$+.\ [1 - (x + y - 1) - 2^{-17}]$$

The sign is plus, and the magnitude is the complement of $(x + y - 1)$.

(3) Partial addition and all carry operations not connected with the signs result in the following:

$$\begin{array}{l} +.\ [x\ ] \\ -.\ [1 - y - 2^{-17}] \\ \hline -.\ [1 + x - y - 2^{-17}] \end{array}$$

Since $y \leq x$, it follows that $(1 + x - y) \geq 1$. Therefore the carry overflows into the sign and the sign change (- to +) causes an end-around carry. The complete result is

$$+.\ [x - y\ ]$$

(4) Partial addition and all carry operations not connected with the signs result in the following:

$$\begin{array}{l} +.\ [x\ ] \\ -.\ [1 - y - 2^{-17}] \\ \hline -.\ [1 + x - y - 2^{-17}] \end{array}$$

Because $y > x$, it follows that $(1 + x - y) < 1$. Thus there is no overflow or end-around carry and the above result is the computer representation of the negative number $x - y$, i.e. $-(y-x)$.

7-3 IN-OUT REGISTER

Data transfers between the central processor and all low-speed or programmed in-out devices are made through the in-out register. In systems including high-speed channels for data transfers between the memory buffer and a high-speed device (such as magnetic tape), the in-out register is still required for transfer of control information.

In addition to its function for in-out operations, the in-out register also serves as the multiplier-quotient register in the arithmetic unit. Because of this arithmetic function, the in-out register is constructed from 18 type 1204 flip-flops. Besides the standard direct clear input and gated 0 and 1 inputs, these flip-flops have a negative pulse complement input. These

complement inputs are not used in the standard computer, but must be available in case the automatic multiply/divide option is included in the system.

The in-out register includes a complete set of shift/rotate gates. The transfer gating allows transfers into IO from the memory buffer or through the input mixer from the peripheral equipment buffers. Both these transfers are 1 transfers requiring a prior clear. The generation of the in-out register control pulses is described in paragraph 7-4e.

a TRANSFER GATES - For arithmetic-unit operations and output operations, words are transferred into the in-out register from the memory buffer. For output operations the contents of IO are made available to the taper pins in in-out transfer control through type 1685 bus drivers.

During input operations, information is transferred from the in-out device buffers through the input mixer to the in-out register. Because the IO flip-flops have complement inputs, they have no direct set inputs. Instead, information is transferred into IO by applying the single-bit negative data pulses from the input mixer to grounded-emitter input gates.

The contents of IO (either the results of arithmetic operations or information brought in from a peripheral device) can be transferred to the memory buffer for deposit in memory.

b SHIFT/ROTATE GATES - The shift/rotate gating system of the in-out register is similar to the accumulator shift/rotate gating system (paragraph 7-2b). There are two shift/rotate pulses, one for left shifts and the other for right shifts. Bit positions vacated during an arithmetic shift are filled from the contents of the sign bit, either IO_0 or AC_0. As in the case of the accumulator, the IO shift/rotate pulses produce a normal bit-to-bit jam transfer in IO_{1-16}. Variations in the shift/rotate operations occur at the ends of the register (bits IO_0 and IO_{17}), depending on the type of operation and on the linkage of IO with AC.

The shift/rotate input gating of IO_0 is shown in figure D7-2A2. In shift/rotate operations IO_0 is unaffected only during arithmetic shifts of IO alone. During cyclic shifts of IO alone, information is gated into IO_0 from IO_1 on a left rotation and from IO_{17} on a right rotation.

During arithmetic or cyclic shifts of AC and IO together, IO_0 receives the contents of AC_{17} and IO_1 on right or left shifts, respectively. This type of double-length shift also occurs in

multiplication and division. In all double-length operations IO is considered to be an 18-bit magnitude extension of AC containing no sign.

The shift/rotate input gating of IO_{17} is shown in figure D7-2C8. During all shift/rotate operations to the right, the contents of IO_{16} are shifted into IO_{17}. On left shift/rotate pulses, the data shifted into IO_{17} may come from several sources. If IO alone is shifted or rotated to the left, the contents of IO_0 are transferred into IO_{17}. When IO and AC together are shifted or rotated to the left, IO_{17} receives the contents of AC_0. However, in a double-length shift that occurs as a part of division, the complement of the contents of AC_0 is loaded into IO_{17}. (If AC_0 is 0, then IO_{17} is set; if AC_0 is 1, then IO_{17} is cleared.)

7-4 ARITHMETIC UNIT CONTROL

Arithmetic unit control includes all the logic nets that generate the control pulses for the accumulator and the in-out register. Also included is the overflow logic and a group of diode decoder nets that allow the computer to sense certain states of the accumulator.

a ACCUMULATOR DECODERS - The diode nets that sense certain states of the accumulator are shown at the lower right of figure D7-3. The net in C8 generates the level AC = +0 when every flip-flop of the accumulator is in the 0 state. For several of the skip instructions the accumulator is sensed for zero contents by the program count logic (paragraph 6-4c).

The diode net in D8 senses for either -1 or -0 in the accumulator. If all bits of AC except the least significant bit contain 1's, then the number in AC is either -1 or -0. The decoder output level is combined with the states of the least significant bit AC_{17} to determine which of these numbers is actually contained in AC. The accumulator is sensed for -1 to control indexing operations; it is sensed for -0 at the end of certain arithmetic instructions. If an addition or division results in an answer of -0, the answer is changed to +0.

b ACCUMULATOR TRANSFER LOGIC - The logic nets that generate the eight accumulator transfer pulses are shown in figure D7-3. These eight pulses include the clear pulse, five pulses that transfer information into AC from other registers, and two pulses that shift information to the right or left in the accumulator.

$MB \xrightarrow{0} AC$, $MB_{0-5} \xrightarrow{1} AC$, $MB_{6-17} \xrightarrow{1} AC$

When all three of these transfer lines are pulsed simultaneously an entire word is jam-transferred

from the memory buffer to the accumulator. This is done immediately after the retrieval of an operand from memory during cycle one of Load Accumulator or of either of the two index instructions. The jam transfer also occurs at the very end of an automatic Divide (MDP-13). In this latter case the quotient is jam-transferred to the accumulator at the same time that the remainder is jam-transferred to MB.

The 0 and 1 transfers are used separately to perform certain logical functions. In these cases the accumulator is not cleared prior to the operation. The 0 transfer, when used alone, produces the AND function of the contents of MB and AC. The full-word 1 transfer, when used alone, produces the inclusive OR function of the contents of MB and AC.

There is one case in which a 1-transfer pulse performs a regular information transfer. This utilizes only the single 1 transfer that loads MB_{6-17} into the corresponding bits of AC. This transfer occurs in the instruction Load Accumulator with N. The accumulator must be cleared before the transfer takes place.

$\overset{0}{\rightarrow}AC$

The accumulator is cleared prior to any transfer of information into AC from the program counter (figure D7-3C2). The clear pulse is also generated prior to a 1 transfer from MB (LAW) and prior to the transfer of a test word into the accumulator in the console operation Deposit. The only pulse generated by the instruction Clear Accumulator is the clear-AC pulse. However, the same condition ($OPR \cdot MB_{10}^{1}$), in the instructions lat and lap, also clears AC prior to transfers from TW and PC respectively.

The accumulator is also cleared at the end of cycle one of Multiply (after the multiplier has been transferred to the in-out register). The accumulator is then free to begin the formation of partial products in the automatic multiplication.

There are also several situations in which AC is cleared because it contains either of the numbers -1 or -0. The clear is generated during an index instruction if the contents of AC have been counted to -1 by the instruction (D2). The accumulator is also cleared if the number -0 appears in AC as the result of either of the instructions Add or Divide Step (C1).

$PC \xrightarrow{1} AC$

The program may transfer an address from PC to AC by the operate instruction Load Accumulator from Program Counter. The current program location is also saved in the

accumulator at the beginning of any sequence break and during the final cycle of any jump instruction that saves the program counter. The final cycle of JDA + CAL is cycle one. The final cycle of the non-memory reference instruction Jump and Save Program Counter is cycle zero if the instruction is not deferred, but is the final defer cycle if the instruction is deferred.

$TW \xrightarrow{1} AC$

The 1 transfer of a word to the accumulator from the console TEST WORD switches occurs in two situations: in the operate instruction Load Accumulator from Test Word; and in the console operation Deposit (B8).

AC SH/RO L

The contents of the accumulator are shifted one place to the left each time a SH/RO pulse is generated during the instruction Divide Step or during a shift group instruction which calls for a shift or rotation of the accumularor to the left. The accumulator is also shifted to the left at each step of an automatic Divide (MDP-1).

AC SH/RO R

The contents of the accumulator are shifted one place to the right at each SH/RO pulse that occurs either during the instruction Multiply Step or during a shift group instruction which calls for a right shift or rotation of the accumulator. In this logic net, the instruction Multiply Step is represented by the logical condition MUS + MUL. However, MUL is a don't-care condition.

The accumulator is also shifted to the right in both of the operations controlled by the automatic multiply/divide option. In Multiply the accumulator is shifted to the right by the multiply shift in each step of the operation (A7). In Divide the accumulator is shifted to the right only once at the end of the instruction (MDP-9) if the complete division has actually been performed (SCR_0^1).

c ARITHMETIC LOGIC - The logic nets that generate the four arithmetic pulses are shown in figure D7-3. The first two of these pulses, partial add and carry, perform the addition algorithm. The other two pulses are the complement pulse and the indexing pulse.

$MB \xrightarrow{PAD} AC, \xrightarrow{CARRY} AC$

When these two functions are pulsed in succession, first the partial add and then the carry,

the contents of the accumulator are replaced by the original contents of the accumulator plus the contents of the memory buffer. This two-stage addition operation occurs in cycle one of all four of the standard arithmetic instructions (B5). The partial addition is performed at TP_5; the carry function follows at TP_6. The operation is always performed in Add, Subtract, and Divide Step. The addition is performed in Multiply Step, however, only if IO_{17} is 1. This means that the addition of a partial product does not take place if the current bit of the multiplier is 0.

This two-stage addition operation is also used for the automatic multiplication and division. The partial addition occurs on multiply/divide pulses 3 and 7; the carry function is generated whenever the carry signal arrives from the multiply/divide logic (B7). In Multiply partial addition is caused by MDP-3; the carry always follows. However, MDP-3 is not generated on every step of Multiply. If the current multiplier bit is 0, MDP-3 is inhibited and no addition takes place. In Divide, the partial addition is generated by MDP-7 in every step of the automatic operation; but the carry follows only if certain other conditions are fulfilled. The automatic multiplication and division are described in detail in paragraph 7-5.

The partial addition, which is equivalent to the exclusive OR logic function, is also used independently of the carry in several nonarithmetic instructions. These are the logical instruction exclusive OR and the two program control instructions that compare the contents of the accumulator with the contents of a memory register. Partial addition can be used as a comparison function because if two words are identical every bit of their exclusive OR function is 0.

The command levels for these three instructions are applied to the same logic net that controls partial addition for the arithmetic instructions. As a result, the partial add occurs at TP_5 of cycle one. The complements of these command levels inhibit the carry function. In the two compare instructions the partial addition is performed a second time at TP_9. This second partial add restores the original contents of the accumulator.

$\overline{IC} \rightarrow AC$

The subtraction algorithm requires that the accumulator be complemented twice in the same memory cycle -- once at TP_4 prior to the addition, and again at TP_9 following the

addition. A subtraction is performed in cycle one of the instruction Subtract (D1). A subtraction is also performed in cycle one of Divide Step provided that the previously generated bit of the quotient is 1.

The other logical conditions that complement the accumulator are not part of the subtraction algorithm. These conditions produce the logical or arithmetic negative of the contents of the accumulator. The complement pulse is generated by the operate instruction Complement Accumulator (D2). It is also generated during LAW if a negative number is being loaded into the accumulator. If the computer includes the automatic multiply/divide option the accumulator is complemented whenever the complement signal arrives from the multiply/divide logic (D3).

$|\xrightarrow{+1}$ AC

The contents of the accumulator are incremented by 1 in indexing operations and in certain steps of a division. In cycle one of either of the index instructions (C3), 1 is added to the contents of the accumulator provided AC does not already contain -1. If AC does contain -1, it is cleared instead.

In certain steps of a division the number in the accumulator must be incremented by 1 because of the characteristics of 1's complement arithmetic. In the standard instruction Divide Step the incrementing occurs at the beginning of cycle one if the previously generated bit of the quotient is 0 (D3). In the automatic Divide the incrementing occurs whenever MDP-2 arrives from the multiply/divide logic (D4).

d OVERFLOW LOGIC - The logic elements that check for overflow in the accumulator are shown in figure D7-3, D5 to D7. The overflow logic includes two flip-flops, OV_1 and OV_2. The setting of OV_2 indicates that overflow can occur in an operation. The setting of OV_1 indicates that overflow has occurred.

If two numbers of opposite signs are added together, the magnitude of the result must be less than the magnitude of the larger number. As a result no overflow can occur when the signs are different. In cycle one of Add or Subtract, OV_2 is set at TP_5 if the sign bits of the memory buffer and the accumulator are the same. In Subtract the accumulator is complemented before TP_5. Therefore, the overflow check is made only on the addition part of the subtraction algorithm.

When two numbers of like sign are added together the result must also have the same sign. Therefore, after the addition algorithm is performed in Add or Subtract, the states of the sign bits are again checked. Flip-flop OV_2 is cleared if the signs of MB and AC are again the same. If the signs are not the same, indicating that overflow has occurred, OV_2 is not cleared. Consequently OV_1 is set at TP_{10}. The state of OV_1 can be sensed by one of the skip group instructions to determine whether or not the accumulator contains the correct result of an addition or a subtraction. Sensing OV_1 also clears it so that it is available for later use.

Both flip-flops are cleared inititially by SC. Furthermore OV_2 is also cleared in every cycle by TP_{10}. A sequence break may interrupt computations in the arithmetic unit, and the overflow flip-flop may be affected during a break routine. Therefore, during break cycle one, the contents of OV_1 are transferred to AC_0 at the same time that the program address is saved in AC. Then, so that OV_1 is available for the break routine, it is cleared at TP_{10}. After the break routine is completed the restoring pulse from the sequence break system returns OV_1 to its original state.

e IN-OUT REGISTER TRANSFER LOGIC - The logic nets that generate the five in-out register control pulses are shown in the lower left of figure D6-5. Information can be transferred into IO from only one other central processor register, the memory buffer. All transfers of information into IO from peripheral devices are made through the input mixer (paragraph 9-3). Transfers from either MB or the input mixer are 1 transfers requiring a prior clear. In addition to the clear pulse and the transfer-from-MB pulse the IO transfer logic includes two standard shift pulses (one left, the other right) and a complement pulse. The pulse amplifiers for the complement pulse are included in the computer only if the automatic multiply/divide option is installed.

$\xrightarrow{|0}$IO

The in-out register is cleared prior to any transfer into the register either from the memory buffer or from peripheral devices through the input mixer. Whenever IO is used for input operations it is cleared by the pulse $\xrightarrow{|0}$IO ON IOT. This clear pulse is generated by in-out transfer control (paragraph 9-2).

The in-out register is also cleared by the operate instruction Clear In-out Register (C1).

$MB \xrightarrow{1} IO$

In the standard computer, the in-out register is loaded from MB only in the instruction Load In-out Register. If the computer includes the automatic multiply/divide option, the $MB \xrightarrow{1} IO$ transfer is also made whenever the corresponding transfer signal arrives from the multiply/divide logic. This occurs once in each of the automatic instructions. In both cases it occurs as the second stage of a transfer from AC to IO, because all such transfers must be made via MB. The transfer $MB \Longrightarrow IO$ is made at the beginning of Multiply to transfer the multiplier from AC to IO. The same transfer is made at the end of Divide to transfer the remainder from AC to IO.

IO SH /RO L

The contents of the in-out register are shifted one place to the left each time a SH /RO pulse is generated during the instruction Divide Step or during a shift group instruction which calls for a shift or a rotation of IO to the left. The in-out register is also shifted to the left at each step of an automatic Divide (MDP-1).

IO SH /RO R

The contents of the in-out register are shifted one place to the right at each SH /RO pulse that occurs either during the instruction Multiply Step or during a shift group instruction which calls for a right shift or rotation of IO. The in-out register is also shifted to the right by the multiply shift at each step of an automatic Multiply.

$\overline{|C} \longrightarrow IO$

The In-out register is complemented whenever the complement signal arrives from the multiply/divide logic. This may occur at the beginning or the end of either of the automatic instructions. During these instructions, the complement pulse performs the necessary adjustments to the sign of the number in IO.

7-5 AUTOMATIC MULTIPLY/DIVIDE LOGIC

The logic circuits for the automatic multiply/divide option are shown in figure D7-4. The multiply/divide logic is a timing and indexing system that generates a series of transfer and arithmetic pulses. These output pulses are in turn applied to the standard logic nets in the computer to produce the appropriate operations within the arithmetic unit.

When the multiply/divide option is included in the machine, the standard instructions Multiply Step and Divide Step are replaced by the instructions Multiply and Divide, respectively. During cycle one of either of these automatic instructions, the regular timing system of the computer serves only to initiate the required sequence of operations. At the end of cycle one, the timing chain stops, and a substitute multiply/divide timing system takes over control of the computer. This substitute timing system is a chain of multiply/divide pulses MDP-1 to MDP-13. Neither instruction uses the entire chain, but the parts used by each instruction overlap.

In some cases the multiply/divide pulses are applied directly to the logic nets in the standard control unit and arithmetic unit. In many cases, however, the transfer pulses require additional gating, and these extra logic nets are shown in the multiply/divide drawing. The outputs of these logic nets are designated by the name of the appropriate transfer preceded by the letters "MD". The transfer pulse from the multiply/divide logic is then applied to the corresponding logic net in the control unit or the arithmetic unit to produce the desired transfer. For example, whenever a transfer from MB to IO is required for either of the automatic instructions, the multiply/divide logic generates the pulse MD:MB→IO. This pulse is applied to the IO transfer logic in the standard computer to generate the corresponding pulse $MB \xrightarrow{1} IO$.

Whenever any pulse produced by the multiply/divide logic is applid to a logic net in some other part of the computer, the number of the figure which contains the receiving logic net is written in an oval beside the output pulse in the multiply/divide figure. These figure numbers are written beside all the special pulses produced by the multiply/divide logic nets and also beside any of the regular multiply/divide pulses that produce operations outside the multiply/divide logic itself.

The multiply/divide timing chain is shown from left to right across the center of figure D7-4. The last few higher-numbered pulses are shown in the lower right of the figure. The logic nets that produce the special transfer pulses are shown at the left. Indexing for the automatic instructions is provided by the step counter (upper left); the contents of the step counter are decoded by the diode nets shown in the upper right.

Since all automatic computations must be performed on positive numbers, the original signs of the operands are stored in the flip-flops shown in A5 and A6. After the instructions are completed, the states of these flip-flops are sensed to determine the sign of the result. At certain stages in multiply/divide operations it is necessary to know whether the accumulator

contains either the number +0 or -0 and whether the in-out register contains the number +0. The decoding of the contents of AC for +0 or -0 is performed by the standard accumulator decoders shown in figure D7-3D8. To decode the in-out register for +0 contents, the two diode decoder nets shown in figure D7-4B7 must be added to the computer. These two diode plug-in units are not mounted in the multiply/divide area of the machine. Instead, they are with IO in mounting panel 2H.

In the standard machine the in-out register and the memory buffer register cannot be complemented. However, these operations are required for automatic multiplication and division. To allow for these operations, the necessary pulse amplifiers must be added to the MB and IO transfer logic (see figure D6-5).

The specific sequence of operations that produces an automatic multiplication or division, and the hardware necessary for the execution of these operations are described in detail in a and c respectively. The proofs of the algorithms utilized for these operations are presented in b and d respectively.

a MULTIPLY SEQUENCE - The complete sequence of operations that execute an automatic Multiply instruction is shown in the flow chart, figure 7-5. The following description of the hardware parallels the sequence shown in the flow chart. All of the hardware described is shown in the multiply/divide drawing, figure D7-4. As each logic net or control pulse is mentioned, the coordinates at which it appears in this figure are also given.

When the multiply/divide option is installed in the machine, wiring connections in instruction control are changed, so that the appearance of op code 54 in the instruction register asserts the command level MUL instead of the command level MUS. During cycle one of the instruction, the multiplicand is retrieved from memory by the usual cycle-one memory reference. However, while memory access is made for the multiplicand, the multiplier is transferred from the accumulator to the in-out register in preparation for execution of the instruction (A3). At TP_2, IO is cleared, and AC is transferred to MB. Then at TP_3, the AC-to-IO transfer is completed by transferring the multiplier from MB to IO.

At TP_9, the step counter and the sign flip-flops, smb and srm, are cleared. Then at TP_{10} the signs of the operands are saved in the sign flip-flops. At the same time, if either of the operands is negative it is complemented. This is done because the partial products must

be formed from positive numbers. If the sign of the multiplicand in MB is negative, flip-flop smb is set (A5) and MB is complemented (B2). If the multiplier in IO is negative, flip-flop srm is set (A6) and IO is complemented (C2).

Furthermore, at TP_{10} of cycle one, several operations are performed to ready the system for the automatic sequence. The accumulator is cleared by applying the signal MUL· CY1 · TP_{10} directly to the AC control logic (A2). Flip-flop run is cleared (A6), halting the normal computer timing system, and the step counter is advanced from 0 to 1 by setting SCR_4 (A4). The automatic Divide sequence requires one more step than the automatic Multiply sequence. Therefore before Multiply begins, the step counter is advanced one position. In this way, the termination of both automatic operations can be controlled by the same set of step counter decoders.

At the end of cycle one, the substitute timing chain is initiated by pulsing MDP-6 through a 0.15-microsecond delay (C5). The values of the delays between the pulses in the chain are shown in the flow chart (figure 7-5). The values given are the delays actually produced by the delay circuits. The delays inherent in the inverters and pulse amplifiers are not included.

With the initial generation of MDP-6 the repeated formation of partial products begins. Each bit of the multiplier is sensed in IO_{17}. The multiplicand is added to the accumulator if IO_{17} contains 1. After the addition, the contents of AC and IO are shifted one place to the right, thus shifting a new bit into IO_{17}. The sequence of operations is then repeated, using the new bit in IO_{17}. If IO_{17} is 1, MDP-6 generates MDP-3 (C3). This pulse is applied directly to the accumulator control circuits, producing a partial addition (paragraph 7-4c). Then, 0.2 microsecond later, MDP-4 generates the carry function (D2). The logic net for the carry pulse includes several other conditions besides MDP-4. These conditions, however, are relevant only to the instruction Divide, and are automatically satisfied during Multiply.

Following the addition, MDP-5 advances the step counter one position (A4) and produces the multiply shift (B5). Note that if IO_{17} contains 0, MDP-5 is generated directly from MDP-6. Thus if the current bit of the multiplier is 0, the addition is skipped and the timing chain skips directly from MDP-6 to MDP-5. The step counter counts the present stage of the sequence and the multiply-shift signal shifts both AC and IO one place to the right (see

AC SH/RO R, paragraph 7-4b; and IO SH/RO R, paragraph 7-4e). In a multiply shift, AC_0 is automatically cleared. This is done because the addition that forms a partial product can overflow. In the shift, the partial product is shifted back into the proper portion of the register, and the overflow bit is removed.

The multiply shift also re-pulses MDP-6, so that the whole sequence of operations is repeated. Finally, when the step counter contains 21 (indicating that 16 steps have been performed), MDP-6 initiates the cycle of operations once more and also restarts the normal computer timing system. The multiply/divide restart pulse sets flip-flop run and pulses TP_0 to begin the timing chain. The restart is inhibited, however, if the computer should stop because it is operating in one of the manual modes, single cycle or single instruction.

While the normal timing chain is starting, the multiply/divide timing chain performs one more cycle of addition and shift, again pulsing MDP-6. But this time the step counter contains 22 (indicating 17 steps), preventing MDP-6 from again starting the partial-product cycle. This last MDP-6 samples the sign flip-flops in order to adjust the sign of the double-length product. If the two sign flip-flops are in different states, indicating that the initial operands had different signs, then the control level SPQ is asserted. The assertion of SPQ indicates that the product, which is now positive, should instead be negative. If SPQ is asserted, and the double-length product is not already equal to +0 (C8), the last MDP-6 generates MDP-11. This final multiply/divide pulse complements both IO and AC (C2, D2), changing the sign of the result.

b MULTIPLICATION ALGORITHM - The actual operation of multiplying two numbers is performed only on positive numbers (a above). In PDP-1, these two positive operands are binary fractions; the fixed binary point is to the left of the most significant bit. The algorithm therefore need only be discussed for the case of two positive fractional operands.

In pencil-and-paper multiplication of two binary fractions, each bit-product (the product of the entire multiplicand by one bit of the multiplier) is shifted to the left before the final addition. For example, the full pencil-and-paper multiplication of .101100 times .100101 (without skipping any steps) looks like this:

```
      .101 100
      .100 101
      -------
       101 100
      0 000 00
     10 110 0
    0 00 000
   0 0 00 00
  10 1 10 0
  ---------
  11 001 011 100
```

Because two six-place binary fractions have been multiplied together, the product formed must be a twelve-place binary fraction; this final product is .011 001 011 100. Note that each consecutive bit-product is shifted left one place before the addition.

The computer cannot add a column of numbers; it adds only two numbers at a time. Consequently, to form the product of two numbers, the computer forms partial products by adding each consecutive bit-product to the sum of all previous bit-products. In other words, the computer uses the relation

$$a + b + c + d + \ldots = (((a + b) + c) + d) + \ldots$$

There is also a second difference between the conventional pencil-and-paper method and the computer method of multiplication: the computer does not shift each bit-product left one place before adding it to the sum of the previous bit-products. Instead, the computer shifts the sum of all previous bit-products right one place, and then adds the current bit-product. A detailed flow chart of computer multiplication is shown in figure 7-5.

The example in table 7-1, using the same numbers as the pencil-and-paper example above, shows the multiplication as performed in the computer. For simplicity, the registers in this example contain only six bits plus a sign bit. Since six-bit registers are used in the example, there are only six steps to the multiplication operation. The step counter, SCR, therefore contains 7 at the completion of the operation (SCR starts the operation containing 1).

At the end of the operation, the combination of the two registers, AC and IO, contains the 12-bit product of the two numbers. The last bit of IO contains the original sign of IO which is 0, because the operation is performed on positive numbers. Multiplication in the computer is performed in just this way except that instead of six-bit registers, the computer uses 17-bit registers.

c DIVIDE SEQUENCE - The complete sequence of operations that execute an automatic Divide instruction is shown in the flow chart, figure 7-6. The following description of the hardware parallels the sequence shown in the flow chart. All of the hardware described is shown in the multiply/divide drawing, figure D7-4. As each logic net or control pulse is mentioned, the coordinates at which it appears in this figure are also given.

When the multiply/divide option is installed in the machine, wiring connections in instruction control are changed, so that the appearance of op code 56 in the instruction register asserts

TABLE 7-1 EXAMPLE OF MULTIPLICATION ALGORITHM

	MB	SCR (octal)	AC	IO	MDP
Start of operation	0 101100	1	0 000000	0 100101	6
Last bit of IO is 1, add MB to AC		1 1	0 101100 0 101100	. . . . . . 0 100101	3,4
Increment SCR and shift right one place. This forms 1st partial product		2	0 010110	0 010010	5
Last bit of IO is 0. Increment SCR and shift right one place. This forms 2nd partial product.		3	0 001011	0 001001	5
Last bit of IO is 1. Add MB to AC.		3 3	0 101100 0 110111	. . . . . . 0 001001	3,4
Increment SCR and shift right one place. This forms 3rd partial product.		4	0 011011	1 000100	5
Last bit of IO is 0. Increment SCR and shift right one place. This forms 4th partial product.		5	0 001101	1 100010	5
Last bit of IO is again 0. Increment SCR and shift right. This forms 5th partial product.		6	0 000110	1 110001	5
Last bit of IO is 1. Add MB to AC.		6 6	0 101100 0 110010	. . . . . . 1 110001	3,4
Increment SCR and shift right. This forms 6th partial product, which is final product.		7	0 011001	0 111000	5

the command level DIV instead of the command level DIS. During cycle one of the instruction, the divisor is retrieved from memory by the usual cycle-one memory reference. No cycle-one transfers are required in preparation for the Divide sequence.

As in the case of Multiply, the step counter and sign flip-flops are cleared at TP_9 and the sign flip-flops are adjusted at TP_{10}. The signs of the operands are not regulated in the same way as they are in the order Multiply. The dividend must be positive, but since the division is performed by repeatedly subtracting the divisor from the dividend, the divisor must be negative. Therefore, if the divisor in MB is negative, its sign is saved by setting flip-flop smb (A5) but the divisor is complemented only if it is positive (B2). If the double-length dividend is negative (as indicated by the state of AC_0), flip-flop srm is set and the entire dividend is made positive by complementing both AC and IO (B2, C2).

At the end of cycle one, the normal computer timing chain is ended by clearing flip-flop run (A6). At the same time, the multiply/divide timing chain is initiated by pulsing MDP-3 through a 0.15-microsecond delay (C2). The delays between the multiply/divide pulses are shown in the flow chart (figure 7-6). The values given in the figure are the delays of the actual delay circuits. The delays inherent in the inverters and pulse amplifiers are not included.

The initial pulse, MDP-3, produces a partial addition (paragraph 7-4c) and also pulses MDP-4. This pulse then produces several operations. If the partial sum in AC is not equal to -0, the standard carry function follows (D1, D2). However, if the partial sum is equal to -0, the accumulator is complemented instead (D1, D2). Then, after a 50-nanosecond delay, if the divisor is negative (which it must be on the first cycle of the Divide sequence) it is complemented (B1).

Furthermore, MDP-4 also pulses MDP-5, which advances the step counter one position. At the same time that MDP-5 increments SCR, it also samples the contents of the counter. If the sign of the accumulator is 0 on the first circuit of the division cycle (when SCR contains 0), then the timing chain continues on to MDP-7 (C6). The condition that the number in AC is positive after the first subtraction indicates that the divisor is less than or equal to the dividend and consequently the division is not possible. The pulsing of MDP-7 at this time skips all further cycles in the sequence and continues the timing

chain into the termination operations.

If the sign of the accumulator is not positive, MDP-5 instead pulses MDP-1 (C1). But if the sign bit of AC is positive (i.e. 0) then MDP-1 complements MB (B1). (This operation can occur only on subsequent cycles. The operation cannot be executed on the first cycle because the condition AC_0^0 prevents the generation of MDP-1 in the first cycle).

Pulse MDP-1 also produces the divide shift by directly pulsing the shift/rotate lines of the accumulator and the in-out register (paragraphs 7-4b, c). The divide shift is a rotation of AC and IO to the left, but it differs in one respect from a normal rotation. In the divide shift, the complement of AC_0 is jam-transferred into IO_{17} (figure (B6).

Following the divide shift, if IO_{17} is 0 (C2), MDP-2 adds 1 to the contents of AC (figure D7-3B5). However, if IO_{17} is 1, MDP-2 is skipped and the timing chain goes directly to MDP-3. This begins the division cycle over again, since the terminating condition for an impossible division now cannot be satisfied. The cycle MDP-3, -4, -5, -1, -2 is repeated over and over again until the step counter contains 22, indicating that the 17th step of the operation is being performed. At this time the 17th step has not been completed, because the termination is at MDP-5, and MDP-1 and -2 of the final cycle have not been performed.

The condition SCR = 22 causes MDP-5 to pulse MDP-7, which in turn pulses MDP-8. These two pulses are utilized for the termination of the automatic operation whether the division is valid or not. Pulse MDP-7 produces a partial addition (paragraph 7-4c). If AC does not contain -0, MDP-8 then produces the standard carry function. (D1, D2). However, if AC does contain -0, MDP-8 complements the accumulator instead (D1). If the timing chain entered the termination sequence from the first cycle because the division was not possible, the actions performed by MDP-7 and MDP-8 restore the original (positive) dividend to the accumulator. If the division is a valid one the actions performed by MDP-7 and MDP-8 complete the final cycle of the Divide sequence.

Many of the terminating operations must be performed only if the Divide sequence has resulted in a valid quotient. Since the step counter is incremented only once if the division is not possible, the occurrence of any number greater than 1 in the step counter indicates that a complete division has been performed. Thus on the condition SCR_3^1,

MDP-7 generates the good-divide signal (B2). The good-divide signal advances the program counter one extra position, causing the program to skip the instruction following Divide. Consequently the program performs the next instruction in sequence only if the division was not possible. This allows the programmer to compensate for an impossible division by jumping to an appropriate subroutine.

After MDP-8, either the original dividend is in AC, or else a correct division has been performed and the quotient is in IO. In the latter case, AC contains an 18-bit remainder. Therefore, on the condition SCR_0^1, MDP-9 shifts AC one place to the right (paragraph 7-4b) putting the remainder into the standard form for a number in the accumulator. This shift automatically clears AC_0 (figure D7-1A1).

Since the Divide instruction is now almost complete, MDP-9 also restarts the regular computer timing chain by setting flip-flop run and pulsing TP_0 (B6). However, this restart pulse is not generated if the computer is stopping because it is in one of the manual modes, single cycle or single instruction.

Following the final shift, MDP-10 clears the memory buffer (paragraph 6-7) and adjusts the signs of AC and IO. The sign of the accumulator is made the same as the sign of the original dividend whether the division is completed or not. This is done because if the division is not performed, AC must contain the original dividend; and if the division is performed, the sign of the remainder must match the sign of the dividend. In either case, if the dividend was originally negative (srm^1) MDP-10 complements AC (D1) provided that AC does not contain +0.

Adjustment of the sign of the in-out register depends on the completion of the division. If the division has not been performed and the dividend was originally negative (the condition $SCR_0^0 \cdot srm^1$), MDP-10 complements IO, returning it to its initial configuration (C1). The sign of a valid quotient depends on the original signs of dividend and divisor. The control level SPQ (sign of product or quotient) is asserted if the signs of the operands were originally different (A5). Thus if IO does not contain +0, a complete division has been performed (SCR_0^1), and SPQ is asserted (B1), then MDP-10 complements IO.

After MDP-10 all signs are correct and either AC and IO contain the original double-length dividend, or else AC and IO contain the remainder and quotient, respectively.

Pulses following MDP-10 occur only if the division has actually been performed; MDP-10 then pulses MDP-12 on the condition SCR_0^1 (D7). Since MDP-10 has already cleared MB, MDP-12 transfers the quotient from IO to MB (paragraph 6-7). Next MDP-13 switches the positions of the quotient and remainder. It does this by jam-transferring the quotient from MB to AC (figure D7-3) and by simultaneously jam-transferring the remainder from AC to MB (A2). Finally, MDP-14 transfers the remainder from MB to IO (A3).

After the entire Divide sequence has been completed the quotient is in the accumulator and the remainder is in the in-out register. The sign of the remainder is the same as the sign of the original dividend.

d DIVISION ALGORITHM - The actual operation of dividing one number by another is performed only on positive fractions (c above). The algorithm for division therefore need only be discussed for the case of two positive fractional operands.

Let A be the divisior, B the dividend, Q the quotient, and R the remainder (if any). Both A and B are positive fractions. Then, by the basic definition of the division process:

1) $$B = QA + R, \qquad 0 \leq R < A.$$

In other words, given any two positive fractions B and A, there is a number Q (which need not be a fraction) such that the product of Q and A comes within the fraction R of equalling B. In the PDP-1 however, all numbers are represented as fractions. Consequently the number Q must also be a fraction. This requires that B, the dividend, be smaller than A, the divisor. In the PDP-1, the division is not performed if the fractional dividend B is larger than the fractional divisor A.

The expression 1), given above, is valid in any number system. If, for example, all the numbers A, B, Q, and R are represented in the binary system, expression 1) still holds true.

In the binary system as well as in the decimal system, pencil-and-paper long division is carried out by determining the digits of Q one at a time. To do this, a number of steps are performed, each step representing a successive application of expression 1). Let $B < A$ as required. Since the division is to be performed in the binary system,

let Q_i be the coefficient of 2^{-i} in the quotient. (That is, Q_i is the i^{th} digit to the right of the binary point in the quotient.) Then the steps representing the division of B by A are:

$$
\begin{aligned}
2)\qquad B &= 2^{-0}Q_0A + 2^{-0}R_0, & 0 \leq R_0 &= B < A \\
2^{-0}R_0 &= 2^{-1}Q_1A + 2^{-1}R_1, & 0 \leq R_1 &< A \\
2^{-1}R_1 &= 2^{-2}Q_2A + 2^{-2}R_2, & 0 \leq R_2 &< A \\
&\;\;\vdots \qquad\quad \vdots \qquad\quad \vdots & &\;\;\vdots \qquad \vdots \qquad \vdots \\
2^{-(k-1)}R_{k-1} &= 2^{-k}Q_kA + 2^{-k}R_k, & 0 \leq R_k &< A
\end{aligned}
$$

The number k indicates the number of quotient digits to the right of the binary point that have been computed. For example, if k = 3, then the last step of 2) is:

$$2^{-2}R_2 = 2^{-3}Q_3A + 2^{-3}R_3.$$

Substituting this expression for $2^{-2}R_2$ into the expression for $2^{-1}R_1$ in 2) gives:

$$2^{-1}R_1 = 2^{-2}Q_2A + 2^{-3}Q_3A + 2^{-3}R_3.$$

Continuing the substitutions for the R's up to the expression for B (in the first line of 2) above) shows that:

$$B = (2^{-0}Q_0 + 2^{-1}Q_1 + 2^{-2}Q_2 + 2^{-3}Q_3) \cdot A + 2^{-3}R_3,$$

which exactly parallels the original expression 1): B = QA + R. And, by definition, every valid division process produces a quotient and remainder which satisfy expression 1). Because $0 \leq R_3 < A$, it follows that $0 \leq 2^{-3}R_3 < 2^{-3}A$. Therefore the remainder left after a division to a three-place quotient is a number whose most significant digit is at least four places to the right of the binary point.

The expression for the result of thefirst k steps of 2) is (by induction):

$$3)\quad B = (2^{-0}Q_0 + 2^{-1}Q_1 + \ldots + 2^{-k}Q_k) \cdot A + 2^{-k}R_k, \quad 0 \leq R_k < A$$

This expression also parallels expression 1) above: B = QA + R. Expression 3) states that B is equal to the product of A times a binary quotient accurate to k places; plus a remainder whose first significant digit is at least k+1 places to the right of the binary point.

Although this pencil-and-paper method for division is valid, there are two reasons why the PDP-1 cannot use it. First, without testing, the computer cannot know whether $B < A$, or whether $A \leq B$. Second, the pencil-and-paper algorithm presented in expression 2) above, requires that in each step a decision be made as to how many times A "goes into" R_i (or into B).

The computer tests whether $B < A$ by using a modification of the initial step of expression 2): Let R_i^* be the contents of the accumulator after the i^{th} division step, so that in particular the result of the test for $B < A$ is R_0^*. Rather than performing the initial step of expression 2),

$$B = 2^{-0}Q_0A + 2^{-0}R_0, \qquad 0 \leq R_0 < A,$$

the computer instead simply subtracts A directly from B in the accumulator:

$$4) \quad B - A = R_0^*, \qquad -A \leq R_0^* < +1$$

Then if R_0^* is negative, $B < A$, and the machine proceeds with the remaining division steps. On the other hand, if R_0^* is positive or 0, then $A \leq B$, and the division sequence is not completed.

The question of how many times A "goes into" B has been partially answered by the initial subtraction, because the machine can now determine the value of Q_0. After the initial subtraction the accumulator contains $R_0^* = B - A$. If this number is negative, then B is not as great as $1 \cdot A$, so the quotient coefficient Q_0 of 2^{-0} (=1) must be 0. Conversely, if R_0^* is positive, then B is at least as great as $1 \cdot A$, so Q_0 must be 1. Thus the value of Q_0 is determined by the sign of the number $R_0^* = B - A$ developed in the accumulator by the initial subtraction.

In a valid division, Q_0 is required to be 0. The digit Q_0 is actually the sign bit of the quotient, so that $Q_0 = 0$ signifies that the quotient is to be a positive number. This follows from the fact that the division sequence is performed only on positive binary fractions. However, if the initial test subtraction reveals that $A \geq B$, then $Q_0 = 1$. A 1 as the sign bit of the quotient indicates that the quotient is a negative number, which is not valid as the result of a division with positive operands.

All division steps performed by the computer subsequent to the initial test subtraction are identical. Suppose that the machine is currently performing the i^{th} division step.

Then as the result of the previous step (the $i-1^{st}$ step), the accumulator contains the number R_{i-1}^*. As the first operation performed in the current step, the computer shifts the contents of the accumulator, R_{i-1}^*, one place to the left, generating the number $2R_{i-1}^*$.

At the same time (as part of the shift operation) the machine senses the sign of R_{i-1}^*: if the sign is positive, then Q_{i-1} is 1; if the sign is negative, then Q_{i-1} is 0. The sign of R_{i-1}^* also determines the next operation performed as part of the present step: if the sign is positive, the machine subtracts A from $2R_{i-1}^*$; however, if the sign of R_{i-1}^* is negative, the machine adds A to $2R_{i-1}^*$. The result of the addition (or subtraction) is the new remainder, R_i^*. The first operation of the next division step then checks the sign of R_i^* to determine the next digit Q_i of the quotient.

The first operation of a division step creates $2R_{i-1}^*$ from R_{i-1}^*, and produces Q_{i-1} according to the sign of R_{i-1}^*. The second operation is either an addition or a subtraction of the divisor A, according to the same sign, that of R_{i-1}^*. The expression corresponding to the i^{th} division step is therefore

$$5) \quad 2R_{i-1}^* + (1-2Q_{i-1})\,A = R_i^*, \qquad -A \leqq R_i^* < +A$$

$Q_i = 1$ if R_i^* is positive;

$Q_i = 0$ if R_i^* is negative.

The term $+(1-2Q_{i-1})A$ determines whether A is added or subtracted in the current step. If R_{i-1}^* is negative, then $Q_{i-1} = 0$, so that $+(1-2Q_{i-1})A = +A$. That is, A is added to a negative $2R_{i-1}^*$. On the other hand, if R_{i-1}^* is positive, then $Q_{i-1} = 1$, so that $+(1-2Q_{i-1})A$ is equal to $-A$. In other words, A is subtracted from a positive $2R_{i-1}^*$.

The first division step subsequent to the test subtraction is just expression 5) with $i = 1$. The accumulator contains R_0^* from the test subtraction. The shift left doubles R_0^*, and since $Q_0 = 0$ (as required for a valid division), A is added, not subtracted. If this first division step is to be performed at all, Q_0 is required to be 0; and R_0^* is required to be negative. However, for the purposes of the following remarks, Q_0 is retained in the expression representing the first division step:

$$5) \quad i = 1, \qquad 2R_0^* + (1-2Q_0)A = R_1^*, \qquad -A \leqq R_1^* < +A$$

Since $R_0^* = B - A$ by 4), then 5) is equivalent to

6) $\quad 2B - 2A + (1-2Q_0)A = 2B - A - 2Q_0A = R_1^*,$

so that $\quad B = 2^{-1}A + 2^{-0}Q_0A + 2^{-1}R_1^*, \; -A \leq R_1^* < +A$

Now collecting terms,

step 1: $\quad B = (2^{-0}Q_0 + 2^{-1})A + 2^{-1}R_1^*, \; -A \leq R_1^* < +A$

Compare this expression with expression 3) above, with k = 1:

3) $\quad k = 1, \quad B = (2^{-0}Q_0 + 2^{-1}Q_1)A + 2^{-1}R_1, \quad 0 \leq R_1 < +A$

Expression 3) states that for given values of A, B, and Q_0, a value for Q_1 may be found such that R_1 is positive and less than A. But the expression for step 1 above implies the assumption that $Q_1 = 1$, and allows R_1^* to be either positive or negative.

The sign of R_1^* is the test for the validity of the assumption that $Q_1 = 1$. If R_1^* is positive, then B is large enough to leave a positive remainder with Q_0 given and $Q_1 = 1$. In other words, from step 1, with R_1^* positive, $B \geq (2^{-0}Q_0 + 2^{-1})A$. However, if R_1^* is negative, then step 1 implies that $B < (2^{-0}Q_0 + 2^{-1})A$, so that Q_1 must be 0. Thus the sign of R_1^* determines the value of Q_1, as stated in the expression for the i^{th} division step, 5) above.

The second division step is just expression 5) again, with i = 2:

5) $\quad i = 2, \quad 2R_1^* + (1-2Q_1)A = R_2^*, \quad -A \leq R_2^* < +A$

Expression 6) above gives $R_1^* = 2B - A - 2Q_0A$; substituting:

$$4B - 2A - 4Q_0A + (1-2Q_1)A = R_2^*$$

Shifting terms in A to the right and clearing powers of 2:

$$B = 2^{-1}A + 2^{-0}Q_0A - 2^{-2}A + 2^{-1}Q_1A + 2^{-2}R_2^*,$$

so that,

step 2: $\quad B = (2^{-0}Q_0 + 2^{-1}Q_1 + 2^{-2}Q_2)A + 2^{-2}R_2^*.$

Now compare expression 3) above with k = 2:

3) $\quad k = 2, \quad B = (2^{-0}Q_0 + 2^{-1}Q_1 + 2^{-2}Q_2)A + 2^{-2}R_2, \quad 0 \leq R_2 < +A$

The comparison is parallel to the comparison made above for step 1. That is, expression 3) with k =2 states that given A, B, Q_0, and Q_1, a value may be found for Q_2 such that R_2 is positive. Step 2 implies the assumption that Q_2 is 1, and develops a number R_2^* whose sign tests the validity of the assumption that Q_2 is 1. The same relation of remainder sign to binary quotient digit holds: if R_2^* is negative, then $Q_2 = 0$; if R_2^* is positive, then $Q_2 = 1$.

The general division step, the i^{th} step, is expression 5):

$$5) \quad 2R_{i-1}^* + (1-2Q_{i-1})A = R_i^*, \qquad -A \leq R_i^* < +A,$$

$$Q_i = 1 \text{ if } R_i^* \text{ is positive;}$$

$$Q_i = 0 \text{ if } R_i^* \text{ is negative.}$$

The induction on i is not difficult; the i^{th} step is equivalent to:

$$7) \quad B = (2^{-0}Q_0 + 2^{-1}Q_1 + \ldots + 2^{-(i-1)}Q_{i-1})A + 2^{-i}A + 2^{-i}R_i^*$$

which is directly comparable to expression 3) with k = i:

$$3) \quad k = i, \quad B = (2^{-0}Q_0 + 2^{-1}Q_1 + \ldots + 2^{-i}Q_i)A + 2^{-i}R_i,$$

$$0 \leq R_i < +A.$$

Comparison shows that the value of Q_i is determined in the same manner as Q_1 and Q_2 were determined in steps 1 and 2 of the division sequence. Expression 7) implies the assumption that Q_i is 1, and produces a remainder R_i^* whose sign tests the validity of the assumption that Q_i is 1.

Expression 3) with k=i states that given values for A, B, and all Q's up to Q_{i-1}, the digit Q_i is determined by the requirement that R_i be positive. That is, expression 3) says that the proper value for Q_i is the value which makes R_i positive and less than A; expression 7) says set $Q_i = 1$, and check whether this is the correct value by checking whether R_i^* is positive; (i.e. if R_i^* is negative then 1 is not the correct value for Q_i, so Q_i must therefore be 0).

This reasoning shows that every division step performed by the computer (i.e. the i^{th} division step for all i) produces the correct value for the digit Q_i, the coefficient of 2^{-i} in the quotient. However, the proof of the algorithm as presented above uses negative powers of 2, i.e., all powers of 2 were divided out in order to show the

similarity of the general division step to the general pencil-and-paper algorithm step. On the far right of expression 7), the last term is $2^{-i}R_i^*$. The computer, however, actually has R_i^* itself in the accumulator, not $2^{-i}R_i^*$. This disparity leads to the question: why are significant bits not lost during the left shift performed as part of each division step?

To show that no bit significant to the division sequence is shifted out of AC during a division step, one preliminary observation must be made: that it is possible for AC to contain a number, say $2R_i^*$, of the range $-2 < 2R_i^* < +2$. The representation of numbers of this range in AC uses the sign bit, AC_0, as the coefficient of 2^{-0}, i.e., as the coefficient of 1. Although most representations of numbers in AC use AC_0 as the bit representing the sign of the number, nevertheless, AC_0 behaves precisely as a next more significant bit of the accumulator.

The computer representation of a negative number in 1's complement arithmetic is the bit-by-bit logical complement of that number. Thus, for example, -0 is represented as a number whose digits are all 1's. This means that if a number in the accumulator is negative, then $AC_0 = 1$, i.e., AC_0 contains the binary digit -0. If a number in AC is positive, then $AC_0 = 0$, i.e., AC_0 contains the binary digit +0.

Suppose that AC contains a 17-bit number, say R_i^*, of either sign. If this number is shifted left one place, then AC contains the 18-bit number $2R_i^*$, and the sign bit is lost out the left end of AC. But the sign bit is just ±0, depending on the sign of R_i^*. The important fact is that the loss of the sign bit does not affect computations, because all logical decisions that depend on the sign of R_i^* are made before the left shift.

As the second operation of any division step, the divisor A is either added to, or subtracted from, the contents $2R_{i-1}^*$ of AC. The result of the addition or subtraction is the new remainder, R_i^*. This new remainder is shifted left one place as the first operation of the next division step. It is therefore necessary to show that R_i^* is a 17-bit number (that the sign bit of R_i^* is the binary digit +0 or -0) so that the subsequent shift left one place may be performed without losing a significant bit.

Expressions 4), 5), and 6) of the algorithm discussion above state without proof that $-A \leq R_i^* < +A$. If this is shown to be true, then the fact that A is a positive binary

fraction implies that R_i^* is contained in the least significant 17 bits of AC, as required.

In the initial test subtraction, the machine developed R_0^* by subtracting A from B. In a valid division, both A and B are positive binary fractions, and B < A. This means that

$$-A \leq (B - A) = R_0^* < 0.$$

Since R_0^* is negative, the machine adds A to $2R_0^*$ as the first division step: $2R_0^* + A = R_1^*$. Thus

$$-A \leq 2(B - A) + A = 2R_0^* + A = R_1^* < +A.$$

Both R_0^* and R_1^* are of smaller magnitude than A; this implies that neither $2R_0^*$ nor $2R_1^*$ exceeds the 18-bit capacity of AC.

The inequality $-A \leq R_i^* < +A$ is therefore true for i = 0 and for i = 1. If it can be shown that the truth of the inequality for i - 1 implies the truth of the inequality for i, then by induction.

$$-A \leq R_i^* < +A \qquad \text{for all } i.$$

Suppose, then, that $-A \leq R_{i-1}^* < +A$. There are two cases to consider:

<u>case I</u>

$$-A \leq R_{i-1}^* < 0$$

In this case, $Q_{i-1} = 0$; then division step i, expression 5) becomes

$$2R_{i-1}^* + A = R_i^*,$$

which shows that

$$-A \leq R_i^* < +A.$$

<u>case II</u>

$$0 \leq R_{i-1}^* < +A$$

In this case, $Q_{i-1} = 1$; then division step i, expression 5) becomes

$$2R_{i-1}^* - A = R_i^*,$$

which shows that

$$-A \leq R_i^* < +A.$$

In both case I and case II, the required inequality holds. Consequently, all remainders R_i^* developed during the division sequence have magnitudes less than A, which is a positive binary fraction. Therefore, the successive AC left shifts, performed during the division sequence, cannot result in the loss of significant bits of the remainders.

The flow chart for the automatic division sequence is shown in figure 7-6. The following portions of this paragraph relate the division algorithm as discussed above to the PDP-1

operations as shown in the flow chart.

The five-bit step counter is used to end the division sequence after all 18 bits (the sign bit and the 17 magnitude bits) of the quotient have been developed. The quotient is formed in IO by shifting both AC and IO left one place, while simultaneously loading IO_0 into AC_{17}, AC_1 into AC_0, and the complement of AC_0 into IO_{17}. The contents of AC and IO are both shifted left as one 36-bit register, with the complement of AC_0 shifted into IO_{17}.

The fact that the complement of AC_0 is transferred into IO_{17} during this division shift reflects the expression

$$Q_i = 0 \text{ if } R_i^* \text{ is negative;}$$
$$Q_i = 1 \text{ if } R_i^* \text{ is positive.}$$

which is given as part of the i^{th} division step, expression 5). This same shift left one place generates $2R_i^*$ from R_i^* in AC.

Addition and subtraction in the PDP-1 are performed by direct addition and addition of the 1's complement, respectively. The divisor A is in MB during the division sequence. Therefore, if the contents of MB are +A and a subtraction is necessary, then MB must be complemented. MB must also be complemented if it contains -A when an addition is necessary. On the other hand, MB must <u>not</u> be complemented if it contains +A before a required addition, or -A before a required subtraction.

Whether the current step requires an addition or a subtraction is determined by the sign of the previous remainder, R_{i-1}^*. If the previous remainder is negative, then an addition is required, and MB must contain +A. If the previous remainder is positive, then a subtraction is required, and MB must contain -A. The computer must therefore complement MB if both AC and MB have the same sign. If the signs of MB and AC are different, however, the computer <u>must not</u> complement MB.

The PDP-1 meets these requirements in the following manner: MB is complemented at MDP-4 (see the flow chart, figure 7-6) if the sign of MB is negative, i.e. if $MB_0 = 1$. Then, at MDP-1 of the following division step, MB is complemented if the sign of AC is positive (if AC_0 is 0 before the shift). This means that if the contents of AC and of MB are both positive, then MB is complemented at MDP-1; whereas if the contents of MB and of AC are both negative, then MB is complemented at MDP-4. Thus the

computer complements MB when MB and AC have the same sign. However, as explained above, it is also necessary that MB not be complemented if the signs of MB and AC differ.

If the signs of the contents of MB and AC are different, then either MB holds a positive number while AC contains a negative number; or else MB contains a negative number while AC holds a positive number. In the first case, MB is not complemented at all; it is not complemented at MDP-4 because MB_0 is 0, and it is not complemented at MDP-1 because AC_0 is 1. In the other case, in which MB contains a negative number and AC a positive number, MB is complemented twice, once at MDP-4 because MB_0 is 1, and again at MDP-1 because AC_0 is 0. Complementing MB twice leaves the contents of MB unchanged.

A further difficulty arises when the remainder produced by the i^{th} division step is negative. The dividend B is a 35-bit number contained initially in both AC and IO. AC_0 is the sign bit, AC_1 through AC_{17} are the first 17 magnitude bits, and IO_0 through IO_{16} are the last 17 magnitude bits. When IO_0 is shifted into AC_{17} during a division shift, the effect is the same as that of the "bring down the next digit" operation done in pencil-and-paper long division. However, the less significant bits of the dividend in IO are always positive. If the contents of AC are negative before the division shift, then after the division shift AC_{17} contains a bit from a positive number, and AC_0 through AC_{16} contain bits representing a negative number. In order to correct for this, the machine adds 1 to the least significant bit, AC_{17}, if the contents of AC are negative before the division shift.

As an illustration of this correction, let the least significant 17 bits of the dividend B be all 0's. Then IO is initially clear. Suppose now that after a given division step the remainder in AC is negative. After the shift, AC_0 through AC_{16} contain twice this negative remainder, and AC_{17} contains a 0 shifted in from IO. But the binary digit -0 is represented in 1's complement arithmetic by a bit containing 1. The machine adds 1 to AC_{17}, so that all 18 bits of AC contain the correct representation for a negative number.

Since the sign of the number in AC is lost out the left end of AC during the shift, the

machine senses the newly developed bit of the quotient in IO_{17} at MDP-2. If IO_{17} is 1, then the contents of AC are positive, and AC_{17} is not affected. However, if IO_{17} is 0, the contents of AC are negative, and 1 is added to AC_{17} at MDP-2.

When the step counter, SCR, contains octal 22 (i.e., decimal 18), all 18 bits of the quotient have been shifted into IO, and no more division steps are performed. Instead, the machine generates the first of several pulses that complete the division sequence. The operations that complete the division sequence are shown on the right of the flow chart (figure 7-6). These operations generate the correct remainder, adjust the signs of the remainder and quotient, and transfer the quotient into AC and the remainder into IO.

The correct positive remainder (that is, positive before the adjustment of signs made during the completion sequence) is generated by MDP-7, -8, and -9. When SCR contains decimal 17, AC contains the product of the 17th division step, R_{17}^*. The value of the last digit, Q_{17}, of the quotient depends on the sign of R_{17}^*. This last digit, Q_{17}, can be transferred into IO_{17} only by a division shift at MDP-1. To generate this last division shift, the machine performs an entire division step, MDP-1 through MDP-5.

At this point, the step counter, SCR, contains decimal 18, and AC contains R_{18}^*. From expression 7) of the algorithm proof above (with i = 18), this 18th division step is equivalent to:

Step 18: $$B = (2^{-0}Q_0 + 2^{-1}Q_1 + \ldots + 2^{-17}Q_{17})A + 2^{-18}A + 2^{-18}R_{18}^*.$$

But the correct remainder after a division to 17 binary places is $2^{-17}R_{17}$. Compare step 18 above with expression 3) with k = 17:

$$B = (2^{-0}Q_0 + 2^{-1}Q_1 + \ldots + 2^{-17}Q_{17})A + 2^{-17}R_{17}.$$

Since all the Q_i in the two expressions are identical by pairs, the remainders are related as follows:

$$2^{-18}A + 2^{-18}R_{18}^* = 2^{-17}R_{17}.$$

Multiplying through by 2^{18},

$$A + R_{18}^{*} = 2R_{17}.$$

This means that to obtain R_{17}, the machine must first add A to R_{18}^{*} (to produce $2R_{17}$) and must then shift right one place to develop R_{17}. Pulses MDP-7, -8 and -9 perform the addition of A and the one-place shift right (figure 7-6). This completes the proof that the PDP-1 division algorithm produces a valid division.

CHAPTER 8

MEMORY

8-1 GENERAL.

The basic core memory system of PDP-1 consists of a memory address register and decoders, a memory buffer register, and a type 12 memory module. The type 12 module is composed of a 4,096-word coincident-current core bank and associated read/write logic, sense amplifiers, and timing circuits. Memory capacity may be expanded by adding the type 15 memory extenstion control to the computer. This control allows expansion of the memory system to 16 type 12 modules (65,536 words).

Because of the system's 12-bit address format and single memory buffer register, module selection logic and a memory buffer mixer are required when using more than one memory module.

All type 12 memory modules are identical in operation. The description of the memory module in bay 3 of the standard computer applies equally to other modules which may be added to the system. All modules share the same memory addressing element and memory buffer register.

The memory system is shown in six logic drawings, figures D8-1 through D8-6. For information on the use and organization of these drawings see paragraph 3-16.

8-2 MEMORY ADDRESSING.

The memory addressing element of the computer is composed of the 12-bit memory address register, MA, and four binary-to-octal memory address decoders.

a MEMORY ADDRESS REGISTER - The 12-bit memory address register is shown in figure D8-1. The flip-flops in the register are labelled MA_6 to MA_{17}. These designations correspond to the memory address portion of the instruction word. Each memory access is made to the location specified by the contents of the MA register. The 12-bit address can

designate any one of the 4,096 (2^{12}) locations within a single memory module. The address is decoded to select the particular pair of X and Y windings within the core bank which designate the desired memory location.

In addition to the standard transfer gating at the 1 input, each of the type 1209 flip-flops in MA also includes direct clear and direct set inputs. The direct clear inputs are all pulsed when the register is cleared by $\overset{10}{\rightarrow}$MA. Since all transfers into MA are 1 transfers the register must be cleared prior to every address transfer.

Flip-flops in MA are set individually through the direct set inputs by pulse outputs from the corresponding bits in the high-speed channel address mixer (paragraph 6-10b).

The logical conditions that govern address transfers through the standard MA input gating are described in paragraph 6-6. Full addresses are transferred into MA from PC or MB_{6-17}. Four-bit numbers are transferred into MA_{12-15} from the break encoder. Transfer pulse $100\overset{1}{\rightarrow}$ MA loads address 100 directly into MA by setting MA_{11}.

b MEMORY ADDRESSES - The allocation of memory space for information storage is in most cases entirely at the discretion of the programmer. However, in certain operations memory locations are fixed by the computer hardware. The origin of each address transfer and the use of fixed addresses depends on the type of computer cycle being executed.

An address is loaded into the memory address register at the beginning of every memory cycle. The address is held in MA throughout the cycle to govern the location of the memory access. At the end of each cycle (TP_{10}) MA is cleared in preparation for the next cycle. However, the clear pulse is inhibited if the computer is to halt at the end of the current cycle. Inhibition of the clear pulse preserves the address of the final memory access. This address is then available for the operator at the control panel indicator lights. When the computer is restarted, register MA is cleared by SP_1. This pulse precedes any memory cycle initiated by a console operation.

All address transfers are executed at TP_0 except in the console operations Examine and Deposit. The memory cycle in these operations is performed as cycle one of lac and dac, respectively. Since the address originates at the console ADDRESS switches the computer skips the normal TP_0 address transfer and begins the cycle at TP_1. For these operations the address is transferred from TA to PC at SP_2 and thence to MA at SP_3.

In every cycle zero an address is transferred into MA from the program counter. During any defer cycle or cycle one, the transfer is made from the address portion of the memory buffer register. A single exception to this procedure is the instruction Call Subroutine. In cycle one of Call Subroutine, the programmed address is ignored, and address 100 is used instead. This instruction is equivalent to Jump and Deposit Accumulator with address 100. During cycle one of cal, the normal transfer MB $\xrightarrow{1}$ MA is inhibited while 100 $\xrightarrow{1}$ MA is generated instead. At the end of the cycle the address in MA is transferred to PC, incremented, and transferred back to MA at the beginning of the following cycle zero. The computer then performs the instruction located at address 101.

During a high-speed channel cycle, an address is transferred into MA from the address lines of the selected channel. Transfer takes place through the high-speed channel address mixer.

In a sequence break (with the sequence break system type 20) the initial break address is transferred into MA from the break encoder during break cycle one. The number of the channel is encoded into binary (four bits for 16 channels) and loaded into MA_{12-15}. Since bits 16 and 17 are 0, for a break on channel n access is made to address 4n during break cycle one. This makes four consecutive memory locations available for each channel. The subsequent locations used during break cycles two and three are counted by the program counter. Systems having the type 20 sequence-break system therefore reserve the first 66 memory locations for use in fixed operations. The 16-channel sequence break system uses locations 0 through 77, and the instruction Call Subroutine uses locations 100 and 101. However, if the computer is not in the sequence-break mode, the programmer may utilize the sequence-break locations for other purposes.

If the optional break system is not installed, only addresses 0 through 3, 100 and 101 are fixed. The standard one-channel sequence break system uses only the first four memory registers. There is no break encoder and the AND gates shown in figure D8-1, D5 to D7, are not present. Break cycle one does not include an address transfer; but because MA is previously cleared, the absence of an address transfer is equivalent to the transfer of address 0. Locations 1, 2 and 3 are then used in subsequent cycles of the sequence break.

c MEMORY ADDRESS DECODERS - Memory addresses in MA are decoded to select X

and Y windings corresponding to the addressed memory location. The decoding is accomplished in two stages. First, four memory address decoders convert the 12-digit binary address in MA to a four-digit octal address. This octal address is represented by four asserted octal output levels, one from each of the four decoders. Second stage decoding is accomplished by applying the 32 decoder outputs (including the four asserted outputs) to the read/write switches in the memory module.

The four memory address decoders used for the first stage decoding are the type 1150 binary-to-octal decoders shown in figure D8-1. Each of these four decoders receives outputs from three bits of MA, and asserts one of eight output levels. The octal output level asserted by a given decoder corresponds to the binary contents of the three associated MA bits.

The asserted outputs from the two decoders that decode MA_{6-8} and MA_{9-11} address a single Y winding from the 64 Y windings in the core bank. Similarly, the asserted outputs from the other two decoders (which decode MA_{12-14} and MA_{15-17})address a single X winding from the 64 X windings in the core bank. Selection of the appropriate X and Y windings addressed from the memory address decoders is performed by the read/write switches in the memory module (paragraph 8-4b).

The output levels from the four memory address decoders are designated in the following manner. The designation of the levels corresponding to the less significant octal digit of the two-digit X address or the two-digit Y address is preceded by the winding letter (e. g. X000), while the designation of the more significant digit-level is followed by the winding letter (e. g. 000X). The three-digit binary portion of the level designation specifies the state of the three MA bits that enables the decoder output. For example, memory address 5326 enables the four decoder outputs designated 101Y, Y011, 010X, and X110.

If only one memory module is in use the outputs of the memory address decoders are applied directly to the X and Y selection logic in that module. In multi-module memory systems the decoder outputs are applied instead to the memory extension control (paragraph 8-5a).

8-3 MEMORY BUFFER REGISTER.

The 18-bit memory buffer register (figure D8-2) is the only register that functions in all four

sections of the computer logic: control, arithmetic, in-out and memory.

Although op code bits 0 through 5 of an instruction word are decoded from the instruction register, all other instruction word control information is utilized directly from MB. In memory reference instructions only the indirect address bit contains control information, but in the augmented instructions all bits provide control information. In some cases single bits of MB gate specific operations; in others, such as addressing sense switches or program flags, three-bit sections of the register provide octal information through the memory buffer decoders (paragraph 6-3c).

The MB register also functions as an element of the arithmetic unit in all arithmetic and logical instructions. During these instructions MB holds the operand and the MB outputs provide the necessary levels to the arithmetic and logical gating of the accumulator. (The contents of MB are not affected by these operations except in the optional automatic Multiply and Divide instructions.)

In high-speed channel data transfers MB serves as an in-out register, bypassing IO. The outputs of MB are available to the data out lines through taper pins in in-out transfer control. Information can be transferred into MB from the data in lines through the high-speed channel buffer mixer.

As an element of the memory system, MB serves as the buffer between the memory module and the rest of the computer. All transfers of information between the computer and core memory must be made through the memory buffer register.

The type 1204 flip-flops in MB are similar to the 1201 flip-flops in MA (paragraph 8-2a) except that the direct-set input is replaced by a complement input. This complement input is not used in the standard machine but it is necessary for installation of the multiply/divide option. The algorithms used by the multiply/divide option require that MB be complemented at various stages in the execution of the automatic instructions (paragraph 7-5).

As in the case of the memory address register, information can be transferred into MB by single-bit 1-transfer pulses. Each pulse sets a specific flip-flop in the register. The single-bit transfer pulses are designated $\overline{MBM}_0$ to $\overline{MBM}_{17}$ to correspond to the memory buffer mixer outputs. However, the pulses come to MB from the mixer only if the computer includes a memory extension control (paragraph 8-5a). If there is no memory extension control but there is a high-speed

channel control then the pulses come from the HSC buffer mixer (paragraph 6-10b). If neither of these controls is installed the pulses come directly from the sense amplifiers of the single memory module (paragraph 8-4e).

A detailed description of the memory buffer transfer logic is presented in paragraph 6-7. At the beginning of every memory cycle MB is cleared. When the sense amplifiers are strobed (TP_4), the contents of the addressed core register are loaded in MB through the MBM input gates. In most cases the write portion of the memory cycle then writes the same word back into memory.

If new information is loaded into MB after the read operation, then the write portion of the memory cycle writes the new information into memory in place of the old.

For a high-speed channel transfer into memory, MB is cleared at TP_5. Then, at TP_7, data transferred into MB through the MBM inputs. If new information originates at the in-out register, MB is cleared prior to the transfer IO ^{1} MB. Note that an addressed core register can be cleared only by clearing MB without a subsequent transfer in.

If new information originates at the accumulator, no prior MB clear is required. This is because the transfer of AC to MB is a jam transfer. The transfer AC ^{i} MB uses two pulse lines: one line gates MB_{0-5}, the other line gates MB_{6-17}. Full-word transfers are executed by pulsing both lines simultaneously. By pulsing one line only either the op code or the address portion of the word in MB can be replaced independently. This can be done without disturbing the rest of the word.

The outputs of the flip-flops in the memory buffer register are applied to type 1684 bus drivers. The bus drivers provide amplification and buffering. Through these drivers information can be transferred from MB to all other central processor registers and to the core memory. In a computer containing a single memory module, the driver outputs are applied directly to that module. In a multi-module system, the driver outputs are instead applied to the memory extension control (paragraph 8-5a).

If the computer includes the high-speed channel control, the 1 outputs from the type 1684 bus drivers are also applied to type 1685 bus drivers. The outputs of these drivers are, in turn, applied to a taper pin panel in in-out transfer control. Information from MB can then be transferred out of the computer on the HSC data-out lines.

8-4 MEMORY MODULE LOGIC

The standard PDP-1 memory module type 12 is shown in figure D8-3. The memory module is composed of a coincident-current core bank and associated timing, driving and sensing logic. The core bank has a capacity of 4,096 eighteen-bit words. The memory module furnished with the standard computer is located in bay 3. All memory modules operate in the same manner.

a CORE BANK - The memory core bank is composed of 18 core planes, each containing 4,096 ferrite memory cores (64 rows x 64 columns). Every core is threaded by four windings; X and Y selection windings, an inhibit winding, and a sense winding.

The 64 X and 64 Y windings each thread a row or column of 64 cores in each of the 18 core planes. A single X or Y winding continues from one core plane to the next, threading the same row or column in every one of the 18 planes. A single X winding and a single Y winding intersect at a single memory location containing an 18-bit core register. During each memory cycle information is read from or written into the single addressed core register. This addressed register is selected from among the 4,096 registers in the core bank by selecting the single X winding and the single Y winding that intersect at the corresponding memory location.

There are 18 inhibit windings and 18 sense windings; one inhibit winding and one sense winding for each core plane. Both the inhibit windings and the sense windings thread all 4,096 cores in the plane with which they are used. Individual cores within the addressed register are selected by the sense windings during reading, and by the inhibit windings during writing.

> NOTE: The core bank actually includes an extra core plane which is completely wired in. There are therefore 19 planes, and 19 inhibit and sense windings. The extra 19th plane is not ordinarily used, but is provided in case it is wanted for some special application. The following discussion treats only the 18 core planes that are ordinarily used.

b X AND Y SELECTION - Each of the 4,096 locations in the memory core bank is specified by a particular 12-bit address in the memory address register. The memory-module selection logic selects one of the 64 X windings and one of the 64 Y windings

according to the contents of MA. These two windings intersect at the same relative location on each of the 18 core planes; the 18 cores located at these intersection points make up the addressed memory register.

The 32 outputs from the memory address decoders (paragraph 8-2c) are applied to the inverters shown in figure D8-3, B2 to C3. The inverter output designations correspond to the inverter input designations, but with one minor change. The three-digit binary portion of the input designation is replaced in the output designation by the corresponding octal digit.

From the inverters, the 16 X-selection levels are applied to the type 1972 read/write switches shown at the top of figure D8-3. Similarly, the 16 Y-selection levels are applied to the read/write switches shown at the bottom of the figure. Only four of these 32 selection levels are asserted during any given memory cycle; two Y levels corresponding to the two octal digits in MA_{6-11}, and two X levels corresponding to the two octal digits in MA_{12-17}.

The two asserted Y levels enable one of 64 Y read/write switches, and thereby permit the associated Y winding to be pulsed by the output of the read or write bus. Similarly, the two asserted X levels enable one of the 64 X read/write switches, and thereby permit the associated X winding to be pulsed by the output of the read or write bus. Thus the two sets of read/write switches select one X winding and one Y winding, and thereby select the addressed core register for reading or writing.

A type 1972 plug-in unit includes four read/write switches. A detailed circuit description of the read/write switches is included in paragraph 10-7b. There are 64 switches in each of the two sets; both sets are identical in function. Each switch is controlled by an AND-gate input. Each AND gate receives one of the eight more-significant-digit selection levels and one of the eight less-significant-digit selection levels. Only the AND gates that receive two asserted selection levels are enabled. During any memory access, these AND gates correspond to the single X and Y windings designated by the contents of MA.

When both inputs to a read/write switch AND gate are at ground, the switch closes, completing the current path between one end of the associated winding and the read bus.

Since the other end of the winding is permanently connected to the write bus, the read/write switch permits application of the bi-polar core-drive pulses to the two selected windings. (One polarity corresponds to a read pulse; the other polarity corresponds to a write pulse).

The type 1976 resistor cards located between the read/write switches and the core bank provide the necessary loading to produce a core-drive current of appropriate magnitude. Nominal value of the half-read and half-write currents is 190 ma. A single half-current is not sufficient to change the state of a core. However, the intersection of two half currents at the cores of the addressed register is sufficient to switch these cores (see e and f below).

c MEMORY TIMING FUNCTIONS - The timing for the memory cycle read and write operations is controlled by the shift register containing flip-flops R, RS, W and I (figure D8-3B1). This shift register is in turn controlled by the memory control pulses (paragraph 6-2 e). During a memory cycle, timing control information is shifted through the shift register by the shift signal $MOP_{2, 3, 7, 9}$. This signal is equivalent to timing pulses TP_2, TP_3, TP_7, and TP_9. The entire register is cleared at the end of each cycle by $L^0 \rightarrow MEM(TP_{10})$. This memory clear pulse prepares the register for the next cycle.

The control data shifted through the register is self-contained. No external gating levels are applied to the register. The register starts the memory cycle in a cleared state. Because the outputs of the second stage (RS) are applied to the opposite input gates of the first stage (R), the first shift pulse following the memory-clear pulse automatically sets R. At the second shift pulse, R^1 sets RS, asserting the condition RS^1. This causes the third shift pulse to set W and clear R. The condition R^0 in turn clears RS at the fourth pulse. Thus a 0 is shifted through the first three stages of the register two pulses behind the 1.

The final stage of the shift register (I) is set early by the inhibit pulse (TP_8). This grounds the 0 output of flip-flop I. Although the fourth shift pulse (TP_9) clears RS, this pulse does not affect I.

The four timing functions, read, write, inhibit, and strobe, cause the read and write operations to be performed. The read, write, and inhibit functions are levels; the strobe function is a pulse. State changes in the ferrite cores of the memory core bank occur much more slowly than changes of state in the computer logic elements. The core-driving pulses are therefore of relatively long duration (approximately two microseconds). The duration of these pulses is much longer than the duration of computer logic pulses. The core driving pulses are in fact produced from computer logic levels.

The memory timing functions are generated from the shift register outputs by the logic nets in B2 and B3. The logical conditions for these functions are as follows:

Read:	R = 1
Strobe:	0.3 μsec after $\downarrow$ RS
Inhibit:	1 = 1
Write:	(RS = 0) · (W = 1)

Operation of the memory timing network is summarized in Table 8-1 below. This table shows the states of the four shift flip-flops after each timing pulse. The shaded columns at the right of the table show the duration of the timing functions. Note, however, that these functions are shown relative to the irregularly-spaced timing pulses. Therefore the length of each shaded column is not necessarily proportional to the actual duration of the associated pulse. The true duration of these functions is shown in the diagram of the memory cycle (figure 3-2).

TABLE 8-1

MEMORY TIMING

Timing Pulse	Timing Flip-flops R	RS	W	I	Timing Functions Read	Strobe	Inhibit	Write
Initial State	0	0	0	0				
2	1	0	0	0				
3	1	1	0	0				
7	0	1	1	0				
8 (Inhibit)	0	1	1	1				
9	0	0	1	1				
10 (Clear)	0	0	0	0				

d READ AND WRITE DRIVERS (Figure D8-3D8) The type 1973 memory current drivers in the memory module provide the read and write currents applied to the core bank windings. These two currents are identical in magnitude, but of opposite polarity. A detailed description of the type 1973 current driver is included in paragraph 10-7c.

When the read driver is enabled by the -3 vdc level, a core drive read current is applied to the selected X and Y windings. This read current flows through the following path: terminal V of the read driver (-13 vdc), the read bus, the two closed read/write switches (these two switches provide parallel paths, one for X, and one for Y), the selected X winding and Y winding corresponding to the two closed read/write switches, the write bus, and terminal V of the write driver (-3 vdc).

Conversely, when the write driver is enabled by the -3 vdc write level, current is applied to the same X and Y windings, but in the opposite direction. Terminal V of the write driver is then at -13 vdc, and terminal V of the read driver is then at -3 vdc. The current path is exactly the same for both the read pulse and the write pulse. However, during the read pulse, the voltage at the read bus is 10 volts more negative than that at the write bus, while during the write pulse, the write bus is more negative. The type 735 memory power supply furnishes the -3 vdc and -13 vdc used by the type 1973 drivers. Circuit description of this power supply is treated in paragraph 10-11e. A separate power supply is required for each memory module because of possible temperature-induced variations in the core characteristics.

e READ SENSING - When a memory core is magnetized in the 1 direction, it is said to contain a 1. When magnetized in the opposite direction, it is said to contain a 0. During the read operation, a full-read current (i.e. two half-read currents, one on the selected X winding, and one on the selected Y winding) is applied to each of the 18 cores in the addressed memory register. The full read current tends to magnetize the memory cores in the 0 direction, and hence has no effect on those cores of the addressed register which were initially in the 0 state.

However, when the full-read current is applied to a core containing a 1, the core magnetization changes polarity, and the core is switched from the 1 state to the 0 state. This change of state induces an output voltage on the sense winding that threads the core.

(The same sense winding threads all 4,096 cores in the plane containing the affected core). The two ends of this sense winding are connected to the two input terminals of a type 1540 sense amplifier. A circuit description of the 1540 sense amplifier is included in paragraph 10-7a.

There are 18 type 1540 sense amplifiers in the memory module, one for each of the 18 core planes (figure D8-3, B7 and C7). The sense amplifiers are differential amplifiers which reject common-mode signals and amplify difference signals by a factor of 20. This tends to prevent noise voltages on the sense windings (from half-selected cores, etc.) from being erroneously sensed as valid 1 output signals. The actual output signal from a 1 state core which has been switched to the 0 state by a full-read applies a difference signal of approximately 60 millivolts to the sense amplifier inputs.

The sense amplifiers sample the core outputs by means of a 70-nanosecond strobe pulse. This strobe is regulated to occur approximately one microsecond after the beginning of the read level. At this time the sense winding is likely to produce the best signal-to-noise ratio. If the addressed core in a given core plane contains a 1, the read pulse causes it to apply a difference signal output to the sense winding.

The strobe pulse samples this output signal, and causes the sense amplifier to generate a standard logic pulse output (provided that the core output exceeds the required 1-signal threshold at the time of the strobe). This logic pulse output sets the corresponding flip-flop of the memory buffer register. The strobe pulse also restarts the timing chain by pulsing TP_4. This aligns timing pulses 4, 5, and 6 with the retrieval of information from memory (paragraph 6-2c).

The transfer of information from the memory to the memory buffer register is a 1 transfer. The memory buffer is cleared prior to the read-out from the addressed core register. At read-out, the sense amplifiers corresponding to the 1-state cores of the addressed register set the corresponding bits of MB. When a memory buffer mixer is included in the system, the output pulses from the sense amplifiers are applied to MB through the mixer (paragraph 8-5a). If no mixer is included, the pulses are applied directly to the MB input gating (paragraph 8-3).

f WRITE INHIBIT DRIVING - The read operation is destructive; read-out leaves all cores

of the addressed register in the 0 state. During writing, a full-write current (i.e. two half-write currents, one on the selected X winding, and one on the selected Y winding) is applied to all 18 cores in the addressed memory register. The full-write current tends to magnetize the memory cores in the 1 direction. Cores that receive only a full-write current are switched from the 0 state to the 1 state of magnetization.

To write a word from MB into the addressed core register, it is necessary to prevent (or inhibit) this change of state for just those cores of the addressed register that correspond to 0 bits in MB. This is done by applying an inhibit current of opposite polarity (the equivalent of a half-read current) to only those cores which are to remain in the 0 state. Net current to these cores is then equivalent to only one half-write current. Because this current is not sufficient to drive the cores beyond the "knee" of the hysteresis loop, they do not change state, but remain in the 0 state.

The inhibit currents which prevent the writing of 1's into the 0 bits of the addressed register are applied to the core planes through the 18 inhibit windings, each of which threads all cores in a single core plane. Each inhibit winding therefore threads one of the 18 bits in the addressed register. The type 1982 inhibit drivers (figure D8-3, B4 and C4) determine which of the 18 inhibit windings are to be pulsed. Circuit description of the 1982 inhibit drivers is treated in paragraph 10-7f.

The inhibit drivers are switching circuits which are enabled by two ANDed inputs. The inhibit level (c above) is ANDed with the 0 signals from the bits of MB, enabling those inhibit drivers corresponding to MB bits that contain 0. This allows inhibit current to flow through the associated inhibit windings. Inhibit current flows from the inhibit common line (always at -3 vdc) through the enabled inhibit drivers to the inhibit reference line (always at -13 vdc). While the inhibit level is asserted, current paths are completed from the common line through the enabled drivers and the associated inhibit windings to the reference line.

8-5 MEMORY EXTENSION CONTROL TYPE 15.

The standard type 15 memory extension control allows expansion of PDP-1 core memory to a capacity of 32,768 eighteen-bit words contained in eight 4,096-word type 12 memory modules.

However, the type 15 control can be modified to allow further expansion to sixteen type 12 memory modules containing a total of 65,536 eighteen-bit words.

To provide the 15-bit address format necessary to specify one out of 2^{15} memory locations, the program counter and the memory address register are extended three extra bits. The extensions of these registers are included in the type 15 extension control as a pair of three-bit registers, EPC and EMA, respectively. For selection purposes, the eight memory modules are designated as module 0 through module 7. During each memory access, a module is selected according to the three-bit module address contained in EMA, while a specific location within that module is selected according to the normal twelve-bit address contained in MA.

At the beginning of each normal memory cycle, the module address is provided to EMA by the extension of the program counter. Thus, as the program continues, both instructions and operands are retrieved from the same memory module. However, the program may jump to another module and operands may be taken from another module, by performing a defer cycle in the extend mode. The word retrieved from memory during such a defer cycle is interpreted as a 15-bit address instead of the usual 12-bit address. Memory access during the following cycle is then made to the newly addressed module.

In addition to the registers that serve as extensions of PC and MA, the memory extension control also includes buffers and mixers for the transfer of information between the computer and the expanded core memory, module selection logic, and module transfer logic.

a ADDRESS AND DATA TRANSFER CONTROL - If the computer contains a single memory module, the outputs of the memory address decoders and memory buffer register are applied directly to that module. The output pulses from the sense amplifiers are applied directly to the input gating of the memory buffer register.

If the computer includes additional memory modules, the memory address decoder and memory buffer register outputs are applied to all memory modules through the MAD and MB buffers (figure D8-4). The sense amplifier outputs from all modules are applied to MB through the memory buffer mixer (figure D8-5). Data coming into the system over the high-speed channels is also transferred to the memory buffer through the memory buffer mixer.

b ADDRESS EXTENSION - The extensions of the memory address register and the program

counter are shown in the upper right of figure D8-6. The flip-flops in EMA and EPC are numbered 3, 4 and 5, because they form left-hand extensions of MA and PC. The extension of PC stores the current operating module address from cycle to cycle. The extension of MA holds a module address for use during a given cycle.

Whenever either of the normal 12-bit address registers is cleared, its extension is also cleared. At the beginning of every memory cycle, a module address is loaded into EMA at the same time that a regular 12-bit address is loaded into MA. In a normal cycle, the module address originates at EPC; but in an extend-mode cycle, the module address originates at the memory buffer.

During high-speed channel access, a full 15-bit address is provided on the high-speed channel address lines (paragraph 6-10b). The most significant three bits from the extended mixer are loaded into EMA through the direct set inputs.

Whenever a program address is transferred into PC, a module address is transferred into EPC. In a normal cycle, the module address originates at EMA, so that the program continues in the same memory module. However, in an extend mode cycle, the module address originates at the memory buffer, so that the program jumps to a new module. Any module address that is made available to the system from the extension of the console ADDRESS switches is also transferred into EPC.

c MODULE TRANSFER LOGIC - The module transfer logic is shown in the upper left of figure D8-6. The module-address transfer for console operations is shown in B3. A module address is provided from the console to EPC whenever an address is transferred from TA to the program counter. A module address is also transferred from ETA to EPC at SP_2 of Read In. This allows the operator to read information into any memory module.

The address extension transfer pulses for nonconsole operations are generated by the logic nets shown in A1 to A3. Transfers between extended registers always generate the corresponding transfers between the extensions of the registers. That is, any transfer from PC to MA, or MA to PC, is always accompanied by a transfer from EPC to EMA or EMA to EPC, respectively.

For other transfers to EMA and EPC, the origin of the module address depends upon whether or not the computer is performing an extend-mode cycle. An extend-mode cycle is a

cycle which satisfies the following two conditions:

(1) the cycle occurs while the computer is operating in the extend mode, i. e. flip-flop EXD is 1; and

(2) the cycle includes an address transfer that both originates at MB and occurs between TP_5 of a defer cycle and TP_1 of the following cycle.

There are only two types of cycle that can satisfy the second condition. These are the defer cycle of a jump instruction, and cycle one of an indirectly addressed memory reference instruction.

Setting the extend-mode flip-flop, EXD, puts the computer in the extend mode. This limits indirect addressing to a single level. Whenever a defer cycle occurs while the computer is in the extend mode, TP_5 sets the extend-mode cycle flip-flop, emc. The 1 state is emc prevents further indirect addressing and changes the origin of the subsequent module-address transfer.

For normal cycles (emc = 0), an address transfer from MB is accompanied by a module-address transfer between the extension registers. For retrieval of a deferred address or an operand, an address transfer from MB to MA is always accompanied by a module-address transfer from EPC to EMA (A1). For a program jump, an address transfer from MB to PC is always accompanied by a module-address transfer from EMA to EPC (A2).

If, however, the computer is performing an extend-mode cycle (emc = 1), a module address always originates at MB instead of at one of the extension registers. In a memory reference instruction, cycle one is the extend-mode cycle, and the address transfer from MB to MA is accompanied by a module-address transfer from MB to EMA. In a program jump the defer cycle is itself the extend-mode cycle, and the address transfer from MB to PC is accompanied by a module-address transfer from MB to EPC. At TP_1 flip-flop emc is cleared to prevent any further extend-mode transfers.

The rest of the logic shown in the upper left of the figure controls the extend mode of the computer. The extend mode may be controlled either by the operator or by the programmer. Both flip-flops EXD and emc are cleared initially by SC. The computer then enters the extend mode at SP_3 of Start or Read In if the EXTEND switch is on. Control of the extend

mode by the programmer is exercised through the iot instructions Enter Extend Mode and Leave Extend Mode. The command pulse EEM + LEM (IOT 74) sets EXD if MB_6 is 1 and clears EXD if MB_6 is 0.

For sequence breaks the logic provides automatic control of the extend mode. During break cycle one the state of EXD is saved in the accumulator along with the states of OV_1, EPC and PC (paragraph 7-2a). Then, at the end of break cycle one, EXD is cleared so that the computer transfers to the break routine in the normal mode. When a return is made from the break routine, the debreak signal sets EXD at the same time that the break channel is freed (see paragraphs 6-8 and 6-9b). This assures that a transfer may be made to any memory module for a return to the interrupted program. Following the retrieval of the program address from memory, the SBS restoring pulse restores the original state of EXD according to the contents of MB_1.

d MODULE SELECTION - The module-selection decoder is shown in figure D8-6, D4 to D8. This is a standard type 1151 binary-to-octal decoder which decodes the contents of EMA in the same way that the memory address decoders decode the contents of MA. One of the eight memory modules is selected according to the three-bit number contained in EMA. The decoder selects a memory module by enabling one of the sets of pulse amplifiers shown above the decoder. Note, however, that the outputs of the decoder are applied to the sets of pulse amplifiers through a tie-point block. This allows a single decoder output to be connected to any one of the sets of pulse amplifiers. Thus, a given physical memory module can be made to correspond to any one of the eight logical addresses. As a result, if a particular memory module is down, it can be replaced by any one of the other modules without requiring programming changes.

Four control lines connect the control unit and the memory. Three of these lines carry control pulses to the memory (paragraph 6-2e). In the standard computer, these pulses are applied to the single memory module. In multi-module systems, the pulses are applied through the sets of pulse amplifiers in the module selection logic (lower right, figure D8-6). The clear-memory pulse is applied to all memory modules. The memory operate pulses and the inhibit pulse are applied only to the single memory module selected by the module address decoder.

The fourth control line carries the strobe from the memory modules to the control unit. This strobe pulse restarts the timing chain (paragraph 6-2c). In a multi-module system, the strobe from the operating memory module is applied to TP_4 through the one-bit mixer shown in C1.

e MODIFICATIONS FOR FURTHER EXPANSION - The memory system can be further expanded to 16 modules (65,536 eighteen-bit words) by modifying the type 15 memory extension control. An extra bit must be added to each of the extended registers, making EPC and EMA each four bits in length. The extra indicator lights are available at the console. The modified system also requires another set of MAD and MB buffers, a larger memory buffer mixer, and eight more sets of pulse amplifiers in the selection logic.

CHAPTER 9

INPUT-OUTPUT SYSTEM

9-1 GENERAL.

The PDP-1 input-output system includes the peripheral equipment and also two elements of the central processor, the in-out transfer control and the in-out input mixer.

In-out transfer control decodes the second operation code of in-out transfer instructions and provides the various signals necessary for the operation of the peripheral equipment. The operations performed by the command pulses of in-out transfer instructions are listed in a chapter 3 timing chart (table 3-2).

The present chapter includes detailed descriptions of the control units for the three standard in-out devices. These devices are the photoelectric tape reader, the paper tape punch, and the typewriter. Operation of these control units is shown in a chapter 3 flow chart (figure 3-11).

Expansion of in-out transfer control (for the addition of optional peripheral equipment to the computer system) is covered in the present chapter. Also included are a table of the signal connections available at the computer for optional equipment, and a table of the in-out transfer instructions that govern the common input-output options. The in-out transfer control logic needed for the optional devices is treated in the corresponding supplements to the basic manual.

The input-output system is shown in six logic drawings and one wiring diagram, figures D9-1 through D9-7. Extra drawings (figure D9-8, etc.) are furnished if required for control of optional equipment. For information on the use and organization of these drawings see paragraph 3-16.

9-2 IN-OUT TRANSFER CONTROL.

The standard logic circuits for in-out transfer control are shown in figure D9-1. Standard in-out transfer control includes the logic elements that control the reader, punch, typewriter,

display, one-channel sequence break system, and memory extension control. Furthermore, the standard control includes several elements that facilitate the addition of optional equipment to the standard computer.

Decoding of the second op code of in-out transfer instructions utilizes the outputs of the memory buffer decoders (paragraph 6-3c). The outputs of these decoders are applied to in-out transfer control through the 4113R plug-in units shown in the upper left of figure D9-1. The standard secondary op code is six bits in length. This op code is represented by the octal out-puts from MBD_B (bits 12 through 14) and MBD_A (bits 15 through 17). In special cases the op code may be augmented; either by octal information provided through MBD_C or MBD_D, or by binary information provided directly from the bits of the memory buffer register.

The two-digit secondary op code is decoded into one or two command pulses. The more significant octal digit is decoded into a pair of pulses representing the class of in-out transfer instructions (that is, the instructions with op codes in the 30's, the 40's, the 50's, etc.). This pulse decoding is performed by the 4603 plug-in units shown in the upper center of figure D9-1. The single one of the eight possible MBD_B outputs which is asserted gates timing pulses TP_{7-4} and TP_{10-4} to produce a pair of pulses representing the class of iot instructions. Because the entire system is gated by the condition $IOT \cdot ioc^1$, this pair of pulses is produced only during the first cycle of an in-out transfer instruction. The condition $IOT \cdot ioc^1$ is satisfied only when the command level for in-out transfer instructions is asserted from the instruction decoder and the in-out commands flip-flop is in the 1 state.

The pulses representing the class of instructions are then gated by the asserted output of MBD_A (the second octal digit of the secondary op code). This produces the command pulses necessary for execution of a specific in-out transfer instruction. The decoding of class pulses into specific command pulses for the standard iot instructions is shown in the upper right of figure D9-1. Some of these instructions utilize two command pulses; others only one. The standard instructions that are wired into all machines are those instructions that govern the reader, the punch, the typewriter, the display, the one-channel sequence brea system, and the memory extension control.

Note that the command pulse RPB is generated both by the decoding of a second op code in an iot instruction, and by a special read-in mode pulse generated on each cycle of the operation Read In (paragraph 6-2b).

The need-a-completion-pulse logic for the standard iot instructions is shown in the lower left of figure D9-1. The necessity for a completion pulse in an in-out transfer operation is determined by bits 5 and 6 of the instruction word. If bits 5 and 6 are different, a completion pulse is required. The completion pulse flip-flops for the reader, punch, typewriter, and display are included in standard in-out transfer control. The state of each of these four completion pulse flip-flops is indicated by the NAC light located with the indicators for the corresponding device. Completion pulse flip-flops for optional devices (when required) are included in the corresponding device control units. The levels NAC and $\overline{\text{NAC}}$ are available for optional control purposes at the taper pin panels.

The remaining circuits shown in figure D9-1 may be utilized by either the standard equipm or the optional equipment. Any inputs to or outputs from these circuits that may be used by the optional equipment are available at the taper pin panels.

The pulse amplifiers for setting program flags are shown in B 1 to B 3. Any device may set a flag for signaling purposes. The typewriter is the only standard device which sets a program flag. The typewriter SYNC pulse sets pf_1 whenever a typewriter key is struck. The outputs of the program flags, as well as the signals NAC and $\overline{\text{NAC}}$ are buffered by 1685 bus drivers (C1, C2) for use by the optional equipment.

On all low speed or programmed input operations information must be sent into the computer through the in-out register. Since all information transfers into IO are 1 transfers, IO must be cleared prior to the transfer. On in-out transfer instructions, the net shown in B4 clears IO. The in-out register is cleared on the instruction Type In and is also cleared prior to the reader-return signal from the tape reader (provided either a completion pulse has been requested or the computer is in read-in mode). The in-out register is also cleared by any instruction in class 30. All class-30 instructions (e.g., status-checking instructions) clear IO at TP_7 and then load information into IO at TP_{10}. The other inputs to the IO clear net are available for the optional equipment.

Whenever an in-out transfer sets the in-out halt flip-flop, the computer waits for the completion of the in-out operation. The completion of the operation is indicated by the pulse IOT DONE This pulse sets the in-out synchronizer flip-flop. For standard instructions, if the corresponding pulse flip-flop is in the 1 state the synchronization is performed by the completion pulse from the reader, the punch, the typewriter (on output) or the display. For the optional equipment,

additional inputs (both gated and ungated) are available.

In addition to the circuits described above there is a single circuit (shown in C5) which is available for the optional equipment but which is not used by any standard device. This circuit permits any external source to increment the program counter.

The standard inputs of all the circuits described above are shown on the common logic drawing provided with all manuals. Any optional inputs required for a specific computer system are added to the drawing for the manual which accompanies that specific computer. Furthermore any variations in the decoding of the standard instructions required for a specific system are shown in the lower right of figure D9-1. Such variations include decoding for additional typewriters, checking additional registers of status bits, and so forth.

If the optional equipment added to the system requires the generation of only several extra pulses, the decoding for these command pulses is also shown in the lower right of figure D9-1. However, if the optional equipment requires a large number of pulses, the decoding for such equipment is shown in an extra figure added at the end of the present chapter (see paragraph 9-7).

9-3 IN-OUT INPUT MIXER.

The input mixer for the in-out register is shown in figure D9-2. The standard computer includes the pulse amplifiers and the upper row of type 4129 plug-in units. The single row of 4129 units provides four registers of capacitor-diode input gates. If more inputs are required because the computer includes optional in-out equipment, and extra row of 4129 units (which provides four more registers of input gates) can be added to the system.

The transfer pulse RB$\longrightarrow$IO (paragraph 9-4b) loads information from the reader buffer into IO through the first gate in each bit of the mixer. The command pulse CKS loads the status bits through the second gate of each bit in the mixer. The five standard status bits are always loaded into bits 0 to 4 of IO in the order shown in the figure. When extra status bits are required for optional equipment these extra bits are also loaded through the second register of gates using bits 5 through 11. If more than twelve status bits are necessary for the system, additional decoding must be provided for the Check Status instruction and additional registers of gates are used in the mixer. Only twelve gates are available in register 2 for checking

status bits. This is because the instruction Type In transfers the contents of the typewriter buffer into IO through the second gates in bits 12 through 17 of the mixer.

The reader buffer, typewriter buffer and standard status bits are shown on the drawings with all manuals. Any additional status bit inputs or buffer inputs required for a specific computer system are added to the drawing for the manual that accompanies that computer.

9-4 PHOTOELECTRIC TAPE READER CONTROL.

The control unit for the photoelectric paper tape reader is shown in figure D9-3. The 18-bit reader buffer is shown at the top of the figure. The control logic and control flip-flops of the unit are shown at the left of the figure. The reader-buffer status bit (lower right) is a 4113 diode unit connected in a flip-flop configuration.

The outputs of the buffer and control flip-flops of the reader control unit are applied to the lights on the console in-out indicator panel through the indicator drivers at the right. The signal designations of the driver outputs are the same as the designations on the indicator panel.

The inputs from the console READER switch are shown in the lower left of the figure. In earlier machines the reader motor is controlled by a pair of push buttons, START and STOP, mounted on the side of the console operator panel. In more recently delivered machines the motor is controlled by a three-positon toggle switch mounted on the front of the operator panel. When the switch is pushed up the start connection is momentarily closed; when the switch is pushed down the stop connection is momentarily broken. When the start connection is closed, pulse generator 11B11 produces a pulse which duplicates the action of the power-clear pulse, that is, it clears the reader control flip-flops. The momentary closure of the start connection also energizes a relay in the reader, turning on the motor. The -15 volts applied through the normally-closed stop connection keeps the relay energized and the reader motor running. When the stop connection is momentarily opened, the relay is de-energized and the reader motor stops.

a READER BUFFER - The reader buffer shown at the top of figure D9-3 is composed of 18 type 4214 flip-flops. Each flip-flop has a direct clear input and gated 0 and 1 inputs. The input gating to the flip-flops is provided by type 4128 capacitor-diode input gates. These gates utilize ground levels and positive pulses.

When information is read the presence of a hole is indicated by a -3 vdc level. Consequently the ground level utilized by the input gates indicates the absence of a hole. To compensate for this polarity in the input signals the output designations of bits 12 through 17 of the reader buffer are inverted. Thus the clear-RB pulse actually clears bits 0 through 11 of RB, but sets bits 12 through 17. After the buffer is cleared information from holes 6 through 1 on the tape is loaded into RB_{12-17} by a 0 transfer.

Information is loaded into RB by the strobe pulse. This strobe always loads information from holes 6 to 1 into RB_{12-17} by a 0 transfer. If flip-flop rby is 0, indicating that the reader is reading in alphanumeric mode, the strobe also loads the output of holes 8 and 7 into RB_{10} and RB_{11}. The outputs of holes 8 and 7 are inverted, so this transfer is a 1 transfer.

If the computer is reading in alphanumeric mode, only a single line on the tape is read, and the entire line is loaded into RB_{10-17}. If the reader is operating in the binary mode, holes 7 and 8 are ignored but 3 lines are read from the tape. In binary mode, data from holes 6 through 1 is loaded into RB_{12-17} by the strob. Following the strobe pulse a shift pulse shifts the information in RB six places to the left and sets bits 12 through 17. This prepares the buffer for the transfer of information from the next line on the tape. A full-length computer word of 18 bits is assembled in the reader buffer by three strobes and two shifts.

b CONTROL LOGIC - The logic nets and control flip-flops of the reader control unit are shown at the left of figure D9-3. The four reader control flip-flops comprise the two-bit read counter, rc, the read binary flip-flop, rby, and the reader clutch flip-flop, rcl.

When the reader clutch flip-flop is set the reader clutch is engaged, moving the tape. When rcl is cleared the clutch is disengaged and the brake is engaged, stopping the tape. The read binary flip-flop controls the acceptance of information from the tape. If rby is 0 the control unit accepts the first line encountered on the tape and information from all eight holes is loaded into the reader buffer. If rby is 1 the reader accepts information only from lines in which hole 8 is punched. In this case information from holes 1 through 6 is loaded into the buffer while holes 7 and 8 are ignored. The read counter controls the execution of a reader instruction by counting the number of lines read from the tape.

The reader can read the tape in either of two modes, binary or alphanumeric. When the reader is to read binary information command pulse RPB loads 01 into the read counter, sets flip-flops rby and rcl, and clears RB. If the reader is to read information in alphanumeric mode, command pulse RPA loads 11 into the read counter, clears rby, sets rcl, and clears RB.

Command pulse RPA sets rcl by pulsing the complement input. This input allows the program to check the stopping time of the tape by programming two consecutive alphanumeric-mode command pulses. Because the second RPA pulse clears rcl, this command pulse is interpreted by the reader as an order to halt. The program can then check the reader buffer to see if any information was loaded into it. By varying the time between the two RPA command pulses, the stop time of the tape can be measured precisely.

When the feed hole on a tape is encountered, a pulse is produced through the pulse generator shown at C3 in figure D9-3. If certain format and read-counter conditions are fulfilled, the output of this pulse generator produces the strobe that loads data from the holes into the reader buffer. This strobe is produced only if at least one bit of the read counter is 1. The read counter controls the number of lines read from the tape; when the counter is counted to 00 no further information can be accepted. Format control of the strobe depends upon the mode of operation of the reader. If the reader is operating in alphanumeric mode (rby^0) the first feed hole encountered produces the strobe. However, if the reader is operating in binary mode (rby^1) then a feed hole can generate the strobe only if hole 8 is punched. Consequently in the binary mode, the reader searches the tape for lines in which hole 8 is punched.

If the reader is executing the instruction Read Punched Tape, Alphanumeric, the read counter initially contains 11 and the first feed hole encountered on the tape generates the strobe. This loads data from holes 8 to 1 into RB_{10-17}. The strobe also turns off the reader clutch (B4) and produces another pulse through a five-microsecond delay (B1). Since rc contains 11 this delayed pulse does not affect the shift logic. However, it does clear the in-out register provided either a completion pulse has been requested or the computer is in read-in mode. The delayed pulse also produces a second delayed pulse through another five-microsecond delay (B2). This second delayed pulse counts the read counter to 00 and generates the reader-return signal. Since the read counter is then

clear, no further information is accepted from the tape.

If the reader is executing the instruction Read Punched Tape, Binary, the read counter initially contains 01. When a line in which hole 8 is punched is encountered on the tape, the feed hole generates the strobe. This pulse strobes data from holes 6 through 1 into RB_{12-17} and also clears rcl. Five microseconds later the pulse from delay 11A2 generates the shift pulse, shifting the contents of RB_{12-17} into RB_{6-11} and setting flip-flops 12 through 17. This shift pulse also sets flip-flop rcl causing the reader to continue reading the tape.

Five microseconds after the shift, the pulse output of delay 11A3 increments the read counter and the entire process begins again. The tape is searched for another line with hole 8 punched; the strobe loads the data from the holes into the reader buffer; and information is again shifted six places to the left in the buffer. But on this cycle, the pulse that is delayed ten microseconds from the strobe (i.e. the output of 11A3) increments the read counter to 11. The reader then reads a third line from the tape. The strobe transfers the data from the holes into the reader buffer and clears rcl.

After the third line is read, the condition rc = 11 prevents the first delayed pulse from shifting the buffer. Instead the delayed output of 11A2 clears the in-out register (provided either a completion pulse has been requested or the computer is in read-in mode. Five microseconds later the second pulse, which originates at 11A3, generates the reader-return signal and cycles the read counter to the number 00.

When the reader finishes reading the tape in either mode, the appropriate information is in the reader buffer. The last signal produced by the control unit is the reader-return completion pulse. This signal is applied to a pulse amplifier at the in-out input mixer (figure D9-3 D4). If a completion pulse has been requested or the computer is in read-in mode, the reader-return generates RB$\longrightarrow$IO. This pulse transfers the information from the reader buffer through the mixer to the in-out register.

In normal operation, if no completion pulse has been requested the transfer does not take place; instead the reader-return sets the reader-buffer status bit, RBS. This bit then indicates that the buffer contains unretrieved information. The status bit is a 4113 diode unit connected in a flip-flop configuration (C5 to C6). Status bit RBS stays in the 1 state until

the computer executes the instruction Read Reader Buffer. This instruction generates RB⟶IO, which both transfers the information through the input mixer and clears RBS. In read-in mode the transfer automatically takes place and RBS is set. However, the setting of RBS does not matter in this case because no status checking can occur in read-in mode.

9-5 PAPER TAPE PUNCH CONTROL

The punch control unit is shown in figure D9-4. The control unit includes an eight-bit punch buffer, drivers to power the solenoids in the punch, and several control circuits. In addition to the punch and the control unit the punch system includes a type 812 power control panel. This panel is mounted at the top of the console plenum door.

Information may be punched on the tape in either of two modes, binary or alphanumeric. Operation of the control unit is exactly the same in both cases. The only difference between the two modes is in the format of the single line punched in the tape.

Both of the punch instructions require two command pulses. The first command pulse clears the punch buffer and sets flip-flop pun. The second command pulse loads the punch buffer. Command pulse PPA transfers information from IO_{10-17} into PB_{10-17}. Command pulse PPB sets PB_{10}, and transfers information from IO_{0-5} into PB_{12-17}. Flip-flop PB_{10} must be set because a line on the tape is recognized as binary data only if hole 8 is punched.

If either flip-flop pun is in the 1 state, or if the console TAPE FEED switch is held on, then the motor-relay signal is asserted (A4). The motor-relay signal is applied to K1 on the punch motor control panel (paragraph 10-11g.) If the punch motor is on, the punch may be used immediately. However, if the punch is not used for 12 or 13 seconds, the punch motor is turned off. When a punch instruction is initiated with the punch motor off, the control unit must wait for the ready signal from the motor control panel to rise. This occurs one second after the motor-relay signal is asserted; approximately one second is required for the punch motor to reach the required speed.

When the ready signal is asserted and flip-flop pun is in the 1 state, the system is ready to punch. Punching is started by the synchronizing signal from the pick up coil on the punch (B1). This synchronizing signal may arrive any time between 0 and 15.8 milliseconds after pun is set and READY is asserted. The punch provides the synchronizing signal every 15.8 milliseconds.

This means that the maximum operating speed of the punch is one line every 15.8 milliseconds, or 63.3 lines per second.

The synchronizing signal triggers delay 11B9. The control unit utilizes both the level output and the terminating pulse from this delay. For five milliseconds after the delay is triggered, the -3 vdc output on pin J applies the 1 output of the flip-flops in the punch buffer to the solenoid drivers. If a given flip-flop of the buffer contains 1, the corresponding hole on the tape is punched. As a result, the solenoid drivers transfer information from PB_{10-17} to holes 8 through 1 on the tape. At the same time that the data is being punched on the tape, the level out-put of the delay enables the feed hole solenoid drivers (A8). This punches the feed hole and advances the tape to the next position.

At the end of the five-millisecond delay the level output of the delay is disabled and a terminating pulse is produced on pin E. This terminating pulse clears pun and produces the punch completion pulse through the pulse amplifier in A3.

The states of flip-flop pun and the bits of the punch buffer are shown by the lights on the console in-out indicator panel (A4). Whenever a punch instruction is initiated, flip-flop pun lights the ON indicator. There are no designations on the indicator panel punch-buffer lights; these lights are arranged in the same format as the holes appear on the tape.

9-6 TYPEWRITER CONTROL

The PDP-1 typewriter is an IBM Model B equipped with a Soroban electromechanical encoder and decoder. The decoder contains six information solenoids. These solenoids are driven from the typewriter control unit. The positions of the solenoid armatures are mechanically decoded to determine the desired character. Besides providing the six information signals, typewriter control also energizes the typewriter cam magnet, thereby causing the typewriter to print the desired character. The typewriter decodes most control characters (carriage return, backspace, space, tab, and shift), and all print characters, from the six information solenoids. The two color characters, red and black, are decoded by the typewriter control unit, which directly controls the color shift solenoid in the typewriter.

When the operator strikes a typewriter key, the typewriter encoder presents information signals to the typewriter control unit. The encoder mechanically encodes all print characters into six

coded switch closures plus a common switch closure. Control characters are not encoded at the typewriter. When a control character key is struck, a single switch is closed. Closure of such a control character switch is encoded within typewriter control.

The typewriter control unit is shown in figure D9-5. The six-bit typewriter buffer with its associated input gates and output solenoid drivers is shown in the upper right of the figure. The six information solenoid drivers are enabled from the 1 outputs of the typewriter buffer flip-flops whenever the typewriter cam magnet solenoid driver (A5) is also enabled. The color shift solenoid driver (A3) is enabled directly by the 0 state of flip-flop TBB. When TBB is in the 0 state, the color shift solenoid is energized continuously, causing the typewriter to print red. When TBB is set, the color shift solenoid is de-energized, and the typewriter returns to black. The color character decoding nets are shown in B3.

The signal connections between the control unit and the typewriter are shown in the lower left of figure D9-5. The lines to the typewriter information solenoids TM1 through TM6 are at the right end of the connector. The typewriter cam magnet is shown in D4. This solenoid is disabled whenever the margin safety switch is open. If the computer runs the typewriter carriage all the way to the right-hand margin, the margin safety switch opens and the typewriter hangs up.

The color-shift solenoid lines are shown in D1. Only the computer can cause the typewriter to print in red. There is, however, a color shift lever in the typewriter, and the operator can prevent red printing by holding this color shift lever open.

The remaining connector lines provide signals from the typewriter to the control unit. Signals from these lines are applied to the logic through the switch filters shown in C1 and C2. The six print-character switches TC1 through TC6 are shown in C3. The output levels from these switches are applied to the input gates of individual bits of the typewriter buffer (B5 to B8). The common print-character switch TCC, and the various control character switches are shown to the left of the six print characters.

The levels from TCC and the various control character switches are applied to a diode net in A1. An asserted output from this diode net indicates to the control logic that a typewriter key has been struck. The various control characters are encoded by the diode nets at the input gates of the typewriter buffer. Carriage return, tab, and backspace are encoded directly from

the individual control character switch signals. A shift character is partially encoded from the shift cam signal; the decoding is completed by the appropriate directional signal. Since the space character signal is not applied to the encoder nets when the space bar is struck, no buffer input gates are enabled. No inputs are necessary because the code for the space character is 00.

In addition to color shift decoding, two other characters are decoded from the typewriter buffer: carriage return and shift. The carriage return and shift characters are decoded by the diode nets in C3 and C4, respectively. The outputs of these decoders are not applied to the typewriter, but rather are required for the internal logic of the typewriter control unit. Whenever the typewriter executes a carriage return or a shift, extra time must be allowed for type-out operation.

The indicator drivers for the typewriter lights on the console in-out indicator panel are shown in D7. The output designations shown in the figure are the same as the designations engraved on the indicator panel.

The typewriter control unit can execute two sequences of operations, an output sequence and an input sequence. Besides being used during the input sequence, most of the input operations are also utilized during out-put to allow the program to check the correctness of the type-out.

a OUTPUT SEQUENCE - Typewriter output operations are initiated by the in-out transfer instruction Type Out. The preliminary command pulse in Type Out clears the typewriter buffer at TP_7 (figure D9-5B4). At TP_{10} the main command pulse TYO loads the typewriter buffer from bits 12 through 17 of the in-out register (B5). The main command pulse also sets flip-flop tyo and triggers delay D_1 (B4).

The diode net shown at B3 decodes the contents of the buffer to determine whether or not TB contains a color character. If the present character is not a color character, the level output of D_1 enables the inputs to the six information solenoid drivers for 25 milliseconds. A given solenoid driver is enabled if the corresponding bit of TB contains 1. At the same time that the information solenoid drivers are enabled, the typewriter cam-magnet solenoid driver is automatically enabled, causing the typewriter to print.

If the typewriter buffer does contain a color character, the enabling circuit for the solenoid drivers is inhibited. Then, depending on whether the color character is black or red, the terminating pulse output from delay D_1 sets or clears flip-flop TBB. Clearing TBB enables the color shift solenoid driver (A3); this causes the typewriter to print red. The typewriter continues to print red until the color character black appears in the typewriter buffer. When black appears, the pulse output of D_1 sets TBB, disabling the color shift solenoid driver.

After the typewriter has responded to a noncolor character, the return signal from the typewriter switch closures arrives at the diode net in A1. Assertion of the return signal produces a pulse through pulse generator 11C3. This pulse clears the typewriter buffer (B5). The clearing of the buffer cannot affect the print-out, because the return can be asserted no sooner than 80 milliseconds after the type-out instruction is programmed. The 80-millisecond minimum time applies to printed characters; control characters require longer periods.

The pulse that clears TB also triggers delay D_3 (A2). Five microseconds later, the output pulse from D_3 strobes the executed character from the typewriter back into the typewriter buffer (B5). After a typewriter output operation, the program can check the contents of the typewriter buffer to ensure that the correct character was printed.

When the return signal from the typewriter falls, another pulse is produced through pulse generator 11C2 (A1). This pulse can be produced no sooner than 105 milliseconds after the beginning of the print-out. If the output character is neither a shift nor a carriage return (B2), then the pulse from 11C2 generates the type-out completion pulse. This completion pulse clears flip-flop tyo and is applied to in-out transfer control for the standard completion pulse operations.

Since there is no return signal from the typewriter when a color shift is executed, the pulse from 11C2 generates the completion pulse on all characters except shift, carriage return, and a color character. If the presently-executed character is a color character or a shift, the pulse output of D_1 triggers delay D_2 (B4). The 150-millisecond delayed output of D_2 then generates the completion pulse directly (B2). This procedure is necessary for a color shift because there is no return signal from the typewriter. However, it is also required for shift characters because there is a return from the typewriter only

if the shift character is actually executed. If a shift up is programmed when the typewriter is already in upper case, no shift is actually performed and hence there is no return signal. The allowed 150 milliseconds is sufficient for the typewriter to be available again to the computer.

If the typewriter has executed a carriage return, the pulse from 11C2 triggers the 100-millisecond delay D_4 (A3). The pulse output of this delay then produces the completion pulse directly (B2). This additional 100-millisecond delay beyond the fall of the typewriter return signal is necessary to allow the typewriter carriage to settle down, and thus ensure correct execution of further Type Out instructions.

b INPUT SEQUENCE - When a typewriter key is struck, the signal from the appropriate switch closure enables diode net 11C12 (A1). The assertion of the typewriter signal produces a pulse through pulse generator 11C3. The pulse output of 11C3 clears the typewriter buffer (B5) and also triggers delay D_3. Five microseconds later, the pulse output of D_3 strobes the character into the typewriter buffer (B5). The strobe also sets the typewriter buffer status bit TBS (A3) indicating to the computer that the typewriter buffer contains a typed character.

When the signal from the typewriter is disabled, another pulse is produced through pulse generator 11C2 (A1). This pulse in turn produces the typewriter sync pulse through pulse amplifier 11B5 (A3). The typewriter sync pulse provides the sequence break signal and sets program flag 1 as another indication to the computer that a typewriter key has been struck.

Program flag 1 may be sensed by a Skip instruction while the TB status bit may be checked in IO if a sequence break is initiated. In either case the character can be retrieved from the buffer only by executing a Type In instruction. When the Type In command pulse TYI transfers the character from TB through the input mixer to IO, it also clears TBS (A3).

9-7 OPTIONAL IN-OUT TRANSFER CONTROL

Additional peripheral equipment can readily be added to the PDP-1 system by expanding the in-out transfer control section of the computer. The various signal connections required for the addition of extra equipment are available at the taper pin panels shown at the right of

figure D9-6. The signal type, polarity, and direction are shown by arrows and diamonds in the figure. Arrows and diamonds pointing to the left indicate input signals; those pointing to the right indicate output signals.

The various signal connections used for the standard in-out equipment (e.g., signal connections to the input mixer, connections for the completion-pulse logic, and so forth) are shown in the taper-pin layout drawing included with all manuals. When additional peripheral equipment is included in the system, the various signal connections required are added to the layout drawing for the specific computer.

The signal connections shown in the layout drawing are listed in table 9-1. This table includes signal name, direction, number of lines, signal type and polarity, the type of PDP-1 plug-in unit that produces or receives the signal, and general information about the use or meaning of the signal. In the column headed "Number" two numbers are listed. The first number is the number of independent signals; the second number is the number of lines available for each signal.

When extra equipment is added to the system, connection from in-out transfer control to the device control unit is made through 50-pin amphenol connectors located in the in-out plug panels. The six mounting holes (numbered from right to left) for these in-out plugs are available in mounting panel 3F. If additional plugs are required they are mounted in panel 3E. Layouts of the 50-pin in-out plugs for the common in-out equipment options are shown at the left of figure D9-6.

The MBD outputs and class pulses ($KX \cdot TP_{7-4}$, $KX \cdot TP_{10-4}$) needed to decode additional in-out transfer instructions are available from taper pin panels 3J1 and 3J2 (figure D9-6). As is mentioned in paragraph 9-2, when only a few additional command pulses are needed the logic nets that generate these pulses are shown in the lower right of figure D9-1. If a considerable amount of logic is required for the control of optional equipment, this logic is shown in an additional drawing, figure D9-7. This additional drawing shows the generation of command pulses and whatever other signals are required by the optional equipment. The description of the logic required for any individual option is provided in the corresponding supplement to the basic manual.

The in-out transfer instructions required for the common optional devices are lested in table 9-2. This table lists the instruction, its meaning, and appropriate codes. Whenever the instruction

cannot utilize an in-out wait, the entire instruction code is shown in the table. If in-out halting and requirement of a completion pulse are at the discretion of the programmer, only the second operation code is listed. In some cases the second op code is augmented either by octal or binary information. When the "0" and "1" represent binary digits, they are enclosed in parentheses. Variable digits are represented by letters; upper case for octal, lower case for binary. The letter "X" (or "x") indicates a digit that is ignored.

Note that one of the optional instructions actually includes a third operation code. The instruction mic has primary op code 72, indicating an in-out transfer instruction with no in-out wait. The second op code (75) is the instruction that must be executed by the tape control unit to receive control information from the computer. Bits 6 and 7 of the in-out transfer instruction word are decoded to determine which tape control (out of three possible control units) is addressed. Bits 8 through 11 of the instruction word provide a third operation code. This third op code is decoded by the tape control unit to determine the instruction that the control unit must execute in governing an individual tape unit.

Whenever any additional logic circuits are required (either for the generation of signals that are not ordinarily necessary for common options, or for signals that are required for special equipment) the appropriate logic is shown in additional drawings numbered D9-8, D9-9 and so forth. These drawings include the logic circuits, layouts of additional in-out plugs (when required) and explanatory notes. If there is a requirement for additional explanation not provided in the usual supplements to the basic manual, special addenda sheets are provided for a specific computer.

TABLE 9-1 SIGNAL CONNECTIONS BETWEEN CENTRAL PROCESSOR AND PERIPHERAL EQUIPMENT

Signal	Direction	Number	Polarity	DEC Circuit At PDP-1	Remarks
IO_{0-17}	Out	18x9	⟶◇ = 1	1685	General programmed output transfers
MB_{0-17}	Out	18x3	⟶● = 1	1685	For HSC data transfers
AC_{0-11}	Out	12x3	⟶● = 1	4113	For oscilloscope or special equipment
AC_{15-17}	Out	3x3	⟶◇ = 1	1685	To address tape unit with type 52 control
pf_{1-6}	Out	6x3	⟶● = 1	1685	May be sensed for general control purposes
MBD_A	Out	8x3	⟶◇	4113	For decoding 2nd op code
$MBD_B \cdot IOT \cdot ioc^1 \cdot TP_{7-4}$	Out	8x3	⟶●	4603	For command pulses (2.5 and 5.0 μs after beginning of memory cycle)
$MBD_B \cdot IOT \cdot ioc^1 \cdot TP_{10-4}$	Out	8x3	⟶●	4603	
MBD_C	Out	8x3	⟶◇	4113	For forming in-out transfers, addressing sequence break channels, etc.
MBD_D	Out	8x3	⟶◇	4113	
NAC	Out	1x3	⟶◇	1685	Need a completion pulse
NAC	Out	1x3	⟶◇	1685	Do not need a completion pulse
IM_{0-17}	In	18x8	⟶● = 1	4129	For incoming information, levels must be present for 2.0 μs. (4 inputs installed, 4 extra wired and become available with addition of 9 type 4129 plug-in units)

TABLE 9-1 SIGNAL CONNECTIONS BETWEEN CENTRAL PROCESSOR AND PERIPHERAL EQUIPMENT
(Continued)

Signal	Direction	Number	Polarity	DEC Circuit At PDP-1	Remarks
$HSAM_{3-17}$	In	15x3	⟵ = 1	4129	For address in HSC access
$HSBM_{0-17}$	In	18x3	⟵ = 1	4129	For HSC data transfers
HSC Control					
CHAN REQ	In	3x1	⟵	4105	Requests HSC transfer
CHAN IN	In	3x1	⟵	4105	Indicates incoming information
WORD XFER	In	3x1	⟶	4603	Indicates HSC access granted
SBS Break Signals					
1 Channel	In	1x8	⟶	4110	Requests sequence break
	In	1x5	⟵	4110	Requests sequence break
Type 20	In	16x1	⟶	4126	Requests sequence break
Miscellaneous Control Pulses - (Output pulses must be buffered by 4603 before driving separate lines)					
$\vert\xrightarrow{+1}$ PC	Out	1	⟶	4603	Indicates program counter advanced
TP_{4-4}	Out	1	⟶	4603	Occur 1.0, 2.5 and 5.0 μs after beginning of memory cycle, respectively
TP_{7-4}	Out	1	⟶	4603	
TP_{10-4}	Out	1	⟶	4603	
SC-4	Out	1	⟶	4603	Clear pulse in console operation Start (= SP_1)

TABLE 9-1 SIGNAL CONNECTIONS BETWEEN CENTRAL PROCESSOR AND PERIPHERAL EQUIPMENT
(Continued)

Signal	Direction	Number	Polarity	DEC Circuit At PDP-1	Remarks
STOP-4	Out	1	→ 4603	4603	Pulse from console STOP switch
POWER CLEAR	Out	1	→	4603	Occurs when power is turned on or off
$\xrightarrow{+1} pf_n (n = 1, \ldots, 6)$	In	6x2	→	4112	Any device may set flag
$\xrightarrow{0}$ IO	In	1x10	→	4110	Clears in-out register
$\xrightarrow{+1}$ PC	In	1x6	→	4110	Advances program counter
IOT DONE	In	1x6	→	4110	Ends in-out wait

TABLE 9-2 IN-OUT TRANSFER INSTRUCTIONS FOR OPTIONAL PERIPHERAL EQUIPMENT

Instruction	2nd Op Code or Instruction Code	Definition
Visual Display Type 30		
dpy	07	Display one point on CRT. If both displays are present, a 1 in bit 8 selects type 30.
Precision Display Type 31		
dpy	(1)xx07	Display one point on Precision CRT. If intensity option included, bit 7 selects intensity; 1 = high, 0 = low.
Card Punch Type 40		
pac	43	Punch a card.
lag	72XX22	Load a group of 18 columns and index field counter.
Programmed Magnetic Tape Type 51		
mcb	72XX70	Magnetic tape, clear buffer
mwc	72XX71	Magnetic tape, write a character
mrc	72XX72	Magnetic tape, read a character
msm	72XX73	Magnetic tape, select mode
mcs	72XX34	Magnetic tape, check status
Automatic Magnetic Tape Type 52 - All type 52 instructions use op code 72; tape control unit is addressed by bits 6 and 7 (ab = 00, 01, 10).		
muf	72abxX76	Take magnetic tape unit address from AC_{15-17} and final address from IO. Additional tape control information may be contained in bits 10 and 11.
mic	72abnN75	Take magnetic tape initial address from IO and execute command nN.
mri	72abxX66	Magnetic tape, reset initial address from IO.

TABLE 9-2 IN-OUT TRANSFER INSTRUCTIONS FOR OPTIONAL PERIPHERAL EQUIPMENT
(Continued)

Instruction	2nd Op Code or Instruction Code	Definition
mrf	72abxX67	Magnetic tape, reset final address from IO.
mes	72abxX35	Magnetic tape, examine status
mel	72abxX36	Magnetic tape, examine location
Line Printer Type 62		
lpr	(00)X45	Line printer, print
lfb	721X45	Line printer, fill buffer
lsp	(10)X45	Line space the line printer

CHAPTER 10

CIRCUIT DESCRIPTION

10-1 GENERAL

This chapter describes the function and operation of 52 circuits used in the standard PDP-1 computer and in the associated central processor options. Three additional circuits are described. These are power supplies 729 and 742, and power control 810. These three units used in earlier models are replaced in later models by power supply 728 and power control 813. All circuits except power supplies and controls are plug-in modules; i.e., all components are mounted on DEC standard etched circuit boards.

Schematic diagrams are included for all circuits, except that any module which includes a final "R" in its type number shares the same schematic with the unit having the same type number without the "R". Inverters 1103 and 1103R are an example of such a pair. The additional connections of the R type are indicated by dotted lines in the common schematic. The schematic diagrams are grouped at the rear of the manual, in order by type number. No figure reference is made in individual unit descriptions, but references to the applicable schematics are implied.

10-2 INVERTERS

The inverter modules used in PDP-1 are made up of combinations of three basic circuits: a -3 vdc supply, a diode-clamped load resistor, and a basic inverter. The clamped loads and the -3 vdc supplies are all identical. In inverter 1103, diode D1 and resistor R13 make up a typical clamped load, while diodes D7, D8, D9 and D10 and resistor R19 form a typical -3 vdc supply.

There are two types of basic DEC inverter, differing in speed of operation. These are the high-speed (5-mc) inverter and the low-speed (500-kc) inverter. Module 1103 contains a typical high-speed inverter composed of transistor Q1, resistors R1 and R2, and capacitor C1. Module 4105 contains a typical low-speed inverter, composed of Q1, R1, R2 and C1. The two inverter types differ only in transistor type, and in the value of the base input bypass

capacitor (C1). These two differences affect only the switching speed of the circuit. The typical delay time of the high-speed 1000 series is 20 nanoseconds, while that of the low-speed 4000 series is 0.3 microseconds. Both types of inverter are used as level gates or pulse gates, and both are driven by DEC standard levels and negative pulses.

The inverter transistors are operated in two modes, saturation and cut-off. When an inverter transistor is in the saturated state, collector-emitter impedance is very low. Conversely, at cut-off, collector-emitter impedance is very high. If the emitter is at ground, and the collector is connected to a -3 vdc clamped load, the collector output level (or pulse) is an inversion of the base input level (or pulse). For example, if the base input level is ground, the transistor is cut off. The output is then -3 vdc, determined by the clamping voltage. However, if the base input level is -3 vdc, the transistor saturates. The ground level at the emitter is then also present at the output.

Base input loading is determined by the 3K base resistor. With -3 vdc present at the base input and the emitter at ground, a saturating current of 1 ma flows through the transistor. The base input bypass capacitor provides overdriving current to speed transistor switching. When the base input is at ground, the 68K resistor to +10 vdc supplies I_{co} to achieve good dc cut-off of the transistor. This 68K resistor also acts as a voltage divider with the 3K input resistor to shift the base positive, thereby preventing accidental transistor turn-on by noise pulses.

The diode in the clamped load limits the negative voltage at the inverter outputs. It does this by providing a low-impedance path from the -3 vdc supply when the output voltage at the collector of the inverter transistor is more negative than -3 vdc. The clamping diode thus supplies whatever current is needed to maintain a 12-volt rise (from -15 vdc) across the 1.5K load resistor. This current is a maximum of 8 ma under no-load conditions, and decreases to zero as the current drawn from the external load increases to 8 ma. The value of the clamped-load resistor thus determines the maximum external load current at which the inverter can maintain a regulated -3 vdc output.

The -3 vdc supply is established by the four 0.75 vdc forward voltage drops across four series-connected 1N645 silicon diodes. Current flows from ground through the four diodes and then through the parallel combination of the supply load and the 560 ohm resistor. This resistor accepts enough current to maintain a -3 vdc diode voltage even under minimum-

load conditions.

a INVERTER 1103 - This 5-mc module contains six basic inverters, six clamped loads, and a -3 vdc supply. All logic terminals (base input, emitter, and collector) are accessible at the output pins of the module. The clamped loads are not connected to the output pins, or to the basic inverters.

b INVERTER 1103R - Inverter 1103R differs from inverter 1103 in only one respect. Each of the six clamped loads in the 1103R is internally connected to the collector output of the corresponding basic inverter circuit. The schematic for the 1103 inverter also represents the 1103R inverter provided that the dotted lines between each collector output and the associated clamped load are considered to be wiring connections.

c INVERTER 1104 - This 5-mc module contains four basic inverters, four clamped loads, and a -3 vdc supply. Both the logic terminals and the clamped-load terminals are accessible at the output pins of the module. Bias return for transistors Q1 and Q2 is to +10 vdc (A). For transistors Q3 and Q4, bias return is to +10 vdc (B). This division permits submodular marginal testing and thus facilitates troubleshooting.

d INVERTER 1105 - This 5-mc module contains five basic inverters, three clamped loads, and a -3 vdc supply. All logic and clamped-load terminals are accessible at the output pins of the module. Bias return for transistors Q1 and Q2 is to +10 vdc (A). The bias return for transistors Q3, Q4, and Q5 is to +10 vdc (B).

e INVERTER 4105 - This module is the 500-kc equivalent of inverter 1105 (d above).

f INVERTER 4106 - This module is the 500-kc equivalent of inverter 1103 (a above).

g INVERTER 4106R - This module is the 500-kc equivalent of inverter 1103R (b above).

10-3 DIODES

The diode modules used in PDP-1 contain one or more diode logic gates. Each gate is internally connected to an inverter base input. A clamped load is provided for each inverter. A single -3 vdc supply (described in paragraph 10-2) is included in each diode module. Diode modules

1110, 4110., 4112, and 4112R contain negative OR gates (OR gates for negative levels). Diode modules 1111, 4111, 4113, and 4113R contain positive OR gates (AND gates for negative levels). Inputs can be driven either by DEC standard levels or, except for the 1111 and 4111, by negative pulses.

a DIODE 1110 - This 5-mc module contains two identical six-diode negative OR gates (composed of diodes D2-D7 and D9-D14 respectively). The following description of a six-diode negative OR gate refers to the circuit containing diodes D2-D7, but applies equally to the circuit containing diodes D9-D14.

The gate is driven by +10 vdc (A and B) applied through parallel puller resistors R1 and R2. This voltage forward biases the diodes. The voltage drop across the diodes is low. As a result, the voltage at the base-input resistor R4 approaches the lowest voltage present at any of the gate inputs: K, L, M, N, P, or R.

If one or more of the six inputs is a negative logic level (-3 vdc), a negative level is applied to the base input of the inverter transistor Q1. This turns Q1 on, causing the collector output of Q1 to rise to ground (provided that the emitter is connected to ground).

The inverter transistor is cut off only in the event that none of the inputs are negative (i.e., only if all six inputs are ground levels). A ground level is then applied to the base input of the inverter. The switching delay of the negative OR gate is typically 30 nanoseconds. Diode D20 prevents the base of Q1 from being driven too positive when no input connections are made. It serves as a gate diode with a permanently grounded input connection.

Resistors R1 and R2 are connected to the A and B +10 vdc supplies respectively. This prevents excessive sensitivity of circuit operation to the marginal test. Submodular marginal testing is still possible, however, because resistors R3 and R9 are connected to separate +10 vdc lines (A and B respectively).

b DIODE 1111 - This 5-mc module contains two identical six-diode positive OR gates (composed of diodes D1-D6 and D7-D12 respectively). The following description of a six-diode positive OR gate (an AND gate for negative levels) refers to the circuit

containing diodes D1-D6, but applies equally to the circuit containing diodes D7-D12.

Diodes D1 through D6, together with resistor R1, constitute a current-switching diode gate. Any diodes that have their anodes at ground are forward-biased, providing current to R1. If all inputs are at -3 vdc, the gate output is isolated from the input, and current is furnished to R1 by the transistor base through diodes D13 and D14.

If one or more of the six inputs is a ground level, the transistor is cut off. The transistor is turned on only in the event that none of the six gate inputs are positive; i.e., only if all six inputs are negative (-3 vdc) levels. A negative level is then applied to the base input of the transistor. This turns Q1 on, causing the collector output of Q1 to rise to ground. The switching delay of the positive OR gate is typically 30 nanoseconds.

If the source of a nominal ground input level is the collector of a saturated transistor, the true input voltage may be slightly negative. The voltage drop across the gate diodes further reduces the actual voltage applied to the cathode of diode D13. To compensate for these reductions in input ground levels, current from resistor R3 produces a forward voltage across silicon diodes D13 and D14. This voltage is large enough to ensure positive bias at the base of Q1 (and reliable cut-off of the transistor) when any normal ground input is applied to the gate.

To permit sub-modular marginal testing, the two positive OR gates in the 1111 diode module are connected (through resistors R3 and R4) to the A and B +10 vdc supplies respectively.

c DIODE 4110 - This module is the 500-kc equivalent of diode 1110 (a above). The two negative OR gates contained in this module have a switching delay of approximately 0.4 microseconds.

d DIODE 4111 - This module is the 500-kc equivalent of diode 1111 (b above). The positive OR gates contained in this module have a switching delay of approximately 0.4 microseconds. Capacitors C3 and C4 are speed-up capacitors used to reduce switching time.

e DIODE 4112 - This module consists of six identical two-diode negative OR gates. Six clamped loads are also included in the 4112 module, but at present these clamped

loads are not used.

The two-diode negative OR gate functions in a similar manner to the six-diode negative OR gates previously described. If a negative logic level is applied to either of the two input terminals, then the inverter transistor is turned on. The inverter transistor is turned off only when ground levels are applied to both input terminals.

The six negative OR gates contained in the 4112 module have a switching delay of only 0.3 microseconds. The 4112 unit has a shorter turn-on and turn-off time than diode 4110, even though the impedance of its base circuit is higher than that of the 4110. This is because the 4112 diode module uses a much faster transistor. Because both the A and B supplies of the 4112 module are connected to the base resistors of all six gates in this module, submodular marginal checking is not feasible.

f DIODE 4112R - This module is identical to diode 4112 (e above) except that each of the six clamped loads in the 4112R is internally connected to the collector output of the corresponding gate transistor. The schematic for the 4112 module also represents the 4112R module provided that the dotted lines between each collector output and the associated clamped load are considered to be wiring connections.

g DIODE 4113 - This module consists of six identical two-input positive OR gates. Six clamped loads are also included in the 4113 module, but at present these clamped loads are not used.

The two-diode positive OR gate functions in a similar manner to the six-diode positive OR gates previously described (e. g. module 4111 treated in d above). If a ground level is applied to either of the two input terminals, then the inverter transistor is turned off. The inverter transistor is turned on only when negative logic levels (-3 vdc) are applied to both input terminals.

The six positive OR gates contained in the 4113 module have a switching delay of approximately 0.16 microseconds. A speed-up capacitor in the base circuit of each gate (shunting the two series-connected silicon diodes) accelerates the turn-off of the transistor when one or more of the gate inputs is raised to a ground level. The base resistors of three of the six gates are connected to the A supply; the base resistors

of the remaining three gates are connected to the B supply. This permits sub-modular marginal checking of the 4113 circuit.

h DIODE 4113R - This module is identical to diode 4113 (g above) except that each of the six clamped loads in the 4113 is internally connected to the collector output of the corresponding gate transistor. The schematic for the 4113 module also represents the 4113R module provided that the dotted lines between each collector output and the associated clamped load are considered to be wiring connections.

i BINARY-TO-OCTAL DECODER 1150 - Both outputs of each of three flip-flops are applied to the input terminals of the binary-to-octal decoder. The decoder has eight output terminals numbered 0 through 7. For any given combination of states of the three flip-flops which furnish the decoder inputs, one specific output of the decoder is a -3 vdc level, and the remaining seven decoder outputs are ground levels.

The binary-to-octal decoder module is composed of eight identical parts. Each of these parts is a three-diode negative OR gate (which is logically equivalent to a positive AND gate). Except for the number of gate diodes, each of these gates is identical to one of the negative OR gates included in diode module 1110 (a above).

The eight decoder output terminals shown in the 1150 schematic represent, from left to right, the octal numbers 0 through 7. The output signals generated by the decoder always include a single -3 vdc level at one of these eight terminals. The remaining seven output terminals are then at ground. The -3 vdc level is generated at a specific output terminal: that terminal which represents the octal equivalent of the binary number in the three input flip-flops.

The eight diode gates are each connected to a different set of input lines (refer to the lower portion of the 1150 schematic). These connections are arranged so that each of the eight gates responds to one of the eight possible combinations of 0's and 1's that can be generated by three flip-flops.

As in the case of diode module 1110 (a above), a given gate transistor is cut off, thereby producing a -3 vdc output, only when ground levels are applied to all of its gate input diodes. Because of the gate input configuration, only one of the eight

gates receives ground levels at all three input diodes. The remaining seven gates must each have at least one negative input level (-3 vdc). Consequently, the seven associated transistors remain saturated, thereby producing ground output levels at all but one of the decoder output terminals.

Each of the eight sets of three-diode gate inputs is connected to one output terminal (either the 0 terminal or the 1 terminal) of each of the three input flip-flops. When a flip-flop is in the 1 state, its 0 and 1 output terminals are at ground and -3 vdc respectively. For the 0 state of the flip-flop, the polarity of the output terminals is reversed. The decoder logic senses the 1 state of an input flip-flop as a ground level taken from the 0 output terminal of the flip-flop. Conversely, a ground level from the 1 output terminal asserts the 0 state of the flip-flop.

The output terminals of the flip-flop representing the least significant of the three binary digits being decoded are connected to inputs L (1 out) and K (0 out). The flip-flop outputs for the next most significant digit are similarly connected to inputs J and H, while those for the most significant digit are applied to inputs F and E. Because the 1 state of a flip-flop is asserted by a ground level from its 0 terminal, the input connections for each gate are the complement of the three-bit binary number being decoded. For example, the gate which decodes octal 7 (= binary 111) is connected to the 0 output terminals of all three flip-flops. These three terminals are all at ground when the three flip-flops contain the binary number 111. With all three diodes at ground, the gate transistor is cut off, and the terminal 7 output of the binary-to-octal decoder drops to -3 vdc.

i BINARY-TO-OCTAL DECODER 1151 - Operation of this decoder is almost the exact reverse of the 1150 decoder (described in i above). As in the case of the 1150, both outputs of each of three flip-flops are applied to the input terminals of the 1151 decoder. However, positive OR gates (rather than negative OR gates as in the 1150 decoder) are used for decoding the contents of the three input flip-flops.

Consequently, the input connections to the 1151 decoder are the reverse of those required for the 1150 decoder. For example, in the 1151, the gate which decodes octal number 7 (= binary 111) is not connected to the 0 output terminals of the three

input flip-flops, as in the 1150 decoder, but instead is connected to the 1 output terminals of these three flip-flops. Therefore, these three gate terminals are all at -3 vdc when the three flip-flops contain the binary number 111.

With all three gate diodes at -3 vdc, the positive OR gate causes the associated inverter transistor to saturate. This results in a ground output level being applied to output terminal 7 of the binary-to-octal decoder. Because each of the remaining seven gate transistors is held off by one or more ground level inputs, the remaining seven decoder output terminals remain at -3 vdc.

The selected output of the 1150 decoder is the single -3 vdc level among seven ground level outputs; in contrast, the selected output of the 1151 decoder is the single ground level among seven -3 vdc levels.

10-4 CAPACITOR-DIODE GATES

The capacitor-diode gate modules contain pulse gates. Standard 0.4 microsecond negative DEC pulses are generally applied to the pulse inputs of these gates. Sometimes a logic level is applied to the pulse input, a pulse being generated by differentiating the negative transition. Logic levels are applied to the gating inputs. The polarity of the logic level input applied to a specific pulse gate determines whether or not that gate will generate an output pulse when an input pulse is applied to it.

The 4126 and 4128 positive capacitor-diode gate modules contain pulse gates which have an inverter input stage, and an output gating stage. These two modules are only used for reading information into unbuffered flip-flops type 4214. The pulse gates in the 4127 and 4129 negative capacitor-diode gate modules are constructed differently. These modules have an input gating stage and an output inverter stage. All capacitor-diode gates may be used for sampling the outputs of unbuffered flip-flops, where flip-flop output is used as a gating level. The contents of the flip-flop can then be sampled by applying a standard DEC pulse to the pulse input of the capacitor-diode gate module.

a POSITIVE CAPACITOR-DIODE GATE 4126 - The 4126 module contains six identical pulse gates. The following description refers to the pulse gate having pulse input terminal F, but applies equally to the other five gates on the module.

The circuit is operated by applying a standard negative DECpulse to input terminal F. This input pulse is inverted by transistor Q1. The resulting positive pulse at the collector of Q1 is applied to the pulse gate composed of capacitor C4, resistor R8, and diode D4. The rising edge of the pulse is differentiated by C4, and may, if the gate is enabled, be applied through diode D4 to the output of the circuit. The sharp negative spike caused by differentiation of the trailing edge of the pulse never appears at the output of the circuit, but is instead discharged through R8.

The logic level applied to gating input E determines whether or not the pulse gate will pass the positive spike generated by the leading edge of the input pulse. A negative gating level (-3 vdc) prevents the generation of an output pulse by providing dc back-bias to diode D4. A ground gating level permits the generation of an output pulse.

When a -3 vdc level is applied to gate terminal E, the junction of capacitor C4 and diode D4 drops from ground to -3 vdc. The delay required for this change in voltage is determined by the time constant of R8 and C4. The gate is inhibited when the anode of D4 is at -3 vdc. With the gate inhibited, no positive pulse of less than three volts can cause the junction of C4 and D4 to rise above ground to forward bias D4. Therefore no pulse can be applied to the load at output terminal H (which is normally at ground potential).

When a ground level is applied to gate terminal E, the junction of capacitor C4 and diode D4 rises from -3 vdc to ground. The delay required for this change in voltage is determined by the time constant of R8 and C4. The pulse gate is enabled when the anode of D4 is at ground. Any positive pulse is then sufficient to forward bias D4; consequently any positive pulse is passed through D4 to the load.

The 4126 module is only used to read information into an unbuffered flip-flop type 4214. For this use, the pulse output of the gate is connected to either the 0 or the 1 input of the flip-flop. The positive output pulses from the pulse gate then set or clear the flip-flop. The delay built into the capacitor-diode circuits is useful for preventing logical race problems. Delay is necessary to avoid splitting pulses when a flip-flop is sampled at the same time it is pulsed (e.g. if the output of the flip-flop were used as the gate input to a 4126 module which were in turn used to read information into the gating flip-flop). Because of the delay built into the circuit, the ground gate

enabling level must be present at least 1.5 microseconds before a shift or jam transfer, and at least 4.5 microseconds before a standard read-in operation.

b NEGATIVE CAPACITOR-DIODE GATE 4127 - The 4127 module contains six identical pulse gates. The following description refers to the gate containing transistor Q1, but applies equally to the other five gates on the module.

The inverter stage of the gate is located at the output of the gating circuit, rather than at the input as in the 4126 module (a above). The 4127 pulse gates are generally used to sample the outputs of unbuffered flip-flops. The flip-flop output is applied to input terminal F of the gate, and a standard 0.4 microsecond negative DEC pulse is applied to input terminal E. If the flip-flop output is a negative level (-3 vdc) the gate is enabled. The negative input pulse then generates a positive-going output pulse at terminal H. If the flip-flop output is a ground level, the gate is inhibited and no output pulse is generated.

A logical delay is built into the circuit to prevent logical race problems (refer to a above). Because of this delay, the gating level must be present one microsecond before the arrival of the input pulse.

The pulse gate is composed of capacitor C1, diode D1, and resistor R2. Through resistor R5, the output of the gate is referenced to a dc level of -3.75 vdc. This voltage source is provided by an additional series-connected diode, D25, which is added to the -3 vdc supply in the module. Diode D2, and resistor R1 are included to prevent the level gate input from going more negative than -3 vdc. Should the input be driven too negative, diode D1 might be forward biased, causing undue sensitivity of the circuit for noise inputs at E.

When the gate is enabled by the application of a -3 vdc level to input terminal F, capacitor C1 charges to -3 vdc through resistors R1 and R2. Diode D1 is still reverse-biased, because its anode is at -3.75 vdc. But, when a negative pulse is applied to pulse input terminal E, and is differentiated by C1, diode D1 is forward biased. The resulting negative pulse at the junction of D1 and R5 is coupled through capacitor C2 to the base of transistor Q1. This turns on the transistor, causing output terminal H to rise to ground potential.

At the trailing edge of the input pulse, diode D1 is cut off. Some rise in voltage is coupled through C2 before D1 cuts off. However, base of Q1 is clamped to ground through diode D3, so no excessive back-bias is applied to the base of Q1.

The clamped loads in the 4127 module (diode D4 and resistor R4, D8 and R9 etc.) are not used. No connections are made from the output pins of the module to these loads.

c NEGATIVE CAPACITOR-DIODE GATE 4127R - The 4127R module differs from the 4127 in only one respect. Each of the six clamped loads in the 4127R is jumpered to the collector output of the corresponding inverter. The schematic for the 4127 also represents the 4127R provided that the dotted lines between each collector output and the associated clamped load are considered to be wiring connections.

d POSITIVE CAPACITOR-DIODE GATE 4128 - This module contains two identical units. Each of these units is composed of a single pulse inverter and four positive capacitor-diode gates. These gates are identical to the gates used in the 4126 module (a above). The single pulse inverter provides the pulse input for all four of the capacitor-diode gates.

Use of the 4128 module permits economical read-in or shift operations with the type 4214 quadruple flip-flop (paragraph 10-3i). Although each set of four diode gates receives a common pulse input, the four gating levels are independent. This permits independent setting (or clearing) of each individual flip-flop on the quadruple flip-flop module.

As in the case of the 4126 unit, a given gate is enabled by a ground gating level, and inhibited by a -3 vdc level. The level inputs must be present at least 1.5 microseconds before a shift or jam-transfer type of read-in and at least 4.5 microseconds before a standard read-in operation. A single 4128 inverter-capacitor-diode unit can be conveniently used to control both the setting and clearing of all four individual flip-flops on a 4214 quadruple flip-flop module.

e NEGATIVE CAPACITOR-DIODE GATE 4129 - This module contains two identical units. Each of these units is composed of four negative capacitor-diode gates ORed into a pulse inverter output stage. These gates are identical to the gates used in the

4127 module (b above). A 68K resistor returns each gate level input to +10 vdc. This prevents any unused gates from affecting the rest of the circuit. The 4129 module also includes a negative dc supply identical to the supply in module 4127. (This supply provides both -3 vdc and -3.75 vdc outputs.)

The pulse gates of the 4129 module are generally used to sample the outputs of unbuffered flip-flops. The flip-flop output is applied to the level input of the gate, and a standard 0.4 microsecond negative DEC pulse is applied to the pulse input. Because of the logical delay built into the circuit, the gating level must be present one microsecond before the arrival of the input pulse.

10-5 FLIP-FLOPS

The flip-flop modules used in PDP-1 may contain one (modules 1201 and 4201), two (modules 1204, 1209, and 4209) or four (modules 1213, 4213, and 4214) flip-flops. All flip-flops change state when an on transistor is turned off by a positive pulse applied to its base. Except for the flip-flop transistors in modules 4201 and 4209, the collectors of the nonconducting transistors are clamped to -3 vdc. Each module includes a -3 vdc supply for clamping voltage and some contain inverters for use as pulse or level gates.

a FLIP-FLOP 1201 - This module contains a single buffered flip-flop, two inverters, and a -3 vdc supply consisting of four forward-biased diodes connected in series. The voltage-dropping resistor from -15 vdc, incorporated in the -3 vdc supplies of previously described modules is omitted. This can be done because one of the two flip-flop transistors and one of the two output buffer transistors are always off. Current from the voltage-supply diodes can thus flow to -15 vdc through the clamped loads of the non-conductiong transistors.

Inputs to the 1201 flip-flop are set, clear, 0 in, 1 in, and two complement inputs. The set and clear inputs use 70 nanosecond positive (2.5 volt) pulses. The remaining inputs are all driven from the collectors of the pulse gates. These gates in turn are driven by 70 nanosecond negative pulses. Each flip-flop has four outputs: 0 out, 1 out, and two carry pulse outputs. The two carry pulse outputs are the complement input pulses inverted by a transformer.

When the flip-flop contains 0, a -3 vdc level is present at the 0 output terminal (terminal F), and a ground level is present at the output terminal (terminal R). Flip-flop transistor Q4 is then saturated, and flip-flop transistor Q5 is cut off.

Output buffer transistor Q6 is also saturated when the flip-flop contains 0; conduction through Q6 clamps the 1 output to ground. Output buffer transistor Q3 is cut off when the flip-flop contains 0; the 0 output terminal of the flip-flop is then held at -3 vdc by clamped load D3-R5.

The two flip-flop transistors Q4 and Q5 remain stable in the saturated and non-conducting states respectively, so long as the flip-flop is not set by an incoming trigger pulse. The clamped load at the collector of Q5 accepts enough base current from Q4 (through R13) to hold Q4 saturated. The voltage divider network between the collector of Q4 and +10 vdc (composed of resistors R10 and R14) provides a sufficient positive voltage at the base of transistor Q5 to hold Q5 cut off.

The flip-flop is equally stable in the 1 state. A -3 vdc level is then present at the 1 output terminal, and a ground level is then present at the 0 output terminal. Flip-flop transistor Q5 is then saturated, and flip-flop transistor Q4 is then cut off.

Output buffer transistor Q3 is also saturated when the flip-flop contains 0; conduction through Q3 clamps the 0 output to ground. Output buffer transistor Q6 is cut off when the flip-flop contains 1; the 1 output terminal of the flip-flop is then held at -3 vdc by clamped load D12-R17. The output buffer transistors Q3 and Q6 are held on or off by voltage dividers R8-R7 and R15-R16, respectively.

If the flip-flop is to be triggered by negative-going input pulses, then the set input is applied through a pulse-inverter gate such as Q1, rather than being directly applied to the flip-flop input terminal. In this case, the collector terminal of Q1 is jumpered to flip-flop input E (1 in). So long as the gate transistor remains off, the terminal E input remains at -3 vdc. Capacitor C5 isolates this -3 vdc level from the base of Q4.

Unless the gate is inhibited (by applying -3 vdc to emitter terminal Z), the flip-flop may then be set by applying a DEC standard 70 nanosecond negative pulse to the base of gate transistor Q1. This triggering pulse momentarily saturates the gate, thereby raising the voltage at input terminal E from -3 vdc to ground. This voltage rise is coupled

through capacitor C5, diode D5, and the parallel combination of C8 and D19, to the base of Q4. The voltage rise at the base of Q4 back-biases the base-emitter diode of Q4, cutting off the transistor.

When transistor Q4 turns off, its collector voltage drops until it reaches the clamp voltage. This drop in voltage is coupled to the base of transistor Q5 through R10 and speed-up capacitor C7, thereby turning on transistor Q5 and completing the flip-flops change of state.

The output buffer transistors are driven by the flip-flop collectors; these transistors also switch. Transistor Q3 is turned on (applying a ground to the 0 output terminal of the circuit) and transistor Q6 is cut off (causing the 1 output terminal of the circuit to drop to -3 vdc). Transistor Q3 is turned on by the drop in voltage at the collector of Q4. This voltage drop is coupled to the base of transistor Q3 through resistor R7 and speed-up capacitor C6. Similarly, transistor Q6 is cut off by the rise in voltage at the collector of Q5. This voltage rise is coupled to the base of transistor Q6 through R16 and speed-up capacitor C11.

Capacitors C16 and C17 (located at the bottom of the 1201 schematic) synchronize the change in flip-flop outputs during the set switching. If the flip-flop were set without capacitors C16 and C17 in the circuit, the 0 output voltage would tend to change before the 1 output voltage. This lag would occur because transistor Q3 would be turned on directly by the cut-off of Q4. However, transistor Q6 would not be cut off until after the cut-off of Q4 had turned on Q5. Capacitors C16 and C17 avoid this problem. Capacitor C17 delays the turn-on of Q3 by sharing the driving current. When transistor Q3 starts to turn on, the resulting voltage rise at the collector of Q3 is coupled through C16 to the base of transistor Q6, thereby accelerating the cut-off of Q6. When the flip-flop is cleared, the functions of C16 and C17 are reversed; C16 shares the driving current to Q6, and C17 speeds the turn-off of Q3.

At the end of the set trigger pulse, input terminal E returns to -3 vdc. Diode D5 is then reverse-biased, and diode D22 shorts the discharge of capacitor C5 to ground. Resistor R4 determines the discharge time required for the input terminal to return to -3 vdc (thereby preparing the circuit for the arrival of the next set pulse). The circuit is not likely to be spuriously triggered by small noise pulses. Diode D19 is a

silicon diode. Its forward-current threshold voltage is series-combined with that of diode D5 to prevent small positive noise voltages from setting the flip-flop.

Because the flip-flop is a symmetrical circuit, it is cleared in a similar manner to the way it is set. Clearing is accomplished by applying a positive-going pulse to the 0 input terminal (terminal T). If the flip-flop is to be cleared by negative-going input pulses, then the clear-input is usually applied through pulse-inverter gates such as Q2. In this case the collector terminal of Q2 (terminal U) is jumpered to the 0 input terminal of the flip-flop. A positive pulse applied to the direct-set input (pin N) sets the flip-flop to the 1 state. In this case the level shifting components of input E are not needed, and are omitted. Diode D1 isolates the direct-set input from other set inputs with which it is ORed. Similarly, a positive pulse applied to the direct-clear input (pin P) clears the flip-flop to the 0 state.

The type 1201 flip-flop is complemented when a positive pulse from the collector of a pulse gate is applied to either pin J or pin L. (Both complementing inputs function in the same manner. The following description of the complementing circuit including pin J applies equally to the other complementing circuit.) If the flip-flop is in the 0 state, transistor Q4 is on and Q5 is off. Terminal 5 of the lower left secondary winding of T1 is connected to the collector of Q4 and is therefore at ground. Terminal 8 of the lower right secondary of T1 is at -3 vdc because it is connected to the collector of Q5.

A pulse applied to input J is coupled to the secondaries of T1 and appears as a positive-going pulse from ground at terminal 6 and as a positive-going pulse from -3 vdc at terminal 7. Because pulse height does not exceed 3 volts, the pulse from terminal 7 cannot forward bias diode D8 and therefore cannot reach the base of Q5. The pulse from terminal 6 does rise to a positive level and draws current through D6, D19 and the base of Q4, cutting off Q4. This causes the flip-flop to change state by turning on Q5.

The third secondary winding of T1 has its terminal 4 grounded. The input is coupled to this secondary, and appears at terminal 3 as a negative-going pulse from ground. This output produces a 2.5-volt 70-nanosecond output pulse 17 nanoseconds after the base of the complement input inverter is pulsed.

Resistor R23 and diode D18, which shunt the primary of T1, have no effect during the

input pulse rise time, since D18 is back-biased. However, at the collapse of the pulse, D18 shunts the primary of T1 with the resistance of R23. This clamps the overshoot of transformer T1 and prevents the appearance of any substantial second pulse at the secondaries of the transformer.

Output S is used as an indicator driver. Whenever the flip-flop contains a 1, output S supplies power to an indicator lamp transistor. Resistor R20 limits this output current to 1 milliampere.

b FLIP-FLOP 1204 - This module contains two identical buffered flip-flops, two pulse inverters internally connected to the 1 inputs of the flip-flops, and a -3 vdc supply. As in the case of flip-flop 1201, the -3 vdc supply is established by the forward voltage drop of four series-connected silicon diodes, D27 through D30. Because both flip-flops are identical, the following description of flip-flop A (on the left in schematic 1204) applies equally to flip-flop B.

Flip-flop A has a gated 1 input (pin L), a gatable clear input (pin N), a direct clear (pin M) and a complement input (pin K). Pins E and P are the 0 and 1 outputs, respectively, while pin J is the indicator light output.

The flip-flop can be set to the 1 state by applying a 70-nanosecond negative pulse to pin F while pin H is at ground. A positive-going pulse applied to pin N clears the flip-flop. This pulse must come from the collector of a pulse gate similar to Q1. The flip-flop may also be cleared by a 70-nanosecond 2.5-volt positive pulse at input M. A 70-nanosecond negative pulse at input K complements the flip-flop.

With the flip-flop in the 0 state, transistors Q4 and Q6 are on, and Q3 and Q5 are off. Transistor Q1 is cut off at all times except when its emitter is grounded (at H) and a negative pulse is applied to its base (pin F). When Q1 is cut off, its collector is at -3 vdc. If the gate is enabled by a ground level at H and a negative pulse at F, Q1 saturates, and its collector voltage rises to ground. Capacitor C2 differentiates the rise, generating a positive pulse referenced to ground at the anode of D3. This pulse forward biases D3. The diode starts to conduct, thereby turning off Q4. The collector of Q4 drops to -3 volts. Resistors R3 and R8 couple this voltage to the bases of Q3 and Q5, turning these transistors on. The collector of Q4 rises to ground. Resistor R12

couples this voltage to the base of Q6, turning off Q6 and completing the change of state of the flip-flop.

In response to a set input pulse, ordinarily the 0 output buffer amplifier, Q3, changes state before Q6, the 1 output buffer amplifier. However, capacitors C7 and C8 synchronize these changes in state by initially delaying the turn-on of Q3 and speeding up the turn-off of Q6.

The flip-flop may be cleared in a similar manner by applying a positive-going pulse to input N. It may also be cleared through the direct clear input (pin M) by applying a positive pulse at this input. In this case the level-shifting components, D12 and C11, are not needed. Diode D11 isolates input N from inputs M and K by ORing N with these two inputs.

A negative pulse at the complement terminal K produces a positive-going pulse of less than 3 volts peak at terminals 3 and 6 of the two secondaries. If the flip-flop is in the 0 state, the collector of Q6 is at ground. Terminal 4 of the secondary is also at ground since it is connected to the collector of Q6. Therefore, if the flip-flop contains a 0, the pulse generated at terminal 3 rises from ground. Similarly, terminal 5 is connected to the collector of Q3. Because in the 0 state Q3 is cut off, the pulse generated at terminal 6 rises from -3 vdc if the flip-flop is in the 0 state. Only the diode which is in the base circuit of the on transistor passes a current to cut that transistor off.

Output J is used as an indicator driver, and furnishes current to an indicator lamp when the flip-flop contains a 1. Resistor R15 limits this current to 1 milliampere.

c FLIP-FLOP 1209 - This module contains two identical buffered flip-flops, inverter pulse gates for the 1 inputs, and a -3 vdc supply. Except that a direct set input is substituted for the complement input to each flip-flop, the 1209 module is similar to module 1204 (b above). The complement circuitry of each flip-flop of module 1204 is omitted, and single diodes (D6 and D18) connected in their place. As a result, inputs K and V are direct set inputs to their respective flip-flops.

For example, if transistor Q4 of the left-hand flip-flop is on, the flip-flop is in the 0 state. When a standard DEC 70-nanosecond positive pulse appears at pin K, the pulse is coupled through diode D6 and the parallel combination of D4 and capacitor

C6 to the base of Q4, cutting it off. The flip-flop is then in the 1 state.

d FLIP-FLOP 1213 - This module contains four identical flip-flops (1 through 4), eight capacitor-diode gates, one pulse inverter, and a -3 vdc supply. The four flip-flops are designed to operate as a unit. With slight changes in external pin connections, the module can operate as a four-stage shift register or as a four-bit buffer register. Both types of operation are described below.

When its base (input E) is pulsed, the clamped inverter (transistor Q1) may be connected to drive the shift-one and shift-zero pulse inputs (inputs S and V) to all four flip-flops. The eight capacitor-diode gates are used to set and clear each flip-flop. Note that only flip-flop #1 has an external connection for a 0 input (pin P). (The other three flip-flops in the module have internally-connected 0 inputs, but no 0 terminal is brought out to the module connector.) A clear input (pin M) is used to clear all four flip-flops simultaneously. The inverted input pulse signal is available at the pin F pulse output terminal. Each flip-flop has individual 1 and 0 outputs.

Silicon diodes D17 through D20 make up the -3 vdc supply. Resistor R28 to -15 vdc, in series with the diodes, furnishes sufficient current to keep the diodes forward biased. This provides a constant -3 vdc source.

For use as a buffer register, pin F (pulse out) is externally connected to pin S (shift one). When a 70-nanosecond negative pulse arrives at pin E, this pulse is inverted by Q1, and appears at pins F and S as a positive pulse referenced to -3 vdc. This enables the set input gates of all four flip-flops. Gating capacitors C11, C13, C15 and C17 differentiate the pulse, which is referenced to the level present at the one-in terminals of the four flip-flops (pins N, R, T, and U). This level may be either -3 vdc or ground. Since the cathode voltages of diodes D2, D6, D10 and D14 are close to ground, the pulse can pass these diodes only when the corresponding one-in terminals are at ground. For example, if flip-flop #1 is in the 0 state, transistor Q2 is on and transistor Q3 is off. If the one-in terminal (pin N) is at ground, a pulse through pin S can pass diode D2 to the base of Q2, cutting that transistor off. This turns on Q3, switching the flip-flop to the 1 state. If the one-in terminal of the flip-flop is at -3 vdc, no change in state can occur when pin S is pulsed.

When used as a buffer, no individual zero-in transfers take place. The entire register is cleared (by a 70-nanosecond positive pulse to pin M) before the parallel transfer of 1's into the register. The flip-flop side of capacitor C3 is returned to approximately -0.75 vdc to provide noise immunity for the clear input.

For use as a shift register, pin F is externally connected to both pin S and pin V. The 0 outputs of flip-flops #1, #2 and #3 are each externally connected to the 1 inputs of the next-most-significant stages (J to R, L to T, and X to U). These external connections parallel the existing internal connections from the 1 outputs of flip-flops #1, #2 and #3 to the individual 0 inputs of flip-flops #2, #3, and #4, respectively. Although only flip-flop #1 has an external 0 input, all four flip-flops have identical reset capacitor-diode gates. These gates are pulsed by the shift zero line.

As a shift register, flip-flop module 1213 operates in the following manner. If all four flip-flops are initially in the 0 state, the first 70-nanosecond negative pulse arriving at pin E is inverted by Q1 and appears as a positive pulse on both the shift one and shift zero lines. This enables the 0 and 1 inputs to all four flip-flops. If ground is present at the one-in input of flip-flop #1 (pin N), that flip-flop is set to the 1 state. This causes the 0 output of flip-flop #1 to rise to ground (pin J). Because pin J is connected to pin R, the 1 input of flip-flop #2, the second shift pulse sets flip-flop #2 to the 1 state. The second shift pulse may or may not clear flip-flop #1, (depending on the level present at the zero-in terminal of that flip-flop).

The third shift pulse sets flip-flop #3 to the state of flip-flop #2. If the preceding pulse cleared flip-flop #1, its 1 output is at ground. This ground output causes the third shift pulse to clear flip-flop #2. The 1's and 0's injected at flip-flop #1 thus propagate through the entire shift register on successive shift pulses. Each stage assumes the state of the next less significant stage. The delay of the capacitor-diode input gates in accepting level changes prevents any ambiguity in the flip-flop outputs at pulse time. This delay is small compared to the pulse rate.

It is possible to set or reset one of the four flip-flops individually without pulsing the shift lines. For instance, to clear flip-flop #4 independently, the collector of an inverter is connected to the 1 output of the flip-flop. When flip-flop #4 is in the 1 state, transistor Q8 is off and Q9 is on. The flip-flop remains in the 1 state because

the collector of Q8 supplies sufficient current through R26 to keep Q9 saturated. The level shift produced by R23 from the collector of Q9 to the base of Q8 reverse-biases Q8, holding it cut off.

If the emitter of the inverter is at ground, and the inverter is driven to saturation, its collector (together with pin Y) will rise to ground. This causes Q9 to be cut off. The collector voltage of Q9 falls to the clamp voltage and turns on Q8. The flip-flop is then stable in the 0 state. The flip-flop can be set in the same way by grounding its 0 output. A change of state performed in this manner does not affect the other flip-flops in the module, and can occur independently of the shift pulses. It is important that the emitters of inverters that are used for this purpose are wired permanently to ground.

e FLIP-FLOP 4201 - This module is the 500-kc version of flip-flop 1201 (a above). It differs in that flip-flop transistors Q3 and Q4 are unclamped, and the complement inputs are capacitor-coupled to the transistor bases. At cutoff, the collector of Q3 or Q4 is at approximately -4 vdc.

The delay between the complement-pulse input and the carry-pulse output is approximately 0.05 microseconds.

f FLIP-FLOP 4209 - This module is the 500-kc version of flip-flop 1209. It differs from the 1209 module in three respects. First, set and clear are gatable inputs similar to the 1 and 0 inputs. Second, the module has no indicator-light output. Finally, pins J and U are both complement inputs, capacitor-diode gated to the bases of the flip-flop transistors.

g FLIP-FLOP 4213 - This module is the 500-kc version of flip-flop 1213. It operates in a similar manner to that flip-flop.

h FLIP-FLOP 4214 - This module contains four identical flip-flops (#1 through #4) and a -3 vdc supply. Each flip-flop has 0 and 1 inputs and outputs. Moreover, each pair of flip-flops shares a single clear input: pin P for flip-flops #1 and #2, and pin R for #3 and #4. A standard DEC 0.4-microsecond positive pulse, applied to either pin P or pin R, clears the two associated flip-flops. Because all four flip-flops are identical, the following description of flip-flops #1 applies equally to the other three flip-flops.

When flip-flop #1 is in the 0 state, transistor Q1 is on and transistor Q2 is off. The flip-flop is stable in this state because the negative collector voltage of Q2 is coupled to the base of Q1 by R2. Enough current flows through R2 to keep that transistor saturated. The collector of Q1 is therefore at ground. The voltage divider R5-R6 biases Q2 off.

To set the flip-flop to the 1 state, the positive pulse output of a capacitor-diode gate is applied to pin H (one in). This pulse turns off transistor Q1 and thereby causes the turn-on of transistor Q2. The flip-flop remains in the 1 state because the negative voltage at the collector of Q1 drives the base of Q2 into saturation, and the ground at the collector of Q2 keeps Q1 cut off. Silicon diodes D3 and D4 have a forward-bias threshold of over half a volt, and thus block smaller noise-pulse inputs from the transistor bases.

The bases of the two transistors in each flip-flop are returned to separate +10 vdc lines to allow more precise trouble localization through marginal testing.

i FOUR-BIT COUNTER 4215 - This module contains four flip-flops for use as counter bits. The four flip-flops, A, B, C and D, are logically independent. The flip-flops may therefore be connected in any logical configuration. When the flip-flops are connected as a counter, the significance of each flip-flop as a counter bit is determined only by the external connections. The module also contains 12 positive capacitor-diode gates (C14-R30-D1 and C1-R7-D5 are two examples) and a negative dc supply. The supply, consisting of diodes D21 to D24 and resistor R29, is similar to the standard supply (paragraph 10-2), except that -0.75 vdc is tapped from the junction of D21 and D22.

Flip-flops B, C and D have complement inputs at terminals R, K and E respectively. Flip-flop A has separate set and clear inputs, but may be complemented by applying a signal to both inputs simultaneously. Flip-flop C has an inhibit level input at M. When the inhibit is enabled, flip-flop C cannot be complemented from 0 to 1, although it may still be complemented from 1 to 0.

Positive pulses (either a standard 0.4 DEC pulse or the positive-going output of a pulse inverter) or a positive 3-volt step (such as the 1 output of a less significant counter bit when that flip-flop goes from 1 to 0) drive the complement and clear inputs. A -3 vdc level at pin M enables the inhibit to flip-flop C. Carry propagate time per bit is 50 nanoseconds. The following description of flip-flop A also describes the other three flip-flops.

A positive pulse applied to terminal W (set one) or terminal V (set zero) reaches the base of the associated flip-flop transistor only if the transistor is on. The positive pulse changes the state of the flip-flop by turning the on transistor off. When flip-flop A is 1, a positive pulse at input V passes through capacitor-diode gate C15-R31-D4 to the base of Q2, turning Q2 off. The flip-flop switches to the 0 state. Positive pulses at V now have no further effect, since gate C15-R31-D4 is disabled when FFA is 0. However, capacitor-diode gate C14-R30-D1 is enabled when FFA contains 0, so a positive pulse at W passes to the base of Q1, turning Q1 off, and switching FFA to the 1 state.

If W is jumpered to V, the resulting combined input is a complement input. When this input is pulsed, gates C14-R30-D1 and C15-R31-D4 steer the pulse to the base of the on transistor, turning it off, so that the flip-flop switches state. Capacitor-diode gate C1-R7-D5, associated with terminal X, is permanently enabled because R7 is returned to -0.75 vdc. Thus a positive pulse at X clears flip-flop A directly. Resistor R7 is returned to -0.75 vdc rather than to ground in order to prevent spurious noise from affecting the flip-flop. Since signal voltages are greater than -0.75 vdc, they pass through D5, but small noise signals are blocked. The gate time constant (R7 x C1) is longer than that of the other two capacitor-diode input gates, because a 1.0-microsecond pulse is used to clear the counter. This allows carries to die out before the pulse ends.

Terminal R (add FFB) is the complement input to flip-flop B. A positive pulse at R is applied simultaneously to two capacitor-diode gates. These gates, C16-R32-D6 and C17-R33-D9, steer the positive pulse to the base of the on transistor, turning it off. When flip-flop B is 1, gate C17-R33-D9 is enabled, and a positive pulse at R clears FFB. Conversely, when FFB is 0, gate C16-R32-D6 is enabled, and a positive pulse at R sets it. The complement input to flip-flop D, terminal E (add FFD), operates similarly.

Terminal K (add FFC) is the complement input to flip-flop C. Pulses at K are gated by two capacitor-diode gates, like the two input gates of flip-flop B, described above. One of the two input gates of FFC is returned to the collector of Q6. This gate, C19-R35-D14, is enabled when FFC is 1. If FFC is 1, a positive pulse at K is gated to the base of Q6, turning it off, and clearing the flip-flop. The other gate, C18-R34-D11, is returned to the output of a negative OR gate, of which the Q5 collector is one input, rather than directly to that collector.

This OR gate, composed of D25, D26 and puller resistor R38, functions similarly to the gates described in paragraph 10-3e (diode 4112). The anode of D11 is at the more negative level of either the collector voltage of Q5 or the inhibit level at terminal M. When M is ground, FFC functions as the others in the module, and is complemented by each positive pulse at K. However, if M is at -3 volts, a pulse at K cannot reach the Q5 base, even though the Q5 collector is ground (the flip-flop is in the 0 state). The path to the base of Q6 is not affected by an input at M, so the flip-flop may be complemented from 1 to 0 regardless of the inhibit level.

10-6 AMPLIFIERS

The present paragraph describes seven modules which are not PDP-1 logic elements, but instead serve to provide power amplification for PDP-1 logic pulses and levels. Pulse amplifiers 1607, 4603 and 4604 amplify and standardize DEC logic pulses. Bus drivers 1684, 1685 and 1690 amplify the power of logic levels, and also determine the rise and fall time of changing levels. The remaining three modules, indicator amplifier 1669 and solenoid drivers 4680 and 4681, are basically switches which enable a small amount of input power at logic level voltages to control larger amounts of power at higher voltages for use in external circuits.

a PULSE AMPLIFIER 1607 - This module contains three identical pulse amplifiers, three inverters, and a -3 vdc supply. The three pulse amplifiers include transistors Q2-Q3, Q5-Q6, and Q8-Q9. The inputs to the three amplifiers are pins H, L and P, while the outputs are pins E-F, J-K, and M-N, respectively. The three inverters are transistors Q1, Q4 and Q7. Diodes D16 through D19 make up the -3 vdc supply.

A pulse amplifier generates an output pulse whenever its input is grounded. The input may be grounded by connecting it to the collector of one or more pulse gates (such as the Q1 circuit). The input to the combined circuit is then the base input of the pulse gate. Normally the signal applied to the input is a DEC 70-nanosecond negative pulse. However, the input requirement is satisfied by any two to five volt negative pulse having a fall time less than 50 nanoseconds, and a width of at least 50 nanoseconds at minus two volts. When the input is pulsed by a signal meeting these specifications, the output generates a DEC standard 70-nanosecond pulse delayed by 25 nanoseconds. This pulse is capable of driving 16 units of pulse load, or 20 units where the load is near by.

Because all three pulse amplifiers are identical, the following description of the amplifier including transistors Q2 and Q3 applies equally to the other two amplifiers in the module.

Assume that the emitter of Q1 is grounded (pin Z) and that the collector of Q1 (pin X) is connected to the emitter of Q2 (pin H). In the quiescent state, transistors Q1, Q2 and Q3 are cut off. The collector of Q1 and the emitter of Q2 are connected to the junction of resistor R3 and silicon diode D1, and are therefore at approximately -4 vdc. Resistor R3 and diode D1, together with R4 and D2, form a voltage divider between -15 vdc and -3 vdc (with the diodes forward-biased). The base of transistor Q2 is connected to the voltage divider at the junction of D1 and D2, and is held at approximately -3.3 vdc.

Another voltage divider is formed by the series combination of R11 and R16. This voltage divider holds the collectors of transistors Q2 and Q3 at approximately -8 vdc. Diode D4 is forward biased, so that the collector voltages of Q2 and Q3 are separated by only 0.3 volts. The base and emitter of Q3 are at ground. There is no voltage across outputs E and H.

When an input pulse appears at the base of Q1 (pin Y), this transistor saturates and grounds the emitter of Q2. This causes Q2 to saturate; resistor R4 limits the base current. The collector of Q2 drops from -8 vdc to ground. This drop immediately appears across the primary of T1. This voltage remains fairly constant because of the low-resistance voltage divider R11 and R16. When the transformer starts drawing more current than originally flowed through resistor R6, diode D4 disconnects the collector circuits from R11-R16. The voltage across transformer T1 begins to decrease. Capacitor C2 tunes the circuit to give the correct output pulse width. When the transformer voltage has dropped to zero the output pulse ends. Resistor R5 and diode D3 clamp the overshoot in the primary of T1.

The Q3 circuit (including R10, T2, D5, R7 and C4) amplifies the pulse from the secondary of T1. The output pulse may be made negative by grounding pin F of the secondary of T2; or positive, by grounding pin E. A terminating resistor in the range of 82 to 220 ohms is used at the ends of cable distribution lines to prevent signal reflections.

b PULSE AMPLIFIER 4603 - This module contains three identical pulse amplifiers and three inverters. The inverters (Q1, Q4, and Q7) are similar to those in module 4105 (described in paragraph 10-2e). Inputs for the three amplifiers are pins H, L, and P; outputs are pins E-F, J-K, and M-N, respectively.

When properly driven, a pulse amplifier produces a standard DEC 0.4 microsecond 2.5 volt output pulse. The pulse is positive if the negative output is grounded, and negative if the positive output is grounded. To drive the circuit, the emitter of the amplifier input transistor is momentarily grounded. This must be done through the collector of an inverter pulse gate, such as those included in the module. The input to the base of the pulse gate is usually a 0.4 microsecond DEC negative pulse. However, it is only necessary that the input pulse have a negative amplitude between two and five volts, a leading edge less than 0.2 microseconds, and a width greater than 0.3 microseconds.

Because all three pulse amplifiers are identical, the following description of the amplifier with input H applies equally to the other two amplifiers. The description assumes that the collector of Q1 (pin X) is connected to pin H and that the emitter of Q1 is grounded.

When the pulse amplifier circuit is in the quiescent condition, pulse gate transistor Q1 and amplifier transistors Q2 and Q3 are all cut off. Circuit voltages are determined by current flowing through series resistors R3, R5, R6, and the parallel combination of R9 and R12. The base of Q2 is coupled to the voltage divider by R4, and is at -3.0 volts. The emitter of Q2 is connected to a slightly more negative point in the divider chain by R7, and is consequently negative with respect to its base.

The base of Q3 is grounded through the secondary of transformer T1. The emitter of Q3 is also at ground (it is grounded by emitter degenerating resistor R11). The collectors of both transistors are at -7.5 volts, the voltage at the junction of R9, R12, and R6. Since diode D1 is forward-biased and its voltage drop is negligible, it need not be considered in calculating the voltage division. No current flows in either T1 or T2, and there is no output voltage across terminals E and F.

When a negative pulse, meeting the input requirements, is applied to pin Y, transistor Q1 saturates, grounding the emitter of Q2. Current flowing through resistor R4 then saturates Q2, driving terminal 2 of transformer T1 to ground. Terminal 1 of T1 remains at -7.5 volts, since the voltage source is of fairly low impedance. The voltage included in the secondary of T1 is proportional to the voltage appearing across the primary.

Increasing current flows in the primary of T1. However, the voltage across the primary remains nearly constant until the transformer starts drawing more current than originally

flowed through R9. At this time, diode D1 disconnects the collector of Q2 from the voltage divider. Capacitor C5 tunes the primary winding of T1 to give the proper pulse width. When the voltage across the transformer drops to zero, the output pulse ends. Resistor R8 damps the overshoot.

In a similar manner, the circuit of Q3 further amplifies and shapes the pulse. This circuit consists of Q3, an emitter degenerating transistor R11, output transformer T2, damping components D2 and R10, and bypass capacitor C4. The negative output terminal is pin E; the positive output terminal is pin F.

c PULSE AMPLIFIER 4604 - This module contains three identical pulse amplifiers. The first pulse amplifier includes transistors Q1 to Q3, and has its inputs at terminals E and F. Outputs are at J and H. The second PA is composed of Q4, Q5 and Q6; inputs are M and N; outputs are S and T. The third includes Q7 to Q9, with inputs at Y and Z, and outputs at V and X. An additional pair of control terminals is associated with each of the three pulse amplifiers. For the first, these control connections are K and L. Shorting K to L with an external jumper connects an internal capacitor in the amplifier circuit. When connected, this additional capacitance lengthens the duration of the output pulse to 1 microsecond. The corresponding control connections for the second and third pulse amplifiers are terminal pairs P-R and U-W.

Negative-going signals with an amplitude of 2.5 to 4 volts, a fall time of less than 0.5 microseconds, and a width greater than 60 nanoseconds drive inputs E, N and Z. Positive-goind signals with an amplitude of 2.5 to 4 volts, a rise time of less than 0.5 microseconds, and a width greater than 60 nanoseconds drive inputs F, M and Y.

When properly driven, each amplifier produces a DEC standard 0.4-microsecond pulse across its outputs. If the external jumpers are added to the circuits, the outputs produce 1-microsecond pulses. Because the three amplifiers are identical, the following description of the circuit including Q1, Q2 and Q3 applies to all three.

The pulse amplifier consists of a monostable multivibrator (Q1 and Q2), and an output pulse amplifier (Q3). Capacitor-diode C4-D6 couples a negative input at E to the Q2 base. Capacitor-diode C3-D3 couples a positive input to the primary of transformer T1. This transformer inverts the positive input, and D5 couples the resulting negative pulse to

the base of Q2. An appropriate pulse at either input thus provides a negative pulse at the Q2 base. This negative pulse triggers the multivibrator. The multivibrator generates a negative output pulse, which is amplified by the Q3 circuit. The output is a negative pulse at J if H is grounded, or a positive pulse at H if J is grounded.

In the quiescent state Q1 is on, and Q2 and Q3 are off. Base current for Q1 flows through R1, holding Q1 in saturation. Voltage divider R7-R9 shifts the slightly negative Q1 collector voltage positive at the base of Q2, keeping Q2 at cut-off. Diode D1 clamps the Q2 collector to -3 vdc. Voltage divider R8-R10-R13 biases the base of Q3 positive, holding Q3 off. The Q3 collector is somewhat more negative than -7 volts, as determined by voltage divider R11-R14. No current flows in the primary of T2, and there is no output across the secondary.

When a negative pulse is applied to input E, C4 differentiates the leading edge of this input signal, generating a negative pulse at the cathode of D6. This pulse forward biases D6, and passes to the Q2 base. Q2 turns on, and its collector voltage jumps from -3 volts to ground. This positive step is coupled by C2 (or C1 in parallel with C2, if pins K and L are jumpered), to the Q1 base. Q1 cuts off and its collector voltage drops to -3 volts. Current flows from the base of Q2 through R7, holding Q2 on even though the input pulse has ended. The multivibrator remains in this state until the coupling capacitance (C2, or C1 and C2) from the Q2 collector to the Q1 base discharges. This discharge time is proportional to the capacitance. Hence, the multivibrator stays in its temporary state 0.4 microsecond if only C2 is in the circuit, or 1 microsecond if both C1 and C2 are in the circuit. After the appropriate time, Q1 turns on, cutting Q2 off. The multivibrator is back in its quiescent state.

The negative pulse generated at the Q1 collector turns on Q3. The Q3 collector rises to ground, placing approximately 7 volts across the primary of T2. Resistors R11, R14 and capacitor C6 stabilize the voltage at terminal 1 of T2 so that the primary voltage does not diminish appreciably during the pulse. The output voltage at the secondary is proportional to the primary voltage. The pulse terminates when the multivibrator returns to its quiescent state, cutting off Q3. D8 and R12 damp the overshoot in the primary of T2. D7 clips the overshoot at -15 volts, so that excessive voltage is not applied to the Q3 collector.

A positive pulse at terminal F triggers the same chain of events to produce an output pulse

across J and H. However, the pulse is inverted by T1 before being applied to the Q2 base. D2 and R5 damp the transformer during recovery. D5 blocks the positive recovery pulse from the Q2 base.

d BUS DRIVER 1684 - This module contains four non-inverting level amplifiers, and a -3.75 vdc supply. Each amplifier output provides logic levels at low impedance, for use in heavily loaded logic lines. The inputs are pins K, M, U, and S (amplifiers 1 through 4 respectively), each input representing 1 unit of base load. The respective outputs are pins L, N, T, and R. The output impedance of each unit is 22 ohms, with output capability up to 5 units of base load. This maximum output can be increased to 20 units of base load by connecting a two watt, 120 ohm resistor between pins C and E. Switching time is less than 75 nanoseconds. Because the four amplifiers are identical, only the first amplifier, with input at K and output at L, is described

When a ground level is applied to input K, resistors R1 and R2 act as a positive voltage divider, biasing transistor Q1 off. With Q1 off, the collector of Q1 is at approximately -4 vdc. Resistor R3 then supplies turn-on current to the bases of transistors Q2 and Q4 through R5 and R4. When Q4 is on, its collector is slightly below ground (provided that output loading does not bring the transistor out of saturation). Transistor Q3 is cut off, since its base is connected to the collector of Q2. This isolates the output from the collector voltage of Q3. No-load output is essentially ground, with an output impedance of 22 ohms.

When the input level changes to -3 vdc, transistor Q1 saturates, turning off Q2 and Q4. Diode D1 limits reverse bias to the base of Q2 and quickly discharges C3. The collector of Q2 falls towards -15 vdc. This turns on transistor Q3. Collector-to-base conduction of transistor Q3 then clamps the voltage at the base of Q3 to approximately -4 vdc. For any load that permits Q3 to operate at saturation, the emitter of Q3 is about -3.75 vdc. Because Q4 is cut off, the no-load voltage at output H is therefore -3.75 vdc. Output impedance is 22 ohms, the value of resistor R9.

The -3.75 vdc supply, consisting of resistor R41 and diodes D5 through D9, is similar to the -3 vdc supplies included in previously described modules. However, an additional series-connected diode is included, increasing the supply voltage three-quarters of a volt.

Moreover, R41 is in series with the supply load, rather than in parallel. Consequently, R41 limits the regulated supply output to approximately 20 milliamperes (5 milliamperes per amplifier). This maximum output can be raised to 100 milliamperes by adding a 2 watt, 120 ohm resistor in parallel with R41.

e BUS DRIVER 1685 - This module contains four identical non-inverting level amplifiers and a -3.75 vdc supply. Each amplifier performs in the same way as an amplifier of bus driver 1684 except that the output rise and fall times are extended to 1.0 microsecond. This slow-switching characteristic makes the 1685 amplifiers useful in circuits where rapid changes of level could produce unwanted ringing.

Logic levels (0 and -3 volts) are applied at inputs K, M, U, and S. The corresponding outputs at pins L, N, T, and R are logically equivalent to the inputs, but are power-amplified. Each input represents approximately one-half unit of base load. The maximum output capability per amplifier is 15 units of base load. Each amplifier is identical, and only the amplifier with input K and output L is described below.

When a ground level is applied to input K, the voltage divider R1-R2 biases the base of transistor Q1 positive. With its emitter grounded, Q1 is cut off; its collector voltage then drops toward -15 volts. This causes transistor Q2 to saturate, and the collector of Q2 rises to ground. This voltage rise is applied to the bases of Q3 and Q4 through diode D1. Transistors Q3 and Q4 are complementary emitter followers. One of these two transistors always conducts (Q3 when the output current flows to the load, and Q4 when the current flows from the load); therefore output L at the emitters of Q3 and Q4 also rises to ground.

When a -3 vdc level is applied to input K, the circuit switches state. Transistor Q1 then saturates, and the collector of Q1 rises to ground. This ground level is applied through R4 to the base of Q2, thereby turning off Q2. The base voltages of transistors Q3 and Q4 are then clamped to approximately -3.75 vdc by diode D2. As in the case of the ground input, either Q3 or Q4 always conducts, and therefore the terminal L output follows the base voltage applied to Q3 and Q4. Consequently, a -3 vdc input to the bus driver always results in a logically equivalent (although amplified) -3 vdc output level.

Capacitor C3 delays the changes in output levels. When the input drops from ground to

-3 vdc, C3 must charge through R6 before the output can change. Conversely, when the input rises from -3 vdc to ground, C3 must discharge through R7 and R8. Diode D1 prevents the low collector resistance of Q2 at saturation from shunting the discharge of C3. Diode D14 compensates for the level shift introduced by diode D1.

f BUS DRIVER 1690 - This module may be used in place of the Type 1685 as a bus driver for the LC and DB outputs. It contains four identical inverting level amplifiers, and a -3.75 vdc supply. Each amplifier output provides logic levels at low impedance, for use in heavily loaded logic lines. The output rise and fall times of level changes are extended to 1.0 microsecond. This slow-switching characteristic makes the 1690 amplifiers useful in circuits where rapid changes of level could produce unwanted ringing.

Logic levels (0 and -3 volts) are applied at inputs K, M, U and S. The corresponding outputs at terminals L, N, T and R are the inversions of the input levels. Each input represents approximately one-half unit of 5-megacycle base load. The maximum output capability per amplifier is 15 units of base load. The amplifier with input K and output L is described here.

When a -3 volt level is applied to input K, transistor Q1 turns on and its collector rises to ground. This voltage rise cuts off D1, and allows R4 and R5 to bring the bases of Q2 and Q3 toward +10 vdc. The positive-going rise ends at ground when D1 again conducts. Transistors Q2 and Q3 are complementary emitter followers. One of these two transistors always conducts (Q2 when output current flows to the load, Q3 when current flows from the load). Output L at the emitters of Q2 and Q3 also rises to ground.

When ground level is applied to input K, the circuit switches state. Q1 cuts off, and its collector voltage falls toward -15 vdc. However, D1 and D3 clamp the voltage at -3.75 vdc. The -3.75 volt level is applied to the bases of Q2 and Q3. Output L follows the base voltage applied to Q2 and Q3. Consequently, the output at L falls to approximately -3.5 volts.

Capacitor C2 delays the changes in output levels. When the input rises from -3 vdc to ground, C2 must charge through R3 and D1. Conversely, when the input drops from ground to -3 vdc, C2 must discharge through R4 and R5. Diode D1 prevents the low collector resistance of Q1 at saturation from shunting the discharge of C3. D2 compensates for the level shift introduced by D1.

g INDICATOR DRIVER 1669 - This module contains nine identical inverters, used as transistor switches in indicator lamp circuits. The following description of the inverter including transistor Q1 applies equally to the other eight inverters in the module.

The input at pin E may be either ground or -3 vdc. When the input is -3 vdc, resistor R2 allows sufficient current to flow through the base of transistor Q1 so that the transistor saturates when it is connected to the intended load. When the input is ground, the voltage divider R1-R2 biases the transistor base positive, thus preventing accidental turn-on by noise pulses. Resistor R1 also supplies cutoff current. No current flows in the collector load circuit when the input is ground, but when the input is at -3 vdc, the transistor saturates. Transistor turn-on completes the load circuit. The usual load is a GE327 indicator light, connected between the collector output (pin F) and -15 vdc. Nominal current output is approximately 30 milliamperes.

h SOLENOID DRIVER 4680 - This module contains three identical driver circuits. Each circuit operates as a switch, capable of switching a 500-milliampere current in a 35-volt (maximum) circuit. Switch control inputs are standard DEC logic levels. Each solenoid driver can control an inductive load, such as punch control relays or typewriter relays.

The three switch inputs are pins K (1 input), M (2 input) and R (3 input). Outputs are L (output 1), N (output 2), and P (output 3). Pin E is connected to the external load return voltage source. Because all three solenoid drivers are identical, the following description of circuit 1 applies equally to the other two drivers.

When the solenoid driver is in the quiescent condition, the pin K input is -3 vdc. Since the emitter of Q1 is at ground potential, Q1 is saturated, and its collector is at ground. The emitter of Q2 is at -2.5 volts (the forward voltage drop across diode D1 through D3). The base of Q2, at ground, is positive with respect to its emitter, and Q2 is cut off. Under these conditions, the load circuit is open, and the inductive load is de-energized. Diode D4 connects the collector of Q2 to the external negative supply, protecting Q2 from highly negative transient voltages at the time the load is de-energized. No current flows in the external circuit, and the output is at the load return voltage. When input K is grounded, transistor Q1 cuts off. The collector voltage of Q1 drops toward -15 volts, turning off Q2. This completes the load circuit, energizing the inductive load.

i SOLENOID DRIVER 4681 - This module contains three identical solenoid drivers similar to those contained in solenoid driver module 4680. The 4681 module is similar to the 4680 module, but differs in having clamping diodes (D4, D8, and D12) and output transistors (Q2, Q4, and Q6) with a higher inverse voltage rating than the comparable diodes in the 4680 module. As a result, the maximum voltage which the circuit can switch is increased from -35 vdc to -70 vdc.

10-7 MEMORY ELEMENTS

This paragraph describes six different memory plug-in units: 1) the sense amplifier 1540; 2) read-write switch 1972; 3) memory driver 1973; 4) resistor board 1976; 5) resistor board 1978; and 6) inhibit driver 1982. These six units are all used in each type 12 memory module of the PDP-1 memory.

The sense amplifier determines when memory cores change state (refer to paragraph 8-4e). Read-write switch 1972, memory driver 1973, and resistor board 1976 are used in series with the X and Y core windings to form the read-write current path (paragraph 8-4d). Inhibit driver 1982 and resistor board 1978 are used in series with the inhibit windings to form the inhibit current path)paragraph 8-4f).

a SENSE AMPLIFIER 1540 - This module contains a difference preamplifier, a rectifying slicer, and a gated pulse amplifier. A balanced input, generated when a memory core changes state, is applied to the input of the preamplifier. Here the input is amplified enough to reach the slicing voltage. The preamplifier also discriminates against common-mode noise signals. Differential signal gain of the preamplifier is 20, while the common mode gain is 0.5.

The pulse amplifier gate is enabled when the preamplifier output reaches a predetermined slice level. A strobe pulse is applied to the pulse input of the amplifier during the specific time interval when the memory cores are read and may change state. An output pulse from the sense amplifier indicates that during the strobe time a core changed state, and thereby produced a slice level which enabled the pulse amplifier gate.

The strobe pulse permits sampling the memory sense-winding at the particular instant of the read operation when the signal-to-noise ratio is best. This accurate timing increases the certainty that every time a memory core being read actually switches from the 1 state to the 0 state, this change of state will indeed be sensed; and conversely, that spurious noise signals will not be wrongly interpreted as a change in core state.

The two ends of the memory sense winding are connected to sense amplifier inputs H and F. The output pulses induced on the sense winding when the memory cores

change state are applied across these two inputs. A 70-nanosecond standard DEC negative pulse is applied to strobe input R. The polarity of the sense amplifier output depends upon the output terminal wiring. When terminal L is grounded, a positive output pulse appears at terminal P; conversely when terminal P is grounded, a negative output pulse appears at terminal L. Neither the preamplifier outputs (terminals S and U) nor the terminal M gating level output are used except for trouble-shooting.

Balance potentiometer R2 adjusts the preamplifier for maximum discrimination against noise signals. Slice potentiometer R12 varies the predetermined threshold level at which the pulse amplifier is enabled. This threshold level can be varied from 0 to 7.5 millivolts referred to the input. A detailed description of these two adjustments is provided in paragraph 11-3c of the maintenance chapter.

Before the memory cores are read, no input is applied to the sense amplifier from the memory sense winding. Therefore the bases of transistors Q1 and Q2 are grounded by R1, R2, and R3. The collectors of Q1 and Q2 are at -5 volts. Capacitors C3 and C4 isolate this voltage from the bases of Q3 and Q4. The quiescent voltage at the bases of transistors Q3 and Q4 is determined by the setting of slice potentiometer R12. Generally R12 is set so this voltage is slightly positive. Transistors Q3 and Q4 are cut off.

Transistor Q5, which shares a common emitter connection with Q3 and Q4 is saturated since its base is at ground. Since Q5 is saturated, the collector of Q5 is slightly positive, holding Q6 off. Voltage divider R18 and R19 keeps the emitter of Q7 sufficiently negative to prevent the enabling of the output pulse amplifier.

When a memory core changes state, a voltage is induced in the sense winding. Input voltages of opposite polarity are applied from input terminals H and F to the bases of transistors Q1 and Q2. For out-of-phase signals, capacitors C1 and C2, and resistor R28 bypass emitter resistors R5 and R7. As a result, the Q1-Q2 preamplifier stage produces a voltage gain for such difference input signals. But for in-phase input signals, the emitter resistance of the preamplifier stage is higher than the collector resistance, and the gain of the stage is less than unity. This feature provides the common-mode noise rejection feature of the sense amplifier circuit.

At the arrival of the sense-winding input signal, an amplified negative voltage swing is capacitor coupled to the base of either transistor Q3 or transistor Q4 (depending upon the direction of the sense winding through that core).

When the base of Q3 becomes more negative than the base of Q5, Q3 conducts. Should the base of Q4 become more negative than the base of Q5, Q4 conducts. In either case, the voltage at the emitter of Q5 follows the base voltage of the conducting transistor (Q3 or Q4). This turns off transistor Q5. Turn-off of Q5 causes a drop in the base voltage of Q6, saturating Q6, and thereby grounding the emitter of Q7. The emitter ground at Q7 enables the pulse amplifier input gating. Pulse amplifier operation is similar to the 1607 pulse amplifier (paragraph 10-6a).

The pulse amplifier remains enabled until the preamplifier output returns to its normal quiescent level. During the time the gate is enabled, a pulse applied to strobe input R generates an output pulse across terminals P and L. Although the pulse amplifier is enabled when the memory core changes state in either direction (during write operations as well as read operations) the strobe pulse occurs only during read operations. Consequently, the sense amplifier generates an output pulse only during read operations.

b READ/WRITE SWITCH 1972 - This module contains four identical switch circuits with outputs numbered 1 through 4. Each circuit is a switch with an AND-gate input used to control the application of drive current to a memory core winding. The following description refers to read/write switch #1 (with gate inputs E and F), but applies equally to the other three switches in the module.

When -3 vdc is present at either gate input, the circuit acts as an open switch, preventing the flow of core drive current. However, when both the terminal E and terminal F gate inputs are grounded, the switch is enabled, permitting core drive current to flow through the associated memory core winding. The core drive current can then flow between bus terminal V and output #1 (terminal W).

If one or both of the gate inputs is at -3 vdc, the D1-D2 AND gate causes grounded-emitter transistor Q1 to saturated. The comparatively small size of resistor R1 provides

fast turn-on. For fast turn-off, germanium diode D4 limits the excursion of Q1 into saturation.

When Q1 is conducting, its collector is at about -1 vdc. This voltage is direct-coupled to the base of transistor Q2, holding Q2 off. This opens the forward-bias current path between -35 vdc and transitors Q3 and Q4. These two transistors are then held off by forward-biased diodes D5 and D6 respectively. With Q3 and Q4 cut off, the switch is open, and the drive current path between terminals V and W is interrupted.

The switch is enabled by grounding both the terminal E and terminal F gate inputs. This causes the D1-D2 AND gate to cut off transistor Q1. The collector of Q1 is then driven negative by resistor R3 toward -35 vdc. The collector of Q1 drops to a voltage more negative than the highest voltage at either pin V or pin W. During the read/write cycle, one of these pins is returned to -3 volts, and the other to -13 volts.

Transistors Q2, Q3, and Q4 turn on. The circuit between pin V and pin W closes, permitting a read or write current to flow. During the read portion of a memory cycle, pin V is at -13 vdc and pin W is returned through the memory core winding to -3 vdc. Core drive current then flows primarily through transistor Q4. During the write portion of the memory cycle, the polarity is reversed; pin V is at -3 vdc and pin W is returned to -13 vdc through the core winding. The core drive current flows primarily through transistor Q3 during this portion of the cycle.

c MEMORY DRIVER 1973 - Each type 12 memory includes two identical type 1973 memory driver modules. These drivers serve as both sources and sinks for the memory core-drive currents.

The core-drive current path runs from one 1973 module (called the read driver) through the enabled 1972 read/write switches (b above) and their associated 1976 resistor board circuits (d below) and core windings, to the second 1973 module (called the write driver).

In the quiescent state, the input of either driver is a ground level and the output is -3 vdc. When a -3 vdc level is applied to the input of a driver, ie, to the read

driver during a read cyle or to the write driver during a write cycle, the output falls to -13 vdc. A 10-volt potential then exists between the enabled driver and the quiescent driver. This potential causes a core-drive current to flow through the specific X or Y memory winding selected by the read/write switches. The logic level input is applied to input terminal J of the memory driver. The -3 vdc or -13 vdc output is taken from output terminal V.

In the quiescent state, a ground level is applied to input terminal J. This input cuts off grounded-emitter transistor Q1. The resulting drop in the collector voltage of Q1 permits current through R3 to turn on transistor Q2. The Q2 emitter current in turn saturates parallel transistors Q5 and Q7.

Current through resistor R3 also saturates transistor Q3, thereby grounding both the base of transistor Q4 and the anode of diode D3. The ground at the base of transistor Q4 turns off Q4. Diode D3 supplies cut-off current to parallel transistors Q6 and Q8, turning these transistors off also.

With parallel transistors Q5 and Q7 conducting, and parallel transistors Q6 and Q8 cut off, output terminal V is coupled to the -3 vdc source at terminal R (and isolated from the -13 vdc source at terminal W). The memory driver output in the quiescent state is, therefore, -3 vdc.

In the active state, a -3 vdc level is applied to input terminal J. Transistor Q1 then saturates, grounding the base of transistor Q2 and the anodes of diodes D1 and D2. The ground at the base of transistor Q2 turns off Q2. Diodes D1 and D2 supply cut-off current to parallel transistors Q5 and Q7, turning these transistors off.

The ground at the collector of transistor Q1 also turns off Q3. Resistor R9 can then drive transistor Q4 into saturation. The emitter current of Q4 in turn saturates transistors Q6 and Q8.

With parallel transistors Q6 and Q8 conducting, and parallel transistors Q5 and Q7 cut off, output terminal V is coupled to the -13 vdc source at terminal W (and isolated from the -3 vdc source at terminal R). The memory driver output in the active state is therefore -13 vdc.

d RESISTOR BOARD 1976 - This module contains eight 50-ohm, 3-watt resistors with 1/2 % tolerance. A series capacitor and resistor are added in parallel with each of the original 50-ohm resistors; they are shown by dotted lines in the schematic. The capacitor is 4700 pf with 1% tolerance. The series resistor is 47-ohms with 1% tolerance.

Each of the eight parallel combinations shown on the schematic is connected as a termination load to a single X or Y winding of the memory core bank. The other end of the parallel combination is connected to one of the 1972 read/write switch outputs. The relatively high impedance of this load (compared to the impedance of the core winding) helps to ensure a constant core drive current regardless of the magnetization states of the cores threaded by a single winding.

e RESISTOR BOARD 1978 - This module contains eight 50-ohm, 3-watt resistors with 1/2% tolerance. For use in the PDP-1 memory, only six of these eight resistors are used (resistors MT and NS are not used). The six resistors which are used have a capacitor and resistor added in parallel with each of the original 50-ohm resistors. This capacitor and resistor are connected in series with each other; they are shown by dotted lines on the schematic. The capacitor is 4700 pf with 1% tolerance. The series resistor is 47-ohms with 1% tolerance.

Each of these parallel combinations is connected to -3 vdc (at terminal P) by 220-ohm 1% resistor. A 39 uf capacitor, C9, provides an ac shunt to ground. Each of these circuits is connected as a termination load to a single inhibit winding of the memory core bank. The other end of the parallel combination is connected to the corresponding inhibit driver.

f INHIBIT DRIVER 1982 - Each type 12 memory contains 18 identical inhibit drivers. Each of these inhibit drivers is a switch with an AND-gate input used to control the application of current to the inhibit winding of a single memory core plane. Four inhibit drivers are included in each 1982 plug-in module. The following description refers to inhibit driver #1 (with gate inputs E and F), but applies equally to the other three drivers in the module.

The inhibit driver is similar to the type 1972 read/write switch (b above) except that it is used to control only a unidirectional inhibit current, rather than read and write

currents of opposite polarity.

When -3 vdc is present at either gate input, the circuit acts as an open switch, preventing the flow of inhibit current. However, when both the terminal E and terminal F gate inputs are grounded, the switch is enabled, permitting core driver current to flow through the associated memory inhibit winding. The inhibit current then flows between the terminal V inhibit supply and output #1 (terminal W).

If one or both of the gate inputs is at -3 vdc, the D3-D4 AND gate causes grounded-emitter transistor Q2 to saturate. The comparatively small size of resistor R2 provides fast turn-on. For fast turn-off, germanium diode D7 limits the excursion of Q2 into saturation.

When Q2 is conducting, its collector is at about -1 vdc. This voltage is direct-coupled to the base of transistor Q4, holding Q4 off. This opens the forward-bias current path between the base of Q6 and -35 vdc, cutting off Q6. Diode D10 furnishes reverse bias current to Q6. With Q6 cut off, the current path between terminals V and W is interrupted, and the inhibit driver furnishes no inhibit current.

The inhibit driver is enabled by grounding both the terminal E and terminal F gate inputs. This causes the D3-D4 AND gate to cut off transistor Q2. The collector of Q2 is then driven more negative by resistor R6. Furthermore, R6 supplies turn-on current to transistor Q4, and the emitter current of Q4 saturates Q6. With Q6 saturated, the current path between terminals V and W is completed, so that the driver can furnish inhibit current to the inhibit winding.

10-8 DELAY CIRCUITS

The four plug-in units described in the present paragraph provide adjustable delays for standard DEC negative pulses. Three of these four units (the 1304, 1310, and 1311 delays) are high speed 5 mc circuits. These three units are used to delay 70 nanosecond pulses. The fourth delay unit, the 4301, is a low speed 500 kc circuit. This unit is used to delay 0.4 microsecond pulses.

The 1310 and 1311 units generate comparatively short delays by means of impedance elements used as a transmission line. The 1304 and 4301 delays generate longer delay

times by means of multivibrators.

a DELAY 1304 - This module contains an input pulse gate, a monostable multivibrator, an output level amplifier, and an output pulse amplifier. The pulse gate transistor is Q1; Q2 and Q3 are the multivibrator transistors; Q4 is the level amplifier transistor; and Q5 is the pulse amplifier transistor. Diodes D11 through D14 provide a -3 vdc supply.

Whenever input terminal Y is enabled by a ground level at terminal Z, and triggered by a DEC 70 nanosecond negative pulse, another 70 nanosecond pulse is generated at pulse output terminal E or F after a predetermined adjustable delay. If terminal E is grounded, a positive pulse is generated at terminal F. However, if terminal F is grounded, a negative pulse is generated at terminal E.

In addition to its pulse output, the 1304 delay circuit also has a level output at terminal J. The terminal J level output, which is normally at ground, falls to -3 vdc during the delay. An alternate method of triggering the delay is to ground input terminal X through the collector of an external pulse gate similar to Q1.

Using only internal components, the delay may be varied from 0.25 microseconds to 500 microseconds in three ranges. With terminal U jumpered to terminal T, potentiometer R5 varies the delay within each range. Range selection is determined by jumpering terminal H to one of the terminals L, M, or N, thereby connecting capacitor C3, C4, or C5 into the multivibrator circuit. The delay range for each of these connections is as follows:

H-L	0.25 microseconds to 2.5 microseconds
H-N	2.5 microseconds to 35 microseconds
H-M	35 mircoseconds to 500 microseconds

Circuit recovery time is 20% of the maximum delay in each range. The connection between H and L is wired internally. If external control of the delay is desired, a potentiometer may be connected between pins S and T. Higher ranges may be added to the delay by connecting an additional capacitor between pins L and K.

In the quiescent state, transistors Q1, Q3, and Q5 are off; transistors Q2 and Q4 are on. Transistor Q1 is held off because its base input is at ground, or its emitter is at -3 vdc. A 2.5 ma current flows from the -3 vdc supply through D15, R20, the primary

of T1 and R7 to -15 vdc at pin C. A 5.5 ma current flows from the -3 vdc supply through D2, R21, and R7 to -15 vdc. The resulting 8 ma current through R7 keeps the collector of Q1 at about -4 vdc.

Since only a dc current flows through the primary of T1, no voltage appears across the secondary. Current flowing from the base of Q2 through R3 and the parallel combination of the R5 potentiometer and R6 (when T is connected to U) holds Q2 on.

With the collector of Q2 at ground, voltage divider R4-R9 holds Q3 off. Only a small dc current flows through the primary of T2 (through D5 and R10 to -15 vdc), and the collector of Q3 is at the clamping voltage, determined at the junction of D4 and R12. Current flowing through R13 saturates Q4, and the level output at terminal J is ground.

Since only a dc current flows in the primary of T2, there is no voltage across the secondary. Consequently, transistor Q5 is held off, and the pulse amplifier remains in its quiescent state. There is no pulse output across pins E and F.

The 1304 delay is triggered in the following manner. A -2.5 volt 70 nanosecond pulse is applied through terminal Y to the base of transistor Q1. If the emitter of Q1 is grounded at terminal Z, enabling the input gate, the transistor saturates, grounding its collector.

Initially, terminal 2 of the T1 primary is at -4 vdc. When terminal 1 of the primary is grounded, the resulting 4 volt drop across the primary is coupled to the secondary. Terminal 4 of the secondary then goes negative. Current through the primary increases linearly from the quiescent value of 2.5 ma to 8 ma.

While the primary current increases, the voltage across the primary at first remains nearly constant because of the clamping action of D2 and R21. However, when the T1 primary begins to draw more than 8 ma, terminal 2 has risen to about -3 vdc, and diode D2 becomes back-biased, thus removing resistor R21 from the primary circuit.

The resulting increase in the resistance of the primary circuit (R7 alone) tends to cause the primary voltage to decrease more rapidly; but capacitor C2 provides an ac shunt to R7. This shunt times the primary current to generate the proper pulse width into trigger diode D1.

The secondary pulse at terminal 4 is negative with respect to the ground at terminal 3. The negative pulse from terminal 4 is coupled through diode D1 to the base of Q3. Transistor Q3 then turns on, and its collector rises to ground. The low forward resistance of diodes D5 and D6 shunts the primary of transformer T2. At the turn-on of Q3, the junction of diode D4 and resistor R12 rises to ground, turning off transistor Q4. The terminal J output level then drops to -3 vdc, indicating the beginning of the delay interval.

The ground at the collector of Q3 is applied through diode D5 to terminal H. The ground level at terminal H is in turn coupled to the base of Q2 by one or more of the capacitors C3, C4, or C5 (depending upon the connections made to terminal H). This ground immediately cuts off transistor Q2, causing its collector to drop to -3 vdc. Resistor R4 then draws base current from Q3, holding Q3 on even after the end of the pulse from terminal 4 of transformer T1.

The monostable multivibrator made up of transistors Q2 and Q3 remains in this state (Q2 off and Q3 on) for the time interval required to charge the capacitance in the Q2 base circuit. The RC time constant which determines this interval depends upon the capacitors in use, and the resistance of R3 in series with the parallel combination of R6 and potentiometer R5.

When the time delay capacitors have charged to a sufficiently negative voltage, Q2 turns on. The resulting rise in the collector voltage of Q2 is coupled through resistor R4 and capacitor C6 to the base of Q3. This cuts off Q3. The current in the primary of transformer T2 then falls to its quiescent level. The resulting negative pulse in the T2 secondary is applied to the base of transistor Q5, turning Q5 on. The pulse amplifier composed of Q5 and transformer T3 then generates a standard 70 nanosecond DEC pulse across output terminals E and F. Operation of the pulse amplifier output circuit is explained in the description of module 1607, paragraph 10-6a.

Since, when Q3 turns off, the junction of diode D4 and resistor R12 returns to -3 vdc, transistor Q4 is then turned on. The terminal J level output therefore returns to ground, indicating the end of the delay.

b DELAY 1310 - This module contains a delay line which provides up to one microsecond delay in 50-nanosecond steps, and an inverter driven by the delay line output. The inverter output can then drive an external pulse amplifier (such as the type 1607; paragraph 10-6a). The inverter terminals are brought to the external connector of the module, so they are available for logical gating.

To trigger the delay, a standard DEC 70-nanosecond negative pulse is applied to terminal X. After a predetermined delay, dependent on the external connections made between terminals J through W, the inverter output at terminal E is temporarily grounded, indicating the end of the delay interval.

The inverter adds 20 nanoseconds to the delay of the line. The line delays described below do not include this 20-nanosecond inverter delay.

Two jumpers are usually used to determine the delay; one for coarse adjustment, the other for fine. The coarse range of the delay is selected by one of the following jumper connections.

Jumpered Terminals	Delay Range
U to N	0 - 0.2 microseconds
V to P	0.2 - 0.4 microseconds
V to R	0.4 - 0.6 microseconds
W to S	0.6 to 0.8 microseconds
W to T	0.8 to 1.0 microseconds

Within a coarse delay range, there are available five graduated delays separated by increments of 0.05 microseconds. The fine delay within a given coarse range is selected by the following jumper connection.

Jumpered Terminals	Delay = Low end of Range Plus:
H to N	Nothing
H to M	0.05 microseconds
H to L	0.10 microseconds
H to K	0.15 microseconds
H to J	0.2 microseconds

For example: To produce a delay of exactly 0.95 microseconds, jumper terminals W and T, and terminals H and K (0.8 + 0.15 = 0.95).

When the circuit is in the quiescent state, resistor R1 furnishes the cut-off current which holds transistor Q1 off. Terminating resistors R2 and R3 prevent signal reflections from the ends of the delay line. By attenuating the short-delay output signals, resistors R4 and R5 compensate for the attenuation of long-delay signals traversing a greater length of line. Diode D1 isolates the input from back voltage generated when the delay line is de-energized.

c DELAY 1311 - This module contains two identical delay lines. The following description refers to the delay containing transistor Q1, but applies equally to the other delay on the module.

The 1311 delay operates in a similar manner to delay 1310 (b above) except that the delay intervals available are limited to the lowest range of the 1310 delay. A delay of 200 nanoseconds (not including the additional 20-nanosecond delay introduced by the inverter) is available in 50-nanosecond steps.

To trigger the delay, a standard DEC 70-nanosecond negative pulse is applied to terminal E. After a predetermined delay dependent upon the external connection made between terminal F and terminal H, J, K, or L, the inverter output at terminal N is temporarily grounded, thus indicating the end of the delay interval.

The delays produced by each connection are as follows.

Jumpered Terminals	Delay
F to L or (S to W)	50 nanoseconds
F to K or (S to V)	100 nanoseconds
F to J or (S to U)	150 nanoseconds
F to H or (S to T)	200 nanoseconds

d DELAY 4301 - This module contains an input pulse gate, a monostable multivibrator, an output level amplifier, and an output pulse amplifier. The pulse gate transistor is Q1; Q2 and Q3 are the multivibrator transistors; Q4 is the level amplifier transistor; and Q5 is the pulse amplifier transistor. Diodes D10 through D13 provide a -3 vdc supply.

Whenever input terminal Y is enabled by a ground level at terminal Z, and triggered by a DEC 0.4-microsecond negative pulse, another 0.4 microsecond pulse is generated at pulse output terminal E or F after a pre-determined adjustable delay. If terminal E is grounded, a positive pulse is generated at terminal F. However, if terminal F is grounded, a negative pulse is generated at terminal E.

In addition to its pulse output, the 4301 delay circuit also has a level output at terminal J. The terminal J output which is normally at ground falls to -3 vdc during the delay. An alternate method of triggering the delay is to ground input terminal X through the collector of an external pulse gate similar to Q1.

Using only internal components, the delay may be varied from 2.5 microseconds to 200 milliseconds in 5 ranges. With terminal U jumpered to terminal T, potentiometer R7 varies the delay within each range. Range selection is determined by jumpering terminal H to one of the terminals L, N, M, P, or R, thereby connecting capacitors C4, C5, C6, C7, or C8 into the multivibrator circuit. The delay range with only C4 in the circuit is 2.5 microseconds to 25 microseconds. Connecting each higher valued capacitor in turn raises the delay range by approximately a factor of 10.

Circuit recovery time is 20% of the maximum delay in each range. The connection between H and L is wired internally. If external control of the delay is desired, a potentiometer may be connected between pins S and T. Higher ranges may be added to the delay by connecting an additional capacitor between pins L and K.

Current from the base of Q2 flows through R6 and the parallel combination R7-R8. This current holds Q2 on. With the collector of Q2 close to ground, voltage divider R4-R9 holds Q3 off. The collector of Q3 is then held at about -6 vdc by voltage divider R13-R14. There is no voltage across the primary of transformer T2, and no voltage appears across the secondary.

Current through R13 and R14 saturates transistor Q4. Consequently terminal J level output is at ground. The base of transistor Q5 is grounded through the secondary of transformer T2. This ground holds Q5 off. The output pulse amplifier remains in its quiescent state. There is no pulse output across terminals E and F.

The 4301 delay is triggered in the following manner. A -2.5 volt 0.4 microsecond

pulse is applied through terminal Y to the base of transistor Q1. If the emitter of Q1 is grounded at terminal Z, enabling the input gate, the transistor saturates, grounding its collector.

Terminal 2 of the T1 primary then becomes positive with respect to terminal 1, and an increasing current flows through the primary. (Capacitor C2 bypasses the primary to prevent input noise from spuriously triggering the circuit.) The increasing current in the primary produces a negative voltage at secondary terminal 4. Diode D1 couples this negative voltage to the base of Q3, thereby turning Q3 on.

The collector of Q3 then applies a ground through diode D7 to the junction of R13 and R14. This cuts off transistor Q4. The terminal J output level then drops to -3 vdc, indicating the beginning of the delay interval.

The ground at the collector of Q3 is applied through diodes D5 and D6, resistor R10, and the terminal H to terminal K capacitance to the base of Q2. This ground immediately cuts off transistor Q2, causing its collector voltage to drop. Resistor R4 then draws base current from Q3, holding Q3 on even after the end of the pulse from terminal 4 of transformer T1.

The monostable multivibrator made up of transistors Q2 and Q3 remains in this state (Q2 off and Q3 on) for the time interval required to charge the capacitance in the Q2 base circuit. The RC time constant which determines this interval depends upon the capacitors in use, and the resistance of R6 in series with the parallel combination of R8 and potentiometer R7.

When the time delay capacitors have charged to a sufficiently negative voltage, Q2 turns on. The resulting rise in the collector voltage of Q2 is coupled through resistor R4 and capacitor C3 to the base of transistor Q3. This cuts off Q3. The current in the primary of transformer T2 then falls to its quiescent level. The resulting negative pulse in the T2 secondary is applied to the base of transistor Q5, turning Q5 on.

The pulse amplifier composed of Q5 and transformer T3 then generates a standard 0.4 microsecond DEC pulse across output terminals E and F. Operation of the pulse amplifier output circuit is explained in the description of module 4603 (paragraph 10-6e).

Since, when Q3 turns off, the voltage at the base of Q4 is determined by voltage divider R13-R14, transistor Q4 then turns on. The terminal J output level therefore returns to ground, indicating the end of the delay.

10-9 PULSE CIRCUITS

Three modules, pulse generator 1410, clock 4401, and pulse generator 4410 are described in the present paragraph.

Pulse generators 1410 and 4410 are similar to pulse amplifiers 1607 and 4603 (paragraphs 10-6a and e respectively) in that the pulse generators produce a standard DEC pulse whenever their inputs are triggered. However, the pulse generators differ from the pulse amplifiers in that the generators are designed to be triggered from slow, irregular voltage changes, such as those produced by mechanical switching. The type 1410 pulse generator generates 70-nanosecond pulses for use in high speed 5 mc circuits, while the pulse generator type 4410 generates 0.4 microsecond pulses for use in low speed 500 kc circuits.

The type 4401 clock generates a steady train of 0.4 microsecond pulses at an adjustable, predetermined repetition rate.

a PULSE GENERATOR 1410 - This module contains a Schmitt trigger circuit, an output pulse amplifier, and an R-C filter. The Schmitt trigger includes transistors Q1 and Q2; the pulse amplifier transistor is Q3; resistors R1 and R2, and capacitor C1 make up the filter.

Module 1410 generates a standard DEC 70-nanosecond pulse whenever its input voltage drops from a value more positive than -1 volt to a value more negative than -2.5 volts. If no internal filtering is required, the input is applied to terminal S. However, if the internal filter of the 1410 is needed (for example when the circuit is used in conjuction with a mechanical switch) terminals S and U are jumpered and the input switch is connected between terminals K and Z. The negative trigger input to terminal Z is then derived by mechanically shorting terminal K to terminal Z.

Terminals E and F are the output terminals of the 1410. The output pulse may be either positive or negative. Terminal E generates a negative pulse when terminal F is connected to ground. Terminal F generates a positive pulse when terminal E is grounded.

In the quiescent state, the input (at S or Z) is ground. Diode D4 and resistor R4 couple this input to the base of transistor Q1. (Diode D4 protects the transistor base from excessive positive voltages.) The emitters of Schmitt trigger transistors Q1 and Q2 at about -2 volts as determined by the voltage divider R8, R10, R6, and R5 in conjunction with voltage divider R12, R13, R14, and R15.

The collector of Q2 is at -4 volts determined by the voltage divider. The current through the primary of transformer T1 is held steady at about 10 ma, and no voltage appears across the secondary. The base of Q3 is held at ground by the seconday of T1, and Q3 is cut off. The collector of Q3 is at approximately -7.5 volts, and there is no output across terminals E and F.

To trigger the circuit, a negative voltage is applied to the input. Schmitt trigger transistor Q1 begins to conduct as soon as its base becomes more negative than its emitter. This occurs when the input falls below about -2.5 volts. When transistor Q1 starts to conduct, its collector voltage rises towards ground. Resistor R6 and and capacitor C2 couple the rising collector voltage of Q1 to the base of transistor Q2, thereby cutting off Q2. Turn-off of Q2 makes the common emitter connection of Q1 and Q2 more positive, speeding the turn-on of Q1. This positive feedback of the Schmitt trigger circuit assures a fast change of state, independent of the fall time of the input signal.

With the turn-off of Q2, the current across the primary of transformer T1 collapses, inducing a negative pulse at terminal 4 of the T1 secondary. This negative pulse is amplified by pulse amplifier Q3 and is then applied across output terminals E and F. The pulse amplifier shapes the output pulse to 2.5 volts amplitude and 70 nanoseconds duration. The polarity of the pulse depends on whether pin E is grounded (for a positive pulse) or pin F is grounded (for a negative pulse).

When the base input of Q1 again rises towards ground, the emitter of Q1 follows this voltage until the base is at approximately -1 volt. At this point, diode D2 again starts to conduct, clamping the emitter voltage. Further rise of the input voltage cuts off Q1. As the collector of Q1 goes negative, this voltage drop is coupled through R6 and C2 to the base of transistor Q2, turning Q2 back on, and thereby returning the trigger circuit to its quiescent state with the emitters of Q1

and Q2 at -2 volts. This re-establishes current through the primary of transformer T1, and induces a positive pulse at terminal 4 of the T1 secondary. This positive pulse, however, only drives Q3 further into cutoff and the output pulse amplifier is unaffected. Diodes D1 and resistor R11 prevent ringing in T1.

b CLOCK 4401 - This module consists of an astable multivibrator, a pulse amplifier-shaper, and an output pulse amplifier. The multivibrator includes transistors Q1 and Q2; the pulse shaper transistor is Q3 and the output pulse amplifier transistor is Q4. The type 4401 clock generates standard DEC 0.4 microsecond pulses across output terminals E and F at any frequency from 5 cycles to 500 kc per second. The interval from 5 cycles to 500 kc is divided into 5 overlapping ranges; within each range the output frequency is continuously adjustable.

Potentiometer R4 adjusts the frequency within each range. The range is determined by the amount of capacitance between pins T and V. An external jumper connects one of five capacitors contained in the module into the circuit for this purpose. The frequency range for each of these connections is as follows.

Connection	Frequency Range
T-M	5 cycles to 50 cycles
T-R	50 cycles to 500 cycles
T-P	500 cycles to 5000 cycles
T-N	5 kc to 50 kc
T-U	50 kc to 500 kc

Diodes D1 through D6 determine the operating voltages of the multivibrator. These are silicon diodes with a voltage drop of approximately 0.75 volts each. Because they maintain this voltage drop across a wide range of current flow, multivibrator operation is stable and comparatively independent of the -15 vdc supply. Diodes D1 through D6 hold the base of Q1 at -2.25 vdc and the collector of Q2 at -3.75 vdc.

Multivibrator feedback is obtained by connecting terminal T to one of terminals M, R, P, N, or U, and jumpering terminals X and Y. An alternative arrangement, which extends the multivibrator frequency range, is to connect an external capacitor between terminals V and T. External fine control may be provided by connecting an external

potentiometer between terminals Y and ground (D, L, or Z).

Because the fine control potentiometer R4 varies the operating point of Q1 over a wide range, a dual collector load is provided for Q1. For low operating current resistor R3 is the principal load, and the Q1 gain is sufficient to maintain oscillation. For high operating current, the principal load is through resistor R2, and Q1 is not driven into saturation.

Multivibrator transistors Q1 and Q2 alternate on and off at a rate that is a function of the R-C time constant of the range-determining capacitor (one of C3 through C7) and the series combination of resistor R1 and potentiometer R4. An output pulse is generated during each cycle of the multivibrator when Q2 turns off.

Assume that at a given moment transistor Q1 is turning off and transistor Q2 is turning on. The emitter voltage of Q2 follows the negative-going voltage at the collector of Q1. Capacitor C3 (if terminal T is jumpered to terminal M) couples the negative transient to the emitter of Q1. The feedback from the collector of Q1 to its emitter rapidly triggers the multivibrator to the astable state with Q1 off and Q2 on. The secondary of T1 generates a positive pulse at the base of Q3. However, this pulse only drives Q3 further into cut-off, and the output pulse amplifier is not affected.

The multivibrator remains stable in this state while C3 charges through R1 and R4. The Q1 emitter voltage rises exponentially towards ground. When the emitter voltage becomes more positive than the -2.25 volts at the base of Q1, transistor Q1 begins to turn on. The Q1 collector-emitter feedback triggers the multivibrator to the other state (Q1 on and Q2 off). The multivibrator remains in that state while C3 charges through R5. At the turn-off of Q2, the collapse of current in the primary of transformer T1 causes a negative pulse to be generated at terminal 3 of the T1 secondary. This negative pulse saturates transistor Q3, grounding terminal 2 of the T2 primary. The output pulse amplifier, composed of transistor Q4 and transformers T2 and T3 generates a standard DEC 0.4 microsecond pulse across terminals E and F. Operation of the pulse amplifier output circuit is explained in the description of module 4603, 10-6e.

c PULSE GENERATOR 4410 - This module is the 500-kc version of pulse generator 1410 (a above). Except for differing component values and designations, both circuits

are identical, and serve a similar purpose. This 4410 module generates a pulse output from an irregular level change at the input (such as the voltage change generated by a mechanical switch closure).

Circuit parameters have been varied where necessary to lengthen the output pulse. When a voltage at the input (terminal S) drops from a value more positive than -1 volt to a value more negative than -2.5 volts, a standard DEC 0.4-microsecond pulse is generated across the output.

Output pulse polarity can be positive or negative depending upon which of the two output terminals (E or F) is grounded.

10-10 SWITCH FILTER 1703

The switch filter 1703 module contains nine identical circuits. Each of these nine circuits converts mechanical switch positions to standard DEC logic levels, in the process filtering out voltage irregularities caused by contact bounce. A -15 vdc input (closed switch) produces a -3 vdc output level. An open switch input produces a ground level output.

The module also includes a -3 vdc supply, composed of diodes D19 through D24, and resistor R19. Input current drawn from a closed switch is 10 milliamperes. Maximum output current is three units of base load. Because all nine switch filters are identical, the following description of the circuit with input pin E (IN 1) and output pin F (OUT 1) applies equally to the other eight circuits in the module.

If the input circuit is open (open switch), capacitor C1 discharges through R2. The output then rises exponentially towards +10 vdc. However, the output is not permitted to rise so far; it is clamped at ground by diode D2. The cathode of D2 is connected to the cathodes of diodes D19 and D20, which are one diode-drop below ground potential. This compensates for the drop across diode D2, and ensures that the anode of D2 (the circuit output) is actually clamped at a value close to ground.

When the input circuit is closed (closed switch), a -15 vdc level is applied to input pin E. Capacitor C1 then charges through R1. The output then falls exponentially until it is clamped at -3 vdc by diode D1. The actual circuit output voltage towards which C1 charges is determined by the voltage divider composed of R1 in series with the parallel

combination of R2 and the external load. Maximum output current is fixed by the limitation that this output voltage should not be allowed to rise above -3 vdc.

10-11 POWER SUPPLIES AND CONTROLS

The PDP-1 power supplies convert standard 110 vac to dc power at the appropriate voltages for the computer circuits. The power control units control the application of power to the equipment.

Different models of the computer use different power supplies and controls. Some PDP-1 computers use power supplies 729 and 742. Other models of the computer replace these two power supplies with a single type, the 728 supply. Power control 810, which is used in some computers, is replaced in other models by the 813 power control. All of the power supplies and controls which are used in any model of the computer are described in the present paragraph.

a POWER SUPPLY 728 - In some models of the computer, this unit replaces power supplies 729 and 742. The outputs of the 728 supply are +10 vdc (0 to 7.5 amperes), or -15 vdc (1 to 8.5 amperes). When both outputs are used concurrently, the current limitations are more stringent. All three of the following limitations then apply:

1) +10 vdc limited to between 0 and 7.0 amps
2) -15 vdc limited to between 1 and 8.0 amps
3) Both outputs limited by the relationship:

$$5I_{(+10)} + 6I_{(-15)} \leq 53$$

The +10 volt output is regulated between +9.5 vdc and +11 vdc; the -15 volt output is regulated between -14.5 vdc and -16 vdc. Assuming line voltage variation from 105 to 125 vac, this regualtion holds from minimum to maximum load. Output ripple is less than 350 millivolts.

The line voltage is stepped down to 10-0-10 vac and 15-0-15 vac by resonant transformer T1. Diodes D2 and D3 are connected to the 10 volt secondary taps as a positive full-wave rectifier. Capacitors C2 and C4 filter out the AC component of the output. Resistor R1, in parallel with the 10 volt load, keeps the output within regulation tolerances even though the external load is decreased to the no-load condition.

Diodes D1 and D4 are connected to the -15 volt secondary taps of T1, as a negative full-wave rectifier. Capacitors C1, C3, C5 and C6 filter out the AC component.

Special properties of transformer T1 make possible the simple design of the power supply. Transformer T1 is a saturated-core resonant transformer which provides inherent overload protection. Even with shorted outputs, only a limited output current can be drawn. The self-limiting secondary current (which remains comparatively constant over limited variations in the primary input) eliminates the need for series impedance elements at the filter inputs. The dc output impedance of the supply is thus kept low, rendering regulating devices unnecessary.

b POWER SUPPLY 729 - This power supply furnishes the -15 vdc and +10 vdc power required to operate the logic modules in PDP-1. Input is a nominal 115 vac; outputs are +10 vdc at 0 to 0.5 ampere and -15 vdc at 1 to 6 amperes.

Resonant transformer T1 converts the 110 vac input to 15-0-15 vac at the center-tapped secondary. Diodes D1 and D2 comprise a full-wave rectifier for voltages positive with respect to the center-tap common. The ripple voltage superimposed on the dc output of D1 and D2 is filtered by C1. The +10 vdc output is maintained at a constant voltage by conduction through resistor R1 and Zener diode D5.

Voltages negative with respect to the center-tap common are rectified by diodes D3 and D4. Although filtering of the dc output is limited to a single stage of parallel capacitance (C2 through C6) this capacitance reduces ripple content sufficiently for the worst condition of maximum rated load. Absence of series-resistive components keeps the -15 vdc output almost independent of load. Transformer T1 is a saturated-core resonant transformer which delivers only a limited amount of secondary current even under conditions of shorted output. This built-in overload-protection keeps surge currents within the maximum ratings of diodes D3 and D4.

c POWER SUPPLY 742 - From a nominal 115 vac input, the type 742 power supply generates a -15 vdc output at 1 to 8 amperes. Two type 742 supplies are connected in series to produce the 30 volt input required by solenoid drivers 4680 and 4681 (paragraphs 10-6f and 10-6g). The 742 supply is almost identical to the -15 vdc portion of the 729 supply (b above). The 742 supply differs from the 729 supply

only in having an additional parallel filter capacitor, and a somewhat higher maximum rated current load.

d VARIABLE POWER SUPPLY 734 - This power supply furnishes dc power for marginal checking of PDP-1 modules. For a nominal 115 vac input, output voltage is continuously variable from 0 to ±20 vdc (no load). Maximum voltage output drops 3 volts at full-rated load of 2.5 amperes.

Line power at 115 vac is stepped down by resonant transformer T1. (Only the terminal 3 and 4 half of the secondary is used.) The voltage at the secondary is applied to terminals 1 and 5 of Variac M5. By adjusting the position of the terminal 3 tap, any voltage within the range of 0 to 20 vac is available between terminals 3 and 4. Output voltage is increased by rotating the Variac control clockwise. The Variac output is applied to a bridge rectifier (diodes D1 through D4).

The rectifier diodes are oriented so that the dc output of the bridge at the junction of D2 and D4 is positive with respect to the junction of D1 and D3. Parallel capacitor C1 filters the output. Voltage regulation is improved for small load currents by parallel resistor R1. A slow-blow 5-ampere fuse at the positive output protects the supply against overload. The dc output voltage is indicated on a 0-30 vdc meter across the output.

e POWER SUPPLY 735 - This supply provides power to the PDP-1 memory logic. The input voltage requirement is a nominal 115 vac. Outputs are -3 vdc (pinB), -13 to -16.5 vdc (pins C and D), and -35 vdc (pin E). A +10 vdc level is generated for use by the internal shunt regulator circuits, and for use by power control 1701. Pins C and B are the inhibit voltage supply output; pins D and B are the read-write voltage supply outputs.

Since the read-write and inhibit voltages must be well-regulated, compound connection shunt regulator circuits are used across these outputs. The bases of shunt regulator transistors Q1 and Q3 are brought to terminals F and N (for connection to Power Control 1701) rather than to their respective output voltage points. Besides regulating the output voltages, the connection to the 1701 control serves two other functions. The 1701 circuitry varies the output voltage in accordance with the temperature of the core bank. Furthermore, the 1701 control permits adjusting the output voltage to the individual

requirements of a specific core bank.

The inhibit and read-write supplies are very much alike. They differ, however, in that the inhibit supply output current varies over a wider range. Whereas the read-write supply output current varies only from 0 to 0.4 amperes, the inhibit supply output must vary from 0 to 1.8 amperes. Consequently, both the inhibit supply series dropping resistance (R1-R2) and also the emitter resistor R4 of the principle shunting transistor Q4, are smaller than the corresponding resistors in the read-write supply.

Resonant transformer T1 steps down the 115 vac input to 10-0-10 vac, and to 35-0-35 vac. Diodes D2 and D3 are connected as a positive full-wave rectifier to the 10 volt secondary taps. Capacitor C5 filters the dc output of the rectifier, which is then applied to the emitter circuits of Q1 and Q3, and to pin A of power control 1701. Diodes D1 and D4 are connected to the 35 vac secondary terminals as a negative full-wave rectifier. Capacitor C3 filters the rectified -35 vdc terminal E output.

This -35 vdc also provides the negative input to both the inhibit and read-write supplies. The positive input to these two supplies is -3 vdc from pin B. This voltage is generated by the forward voltage drop across four series-connected diodes D5, D6, D7 and D8. The anode of D5 is connected to the grounded center tap of T1. Because the inhibit and read-write supplies are similar, the following description of the read-write supply also adequately describes the inhibit supply.

The base of the shunt-regulator transistor Q1 is biased from terminal F of the 1701 control. The operation of the 1701 control is fully described in f below. However, to understand the regulating action of transistors Q1 and Q2 it can be assumed for the time being that the base of Q1 is biased through a connection to the terminal D output of the supply. Although this connection is in fact made through the 1701 control circuitry, the bias feedback functions in much the same way as if the feedback connection from output terminal D were instead made through a battery, reference diode, or resistance.

If the output voltage rises, because of either an increase in the supply load, or because of a rise in the supply input voltage, then the base voltage of Q1 also rises. Conduction through Q1 decreases, and the emitter voltage of Q1 rises with the base voltage. Consequently, conduction through transistor Q2 also decreases. The decrease in

conduction through the two transistors (chiefly Q2) tends to restore the original voltage at D. A fall in the output voltage is similarly counteracted by the shunt regulator. Capacitor C6 provides AC filtering for the output voltage.

During normal circuit operation, Zener diode D8 does not conduct. This diode is used solely as a protective device. In the event that conduction through Q1 or Q2 is seriously impaired by a malfunction, the circuit output voltage would, in the absence of diode D8, tend to fall towards -35 vdc. To avoid such a large negative output voltage, and the resulting possibility of damaging other computer memory elements, Zener diode D8 is used to clamp the output to a maximum negative value of -15 vdc, the breakdown voltage of D8.

The read-write voltage output meter (included only in certain early model computers) normally indicates an output voltage in the range from -10 vdc to -13.5 vdc. A reading which falls as low as -15 vdc indicates circuit malfunction.

f POWER SUPPLY CONTROL 1701 - This unit controls power supply 735. The 1701 module contains two identical circuits. One of these two circuits controls the 735 inhibit supply, and the other controls the 735 read-write supply. Since both the inhibit and read-write supplies function in the same way, the following description of the read-write control circuit (in the lower half of the schematic) applies equally to the inhibit control circuit (top half of schematic).

Terminal E of the 1701 control is connected to the -3 vdc common of the 735 power supply. Terminal H of the 1701 control is connected to the nominal -13 vdc output of the read-write supply. Terminal A receives +10 vdc (from the 735 supply). The control output is at terminal F.

As an adjunct to the 735 read-write supply, the control circuit performs three functions. First, the terminal F output biases the base of shunt regulator transistor Q1 in the 735 supply. This bias determines the read-write output voltage. The bias, and the resulting read-write output, can be adjusted by a potentiometer.

Second, a thermistor (placed in the environment of the memory core bank) makes the bias temperature-dependent. Because the thermal coefficient of this thermistor is negative (-4.4% per C^{o}), the read-write output voltage is a negative function of temperature

(-0.5% per C^{o}). As the temperature of the core bank increases, the read-write voltage and current decrease. This temperature compensation corrects for the fact that the higher the core temperature, the smaller the core winding current that is needed to switch a memory core.

The third function of the control circuit is to compensate for changes in the read-write voltage which are caused by variations in the load and in the supply input voltage.

Transistors Q4 and Q5 make up a difference amplifier. The change in voltage at the collector of Q5 is proportional to the voltage difference between the bases of Q4 and Q5. Bias control levels from the potentiometer enter the difference amplifier at the base of Q4. Bias control levels determined by changes in the resistance of the thermistor are applied to the base of Q5. The feedback, or regulation signal also enters the difference amplifier at the base of Q5.

The series combination of control potentiometer R13 and resistor R12 is in parallel with a 6.2 volt Zener reference diode. This double-anode Zener diode provides the basic voltage reference used by the circuit. The reference diode has extremely good temperature stability. Voltage across it remains nearly constant for normal variations in ambient temperature.

Counter-clockwise rotation of the potentiometer varies the base voltage of transistor Q4 from -9.2 volts to approximately -6.5 volts. Rotating the potentiometer counter-clockwise decreases the read-write output voltage; clockwise rotation increases the output voltage. The potentiometer controls the output voltage in the following manner. Assume that the potentiometer is rotated counter-clockwise. The base voltage of transistor Q4 then rises, increasing conduction through Q4. Conduction through Q5 then decreases, raising the base voltage of NPN transistor Q6. This increases conduction through Q6, and thus lowers the bias output at F. The more negative output at F causes increased conduction in the 735 read-write shunt regulator transistors, thereby decreasing the read-write output voltage. Clockwise rotation of the potentiometer increases the supply output in exactly the opposite manner.

To understand the way in which the control circuit compensates for temperature changes, assume that while the base voltage of Q4 remains constant, the temperature

increases. The increasing temperature produces a decrease in the value of the thermistor (connected between terminals J and K). As the resistance of the thermistor decreases, the base voltage of Q5 also decreases, increasing conduction through Q5. The bias output at F decreases thus decreasing the supply output voltage. Decreases in ambient temperature produce an increased output voltage in exactly the opposite manner.

Connected across the supply output is a voltage divider, comprising the thermistor and resistors R16, R17, R18, and R19. The supply output voltage is fed back to the base of Q5 through this voltage divider, thereby regulating the output over variations in load and input voltage. In a sense, therefore, transistor Q5 provides the first stage of a compound shunt regulator. The final stage of this compound regulator is transistor Q2 (in the 735 supply).

If, for example, the supply voltage deviates negatively, then the base of Q5 also goes negative, increasing conduction through Q5. This causes the base voltage of Q6 to rise, and increases conduction through Q6. The output terminal F bias voltage then drops, increasing conduction through the shunt regulator transistors in the 735 supply. This causes the output voltage of the 735 supply to rise to its original correct value. Positive deviations in output voltage are corrected in exactly the opposite manner.

g POWER CONTROL 812 - The main function of this unit is to switch 110 vac line power between input terminals 1 and 2 and output terminals 3 and 4. The switching function is performed by relays controlled by a standard DEC logic level applied to input terminal 8 or 10. Control of the paper tape punch motor is a typical application of the 812 unit.

The 110 vac input to terminals 1 and 2 usually originates at either a type 813 power control unit (h below) or a type 810 power control (i below). The 110 vac output is switched to output terminals 3 and 4 immediately after the application of a -3 vdc turn-on level to input terminal 8 or 10. Approximately one second after the 115 vac output is enabled, a -3 vdc ready signal is applied to terminal 9. The one second delay permits the punch motor to come up to speed before the terminal 9 punch-

ready signal is asserted.

To turn off power, the -3 volt turn-on signal (which may consist of either a level or a train of pulses) is raised to ground. After the turn-on signal is thus ended, power control output terminals 3 and 4 remain at 110 vac for 12 to 13 seconds, and for the same interval, output terminal 9 continues to assert the -3 vdc ready signal. At the completion of this 12 to 13 second turn-off delay, the 110 vac is switched off, and output terminal 9 rises to ground, ending the ready signal.

Instantaneous on and off relay K1 operates on -3 vdc. Relay K2 is instantaneous on, 12- to 13-second delayed off. Relay K3 is one-second delayed on, but instantaneous off. The blocks designated FL1 and FL2 represent low-pass line filters.

In the quiescent state, input terminals 8 and 10 are at ground, relay K1 is de-energized, and the power control is off. Because contacts 1 and 3 of relay K1 are open, relay K2 is also de-energized. Contacts 6-4 and 5-3 of K2 are then open, interrupting the circuit between input terminals 1 and 2 and output terminals 3 and 4. Consequently relay K3 is also de-energized, and contacts 3-5 of K3 are closed, grounding the pin 9 ready output, and thus preventing the ready signal from being asserted.

The 812 power control is turned on by applying a -3 vdc turn-on signal to either input terminal 8 or input terminal 10. When the power control is used as a punch motor control, the turn-on signal is a train of pulses applied to input terminal 8. The input filter comprising C1, R1, and diode D2 prevents contact chatter of relay K1.

The -3 vdc signal applied to terminal 7 of relay K1 energizes that relay. Closure of K1 contacts 1-3 energizes relay K2. The circuit between the power control input terminals 1 and 2 and the power control output terminals 3 and 4 is completed by the closure of K2 contacts 6-4 and 5-3. Application of 110 vac to output terminals 3 and 4 of the power control in turn energizes delayed-on relay K3. Approximately one second later, K3 contacts 3-5 open, removing the ground from output terminal 9, and permitting a -3 vdc ready signal (from voltage divider R3-R4) to be applied to that terminal.

When the -3 vdc turn-on signal at input terminal 8 or 10 ceases, relay K1 is de-energized

immediately. The opening of K1 contacts 1-3 de-energize delayed-off relay K2. After a 12- to 13-second delay, the 6-4 and 5-3 contacts of K2 open. The opening of these contacts immediately removes 110 vac from output terminals 3 and 4 of the power control, and thereby de-energizes K3. Contacts 1-5 of K3 close immediately, grounding terminal 9, and terminating the -3 vdc ready signal.

h POWER CONTROL 813 - For some PDP-1 computers, this unit is used as the main power control. Other PDP-1 computers use the 810 power control (i below) in place of the 813. The 813 power control differs from the 810 in that it may be used with 220 vac input as well as with the 110 vac input used by the 810 control. It is like the 810 in that both controls use relay switching, provide overload protection, and furnish a five-second ground (at pin 7) to enable the power-clear pulse amplifier at computer turn-on and turn-off.

If the type 813 power control is used with 110 vac, the two H input pins are jumpered, and the 110 vac is applied across H and N. For a 220 vac input, terminal N is the ground connection, while the H terminals are the two hot connections to the 220-volt line. Pins G and H are the power output terminals for punch motor control power. The power output for memory power supply 735 is at pins A and B. All other ac power for the computer is furnished across pins C-D and D-F.

Relay D1 is instantaneous on, 5-second delayed off. Relay D2 is 5-second delayed on, but instantaneous off. Both K3 and K4 are instantaneous on-off relays.

Normally ac line voltage is present at input pins H and N. The power control is turned on by closing the POWER switch on the console. Relays D1 and D2 are then energized. Contacts 2-3 of D1 close immediately, energizing K3. All three sets of K3 contacts close at once, applying 110 vac across outputs C-D and D-F. All the power supplies fed by these outputs are then energized.

Although even before the POWER switch is closed, 110 vac is present at output terminals G and H, no load is then connected across these outputs. The application of power to output terminals C-D and D-F permits the computer to switch a load across terminals G and H. (The punch motor can only be turned on when the computer is on)

Five seconds after the POWER switch is closed, contacts 2-4 of relay D2 close. Unless the memory power switch is open, relay K4 then energizes, closing both sets of K4 contacts. The closing of the K4 contacts supplies 110 vac to memory power outputs A and B.

The five-second delay introduced by relay D2 ensures that all turn-on transients in the rest of the computer have ended before memory power is turned on. During these five seconds, while the 7-6 contacts of D1 are closed and before contacts 6-7 of D2 have opened, terminal 7 is grounded. The ground output at terminal 7 enables the computer power-clear pulse amplifier. At the end of five seconds, contacts 6-7 of D2 open, ending the ground output at pin 7.

When the POWER switch on the console is turned off, relays D1 and D2 are de-energized. Contacts 2-4 of D2 open immediately de-energizing K4. Power output to pins A and B is interrupted, thus turning off the memory power supply before any other turn-off transients can occur. Five seconds later, contacts 2-3 of D1 open, de-energizing K3. All three K3 contacts open immediately, interrupting power to outputs C-D and D-F. (The turn-off of power at C-D and D-F also prevents the punch motor control from drawing power from outputs G and H).

During the five-second turn-off delay, contacts 7-6 of D1 and 6-7 of D2 are both closed, grounding terminal 7. This enables the power-clear pulses. At the end of the five seconds, contacts 7-6 of D1 open, ending the ground output at pin 7.

The memory power switch permits the operator to turn off memory power while the rest of computer power is still on. Circuit breakers CB1, CB2, and CB3 provide overload protection (20 amperes) for all power except the punch motor control line. This line is protected by four-ampere circuit breakers CB3 and CB4. All circuit breakers are normally closed.

i POWER CONTROL 810 - For some PDP-1 computers, this unit is used as the main power control. Other PDP-1 computers use the 813 power control (h above) in place of the 810. The 810 power control differs from the 813 in that it may only be operated with 110 vac, and not with 220 vac. Like the 813 control, the 810 control uses relay switching, provides overload protection, and furnishes a five-second ground

to enable the power-clear pulse amplifier at computer turn-on and turn-off.

The principal components of the 810 power control are three relays (K1, K2, and K3). These relays turn computer ac power on and off when actuated by the POWER switch on the console. Relay K1 is an instantaneous on-off type, while relay K2 is an instantaneous on and five-second delayed off type. The on-off characteristics of K3 are opposite to those of K2, that is, K3 is a five-second delayed on, instantaneous off relay.

The ac line enters the circuit at terminals E and F. Terminals G and H furnish power to the type 812 punch motor control. Terminals C and D furnish power to memory power supply 735. All other ac power for the computer is taken from terminals A and B. A five-second momentary ground is established at pin 7, beginning with the turn-on or turn-off of the POWER switch. This ground output enables the computer power-clear pulse amplifier.

The computer is turned on by closing the POWER switch. When the POWER switch is closed, pins 1 and 2 are shorted, thus energizing relays K2 and K3. The K2a contacts close immediately, energizing K1. Contacts K1a and K1b then close, supplying 110 vac to terminals A and B. All computer power supplies and equipment connected to this output are then energized.

Although even before the POWER switch is closed, 110 vac is present at output terminals G and H, no load is then connected across these outputs. The application of power to output terminals C-D and D-F permits the computer to switch a load across terminals G and H. (The punch motor can only be turned on when the computer is on.)

Five seconds after the POWER switch is closed, the K3a contacts close. This supplies 110 vac to outputs C and D for use by the memory power supply. The five-second delay ensures that all turn-on transients in the rest of the computer have ended before memory power is turned on. During these five seconds, while the normally-open contacts K2b are closed, and before normally-closed contacts K3b open, pin 7 is grounded. The ground at pin 7 enables the power-clear pulses that prepare the computer for operation.

The computer is turned off by opening the POWER switch. When the POWER switch on the console is opened, relays K2 and K3 are de-energized. Contacts K3a immediately open, interrupting power to outputs C and D. Thus, memory power is turned off before any turn-off transients occur in the computer logic. Five seconds later, contacts K2a open, de-energizing K1. Contacts K1a and K1b immediately open, interrupting power to outputs A and B. (This also prevents the punch motor control from drawing power from outputs G and H.) During the five-second turn-off delay, contacts K2b and K3b are both closed, grounding pin 7.

Switch S1 permits the operator to turn off memory power while the rest of the computer power is still on. Circuit breakers CB1 and CB2 provide overload protection (20 amperes) for all outputs except the punch motor control line. This line is protected by four-ampere circuit breakers CB3 and CB4. All circuit breakers are normally closed.

CHAPTER 11

MAINTENANCE

11-1 SPECIAL TOOLS AND TEST EQUIPMENT

The following special tools and test equipment are recommended for the efficient maintenance of the PDP-1 computer.

Multimeter	Simpson Model 260A, or Triplett Model 630NA, or equivalent
Subminiature alligator clips	Mueller type 30 or equivalent
Oscilloscope	Tektronix 540 series with type CA plug-in vertical amplifier, or equivalent
Long-lead probes	Tektronix P-6002 or equivalent
Current probe	Tektronix P-6016 or equivalent
Paper Tape Gauge	Friden Type T-18118, or equivalent
Plug-in puller	DEC type 1960*
Plug-in extender	DEC type 1954*
Pigtail plug-in extender	Modified DEC type 1954 (a below)
60-cycle button pusher	DEC type 4900 (b below)
Soldering iron	6 vac iron with isolation transformer

a PIGTAIL PLUG-IN EXTENDER - This maintenance aid can be readily fabricated from a standard DEC type 1954 unit extender. Disconnect the small wire leads to terminals A, B, and C of the unit extender, and solder eight-foot leads to the three terminals. Solder alligator clips to the free ends of the three eight-foot leads. Terminals A and B can then be connected to the 10-volt marginal check power supply. Terminal C can be connected to the 15-volt supply. This permits convenient marginal testing of an individual circuit card. Either the A or the B portion of the card can be separately checked, thereby permitting submodular testing.

* Digital Equipment Corporation furnishes one of each of these units without charge with each PDP-1 computer.

b 60-CYCLE BUTTON PUSHER - This device simulates the effect of operating any console switch at a 60-cycle rate. This uniform repetition is useful for many forms of console troubleshooting, especially for the troubleshooting of logic functions that are initiated by console switches. The button pusher has two leads. One lead clips to any convenient ground (such as the console frame); the other lead clips to the switch terminal corresponding to the desired console function. Button pushers can be ordered directly from DEC.

11-2 EQUIPMENT LAYOUT AND WIRING

The detailed physical layout for all of the logic comprised by the standard PDP-1 system is shown in figure 11-1. Locations of the power control panels and power supplies are shown in figure 2-8. Two module layout drawings, figures 11-2 and 11-3, show the plug-in module types normally installed in every mounting panel location. Figure 11-2 is the module layout for the standard PDP-1; figure 11-3 shows the five central processor options. Four wiring configuration diagrams (figures 11-4 through 11-7) show the cable connections for the computer memory. Figure 11-4 shows the standard machine with one memory; figures 11-5 through 11-7 show machines that include the type 15 memory extension control and the type 19 high-speed channel control.

The detailed logic layout, figure 11-1, shows each mounting panel divided into sections according to logical function. Unlabelled areas bounded by solid lines represent sections of the mounting panel in which no plug-ins are installed. Areas that are labelled by logic function and separated by dotted lines represent circuits appearing on the same logic drawing. Figure references are included as two-part hyphenated numbers that refer to the D-size logic drawings. (The numbers are at the lower left of each logic section bounded by solid lines.) For example, all of the general control function logic in mounting panels 1H, 1J and 1K appears in figure D6-1.

The wiring configuration diagrams, figures 11-4 through 11-7, show wiring and cabling connections between mounting panels. The four mounting panels containing the memory module (panels 3A through 3D) are shown at the right of each of the four wiring configuration diagrams. If more than one memory is used in a machine, the mounting panels containing the additional memories are labelled 3A through 3D regardless of their actual position in the bays.

The 50-pin connector at the left of mounting panel 3C connects the 32 memory address decoder outputs and the 18 memory buffer zero outputs to the memory logic. The connector at the left of mounting panel 3D is also a 50-pin connector; but only 44 connections are used. Of these 44 connections, 36 connections supply the sense amplifier pulse outputs to the memory buffer, and the remaining 8 connections link the timing chain with the memory timing functions.

All cables that terminate in plugs are either flat printed ribbon cable or twisted pair. In cables that carry pulses, each pulse connection alternates with its respective ground connection. The 18 sense amplifier outputs and the 18 corresponding ground connections make up the 36 wires from panel 3D to panel 2D (figure 11-4). Similarly, the 8 wires from panel 3D to panels 1F and 1H comprise four timing pulse circuits.

If the type 15 memory extension is used (figure 11-5), the sense amplifier and memory timing pulse cables originate at the module transfer and selection logic in panel 3Y and at the memory buffer mixer in panel 2Y. The connection to panel 3D of each memory module is made by a separate 50-wire cable from panels 2Y and 3Y. The MAD and MB^0 outputs are supplied through the 50-wire cable originating at the MAD and MB buffers in 2Z. The single cable from panel 2Z connects to the left end connector of panel 3C. These connections are jumpered through panel 3C to the right end connector for connection to additional memory modules.

If the type 19 high-speed channel is used (figure 11-6), sense amplifier and timing pulse connections from the memory are made to the high-speed channel control (panel 1Y). Information from memory is mixed in HSBM and transmitted to MB from panel 1Z. The same cable also carries the control pulses. The MAD and MB^0 outputs to panel 3D come directly from the decoders and the memory buffer (panels 1A and 2C).

In machines including both the type 19 and the type 15 (figure 11-7), connections to memory modules are made exactly as in machines using the type 15 memory extension alone. However, additional cabling is provided from the high speed channel control (in panels 1Y and 1Z) to the memory buffer mixer (panel 2Y) and the memory address register (panel 1A).

11-3 ADJUSTMENT AND CALIBRATION

All DEC systems are designed for maximum reliability under a wide range of operating conditions.

Very little adjustment and calibration is required. The following procedures may be carried out in the course of corrective maintenance, but should not be performed as routine periodic checks.

a ADJUSTABLE DELAYS - Only one type of adjustable delay module is used in the PDP-1. This is the type 4301 delay (1 shot) module. All other delay modules used in the PDP-1 contain distributed-constant delay lines which cannot be adjusted. The duration of the delay in the type 4301 module is adjusted by observing the duration of the level output at pin J. Connect an oscilloscope with a calibrated sweep to pin J. Trigger the sweep internally, and set the sweep time per centimeter adjustment so that the entire duration of the level output is displayed. The duration of the negative going level at pin J is adjusted to the required delay by means of the screwdriver trimpot adjustment. An access hole is provided for the screwdriver in the aluminum frame of the type 4301 module.

b POWER SUPPLIES - The PDP-1 contains two types of variable power supply. These are the 734 marginal check power supply and the type 735 memory power supply. The type 734 power supply provides the marginal check voltage, variable from 0 to +20 vdc. The output polarity of the supply is determined by the setting of the polarity switch on the marginal check switch panel. The adjustment is at the knob at the front panel of the type 734 supply at the top rear of bay 2. This marginal check supply adjustment is used routinely in marginal check procedures.

The type 735 memory power supply adjustments are made at the type 1701 plug-in (part of the supply). The type 1701 plug-in has two access holes for screwdriver adjustments. The adjustment through the center hole is the read/write current adjustment. For each memory module, machines containing the type 15 memory extension provide one complete type 735 power supply with its associated type 1701 plug-in control (10-11e and f). These 735 supplies are each adjusted independently.

Type 735 memory supply adjustments are always made for current output. Never adjust the 735 supply for voltage output. These current adjustments should not be altered unless there has been trouble with the regulation of the supply; both the inhibit current and the read/write current adjustments are set accurately during manufacture. Subsequent adjustment

is seldom required.

If memory malfunction has been isolated to insufficient or excessive read/write current or inhibit current the type 735 memory power supply may be adjusted by the following steps.

1) Deposit the instruction "jump to 0000" (60 0000) in location 0000. Depress the START switch.

2) Attach an oscilloscope current probe to the wire originating at pin W of the type 1972 read/write switch in location 3B1. Set the sweep to 1/2 microsecond per centimeter and trigger the sweep on TP_0. The oscilloscope then displays the read current waveform, followed immediately by the write current waveform which is of opposite polarity.

3) Adjust the current probe to give a calibrated deflection of convenient amplitude. Check both the read and write current waveforms for the current values given in table 11-1 below. Do not yet alter the adjustments at the type 735 memory power supply.

4) Halt the computer and deposit the instruction "jump to 0001" (60 0001) in location 0001. Again depress the START switch.

5) Attach the current probe to the wire originating at pin X of the type 1972 read/write switch in location 3B1. Again check the read and write current waveforms against the values given in table 11-1 below.

6) If both the read current and write current waveforms as observed at both pin W and pin X of the type 1972 read/write switch are incorrect by roughly the same amplitude and in the same direction, then the type 735 memory power supply needs adjustment. Adjust the screwdriver trimpot through the center hole of the type 1701 plug-in so that both the read and the write current waveforms have the value shown in table 11-1 below. This adjustment controls both the read and the write current waveform. The read current may be adjusted two or three milliamps too high if necessary to obtain the proper value for write current (or vice-versa).

7) With the current probe, observe the inhibit current waveform at pins W, X, Y, and Z of the type 1982 inhibit driver located in 3C7. The peak inhibit current

amplitude should correspond to the value shown in table 11-1. The type 735 memory power supply should be readjusted only if the inhibit current at all 4 pins is incorrect by the same amount and in the same direction.

8) If the inhibit current adjustment is required, adjust the screwdriver trimpot through the bottom access hole in the type 1701 power supply control plug-in. The inhibit current amplitude may be checked after adjustment at all inhibit driver output pins except pins W and X of the inhibit driver in location 3C5.

9) Check the inhibit current at pins W and X (location 3C5) by depositing the instruction jda 0000 (17 0000) in location 0001. Since MB_0 and MB_1 are both 0 in this instruction, the inhibit current levels may be checked at pins 3C5W and 3C5X.

TABLE 11-1 NOMINAL VALUES, * MEMORY READ/WRITE AND INHIBIT CURRENTS

Core-stack Manufacturer	Read/write Current	Inhibit Current
RCA	180 ma	165 ma
Ampex	180 ma	165 ma
General Ceramics	180 ma	165 ma
Ferroxcube	200 ma	180 ma

*Optimum current values are listed on a label at the rear of the core-stack. These values set at DEC for best performance under margins.

c SENSE AMPLIFIERS - The type 1540 sense amplifier has two adjustments: the input balance adjustment, and the slice adjustment. These adjustments are made as follows.

1) Using an oscilloscope with a differential preamplifier, observe pins S and U of the sense amplifier. Set the preamplifier of the oscilloscope to display the differential waveform between pins S and U (on Tektronix type CA preamp, set to "add algebraic" and invert one but not both inputs).

2) Run the memory checkerboard program with sense switches set to minimize the base-line drift in the scope display. In particular, do not use the worst pattern portion of the checkerboard program, since this pattern generates severe drift.

3) Set the duration of the scope trace so that one entire memory cycle is displayed: 1/2 microsecond/centimeter. Sync at TP_0.

4) Adjust the balance control through the upper access hole to minimize the noise injected at TP_7 and TP_{10}. These timing pulses correspond to the turn-on and turn-off times of the inhibit current. As the correct adjustment is approached, the sense preamplifier output at TP_4 broadens and increases in amplitude. Often the strobe pulse may be seen as a pip near the center of the sense preamplifier waveform.

5) Remove the oscilloscope probes from pins S and U, and using one probe, observe pin M. This pin provides a -3 vdc logic level from the slicer section of the sense amplifier.

6) At the marginal check switch panel just below the type 734 marginal check power supply, turn on first switch at the left. This switch applies marginal check supply voltage to the +10A line for the sense amplifiers.

7) Have someone stand by the type 734 marginal check supply to vary the voltage and call out meter readings. The lower the voltage of the supply, the greater is the likelihood that spurious bits are generated; the higher the voltage of the supply, the greater is the likelihood of losing bits.

Adjust the slice control through the lower access hole so that bits are lost and "picked up" at marginal check voltages symmetric about the nominal +10 vdc level. As the marginal check voltage is decreased, the duration of the logic level at pin M is seen to increase, until eventually a spurious, thinner -3 vdc logic level appears within, representing the spurious 1 level sensed from a core containing 0.

As the marginal check voltage is increased, the -3 volt logic level at pin M narrows, and eventually either falls or becomes so narrow as to exclude the strobe pulse. When this happens, the sense amplifier pulse output is absent regardless of the state of the sensed core; the bit is lost. The slice control must be adjusted so that the voltages (as provided by the marginal check supply) at which bits are lost and picked up, are symmetric about the nominal +10 volts.

d TAPE READER AMPLIFIERS - The nine reader amplifiers are located on the reader chassis. They are shown in figure 1-2 of the Digitronics Perforated Tape Reader Model 3500 manual dated May 1962.

Before making adjustments on the reader amplifiers, the reader itself should be checked for proper mechanical operation. The reader amplifier adjustments affect the timing and duration of the reader amplifier output levels. However, the intensity and duration of the light impulse sensed by the reader photodiodes also affects these same output levels. Therefore, before adjusting the reader amplifiers, the following four steps should be performed.

1) Check the lateral registration of the punched holes in the tape with respect to the tape edge. Use the Friden tape gauge type T-18118. Insert the gauge pins into the tape feed holes and check that the tape lies in the gauge with the edge nearer the feed holes snug against the raised shoulder of the gauge.

2) Remove the read head cover. Thread the tape through the tape guides on both sides of the read head, but pass the tape over the capstan on the left, and over (not through) the brake assembly on the right. Position the tape lengthwise so that the feed hole of the tape is over the feed-hole photodiode in the read head. Check for lateral registration at the feed holes. The tape feed hole should not be out of registration laterally by more than 10 percent of its diameter. The lateral tape position adjustment is described in e below.

3) With reader power on, check the exciter lamp for yellowing. Check that the light beam falls directly on the row of photodioedes (the tape need not be loaded). If necessary, replace the lamp. The light beam adjustment procedure is described in e below.

4) Replace the read head cover, making sure that the springs at the cover base properly flatten tape folds as the tape passes through the head.

Once the mechanical operation of the reader is satisfactory, the tape reader amplifiers may be adjusted as described below.

Adjustments to the tape reader amplifiers are made at the screwdriver trimpot mounted on each amplifier card. The amplifier output levels should be observed with an oscilloscope

having a dual-trace preamplifier. A closed loop of tape having alternate lines punched with 1's and 0's should be used. The adjustment procedure is as follows.

1) Sync the oscilloscope sweep to the positive-going leading edge of the waveform at pin S of the pulse generator in location 11A1. This signal is the output of the tape reader feed-hole amplifier. Set the sweep to 0.5 ms/cm, so that two lines on the tape are displayed.

2) Run the closed tape loop of 1's and 0's continuously. This may be accomplished by putting the instructions rpa (73 0001) and "jump to 0000" (60 0000) in memory locations 0000 and 0001 respectively; resetting the address switches to 0000; and starting the computer.

3) With one oscilloscope channel, observe the waveform at pin 11A1S (in parallel with the sync probe). Two positive-going levels are displayed. These are the feed-hole levels.

4) Adjust the feed-hole amplifier on the reader chassis to provide positive-going output levels of 1 ms. duration.

5) Set the oscilloscope preamplifier to switch channels on alternate sweeps. With the second scope channel, observe the data channel amplifier outputs (one at a time) at pins F, L, V, Z, J, and N at location 11A20, and at pins K and V at location 11A21. The display should show the two 1-ms feed-hole levels and one data-channel negative-going level of longer duration, corresponding to the line of holes on the tape.

6) Adjust the data-channel amplifiers for a logic level duration of 1.7 to 1.8 ms.

7) Check that the trailing edge of the negative data-channel level is at least 0.3 ms. later than the trailing edge of the positive feed-hole level. This completes the adjustment procedure.

e TAPE READER MECHANICAL ADJUSTMENTS - Three adjustments are described: lateral tape position (registration), pinch-roller clearance, and light-beam position.

(1) Lateral Tape Position - The lateral position of the tape is correct when the tape feed hole and the feed hole photodiode are in perfect registration. Check

registration by removing the read head cover, positioning the tape in both tape guides, and looking through the tape feed hole at the feed hole photodiode. If the registration is incorrect by more than 10 percent of the feed hole diameter, the lateral tape position must be adjusted.

The tape is positioned by the tape guides on each side of the read head. If the tape is too close to the reader front panel, the tape guides may be shimmed. If the tape is too far from the reader front panel, the read head is shimmed out to register with the tape. This procedure is necessary because the guides can only be shortened by machining the guide body shoulder.

To move the lateral position of the tape away from the reader front panel, first remove the retaining rings for both tape guides (see figure 6-2, Digitronics 3500 manual). Remove both tape guides, and insert shims for the required thickness behind the shoulder of the tape guide body (if shim stock is not available, temporary shims may be cut from typewriter paper). Before replacing the shimmed guides, be sure that the tape passes freely but without lateral play between the guide nose and the guide body. If the tape is pinched or there is lateral play, adjust the clearance by loosening the set screw in the guide nose, moving the nose in or out as required, and then retightening the set screw.

If the lateral position of the tape is too far from the reader front panel, the read head is shimmed out to register with the tape. Loosen the four mounting screws of the read head base plate and pull the base plate somewhat away from the front panel (see figure 6-1, Digitronics 3500 manual). Insert the required thickness of shims under the read head base plate, and tighten the four mounting screws. Shims used under the base plate should be large in area, to avoid warping the plate when the mounting screws are retightened.

(2) Pinch-roller Clearance - With power applied to the reader and connector P4 (of the reader) disconnected, the clearance between the pinch roller and the capstan should be between 0.0015 and 0.002 inch. Since the capstan is often out of round by the same order of magnitude, the following procedure should be substituted for the adjustment procedure given in paragraph 4.7 of the Digitronics 3500 manual.

Disconnect connector P4 of the reader and turn reader power on. Loosen the two screws holding the solenoid mount to the reader front panel. Let one finger ride lightly against the pinch roller. Then bring the solenoid mount slowly upwards (decreasing the pinch-roller clearance) until the pinch roller just begins to "kick".

Holding the solenoid immovable against the reader front panel, gradually and evenly tighten the two mounting screws.

After tightening the screws, recheck that the pinch roller is easily stopped by light pressure of one finger (tightening the screws often alters the position of the solenoid mount). Readjust if necessary.

(3) Light Beam Position - If the beam from the exciter lamp does not fall directly on the row of photodiodes, the position of the beam must be adjusted. The beam invariably requires adjustment when the exciter lamp is replaced.

The beam is adjusted by altering the position of the collimating lens (see figure 6-1 in the Digitronics 3500 manual). This lens is secured from the back of the reader front panel by a single 8-32 screw. Loosening this screw allows rotation of the lens about its long axis. The lens should be positioned so that the row of photodiodes lies precisely in the center of the beam of light.

When a replacement exciter lamp has a slightly off-center filament, the lamp can often be rotated within the lamp holder to properly position the light beam. If movement of the lamp is not sufficient to position the beam, the lens must be adjusted as described above.

11-4 RECOMMENDED SPARE PARTS

The most economical quantity of spare parts to be maintained depends on the requirements of the individual user. Spare module stocks for the PDP-1 used one shift per day need not be as large as spare stocks for the PDP-1 used two or three shifts per day. Similarly, in applications that permit only minimal down-time, the stock of spares must be greater than the stock required when more down-time can be tolerated. Paragraphs a, b, and c below discuss recommended spare allowances for modules, circuit components, and in-out equipment respectively. Paragraph d gives recommended mechanical spare allowances.

a MODULE SPARES - For single-shift applications, one spare module of each type usually consitutes a sufficient stock of spares. A spare module of each type permits testing by substitution during off-hours. When a defective module is removed and replaced by the corresponding spare, the defective module can be repaired to create a new spare. Defective transistors, diodes, and other easily detected faulty components can be rapidly removed and replaced. Seldom, during single-shift operations, will two modules of the same type fail before one of the two can be repaired. Table 11-2 gives the DEC module types used in the PDP-1, including standard in-out transfer control; memory; reader, punch and typewriter control logic; and all central processor options.

For PDP-1 applications in which down-time must be minimized, and for installations used for more than one shift per day, additional stocks of the more complex modules are desirable. The more complex modules may require considerable time for diagnosis and repair of faults. To minimize down-time, insurance (in the form of additional spares) should be provided against the possibility of two failures within the time required for repair. Additional spares are also desirable for module types used in large quantities.

Table 11-3 gives recommended additional spare allowances by module type for PDP-1 applications requiring minimal down-time or multiple-shift operation. The spares listed in table 11-3 are recommended in addition to the minimum stock of table 11-2. All PDP-1 installations should stock the table 11-2 listing. High-usage or high-priority installations should stock both the 11-2 allowance and the table 11-3 allowance as well.

For example, the spares allowance of module type 1103 are as follows: in a single-shift installation where moderate down-time can be accepted, only the single 1103 spare listed in table 11-2 need be stocked. In a multiple-shift installation, or in an installation where only minimal down-time can be tolerated, two type 1103 spares are recommended. (The additional spare is listed in table 11-3.) Similarly, the same installation should, if equipped with an additional type 12 memory and type 15 memory extension control, stock three type 1103 spares (see table 11-3).

TABLE 11-2 SPARE MODULE LIST FOR PDP-1

(includes memory, all in-out device controls, and all central processor options)

1103	1209	1684	4105	4201
1104	1213	1685	4106	4209
1105	1304	1701	4110	4214
1110	1310	1703	4111	4301
1111	1311	1972	4112	4401
1150	1410	1973	4113	4410
1151*	1540	1976	4126	4603
1201	1607	1978	4128	4680
1204	1669	1982	4129	4681

* required only if memory extension control type 15 is used.

TABLE 11-3 RECOMMENDED ADDITIONAL SPARES

(for minimal down-time or multiple shift use)

Standard PDP-1 (includes basic memory and standard in-out equipment)

1103	1204	1607	4209	4410
1105	1209	1972	4214	4603
1201	1540	4201	4301	

Central Processor Options

Each additional Type 12 Memory:	1540	2-1972
Memory Extension Control Type 15:	1103	1607
High Speed Channel Type 19:	4603	
Sequence Break System Type 20:	4214	

b IN-OUT DEVICE SPARE PARTS - Recommended spare parts allowances for the Digitronics photoelectric paper tape reader, the Teletype BRPE 11 paper tape punch, and the Soroban computer typewriter are described below.

Table 11-4 gives the recommended spare parts allowance for the Digitronics Type 3500 reader. Installations at which down-time must be minimized should also have a complete Type 3500 reader available for immediate replacement. Table 11-4 is divided into two parts: the first part show items that can be routinely replaced upon failure; the second part shows items that require considerable technical skill for replacement.

TABLE 11-4 TAPE READER SPARES

I: ROUTINE SPARES

Part Number	Description	Quantity
Osram 10-6411	Bulb	1*
B-A2253-8	Drive Belt	1
PGF 1106A	Data Sense Amplifier Cord	1
PSE 1101A	Sprocket Sense Amplifier Card	1
PGE-A-BC1403	SCM Card	1
B-C417	SSA Card	1
B-C412	FRA Card	1
B-C413	SDA Card	1
C-C1367	SPA Card	1

* in addition to the one provided in clip, inside read-head assembly

II: ADDITIONAL SPARES (see paragraph 11-4b)

B-C462-2	Photo Head Assembly	1
A 1073-1	Bearing, Capstan	1
A 1072-2	Bearing, Pinch Roller	1

Experience has shown that the Teletype BPRE 11 punch is an extremely reliable unit. With proper preventive maintenance and lubrication, trouble-free operation can be expected for long periods of time. Apart from lubrication and checkout procedures (described in the Teletype manual), all adjustment and repair should be performed by Teletype or DEC personnel. One spare timing belt should be on hand for each punch. To ensure minimum down time, it is desirable to have available a complete

spare punch. The punch requires two types of lubricant, both available from either the Teletype Corporation or DEC. These lubricants are: Teletype KS7470 oil, and 145867 grease.

The Soroban computer typewriter is a particularly complex piece of equipment. Replacement of any part of the Soroban decoder and encoder assemblies requires delicate and highly precise readjustments. It is highly recommended that a complete spare Computeriter be stocked. Should a fault in the Soroban encoding and decoding equipment occur, the Computeriter should be returned to DEC for repair and adjustment. Spare parts for the typewriter itself (platen, ribbons, etc.) are identical to the corresponding parts for the IBM Model B electric typewriter. The Soroban encoding and decoding equipment requires two types of lubricant: a light silicone machine oil (used in place of Lubri-Plate #2), and Molub-Alloy #3 or equivalent. Molub-Alloy #3 may be obtained from the Imperial Oil and Grease Co., Los Angeles 48, California.

c CIRCUIT COMPONENT SPARES - All circuit components except semiconductor, inductive, and distributive-constant delay components are available through local electronics parts distributors. The semiconductor, inductive, and distributive-constant delay components may be ordered directly from DEC.

Table 11-5 shows the total number of each type of transistor, diode, transformer, and delay line used in the basic PDP-1 and the five central processor options. The quantities given for the basic PDP-1 include the components used in the basic memory module and in the control logic for the standard in-out equipment (reader, punch and typewriter).

The quantities listed for the type 12 memory module apply to each additional type 12 memory module; these quantities must be multiplied by the number of additional type 12 memory modules used. The semiconductors and transformers used in the power supplies are also included in table 11-5.

Although table 11-5 shows the delay lines used in the DEC pulse delay circuits, replacement of delay lines is not recommended. If a malfunction is directly attributable to a faulty delay line, the module should be returned to DEC for repair. The right hand column of table 11-5 shows the module in which the component appears or, in the event of components common to a number of modules, the purpose for which it is used.

The recommended quantity of circuit component spares is listed in table 11-6. The quantities listed under the column headed basic PDP-1 comprise a recommended minimum stock for installations contemplating module repair. The quantities listed for the five central processor options should be added as appropriate to the basic minimum stock. For example, a PDP-1 installation that includes a type 19 high speed channel should stock three 2N393 transistors.

As in table 11-5, the quantities listed for the type 12 memory module apply to each additional memory module. These quantities should be multiplied by the number of additional type 12 memory modules used in the installation.

Since delay line replacement is not recommended, the delay lines do not appear in table 11-6. The three power transformers are also omitted from table 11-6 because power transformer failure is extremely rare. In installations where down time must be kept to an absolute minimum, it is preferable to stock one complete spare of each power supply type, rather than stocking replacement power transformers.

d MECHANICAL SPARES - Table 11-7 gives quantities of mechanical spares recommended for a PDP-1 installation. Except for the quantity of air filters, which varies with the number of equipment bays, the quantities of spares listed are sufficient for most PDP-1 installations. However, table 11-7 lists mechanical spares required only for the optional in-out equipment that is mentioned in paragraph 2-3b.

11-5 PREVENTIVE MAINTENANCE

This paragraph lists recommended preventive maintenance procedures for the standard PDP-1 installation, for all central processor options, and for the standard PDP-1 in-out equipment.

Preventive maintenance procedures should be performed on a rigorously regular basis. By appropriate use of regularly scheduled preventive maintenance techniques, most potential computer malfunctions can be detected before occurrence. In order to minimize computer down-time, this advance detection capability should be used faithfully. Good maintenance is preventive maintenance; corrective maintenance is a costly last resort.

In preventive maintenance procedures involving marginal power levels, the use of maintenance logs is very important. When accurate logs are kept, long term drifts in the values of margin

TABLE 11-5 CIRCUIT COMPONENT TOTALS, BASIC PDP-1 AND OPTIONS

Component	Basic PDP-1	Type 10	Type 12	Type 15	Type 19	Type 20	Modules
TRANSISTORS							
2N 393	56	-	-	-	40	-	1685
2N 456A	4	-	4	-	-	-	735
2N 2485	8	-	6	-	-	-	1973, 1410
2N 2489	170	72	24	102	36	-	1304, 1410 1540, 1607
2N 599	256	-	256	-	-	-	1972
2N 711A	88	-	-	104	-	-	1684
2N 1184	18	-	-	-	-	-	4680
2N 1184B	9	-	-	-	-	-	4681
2N 1204 or 2N 2099	10	-	10	-	-	-	1973
2N 1304	30	-	2	-	20	-	1685, 1701
2N 1305	470	-	4	-	125	78	500 kc logic
2N 527	72	-	-	-	-	-	1669, 4680, 4681
2N 2451	1977	151	149	235	75	85	5-meg. logic
2N 1754	148	-	148	-	-	-	*1972, 1982
2N 1754	346	12	36	52	34	210	500 kc flip-flops, delays, etc.
2N 1204	20	-	20	-	-	-	1982
DIODES							
1N 270	4	-	-	-	-	-	1703
1N 276	5301	361	953	592	612	714	Logic diodes
1N 645	2274	180	234	253	168	336	Reference Diodes
1N 914	179	36	3	51	18	128	Bias & Logic Level
1N 994	169	37	7	-	22	-	Bias & Logic Level
1N 1217	99	-	-	-	-	-	4680, 4681
1N 1220	9	-	-	-	-	-	4681

*BVCES $\geq$ 40v at 100μa

TABLE 11-5 CIRCUIT COMPONENT TOTALS, BASIC PDP-1 AND OPTIONS

(Continued)

Component	Basic PDP-1	Type 10	Type 12	Type 15	Type 19	Type 20	Modules
1N 429 (Zener)	2	-	2	-	-	-	1701
1N 3316B (Zener)	2	-	2	-	-	-	735
GRS 20 SP-4B4	4	-	-	-	-	-	(GE Thyrector) 812 power control
PULSE TRANSFORMERS (all are DEC types)							
T 2003	114	36	39	51	18	-	1304, 1410 1540, 1607
T 2006	1	-	-	-	-	-	1304
T 2010	75	36	3	51	18	-	1607
T 2012	1	-	-	-	-	-	1410
T 2017	4	-	-	-	-	-	4201
T 2018	90	-	-	-	30	21	4603
T 2019	1	-	-	-	-	-	4401
T 2020	19	-	-	-	-	1	4410, 4301
T 2021	2	-	-	-	-	-	4401
T 2023	9	-	-	-	-	1	4301
T 2024	109	-	-	-	30	22	4603, 4410, 4301
T 2029	64	10	-	-	-	-	1201
T 2033	38	-	-	-	-	-	1204
POWER TRANSFORMERS (all are DEC types)							
100-X-1010	1	-	-	-	-	-	734 Supply
100-X-1015	1	-	1	-	-	-	735 Supply
100-X-1016	9	-	1/3*	-	-	-	728 Supply
DELAY LINES (all are DEC types)							
330-25E-3	40	12	4	-	-	-	1310
330-25E-6	16	17	1	-	-	-	1310, 1311

*One type 728 supply required for up to three additional Type 12 memory modules; e.g. four Type 12's require two Type 728's; however, two Type 728's can power up to six Type 12's etc.

TABLE 11-6 RECOMMENDED CIRCUIT COMPONENT SPARES

	Basic PDP-1	Type 10	Type 12	Type 15	Type 19	Type 20
TRANSISTORS						
2N 393	2	-	-	-	1	-
2N 456A	1	-	1	-	-	-
2N 2488	1	-	1	-	-	-
2N 2489	4	2	1	2	1	-
2N 599	10	-	6	-	-	-
2N 711A	2	-	-	1	-	-
2N 1184	1	-	-	-	-	-
2N 1184B	1	-	-	-	-	-
2N 1204 or 2N 2099	2	-	1	-	-	-
2N 1304	2	-	-	-	1	-
2N 1305	6	-	-	-	3	2
2N 527	2	-	-	-	-	-
2N 2451	10	1	2	1	1	1
2N 1754*	4	-	2	-	-	-
2N 1754	5	-	1	1	1	2
2N 2099 or 2N 1204	2	-	2	-	-	-
DIODES						
1N 270	1	-	-	-	-	-
1N 276	40	2	5	3	4	4
1N 645	25	2	4	1	3	4
1N 994	4	1	1	-	1	-
1N 1217	2	-	-	-	-	-
1N 1220	1	-	-	-	-	-
1N 429 (Zener)	1	-	-	-	-	-
1N 3316B (Zener)	1	-	-	-	-	-
GRS 20SP-4B4	1	-	-	-	-	-

*BVCES $\geq$ 40V at 100 μa

TABLE 11-6 RECOMMENDED CIRCUIT COMPONENT SPARES

(Continued)

	Basic PDP-1	Type 10	Type 12	Type 15	Type 19	Type 20
PULSE TRANSFORMERS						
T 2003	2	-	1	1	1	-
T 2006	1	-	-	-	-	-
T 2010	2	1	-	1	-	-
T 2012	1	-	-	-	-	-
T 2017	1	-	-	-	-	-
T 2018	2	-	-	-	1	1
T 2019	1	-	-	-	-	-
T 2020	1	-	-	-	-	-
T 2021	1	-	-	-	-	-
T 2023	1	-	-	-	-	-
T 2024	2	-	-	-	1	1
T 2029	2	-	-	-	-	-
T 2033	1	-	-	-	-	-

TABLE 11-7 MECHANICAL SPARE PARTS

Part Number and Description	Quantity of Spares
53E168, Type CFG: Rotron fan with #2R blade	1
Rotron Venture: Muffin fan with mounting clips	1
10" x 10" x 2" EZ Kleen Filters	*
Type 418 Super Filter Coat, Pints	2

*One filter unit required for each bay of the installation; Example: the standard PDP-1 requires four filters.

voltage that cause malfunction are readily apparent. It is especially useful to make a log entry noting any temporary malfunction which may occur during either testing or actual operation. Such entries can be invaluable in isolating intermittent failures. When a

malfunction occurs, it is useful to note the portion of the program at which the malfunction is noticed, as well as any control settings and panel indications which may be relevant to the difficulty. Location of an intermittent failure is frequently accomplished by logging two or more malfunctions which all intersect at a common defective component.

Component deterioration is often evidenced by malfunction of a particular module at a steadily decreasing margin voltage. Replacement may be indicated when such a long term drift in margin voltage is detected, even though the margin at which failure occurs has not yet exceeded normal limits. It is always better to perform such preventive replacement before an actual operationg malfunction occurs. Prompt replacement of deteriorating modules often forestalls computer errors, thereby reducing both error down time and diagnostic down time.

To minimize computer malfunctions, the following schedule of preventive maintenance procedures is strongly recommended.

a EVERY DAY (OPERATOR'S MAINTENANCE):

1) Run the DEC maintenance programs (MAINDEC) without margins. Log all error halts, noting the reason for the error halt if known.

2) Inspect and clean the tape-handling surfaces of the tape reader. These include the read head, tape guides and rollers, the pinch roller, the capstan, and the brake. Use a lint-free cloth or a cotton swab (eg a Q-tip) moistened with denatured alcohol if necessary.

3) Inspect and clean the tape-handling surfaces of the paper tape punch. Use a lint-free cloth, a cloth strip, or a soft toothbrush, as convenient. Do not use alcohol or other solvents near the feed pawl or the die block since such solvents remove the light lubricating film. Empty the chad container.

4) Inspect and clean the platen and paper guides of the Computeriter as necessary. (The platen need be cleaned only if typing has run off the page or if the Computeriter has run without paper.) Clean the type, using a carbon-eating putty (such as Eberhard Faber "Star Type Cleaner"). Remove lint and other fouling from ribbon guides; replace ribbon if necessary.

5) Check that all cooling fans (bottom of each bay and back of each core-stack) are operating properly. Check for free flow of air.

6) Replace any non-critical malfunctioning components which can be detected by observing the console (eg. indicator lamps, etc).

NOTE

The remainder of the scheduled preventive maintenance procedures should be performed by qualified technicians only.

b EVERY WEEK (IN MULTIPLE-SHIFT OPERATION, EVERY 80 HOURS):

1) Check the operator's maintenance logs. Note malfunctions and error halts detected by operator's MAINDEC runs. Note non-critical component replacements made by the operator (Does a particular fuse blow too often? Does an indicator light burn out too often?)

2) Reader inspection: Check for effects of vibration, and for wear at the drive belt. Check drive belt tension and adjust if necessary. With the reader power off, gently rotate the motor pulley, feeling for stickiness or bind in bearings.

3) Punch inspection: Check for the effects of vibration, for tightness of wiring connections, and for tightness of the nuts and screws that lock the adjustments. Check for the presence of oxidized (red) metal dust near bearing surfaces, indicating insufficient clearance; this condition must be rectified immediately. With the punch unit cover removed (by removing the four mounting screws) rotate the main shaft slowly in the normal direction (clockwise as viewed from the front). During rotation activate all movable elements checking for freedom of movement. Check that all contact points meet squarely.

4) Computeriter inspection: Check for effects of vibration: for tightness of cable plugs, mounting screws, etc. Check decoder solenoids for tightness of mounting, and check both the decoder solenoids and the accelerator cam trip magnet (TCM) for signs of overheating. Check the surface of the typewriter power roll for signs of excessive wear or presence of lubricant. If any oil or grease has deposited on the power roll surface it must be cleaned thorougly (use a cloth dampened with alcohol or trichloro-ethylene).

Run the "QUICK BROWN FOX" maintenance program (MAINDEC 14). Check that the type is even on the line and of relatively equal impression. If a character

prints weak, is askew, or is out of line, check that the type bar does not strike the type bar guide (turn the power off and lift the type bar manually).

5) Memory checkout: Run the memory checkerboard maintenance program (MAINDEC 2). Using the worst-case configuration, increase the sense-amplifier +10 A margin until the first bit is lost. Note the meter reading at the type 734 marginal check supply. Decrease the sense-amplifier margin until the first bit is picked up; note the meter reading. The voltage at which the first bit is picked up and the voltage at which the first bit is lost should be symmetric about +10 vdc.

If the bit lost at high margin is the same bit as that lost at low margin, the corresponding sense amplifier may be weak. In this case, increase margins slightly further until a second bit is lost and note the reading. Similarly decrease the lower margin further until a second bit is picked up. Again these margins should be symmetric about +10 vdc; however, the difference in margin voltage between the first bit error and the second is the criterion for judging the sense amplifier.

If the margin voltages causing bit error are not symmetric about +10 vdc, the sense amplifier slice adjustment should be performed. If one sense amplifier appears weak, the sense amplifier balance adjustment should be made before summarily replacing the sense amplifier. Procedures for both adjustments are given in c above.

c EVERY MONTH (IN MULTIPLE SHIFT OPERATION EVERY 160 HOURS):

1) Check the operator's maintenance logs. Note malfunctions and error halts detected by operator's MAINDEC utility program runs. Also review noncritical component replacements by the operator, checking for excessive replacement of a particular component.

2) Run all MAINDEC programs with margins. The procedure for running MAINDEC with margins is discussed in paragraph 11-7 below. It is extremely important to log all malfunctions caused by the application of marginal voltage.

3) Change and clean the air filters at the bottom of each bay, using the following procedure. Loosen the two thumbscrews holding the fan and filter housing to the floor of the cabinet. Remove the fan and filter housing. The filter can then be taken out of the housing and the clean spare filter installed. Replace the fan and filter

housing containing the clean filter, and tighten the two thumbscrews.

Clean the filters by thoroughly flushing them with hot tapwater in a direction opposite to that of airflow. When all dust and lint is removed, shake out excess moisture. Stand the filter on one end for ten or fifteen minutes to allow remaining mositure to evaporate. If the flush water is sufficiently hot, the filter should dry completely in about fifteen minutes. Finally, spray the filter with aerosol Super Filter Coat or an equivalent product. This spray serves both as a dirt-capturing medium and as a detergent which helps wash out the dust and lint during the next reverse flushing.

4) Check all moving parts of the reader for freedom of movement and for wear. Check the outputs of the data channel amplifiers and the sprocket channel amplifier, using the procedure of paragraph 11-3d above. Do this regardless of whether or not the tape reader system passed the MAINDEC reader test program. The reader does not require lubrication; all bearings, including those of the drive motor, are permanently lubricated and require little attention. If a bearing shows any sign of sticking, it should be replaced.

5) Lubricate the high-speed tape punch according to procedures given in section five of Teletype Bulletin 215B. The teletype lubrication procedures cover lubrication of several different types of punch. It is important to follow the lubrication procedure that applies to the particular punch type supplied with the PDP-1. It is particularly important to let no oil or grease accumulate between the armatures and the magnet pole faces, or between contact points. Always wipe off excess lubricant.

6) Lubricate the Computeriter following procedures given in the two Soroban manuals. One of the two manuals covers lubrication of the decoder and power unit; the other covers lubrication of the encoder unit. It is essential to avoid contaminating the power roll surface with lubricant. In order to eliminate possible lubricant oxidation, use a light silicone machine oil for all applications where Lubri-Plate #2 is specified in the Soroban manuals.

11-6 OPERATOR'S MAINTENANCE

The computer operator at a PDP-1 installation has three maintenance responsibilities: 1) performing

the daily machine checkout; 2) keeping an accurate Operator's Log; 3) acting as an aid to the technician in troubleshooting machine malfunctions.

The daily machine checkout procedure is described in paragraph 11-5a above. Preferably, this checkout procedure should be performed as the first operation after the machine is turned on in the morning. Running the MAINDEC test programs is the most important part of the daily checkout procedure.

One good reason for running the MAINDEC programs every day is that if there is a fault in the computer logic, one of the MAINDEC test programs is nearly certain to reveal it. This means that the fault in the computer is encountered while running a program which is known to be good. On the other hand, if the fault should occur during the running of a normal operating program later in the day, the fault might easily be dismissed as a program error, on the grounds that the program had not been sufficiently debugged.

In this case, the fault in the machine remains undiscovered, and valuable time is wasted debugging a program which might be good. However, to ensure that the machine is not at fault, the operator might run the MAINDEC. If this is done, when MAINDEC discovers the fault, the operator should of course notify the author of the operating program, so that at least he does not waste further time debugging a good program.

For efficient operation, it is obviously desirable to discover faults by running the MAINDEC rather than to encounter them during operating programs. Otherwise all computing time, from the start of the operating program which originally encountered the fault until the discovery of the fault, is wasted. During this time the machine produces no useful results.

Another good reason for running the MAINDEC as the first operation of the machine shift is that MAINDEC test programs have a diagnostic value. If a MAINDEC program discovers a fault in the computer logic, it simultaneously gives indications as to the general location of the fault. This is certainly not true of operating programs. A great deal of time that would otherwise be spent in diagnosing the location of a fault can be saved if the fault is first discovered by MAINDEC.

The Operator's Log, if kept properly and in sufficient detail, can be a valuable aid in subsequent diagnosis of machine malfunction. Nearly every computer installation maintains an operating time log for purposes of charging machine time to the various departments or divisions

using the machine. The operator's maintenance log can easily be combined with this operating time log. While usually noted as a part of the machine operating time, the daily running of the MAINDEC utility programs is most often charged directly to maintenance.

From a maintenance point of view, the most important items to be logged are:

1) The times at which computer power was turned on and off.

2) The name or type of program being run during operating time.

3) The exact times of beginning and end of computer down-time.

4) The reason for down-time, if any, and the corrective measures taken to restore the computer to operating condition.

5) All replacements of minor components made by the operator in the absence of the technician.

6) In case of any computer malfunction and, in particular, in case of a fault discovered by MAINDEC, the register indications and settings of all sense and address switches.

A sample format for an operation log is given as table 11-8. This format is included only as an example. Each user should modify the sample format to best suit his own needs. Any operation log format should, however, have two desirable characteristics: first, entries during operating time should be simple and require little time to complete because the operator is busy during operating time; second, there should be plenty of space for operator's remarks and comments as well as for entries of register indications and sense switch positions in case of malfunction.

In most new computer installations, the operator has an opportunity to become familiar with programming and machine language before the technician. In addition, the operator rapidly becomes familiar with the frequently used operating programs and routines. This knowledge is a valuable aid in troubleshooting malfunctions that develop during normal operation. Since the operator is so often familiar with the internal workings of commonly run programs, he is often able to make a preliminary diagnosis of malfunction location.

Even when a preliminary diagnosis cannot be made, thoughful observation of register indicators and positions of test word address and sense switches generally enables the operator to eliminate vast sections of computer logic from suspicion. The operator is also in a better position than

PDP LOG

POWER	TIME ON					DATE
	TIME OFF					

USER	TIME ON	TIME OFF	COMMENTS	ELAPSED TIME METER READING AT POWER OFF

TABLE 11-8 SAMPLE OPERATION LOG FORMAT

the technician to discriminate between faults caused by machine logic malfunction and errors caused by program bugs.

In many troubleshooting problems it is convenient to use small program loops containing only a few instructions. These loops may be used either for exercising certain portions of the machine logic, or for diagnostic purposes within a small section of machine logic. Diagnostic and exercise loops are, generally, extremely simple; ie, an exercise loop could consist of only a single instruction. The adjustment procedure for the type 735 memory power supply, for example, uses three such one-instruction loops (shown in steps 1, 4, and 9 of paragraph 11-3b). Other examples of exercise and diagnostic loops are given in paragraph 11-8c. The operator is often able to aid the technician considerably by producing simple program loops for specific troubleshooting applications.

11-7 MAINDEC PROGRAMS

The DEC Maintenance Programs (MAINDEC) permit effective use of PDP-1 for self-testing. For the majority of possible equipment malfunctions, intelligent use of these programs provides efficient trouble detection and location. Complete descriptions of the MAINDEC programs and procedures for their use are given in MAINDEC Program Guides (a below).

Before loading a MAINDEC program tape into the tape reader, the reader motor must be turned off by pushing READER switch down. This releases the brake, allowing the tape to be loaded. The fan-folded MAINDEC tape stack is placed in the right-hand tape bin, oriented so that the tape unfolds from the top of the stack. Tape movement through the reader is from right to left. Looking in the direction of tape movement, the five data holes are to the left of the sprocket hole and the remaining three data holes are to the right of the sprocket hole. The centerline of the sprocket hole leads the centerline of the data holes through the reader. Figure 11-8 shows the appearance of the top surface of the tape and the direction of tape movement when the tape is properly loaded.

a MAINDEC PROGRAM GUIDES - The MAINDEC program guides are a separate set of self-contained technical memoranda, each of which is designed as an aid to learing the function and application of a single MAINDEC program. For rapid reference, all MAINDEC program guides are written in the same format. Each guide contains three

major sections:

1) Console operating procedure.
2) Suggested applications of program.
3) Program description.

Both the first and second sections are intended for reference and should usually be consulted each time the test program is used. The third section, program description, is designed as an aid to understanding the program rather than for repeated reference. Each program guide starts with a cover-page abstract which permits convenient identification of the program. Immediately following the abstract is the console operating procedure.

(1) Console Operating Procedure - This section is composed of the following five tables:

Table	Contents
1) Tapes Required for Test	Lists tapes which are required to run the program.
2) Switches	Lists console switches applicable to the program, and specifies appropriate settings.
3) Load Sequence	Gives detailed step-by-step instructions for loading and starting the program.
4) Error Halts	Lists addresses of the programmed error halts, the contents of relevant registers, and the meaning or cause of the error halt.
5) Post-Error Restart Procedures	Specifies correct procedure for restarting the program after each type of error halt.

(2) Suggested Applications of Program - The console operating procedure is followed by a section covering suggested program usage. This section of each program guide presents a generally useful test sequence which the operator may perform. It by no means exhausts the capabilities of the program,and is not intended

to limit the freedom of the operator to modify program use where appropriate.

(3) Program Description - The third and final section of each program guide contains a detailed description of program structure and operation. The operator does not need to read this section each time he runs the test. It would in fact be possible to execute the suggested test without understanding the program at all. However, a good understanding of the program yields the ability to modify program usage as required by actual computer malfunctions.

In addition to a detailed description of the program, the third section of each guide also includes a program flow chart and listing.

b USE OF MARGINAL CHECK - Variable power supply type 734 produces all marginal check voltages used in PDP-1. This supply and the associated marginal check switch panel are located at the top of the bay 2 plenum door. The 734 supply furnishes voltages which vary from 0 to -20 vdc, or from 0 to +20 vdc, depending upon the setting of the polarity switch. Voltage amplitude variation between 0 and 20 volts is controlled by the large black knob on the 734 supply. The output voltage is shown by the MARGINAL CHECK voltage meter.

Marginal voltage can be applied to the A lines of any mounting panel by pushing up the top toggle switch on that panel. Marginal voltage can be applied to the B lines by pushing up the center toggle switch. When marginal checking the A and B lines, the polarity switch must be in the +10 MC position. To marginal check the C lines (-15 vdc), set the polarity switch to -15 MC and push up the bottom toggle switch at the left front of the panel to be tested.

When the polarity switch is in the OFF position, normal voltages are applied to all three power lines of every panel, regardless of the settings of the toggle switches. The OFF position of the polarity switch is provided as a convenience to the technician during troubleshooting, not as a substitute for turning off the toggle switches at the mounting panels. To minimize the effects of unauthorized tampering (for example, if someone should inadvertently turn on the 734 supply during an operating program) be sure to turn off all three marginal check toggle switches at the left of every mounting panel at the completion of marginal check procedures.

Marginal check voltages may be applied to a single module alone by means of the pig-tail plug-in extender (paragraph 9b below).

The two +10 vdc power lines are used principally as the base bias supplies for transistor logic. Making the +10 vdc supply more positive checks transistor current gain (β). Reducing the +10 vdc supply tests for excessive transistor leakage.

The -15 vdc power line is used chiefly as the collector supply. The -15 vdc line is applied only to pulse amplifiers (module type 1607 and 4603). Making the -15 vdc supply more negative increases pulse amplifier gain; making the -15 vdc supply less negative decreases pulse amplifier gain. Marginal voltage should not be applied simultaneously to all -15 vdc supply lines throughout the computer, because the resulting load exceeds the rating of the type 734 supply.

During MAINDEC runs, the particular mounting panels to which marginal check voltages might be applied depend upon which maintenance program is selected. Any program, whether used for maintenance purposes or not, consists of instructions which must come from memory locations. Regardless of the particular section of machine logic which a MAINDEC program tests, it must always depend upon the memory, the timing, and the instruction logic for its operation. Routine MAINDEC runs with marginal checking should, therefore, be performed in the following order:

1) ADDRESS TEST (MAINDEC 3)
2) MEMORY CHECKERBOARD (MAINDEC 2)
3) INSTRUCTION TEST (MAINDEC 1)
4) MUS-DIS or MUL-DIV (MAINDEC 4 or 4A)
5) RPA TEST (MAINDEC 11)
6) QUICK BROWN FOX (MAINDEC 14)
7) PUNCH TEST (MAINDEC 12)
8) SEQUENCE BREAK (MAINDEC 5)

If desired, the RPB TEST program (MAINDEC 10) may be substituted for the RPA TEST program, and the TYPE-ECHO program (MAINDEC 13) may be substituted for the QUICK BROWN FOX program.

When running MAINDEC programs that test the memory, marginal check voltage should be applied only to the memory module and the memory control logic. The memory

checkerboard program, for example, primarily tests the core bank and the sense amplifiers. If marginal check voltages were applied to control logic (bay 1) while running the memory checkerboard program, control malfunctions might be introduced which the checkerboard program could not diagnose. Conversely, if marginal check voltages were applied to the memory module while running the instruction test program, memory errors might be introduced which the instruction test program could not diagnose.

Suggested applications of marginal voltage (mounting panel locations) for each of the eight basic MAINDEC programs are presented in table 11-9. This table applies primarily to routine MAINDEC runs. While running a MAINDEC program to solve a specific troubleshooting problem, the application of marginal check voltage is made at locations determined by the nature of the problem. To aid in selecting mounting panels for marginal checking, the physical location of the various logic sections of the computer are shown in figure 11-1.

c LOG ENTRIES - During the running of MAINDEC with marginal check voltages, it is extremely important to keep detailed and accurate logs. An accurate set of logs, when combined with the production test marginal check record (supplied with the PDP-1) makes up a complete operating history of all machine logic under marginal conditions. When accurate and complete logs are kept faithfully, any deterioration of circuit components is easily detected. A deterioriating component is revealed in the logs as an error of a particular type which occurs at steadily decreasing margin voltages over a period of months. Without logs, a symptom of this type is likely to pass completely unnoticed.

When accurate logs are kept, a glance at the previous history of the failure pattern in certain sections of computer logic often saves unnecessary replacement of plug-in units. For example, a certain sense amplifier may operate in a completely reliable manner under normal conditions, yet it may fail at a narrower margin than the others. Normally this would indicate that the sense amplifier was weak; however, when a logged history of sense amplifier performance under marginal check voltages is available, it could immediately be determined whether the narrower margin at which this sense amplifier failed is the result of a gradual drift (indicating weakness) or is merely the voltage at which this sense amplifier has always failed. In the first case, the sense

TABLE 11-9 MAINDEC: APPLICATION OF MARGINAL CHECK

Program	A-Lines	B-Lines	C-Lines	Remarks
MEMORY CHECKERBOARD		SENSE AMPS		
	3A	3C	3C	one panel
	3B	3D		at a time
	3C			
	3D			
ADDRESS TEST	3A	3D		one panel
	3B			at a time
	3D			
INSTRUCTION TEST	BAY 1	BAY 1	BAY 1	check 1B,
	except	except	except	1H separately
	1B, 1H	1B, 1H	1B, 1H	
	1B	1B	1B	alone
	1H	1H	1H	together
	3D	3D	3D	
	BAY 2	BAY 2	BAY 2	check 2B
	except	except	except	separately
	2B	2B	2B	
	2B	2B	2B	
MUS -DIS or MUL-DIV	SAME AS FOR INSTRUCTION TEST			
RPA TEST	FEED HOLE; INFO HOLES			one at a time
	1B	1B	1B	simultaneously
	1D	1D	1D	
	1H	1H	1H	
	1J	1J	1J	
	1K	1K	1K	
	2H	2H	2H	
	2J	2J	2J	
	3H	3H	3H	

TABLE 11-9 MAINDEC: APPLICATION OF MARGINAL CHECK

Program	A-Lines	B-Lines	C-Lines	Remarks
RPA TEST (continued)	3J	3J	3J	simultaneously
	3K	3K	3K	
	11A	11A	11A	
	11B	11B	11B	
QUICK BROWN FOX	11B	11B	11B	simultaneously
	1B	1B	1B	
	1D	1D	1D	
	1H	1H	1H	
	1J	1J	1J	
	1K	1K	1K	
	2H	2H	2H	
	2J	2J	2J	
	3H	3H	3H	
	3J	3J	3J	
	3K	3K	3K	
	Substitute 11C for 11B and restart			
PUNCH TEST	Same as for QUICK BROWN FOX except no 11C.			
SEQUENCE BREAK TEST	BAY 1	BAY 1	BAY 1	simultaneously; check 1B separately
	except	except	except	
	1B, 1H	1B, 1H	1B, 1H	
	3H	3H	3H	
	3J	3J	3J	
	3K	3K	3K	
			11A	
			11B	
			11C	
	ALL R1	ALL R1	ALL R1	

TABLE 11-9 MAINDEC: APPLICATION OF MARGINAL CHECK

Program	A-Lines	B-Lines	C-Lines	Remarks
SEQUENCE BREAK TEST (continued)	1B	1B	1B	
	3D	3D	3D	} simultaneously
	3H	3H	3H	
	BAY 2	BAY 2	BAY 2	check 2B
	except 2B	except 2B	except 2B	separately
	2B	2B	2B	

amplifier would be replaced. In the second, however, since no drift in margin voltage is taking place, it can be assumed that the sense amplifier will continue indefinitely to give reliable performance under normal conditions.

11-8 TROUBLESHOOTING

There are many ways to troubleshoot the PDP-1. Most of them are slow, inefficient, and frustrating. A few however are fast, efficient, and usually rewarding. One such method is described in this section.

The troubleshooting procedure presented below does not involve the use of malfunction tables or symptom-cause-remedy charts. For a full-scale general purpose digital computer like PDP-1, such tables or charts would be so cumbersome and inconvenient as to be useless. Instead, the method suggested depends upon logical thinking, common sense, and an organized step-by-step procedure. Troubleshooting has been likened to detective work; the analogy is accurate. In troubleshooting, just as in detective work, the leg-work is vital.

For efficient troubleshooting, the technician must be completely familiar with the PDP-1 system function and machine logic. When confronted with a malfunction, a technician who is not familiar with the machine wastes valuable time poring over prints and elementary system description, thus unnecessarily prolonging down-time. It is essential to have a good underlying knowledge of system function (chapter 3), operation (chapter 5), the machine logic (chapters 6-9), and, to a lesser degree, plug-in unit circuit theory (chapter 10).

It is equally important to be familiar with the logic prints (D-size block schematics).

Most people find it impractical to memorize the prints completely; but, after a reasonable learning period, the competent technician will probably know which print contains what logic, and even the approximate area of the print which show given sections of the logic. All of the machine logic is on the prints. The description of the logic (chapters 6 through 9) is tied to the prints. The flow of logic levels and pulses can be traced from the prints. Location and identification of circuits by mounting panel location, pin connection, and module type is also given on the prints.

This maintenance manual is intended primarily as an aid to learning the prints. Once the system logic is well understood, the prints are usually used much more often than the manual. The prints are the best available source of reference information, and the prudent technician should make the time to become thoroughly at home with them.

The technician should, for example, know that the pulses and levels for accumulator shifts and rotations originate at the logic shown in figure 6-4, that these signals are gated and amplified by the logic shown in figure 7-3, and that they are applied to the AC shift and rotate gates shown on figure 7-1. After working with the machine for a reasonable period, it will become almost second nature to recall that although the IO transfer logic is shown in figure 6-5, the pulse controlling the transfer from the reader buffer into IO is generated by the logic on figure 9-3. This is much easier than it sounds; once the function of the machine is well understood, the details of the machine logic follow quite logically, making the learning task less formidable than the sheer bulk of the data would suggest.

In addition to the prints and this maintenance manual, the technician should have four manufacturer's manuals for the standard in-out equipment: the Digitronics manual for the paper tape reader; the Teletype manual for the punch; and the two Soroban manuals for the Computeriter. It is helpful to read these manual, and learn what information is in them, and what information is not. Much computer down-time can be saved by knowing in advance where to go for information.

In troubleshooting a malfunction, particularly one which is encountered during an operating program, it is essential to maintain a calm, logical attitude. Malfunctions that occur during operating runs are generally accompanied by continuous high-level noise, both from the

operator, and from the author of the program. Although this atmosphere is conducive neither to a calm attitude nor to logical thinking (the emotional strain is frequently contagious), nevertheless, efficient troubleshooting must be done calmly, logically, and with common sense. The noise invariably damps out, given time.

When confronting a new malfunction in the machine, the following sequential plan of attack is usually effective:

1) Initial investigation: gather all available information on the problem.

2) Preliminary check: see if the malfunction presents obvious physical symptoms.

3) Console troubleshooting: use the MAINDEC utility programs and, if indicated, marginal checking procedures; attempt to localize the problem to a particular section of logic.

4) Logic troubleshooting: further localize the malfunction to a particular module, power supply, or power control unit.

5) Module troubleshooting: locate the specific malfunctioning component within a plug-in unit or power unit.

6) Testing after repair: ensure that the machine is really back on the line.

7) Logging the trouble: note what went wrong, and how it was fixed.

Steps (1) through (4) are further discussed in paragraphs a through d below. Step (5), module troubleshooting, is treated in 11-9b. Step (6) is explained in e below. Use of maintenance logs (step 7) is covered in f below.

a INITIAL INVESTIGATION - As the first step in troubleshooting a malfunction, before even touching the PDP-1, the technician should find out as much as possible about the nature of the malfunction. Question the operator mercilessly. Consult the operator's log: did the malfunction also occur during the operator's MAINDEC runs? Has the operator noticed any unusual machine behavior as a prelude to this malfunction? If this malfunction occurred during a MAINDEC run, the operator should have noted the readings of the console indicators, and the settings of the test word, address, and sense switches. Also, if the malfunction occurs during a MAINDEC run, these indications and settings are very significant and should be carefully noted.

Look for a possible history of this malfunction among previous entries in the maintenance logs. This step is particularly important if more than one technician works on the machine. Has the same malfunction, or one related to it, occurred before. If so, how was it remedied? Look also at the last half-dozen monthly entries for MAINDEC runs with marginal check. Is there a deteriorating module (one that fails at steadily decreasing margin voltages) that seems related to this malfunction? If so, compare notes; do the register indications and switch settings for that failure resemble the indications and settings for the present malfunction?

The more information the technician can gather, the more rapidly he can make his diagnosis, and the sooner the machine can be returned to operation. Every available source of information should be explored. Do not try to troubleshoot a computer malfunction cold; usually this just wastes time.

b PRELIMINARY CHECK - The second step is to check for physical symptoms of malfunction. Look first at the test word, address, and sense switches. Make sure that the operator is not running the program incorrectly. Open the plenum doors. Look for blown fuses, broken cords or plugs, tripped circuit breakers. Has someone inadvertently placed half the machine on marginal check voltages? Are all plug-in units plugged in all the way? Are all memory cables plugged in all the way? Is there power?

This preliminary check is useful far more often in the case of a catastrophic malfunction than in the case of an intermittent one. Except for cable and plug-in unit connections which may be intermittent, most intermittent malfunctions are due to cold solder joints, or faulty circuit components. Unless the malfunction is almost certainly isolated to within two or three modules by the initial investigation (a above), it is poor strategy to start checking arbitrary modules for cold solder joints or bad components. More sophisticated troubleshooting procedures must then be used (c and d below). Nevertheless, because it eliminates many common sources of trouble which might otherwise be overlooked, the preliminary check should not be omitted. Few things are more annoying than to through complex time-consuming troubleshooting procedures only to discover that the malfunction is actually caused by some obvious cable connection not making proper contact.

c CONSOLE TROUBLESHOOTING - In many cases, the initial investigation discloses an appropriate line of attack, but does not in itself pinpoint the location of the trouble. The third step in the troubleshooting sequence, troubleshooting from the console, is used to localize the malfunction within a small section of the machine logic.

Console troubleshooting most often requires the use of the MAINDEC programs (with or without marginal checking, depending on the nature of the malfunction.

Intermittent malfunctions caused by weak components can almost always be aggravated and so transformed to relatively consistent malfunctions by the use of marginal checking with MAINDEC. Unfortunately, use of marginal check won't reveal intermittent connections (such as cold solder joints or interruptions in cable wiring). Consistent (catastrophic) malfunctions are, however, easy to locate. Simply run the appropriate MAINDEC program, wait for the error halt, and consult the MAINDEC program guide.

(1) MAINDEC Program Selection - Normally the initial investigation (a above) restricts a malfunction to within some large section of machine logic. For example, a particular in-out device does not work properly; or it is impossible to transfer into the accumulator; or some operating program which is known to be good invariably halts at a certain memory location; or the automatic divide instruction produces gibberish at the accumulator; etc.

Even if the program does not run at all, the malfunction can usually be well localized by initial investigation. In this case the trouble is likely to be located in the general control functions (ie, in the timing chain; or in the control logic for run, defer, in-out halt, or memory; or in the instruction register and decoder; or possibly in the program counter and program control logic).

For nearly any computer malfunction, the initial investigation usually determines which of the MAINDEC programs is the proper one to use. The MAINDEC program used depends on the character of the malfunction. Suppose for example that the complaint involves unexplained halts in operating programs. The operator says that attempts to restart the programs result merely in another unexplained halt, and he then produces a list of memory locations at which the machine frequently halts. A logical choice of MAINDEC program for investigating this malfunction

would be the memory address test.

(2) Use of Marginal Check - The particular MAINDEC program most likely to discover a malfunction should first be run without marginal check voltages. If the malfunction is a consistent one it shows up immediately during this first run. If the malfunction does not show up during the first run without marginal voltages, there are two alternate possiblities, one of which probably explains the situation. First, the malfunction may be an intermittent one, for example, a loose connection. Second, there may be some loading condition imposed on the logic circuits which is present during the operating program, but not during the MAINDEC.

If the malfunction is caused by loading conditions unique to the operating program, it can nevertheless be detected during a MAINDEC run. This is done by applying marginal check voltages to the suspected sections. From the last few entries in the MAINDEC marginal check log, the technician can determine the marginal check voltages which normally cause failure during the selected MAINDEC program. A malfunction caused by differences in loading conditions is likely to show up at considerably smaller margins than those listed in the last few log entries.

If MAINDEC discovers the error during application of marginal check voltages, do not restart the computer. First look at the register indications and at the test word and sense switch settings for hints locating the malfunction.

(3) Procedure - Console troubleshooting procedures for locating catastrophic malfunctions should be directed toward discovering a pattern of consistency among the errors. Two examples of procedures which locate catastrophic malfunctions through the discovery of error patterns are presented in (4) below.

Location of an intermittent malfunction is extremely difficult using console troubleshooting alone. For this reason, procedures for locating intermittent malfunctions are explained under logic troubleshooting (d below).

Malfunctions (either catastrophic or intermittent) in the start-clear and special pulses, and in other control logic initiated by console switches can easily be located with the 60-cycle button pusher (paragraph 11-1b), and, if necessary, with short exercise

loops (d below).

(4) Examples - Two examples illustrating the discovery of error patterns are given; the first a continuation of the memory address trouble hypothesized in (1) above; the second an example of trouble involving the in-out equipment.

In (1) above, the memory address test was selected as the MAINDEC program most likely to give an indication of the malfunction location.

A number of possible malfunction locations may occur to the technician: the X and Y selection inverters, the memory address register, the memory address decoders, the memory address transfer logic, the program counter, etc. However, before running the memory address test, the operating program should be closely examined: do the memory locations at which the computer likes to halt all contain the same instruction? If they do, the instruction register or the instruction decoders could be at fault.

In the present example, suppose that examination of the operating program discloses that the troublesome addresses do not all contain the same instruction. The logical first step in console troubleshooting is then running the memory address test.

If the memory address test detects an error, the failing address is indicated in the in-out register lights and its contents are indicated in the accumulator lights. Suppose that this is an intermittent malfunction, i.e. suppose that the entire memory address test proceeds to completion with no error halt.

The fact that the machine completes the test without an error halt suggests that circuit loading conditions during the normal operating program are significantly different from conditions during the test run. Do not despair--use marginal check!

There is one fortunate characteristic which every malfunction displays. No matter how esoteric it may be, there is always a thread of consistency in the errors it produces.

Returning to the present example, suppose that the marginal check voltages cause recurrence of the malfunction, and that every error halt of the memory address test occurs at an address (shown by the in-out register lights) in which bits 9 through 11

contain 5 octal (101). Now there is something to work on.

The binary-to-octal decoder for MA_{9-11} is in location 1A17 (see figure 8-1). The output level labelled 5 is asserted negative. This level feeds a 50-pin Cannon connector to the memory module. Figure 8-3 shows the level arriving at pin F of the Y selection inverter module in location 3D2. The ground assertion level comes out on pin T of the same module. We can check for the positive assertion level at the memory read/write switches in locations 3B17 and 3B19. Except for an extremely pathological situation (such as simultaneous malfunction of all eight read/write switches in 3B17 and 3B19), the trouble must be located somewhere along the line of connections listed above.

The console troubleshooting procedures illustrated by the foregoing example have isolated the trouble to a manageable section of the computer logic. The technician can now proceed with the signal tracing procedures described in d below.

As a second example, suppose that the operator reports that the typewriter prints nonsense. It is possible, by spending a great deal of time, to make some sense of the remarks printed out as part of the operating program's output subroutine; however, the data output is entirely meaningless. If the preliminary check discloses nothing useful, the logical first step is to run the MAINDEC Type Echo program (MAINDEC-100).

Start the type echo program at register 30, so that the typewriter alternately types out the left half, and then the right half of the test word switches, returning the carriage after every 100 octal characters. Using table 5-1, the numeric code for every character and typewriter function can be set into the test word switches. Suppose during this investigation it is discovered that TW_6 and TW_{15} have no effect. Figure 9-5 indicates that signal tracing might logically start with the modules in locations 11C18, 11C19, 11C21, and 11C23; and from there might proceed through the cable to the solenoids in the typewriter.

d LOGIC TROUBLESHOOTING - After console troubleshooting procedures (c above) have located the trouble to within a small section of the computer, logic troubleshooting methods are employed to further isolate the malfunction to within a single module or module connection. Logic troubleshooting is most often done using the oscilloscope and

small diagnostic or exercise loops consisting of only a few instructions.

Logic troubleshooting is normally the fourth step in the troubleshooting sequence; it relies heavily upon successful completion of the first three steps. Logic troubleshooting is detail work, normally performed only on small sections of logic or particular discrete strings of connections between panels. With the possible exception of diagnostic and exercise loops the procedures outlines in this paragraph are applicable only to small subsystems. These procedures should not be substituted for the console troubleshooting methods outlined in c above. To avoid wasted time from widespread detail work, use console troubleshooting to isolate the malfunction to the smallest possible section of machine logic before logic troubleshooting procedures are begun.

The remaining portions of this paragraph discuss the construction and development of both diagnostic and exercise loops, as well as suggested procedures for logic troubleshooting.

(1) Diagnostic and Exercise Loops - A program loop is a set of instructions, one characteristic of which causes the computer to repeat the set of instructions over and over again. A program loop may contain any number of instructions, and generally incorporates some method of indexing. The diagnostic and exercise loops discussed here, however, contain only a few instructions, and are designed to repeat indefinitely (ie, no indexing is used).

Exercise loops are designed specifically to repetitively pulse some small specific section of the computer logic. Some examples of exercise routines, used in adjusting memory circuits, are included in paragraph 11-3b, steps 1, 4, and 9, and in step 2 of paragraph 11-3d.

A diagnostic loop is simply an exercise loop which includes some method of sensing for error. A diagnostic loop can be set up to halt the computer in case of error, in such a way that the console indicator lights give some indication of the location of the malfunction causing the error. Good familiarity with the PDP-1 instruction list is required for the development of small exercise and diagnostic loops.

Exercise and diagnostic loops are used generally as a way of keeping some section of computer logic operating repetitively in a predictable manner. When repetitive

operation is set up in this way, oscilloscope signal tracing techniques can be used to determine whether the correct pulses and levels are being generated. Diagnostic loops are also used during console troubleshooting procedures as an aid in further narrowing the possible trouble area.

The general exercise loop consists of instructions which set up initial conditions, followed by an instruction or instructions which generate the desired pulse or level, followed by a jump instruction which returns control to the beginning of the exercise loop. The simplest possible exercise loop is just the jump instruction itself. If a jump instruction is deposited in the memory location corresponding to its own address portion, and the computer is started at that memory location, the computer simply performs the jump instruction again and again. The jump instruction itself provides the necessary repetition, as well as being in itself the desired exercise operation.

In general, the set of instructions which sets up desired initial conditions should be very short, certainly no more than three instructions. If a particular repetitive operation requires a complicated or involved pattern of initializing, then nearly all the preparation should be done at the console before depositing the exercise loop. Instructions that set up initial conditions in exercise loops are normally taken mainly from the operate group of instructions (cli, lat, cla). Occasionally the instructions law (an augmented instruction) and lac (a memory reference instruction) are useful.

The instruction or instructions which generate the desired operating pulse or level in an exercise loop naturally depend on the specific repetitive pulse or level desired. For example, the pulse that generates in AC the exclusive OR function between the contents of AC and the contents of MB is produced by a number of instructions, ie add, xor, sad, sas. Note that if the instructions sad or sas are used, it is also possible to make the machine diagnose itself by skipping on an error.

One of the jump instructions, jmp, jsp, or jda must be used to make the exercise routine repeat itself indefinitely. The following example (table 11-10) illustrates an exercise loop of six instructions and two data words which tests the reaction of each bit of the accumulator to the exclusive OR pulse under each of the four possible initial conditions. In other words, the loop repetitively checks each bit of the

accumulator for behavior corresponding to the exclusive OR truth table at the end of paragraph 7-2c.

TABLE 11-10 SAMPLE EXERCISE LOOP

Exclusive OR pulse loop	Instructions	Loc.	Contents
Clear accumulator	cla	1000	76 0200
Exclusive OR with all 1s	xor 1s	1001	06 1006
Exclusive OR with all 0s	xor 0s	1002	06 1007
Exclusive OR with all 1s	xor 1s	1003	06 1006
Exclusive OR with all 0s	xor 0s	1004	06 1007
Repeat	jmp xor-1s	1005	60 1001
	all 1s	1006	77 7777
	all 0s	1007	00 0000

Alternatively, the instruction clear accumulator could be replaced by the instruction load accumulator from test word switches (lat: 76 2200).

In a diagnostic loop, the set of operations to be repeated includes some instruction or instructions for sensing an error. Generally, these are the skip instructions. All of the skip instructions are in the skip group except for the memory reference instructions sad and sas. Table 11-11 gives an example of a simple diagnostic loop to test the response of accumulator bits to the exclusive OR pulse.

TABLE 11-11 SAMPLE DIAGNOSTIC LOOP

Exclusive OR pulse loop	Instructions	Loc.	Contents
Load Acc. from Test Word Switches	lat	1000	76 2200
Deposit Acc. in loc. CTW	dac CTW	1001	24 1006
Exclusive OR with Test Word	xor CTW	1002	06 1006
Skip on +0 in Acc.	sza	1003	64 0100
Halt	hlt	1004	76 0400
Jump to beginning	jmp lat	1005	60 1000
Test Word location	CTW	1006	C(TW)

This example of a diagnostic loop does not test for all possible combinations of 1's and 0's in AC and MB as did the exercise loop of table 11-10. However, the loop does halt the computer on an error, and shows the malfunctioning bit of AC in the accumulator indicator lights (that bit which is different from the corresponding bit of the test word switches).

The table 11-1- and table 11-11 examples are given only to illustrate form and organization. Diagnostic and exercise loops take on endless variety. A technician should be able to generate applicable loops to troubleshoot any section of the computer.

Although the examples given are typical maintenance loops, they are designed for one specific application. Even if a similar problem should happen to occur, the particular circumstances are not likely to be identical. For example, it may not be necessary to test for every combination of 1's and 0's (as does the exercise loop of table 10-10). It may not even be necessary to test every bit of the accumulator. If the problem seems confined to the exclusive OR pulse logic alone, and is unrelated to the response of the AC bits, the simple exercise loop composed of the instruction xor, followed by jmp to the location containing xor, would be sufficient. Ingenuity, common sense, and familiarity with the instruction list enable an alert technician to develop diagnostic or exercise loops to suit any specific troubleshooting problem.

(2) Suggested Procedures - Logic troubleshooting procedures should be undertaken only after the preliminary check or console troubleshooting has isolated the malfunction to a small section of the computer logic.

Logic troubleshooting is performed inside the computer. It always consists of a number of steps designed to narrow down the location of a malfunction to within a particular plug-in unit, connection, or power unit. The specific steps required and the order in which they are carried out always depend on the problem. In general, however, logic troubleshooting steps fall into three broad categories: 1) signal tracing, 2) substitution, and 3) aggravation.

Only one good method of signal tracing is available: the use of the oscilloscope. Since component troubleshooting has (hopefully) isolated the trouble to within a small section of computer logic, an appropriate exercise or diagnostic loop can

be used to operate the suspected section of logic repetitively. When the machine is running in a closed exercise or diagnostic loop, the desired operating pulse or level is generated at intervals that are always multiples of the 5-microsecond memory cycle. The oscilloscope sweep may be synchronized to any of the timing pulses (each generated once per memory cycle). The duration of the sweep may be set either to 1/2 microsecond per centimeter (so that the entire sweep displays one memory cycle), or to a value which displays one complete performance of the exercise or diagnostic loop.

Every pulse or level generated in a given section of machine logic is available at some plug-in module output pin. The output pin locations are shown on the print corresponding to that section of machine logic. In the case of a catastrophic malfunction, the signal tracing method determines with absolute certainty whether a pulse of good quality (amplitude, duration, and rise time) is being generated at the right time. In the case of an intermittent malfunction this signal tracing technique must be combined with some appropriate form of aggravation (discussed below).

Substitution is the technique which first occurs to most technicians. Usually, a spare plug-in module is substituted for a suspected module, to see whether the malfunction is thereby cured. When troubleshooting registers and counters, however, it is often more useful to exchange bits of the register or counter rather than substituting a spare module. After the exchange, if the malfunction has moved to the new location, the trouble is probably in the exchanged module. However, if the malfunction still affects the original location, the malfunction is more likely to be in some logic network supplying pulses or levels to that location.

Aggravation, as an electronics maintenance technique, sounds as though it should be scrupulously avoided; it is actually quite useful. In the troubleshooting of intermittent malfunctions, aggravation is often the only technique which gives any indication of malfunction location. The two main types of aggravation are vibration and the application of marginal check voltage. Marginal check voltage application as a method of finding weak components or malfunctions due

to program-linked differences in circuit loading is discussed in c above. Vibration, as a malfunction locating technique, is primarily used to locate intermittent connections.

As long as reasonable care is used to avoid inflicting permanent damage, the technician, in his search for a malfunction, should not hesitate to twist, probe, worry, or poke at connections, cables, plugs or plug-in units. All connections in PDP-1 are designed for excellent reliability. Connections through plugs and sockets and through cables should be impervious to any reasonable amount of pulling, twisting, or flexing. If such aggravation produces a computer error, an intermittent logic connection is probably causing the malfunction. Procedures for finding intermittent connections in individual plug-in units are described in paragraph 11-9.

Intermittent failures caused by poor connections cannot be located merely by using marginal check with the MAINDEC. However, an intermittent connection can often be revealed by vibrating the modules while running an appropriate MAINDEC program. Wipe the handle of a plastic screwdriver across the back of the suspected row of modules. The resulting vibration generally interrupts an intermittent connection.

By repeatedly restarting the MAINDEC program and narrowing the area of vibration (tapping fewer and fewer modules) the malfunction can be localized within one or two modules. After localizing the malfunction in this way, try wiggling the suspected module up and down within the mounting panel. If wiggling the module cause a computer halt, before removing the module closely inspect the associated mounting panel wiring.

Although each PDP-1 system is thoroughly tested before it leaves the factory, nevertheless, one or two poorly soldered connections occasionally show up later. This type of malfunction appears as an intermittent failure, and is sometimes very difficult to locate. Poorly soldered connections, if any, are more likely to appear in mounting panel or plug and cable connections than within the modules themselves.

e TESTING AFTER REPAIR - After a malfunction has been located and the defective plug-in unit or connection replaced or repaired, a complete test should be made of the entire system.

The procedures described above (console troubleshooting and logic troubleshooting) are usually followed under the tacit assumption that only one malfunction is present in the machine. As a matter of fact, even after faulty parts related to one malfunction are located and repaired or replaced, the system may still contain other faults. In order to ensure that the PDP-1 is definitely in perfect operating condition, the entire MAINDEC should be run with the application of marginal check voltage after the completion of corrective maintenance procedures. Particular emphasis should of course be placed on those portions of MAINDEC which check the originally malfunctioning section of the computer logic.

A record of the computer malfunction and the way in which it was repaired is entered in the maintenance log. The record of the final test, however, should form part of the marginal check log. It should include a notation showing that this unscheduled test was performed because of computer malfunction and give a page reference to the malfunction writeup in the maintenance log.

f MAINTENANCE LOGS - The first step of any troubleshooting procedure is an initial investigation of the malfunction (a above). This entails gathering all available information which might apply to the problem. The maintenance log forms a vital part of that information. It must therefore contain a clear, detailed, accurate description of every computer malfunction, giving the cause of the malfunction, the steps taken to isolate the malfunction, and the way in which the malfunction was finally repaired.

A maintenance log should not be constrained into a standardized and rigid format; rather, it should look much like a diary. Computer malfunctions are much too diverse to categorize into any standard-form questionnaire. The date and time of each entry should be followed by comments describing everything that the technician does to the computer for whatever reason.

In addition to malfunction writeups, the maintenance log should also contain the

technician's observations of other pertinent maintenance information. For example, if the technician notices excessive vibration in the punch, even if such vibration is not accompanied by malfunction, it should be noted in the log.

The operator may mention that some piece of in-out equipment seemed sluggish for a minute or two; this fact should also be noted in the maintenance log. If the operation log should indicate that some particular light or fuse has been replaced too many times over a short period, this should be noted in the maintenance log as well. Maintenance log entries of this type may seem insignificant or even silly at the time they are entered; but later if a malfunction turns out to be related to these symptoms the entries do not appear silly in the least.

Accurate logs reveal at a glance the previous history of failures throughout the entire system. No one can be aware of every possible failure pattern of a large system such as PDP-1. Properly kept maintenance logs can often reveal patterns of consistency among failures that seem totally unrelated. In troubleshooting, completely new lines of attack can often be suggested by such patterns of consistency. To take best advantage of this information, the maintenance log must be kept accurately and faithfully. The more information that is available on a trouble, the less computer down-time before it is isolated and repaired.

11-9 MODULE REPAIR

When the location of a malfunction has been narrowed to within a specific module , it may be worthwhile to continue troubleshooting within the module. In many cases, a minute or two of additional oscilloscope signal tracing can isolate a malfunction to a particular transistor, diode, or connection. Considerable bench testing time can often be saved in this way, even if the module must be replaced. The following portions of this paragraph describe removal and replacement of modules, troubleshooting within modules, and circuit-component replacement.

a REMOVAL AND REPLACEMENT - It is advisable to turn off system power before extracting or inserting modules.

Modules are extracted by means of the DEC Type 1960 Module Puller furnished with the PDP-1. Carefully hook the small flange of the puller over the center of the module rim, and gently pull the unit out of the mounting panel. Use a straight, even pull to avoid damaging the plug connections or twisting the etched circuit. Since the puller does not fasten to the module, prevent the unit from falling by grasping the rim of the unit in your other hand before the unit comes all the way out of the mounting panel.

When replacing a module, always position it so that the component side of the board is to the right, and the printed wiring side of the board is to the left. The aluminum rim of a module extends along the bottom edge beyond the plug. When a module is properly installed, this aluminum extension fits into a matching slot in the mounting panel. Should a module be installed with the bottom edge up, this aluminum extension prevents the module from making contact with the socket in the mounting panel.

Carefully slide the module in between the guide ridges embossed on the mounting panel surfaces until the plug just begins to make contact with the socket. If the plug and socket are properly aligned, a gentle pressure is sufficient to fully insert the module. If the plug and socket are not aligned, do not force the connections. Occasionally, slight movements of the module within the guide ridges may be necessary to match the plug with its connector. After a little practice, rapid removal and replacement of modules is very easy.

The tape reader printed circuit cards (mounted on top of the reader chassis) are removed by grasping the top of the card with the thumb and index finger of one hand, and the edges of the card with the thumb and fingers of the other hand. A gentle pull disengages the card from its socket. The card then slides straight up and out of the mounting panel. These cards are inserted by guiding the card into the two mounting panel slots and letting it slide downward until it just makes contact with the socket. A slight push downward fully engages the connections. When installing reader cards, note that the component side of the card is on the right, when looking from the front of the reader.

Connections to the power supply and power controls are made both by connections at barrier terminal strips, and by cables terminating in plugs. Although the wiring to barrier terminal strip connections is color-coded, color-code markings denoting the proper connections may be absent from the chassis of the power unit. Before removing

or replacing a power unit, clearly mark all unlabeled connections both on the unit to be removed and on the spare to be installed. After disconnecting the unit, release it by removing the Phillips-head mounting screws on both sides. (The power units are fairly heavy, so get a good hold to avoid dropping them when they are removed.)

b MODULE TROUBLESHOOTING - Locating a malfunction within a single module can be done in two ways. The first (active circuit troubleshooting) involves use of the module extender (DEC Type 1954), the pig-tail plug-in extender, the oscilloscope, and small two- or three-instruction exercise loops of the type described in paragraph 11-8c above.

The second method is bench troubleshooting. This involves use of a suitable multimeter (paragraph 11-1) and other applicable bench test equipment such as an in-circuit transistor and diode checker, a regulated bench power supply, etc.

In complex installations, or in multiple installations, an independent bench module tester may be desirable. A bench tester for DEC plug-in modules can be made up from a Type 722 power supply, standard DEC signal-generating modules (clocks, pulse generators), and suitable switching circuits.

Such a bench tester, when used with an oscilloscope, can provide active-circuit troubleshooting (signal tracing) independently of the computer. A desirable addition to the bench module tester is a type 734 power supply. This supply permits marginal checking of plug-in units at the bench.

Information on system design with DEC modules, helpful in assembly of a bench tester, is contained in the Digital Modules Catalog (A-705) and the Digital Logic Handbook (A-400B), both available from DEC without charge.

(1) Module Extender - The DEC Type 1954 module extender permits troubleshooting a plug-in unit while the system is operating. First, turn off the system power. Next, remove the plug-in unit to be tested and insert the extender in its place. Then plug the unit into the exposed end of the extender. Finally, turn the power back on. The unit is then accessible for troubleshooting.

If it is desired, during troubleshooting, to apply marginal voltages to an individual plug-in unit, the pig-tail plug-in extender (paragraph 11-1a above) may be used instead. Pins A, B, and C of the exposed plug-in unit can then be furnished appropriate marginal check voltages through the three alligator clip leads.

CAUTION

Although DEC circuits are designed with internal safeguards which prevent damage from opening or shorting the output terminals on a single unit, they are not proof against all the accidental shorts which might be produced while testing the unit on an extender card. Care must also be exercised when testing terminals on the wiring side of the racks.

(2) Bench Troubleshooting - If simple inspection fails to reveal the cause of trouble, the use of multimeter resistance readings can usually isolate the trouble to a specific defective component.

Resistance readings may be taken to check the emitter-base and the collector-base diodes of transistors in both the forward and reverse directions. It is essential to determine the internal battery polarity of the multimeter. Often this polarity is opposite to the normal polarity of the leads used for voltage and current measurements.

Resistance readings for both the emitter-base and collector-base diodes of a transistor should be relatively low in the forward direction, and relatively high in the reverse direction. Note that although incorrect resistance readings are a sure indication that the transistor is defective, correct readings give no guarantee that the transistor is good. It may have other troubles.

Several types of inexpensive in-circuit transistor and diode checkers are on the market. These can generally provide a more reliable indication of diode or transistor malfunction.

Damaged or cold-soldered connections can also be located with the multimeter. Set the multimeter to the lowest resistance range, and connect it across the suspected connection. Poke at wires and components around the suspected connection with a probe or with the fingers. Alternatively, rap the module sharply (but not too sharply) on a wooden surface. Often the response time of a multimeter is too

slow to detect the rapid transients produced by intermittent connections. Current interruptions of very short duration, caused by intermittent connections, can be detected by placing a 1.5-volt flashlight battery in series with a 1500-ohm resistor across the suspected connection, and observing the voltage across the 1500-ohm resistor with an oscilloscope while probing the connection.

c COMPONENT REPLACEMENT - Use a six-volt soldering iron with an isolating transformer for removal and replacement of defective components. Avoid excessive heat which can cause damage to components and may even cause delamination of the etched wiring. Transistors and diodes require special care. Whenever possible, attach a copper alligator clip, or other heat sink, to the lead being soldered, thus reducing the amount of heat transferred to the component.

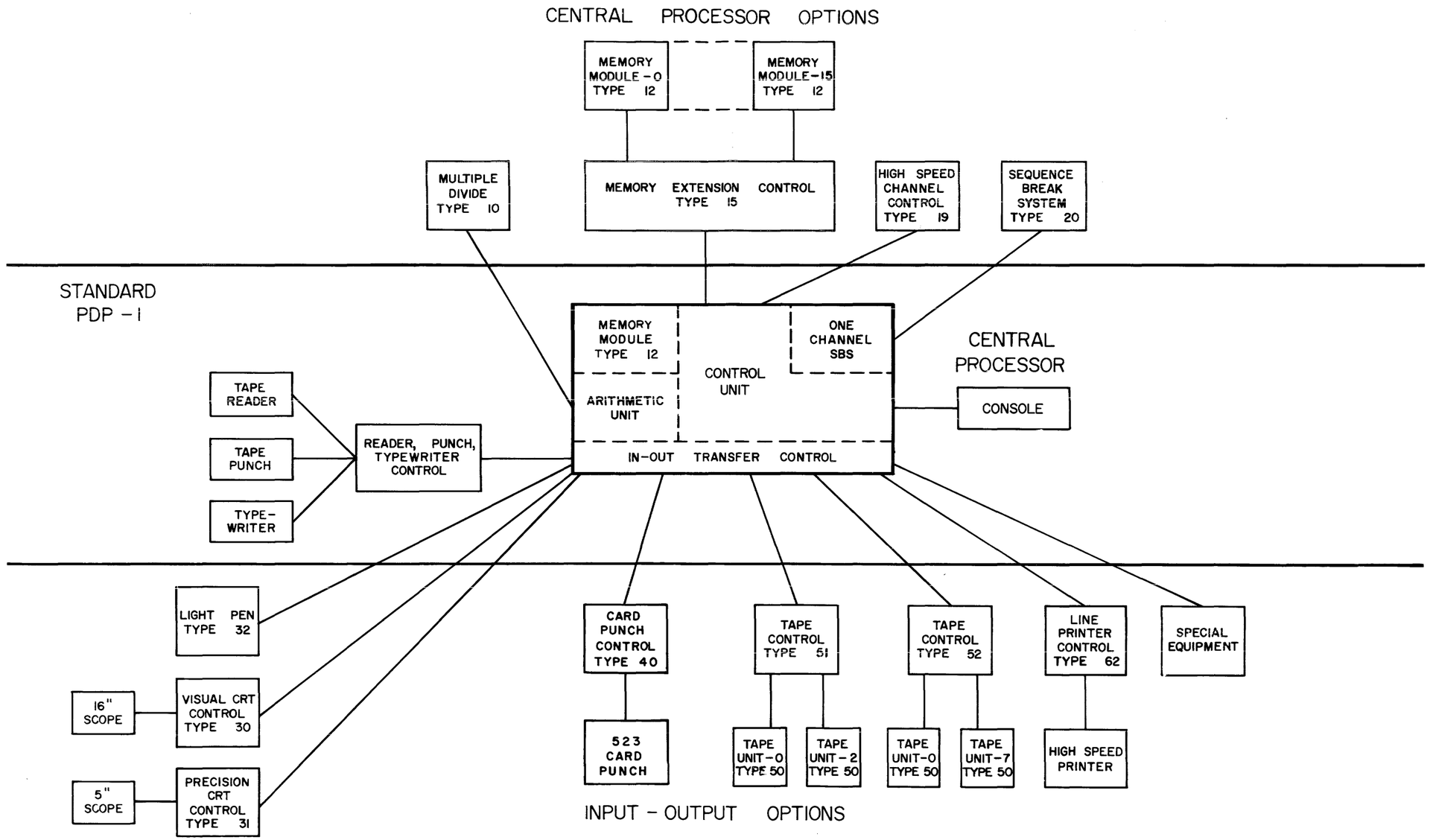


Figure 2-1 PDP-1 System Configuration Diagram

Figure 2-2 Standard PDP-1

Figure 2-2

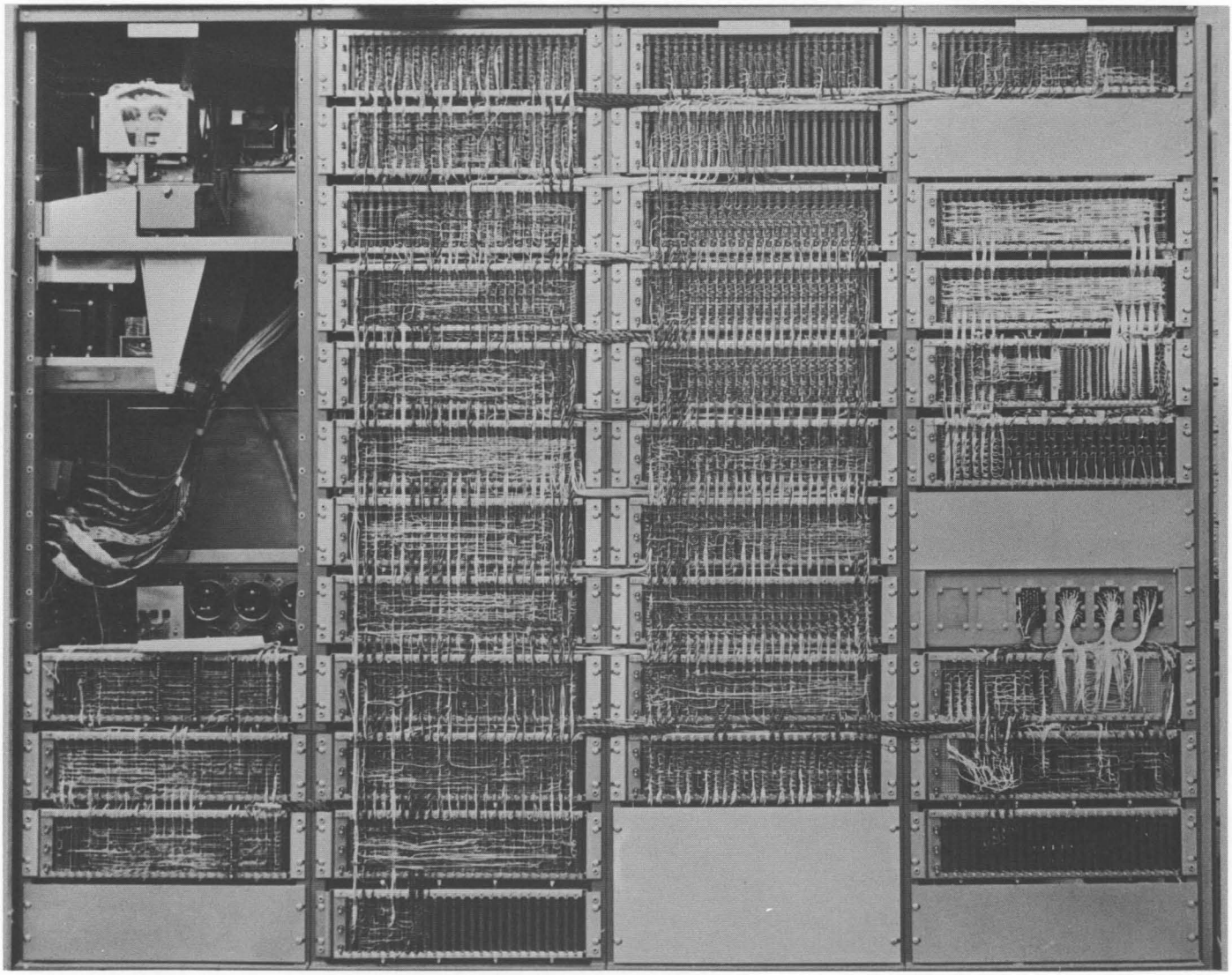


Figure 2-3 Central Frame Logic Panels

Figure 2-3

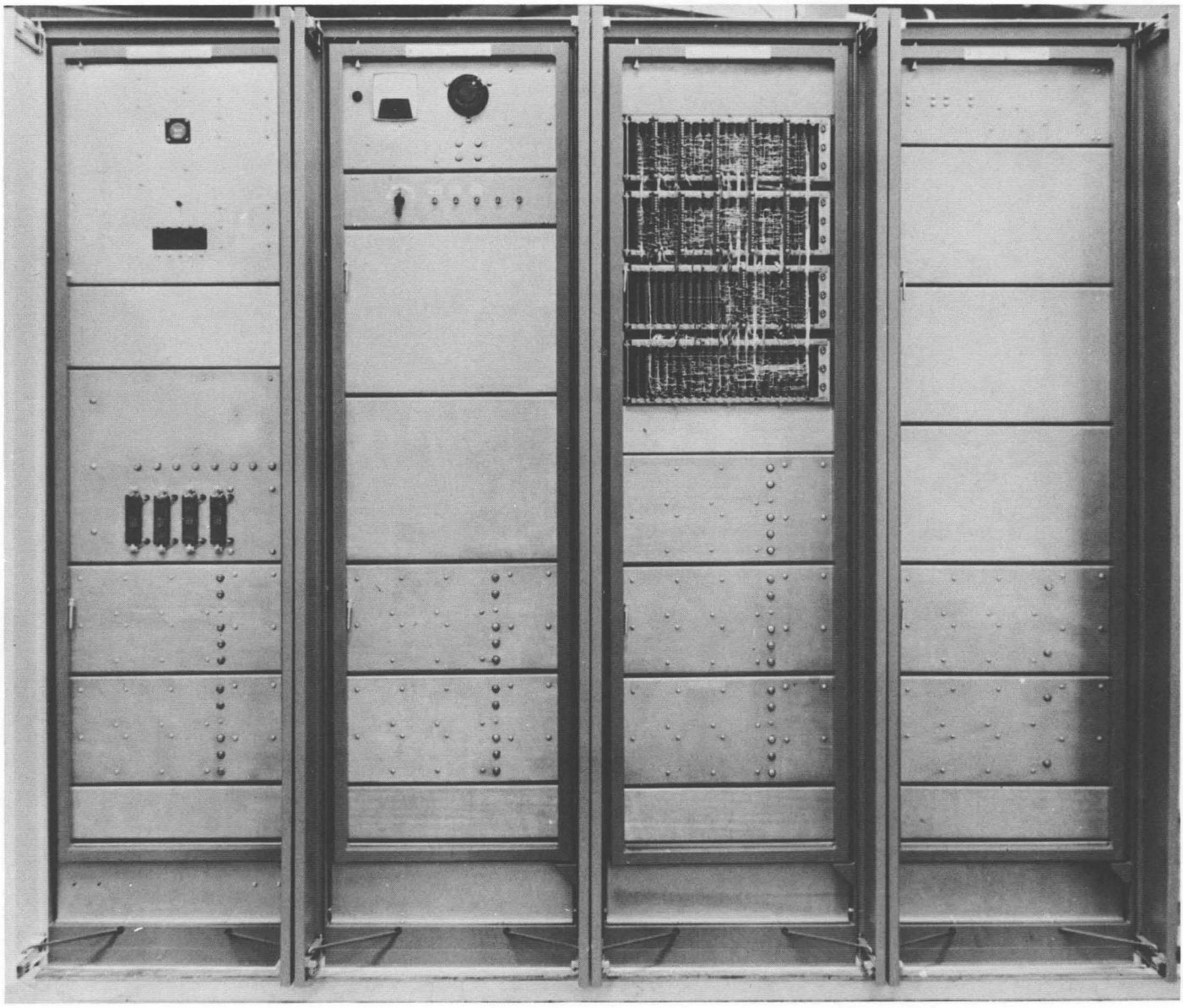


Figure 2-4 Plenum Doors, Back View of Central Frame Bays

Figure 2-4

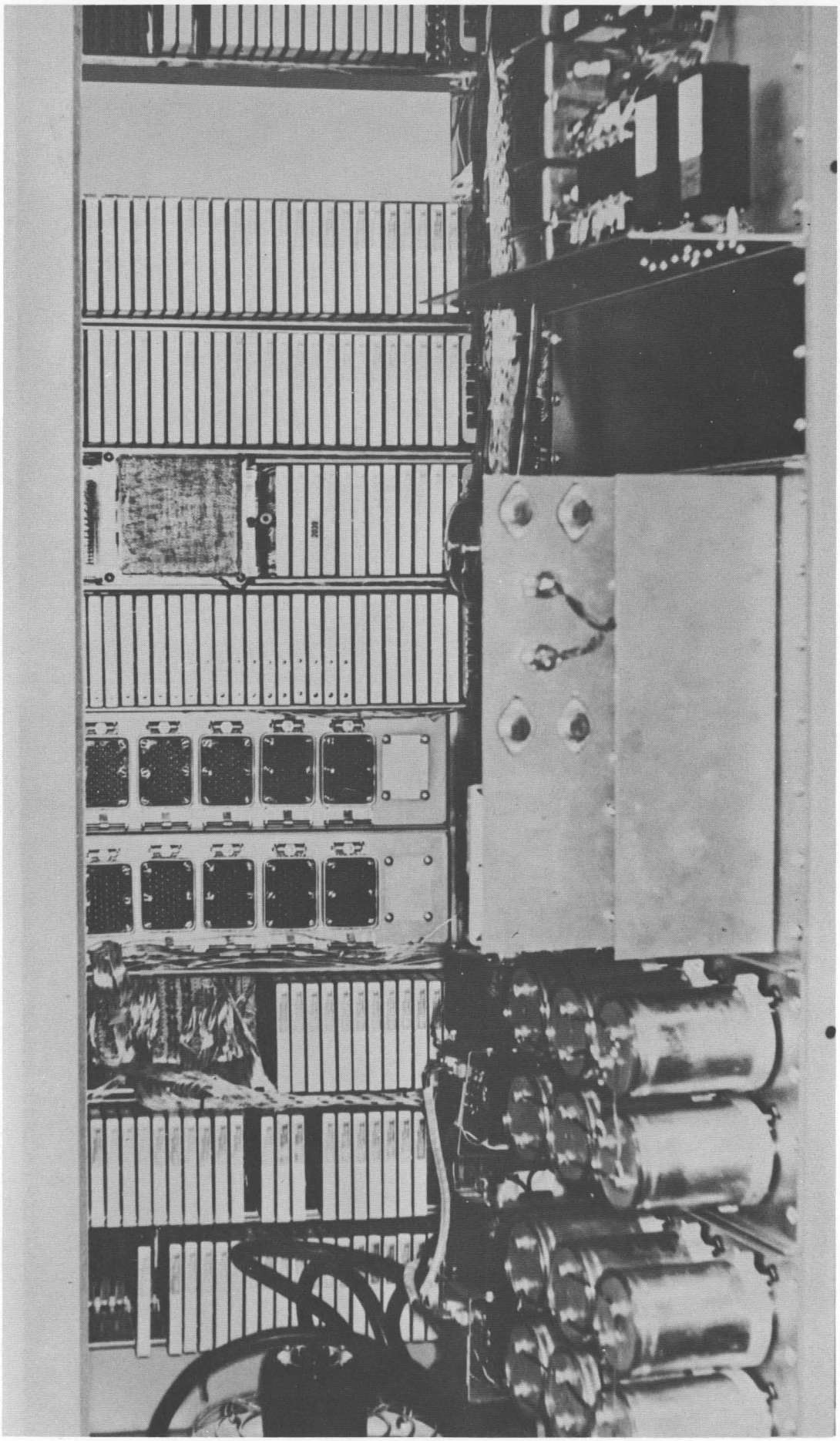


Figure 2-5 Back View of Mounting Panels, Bay 3

Figure 2-5

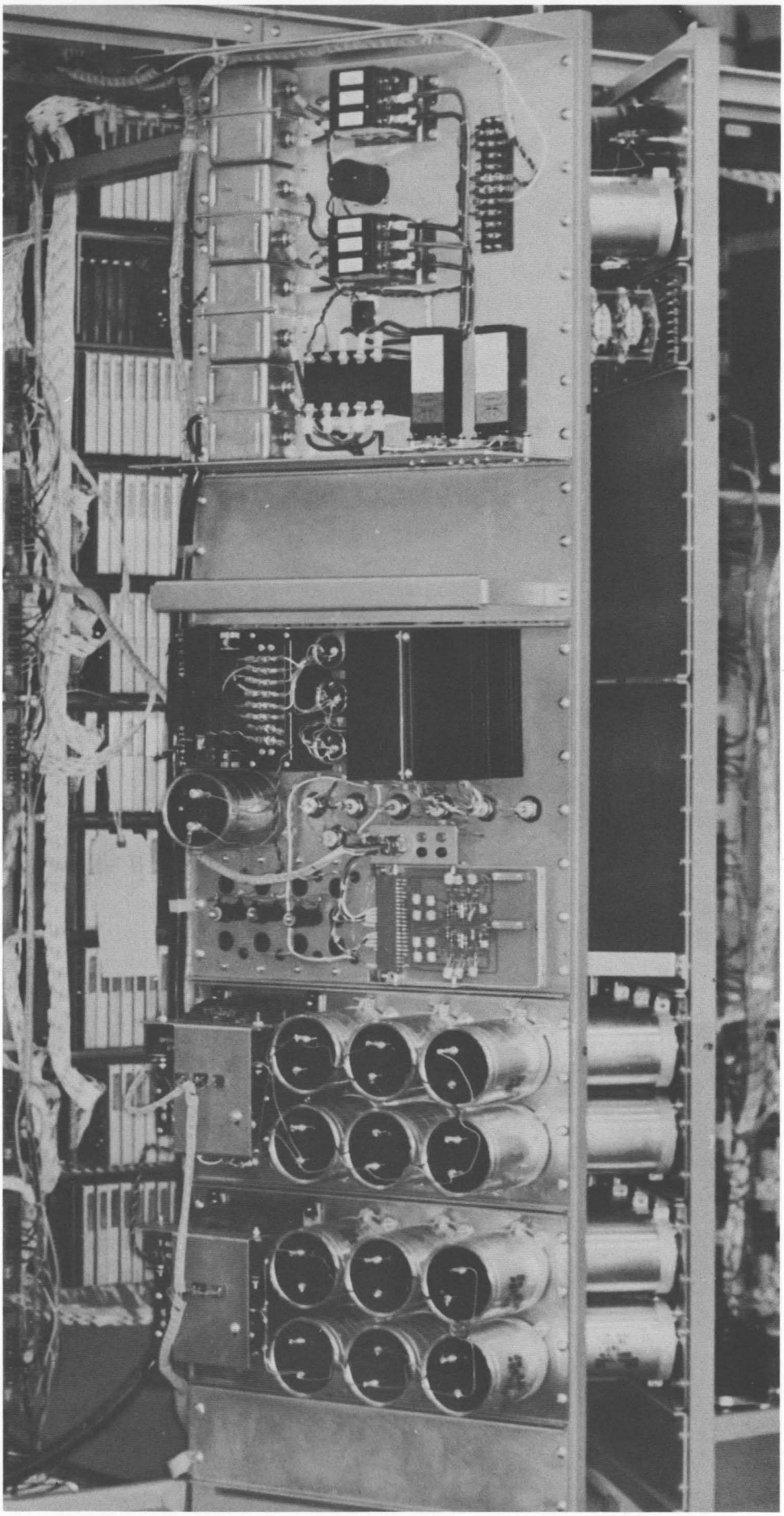


Figure 2-6 Inside of Plenum Door, Bay 3

Figure 2-6

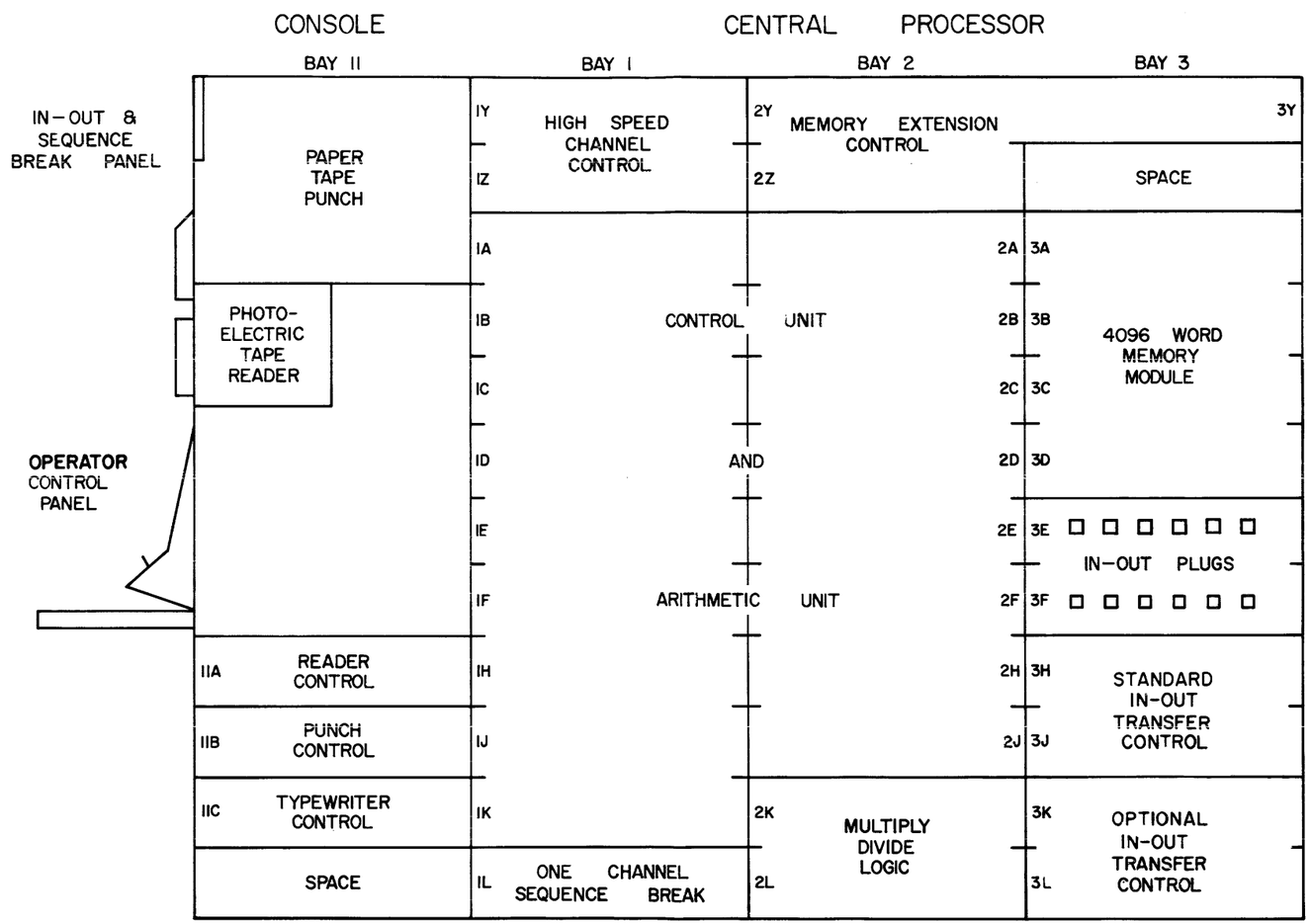


Figure 2-7 Logic Layout, Control Frame

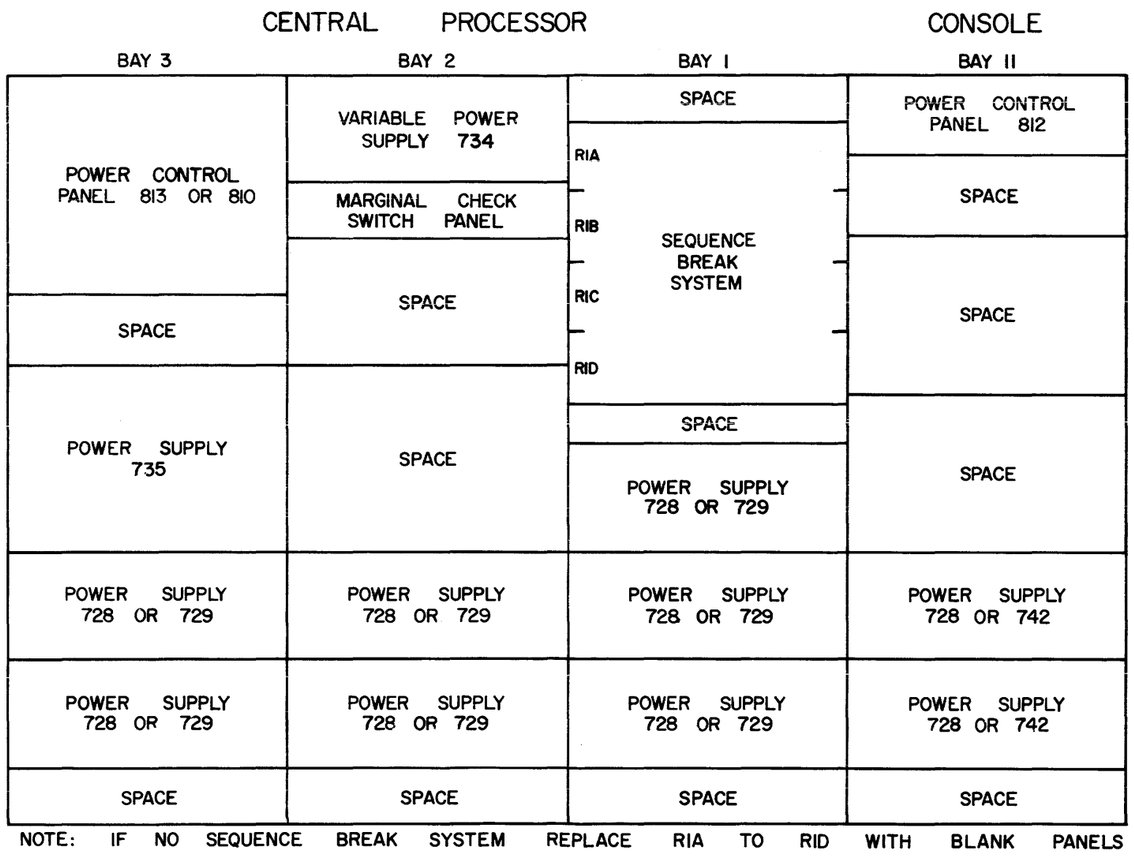


Figure 2-8 Plenum Door Layout, Central Frame

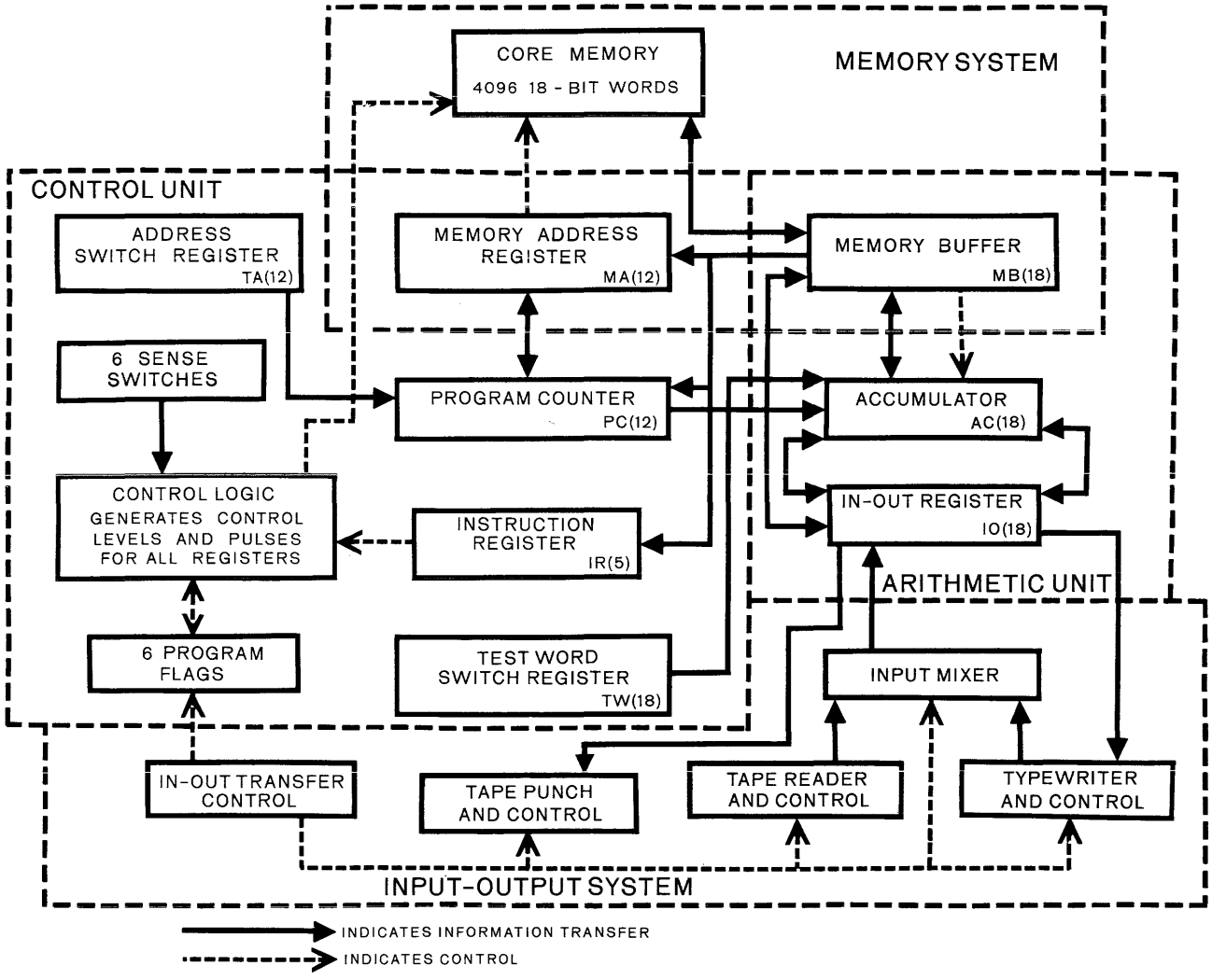


Figure 3-1 PDP-1 Logical Organization

Figure 3-1

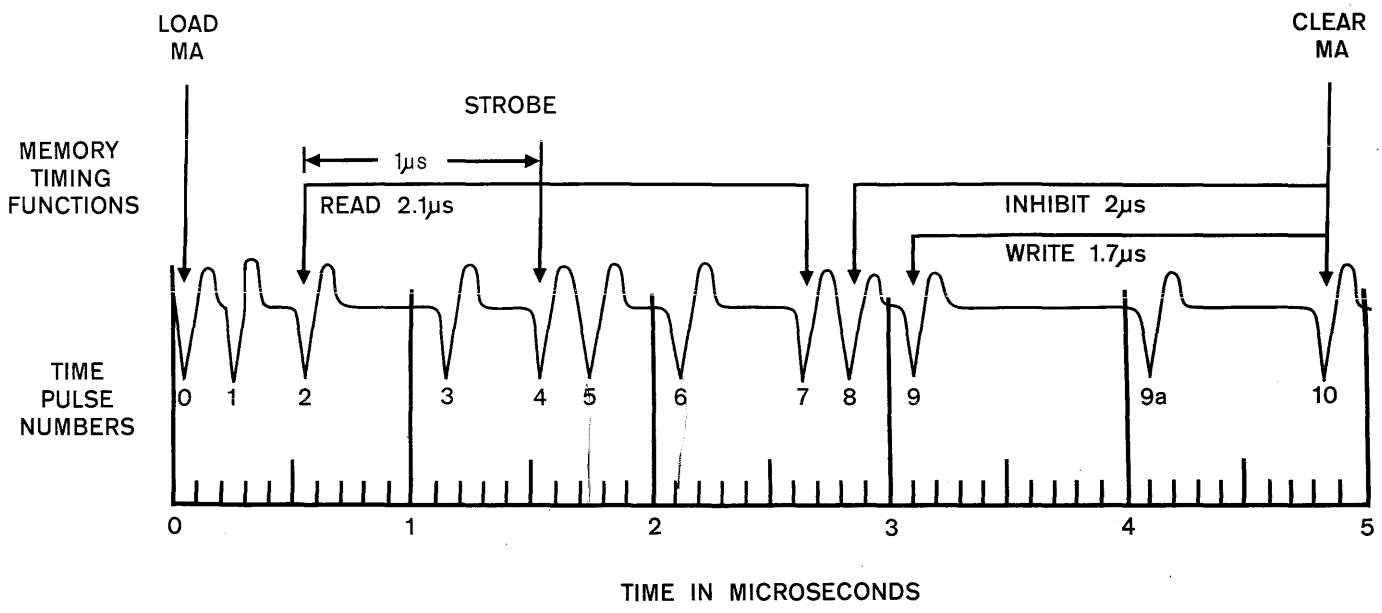


Figure 3-2 Computer Memory Cycle

Figure 3-2

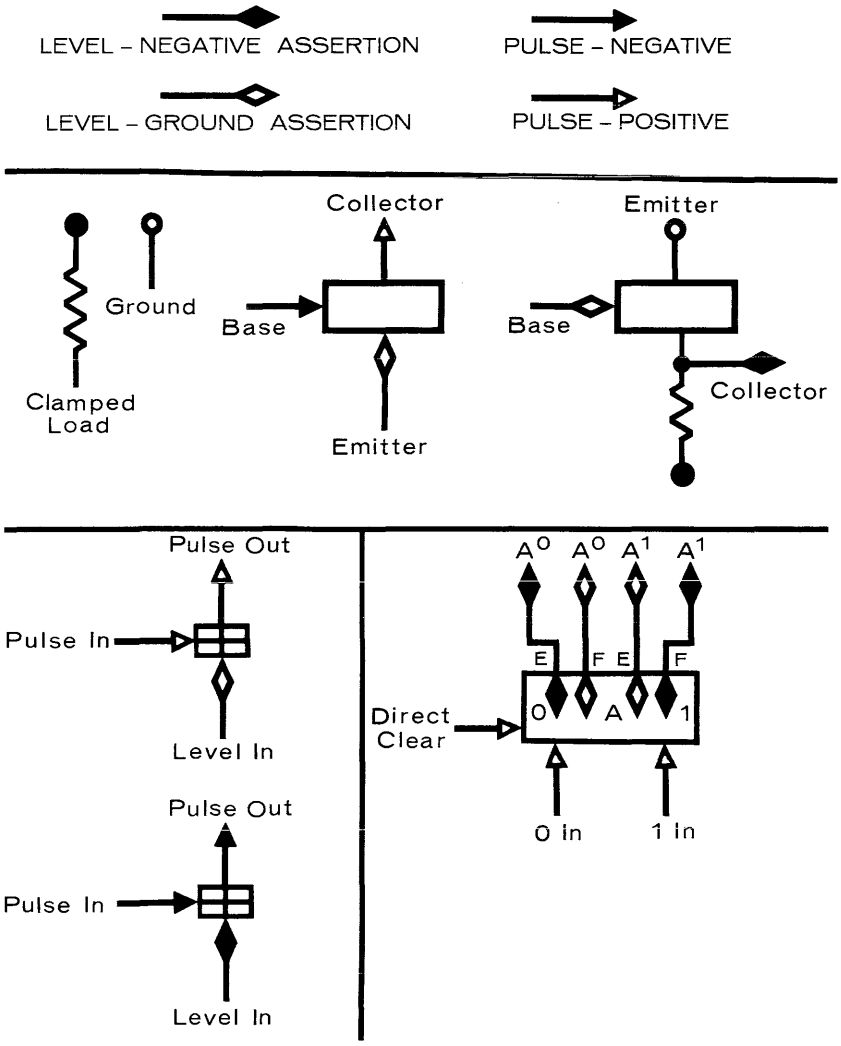


Figure 3-3 PDP-1 Logic Symbols

Figure 3-3

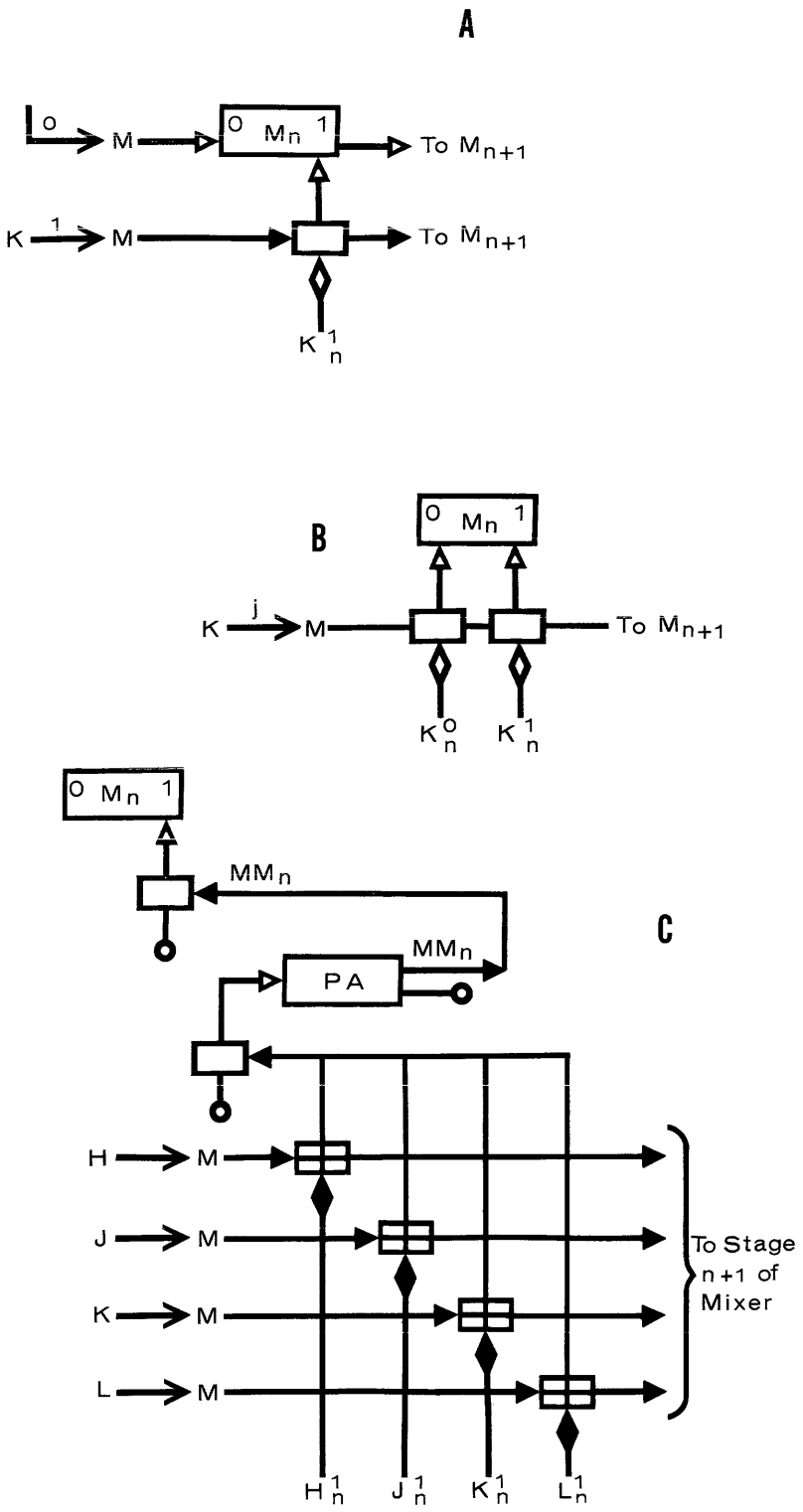


Figure 3-4 Transfer Logic

Figure 3-4

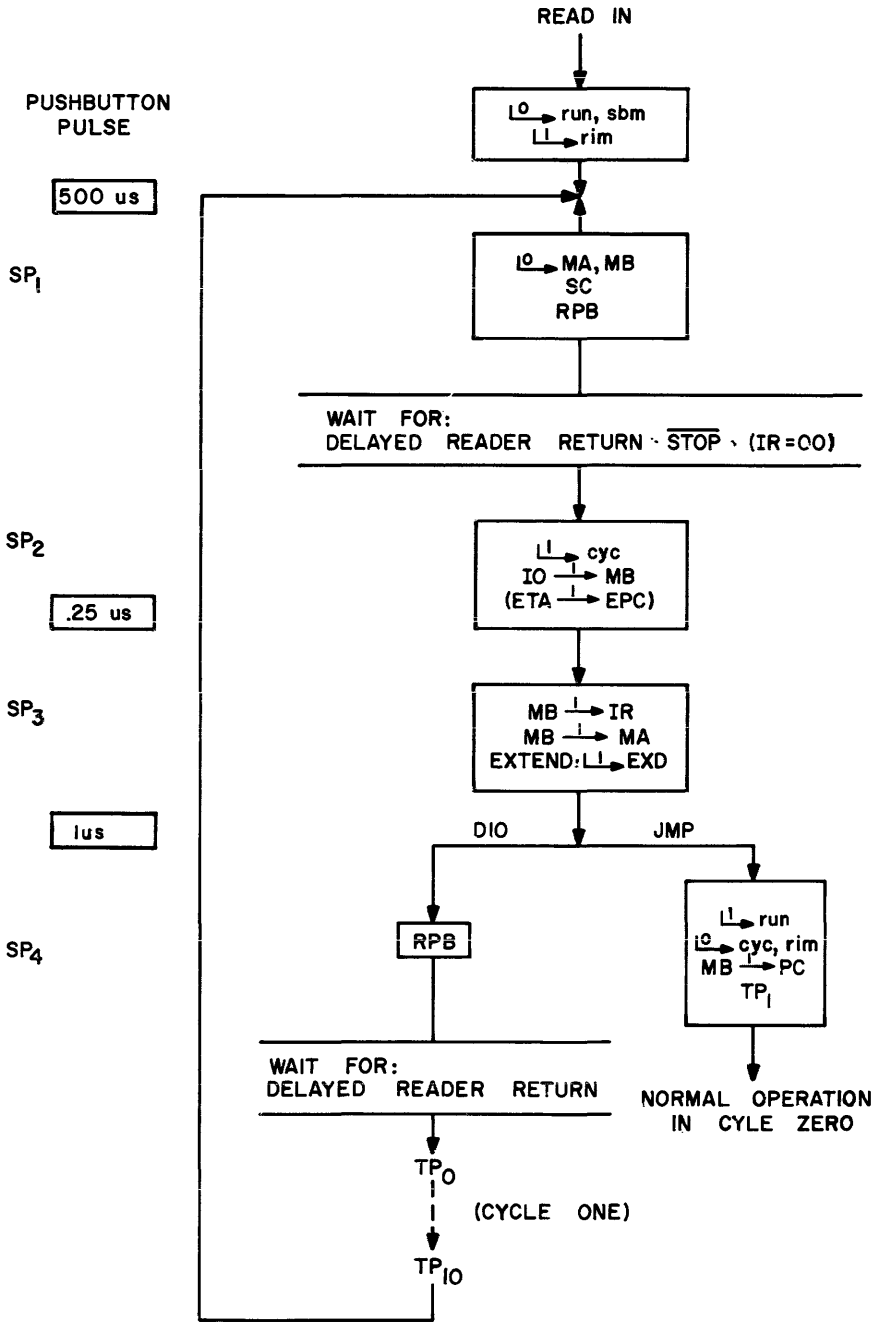


Figure 3-5 Flow Chart Read In

Figure 3-5

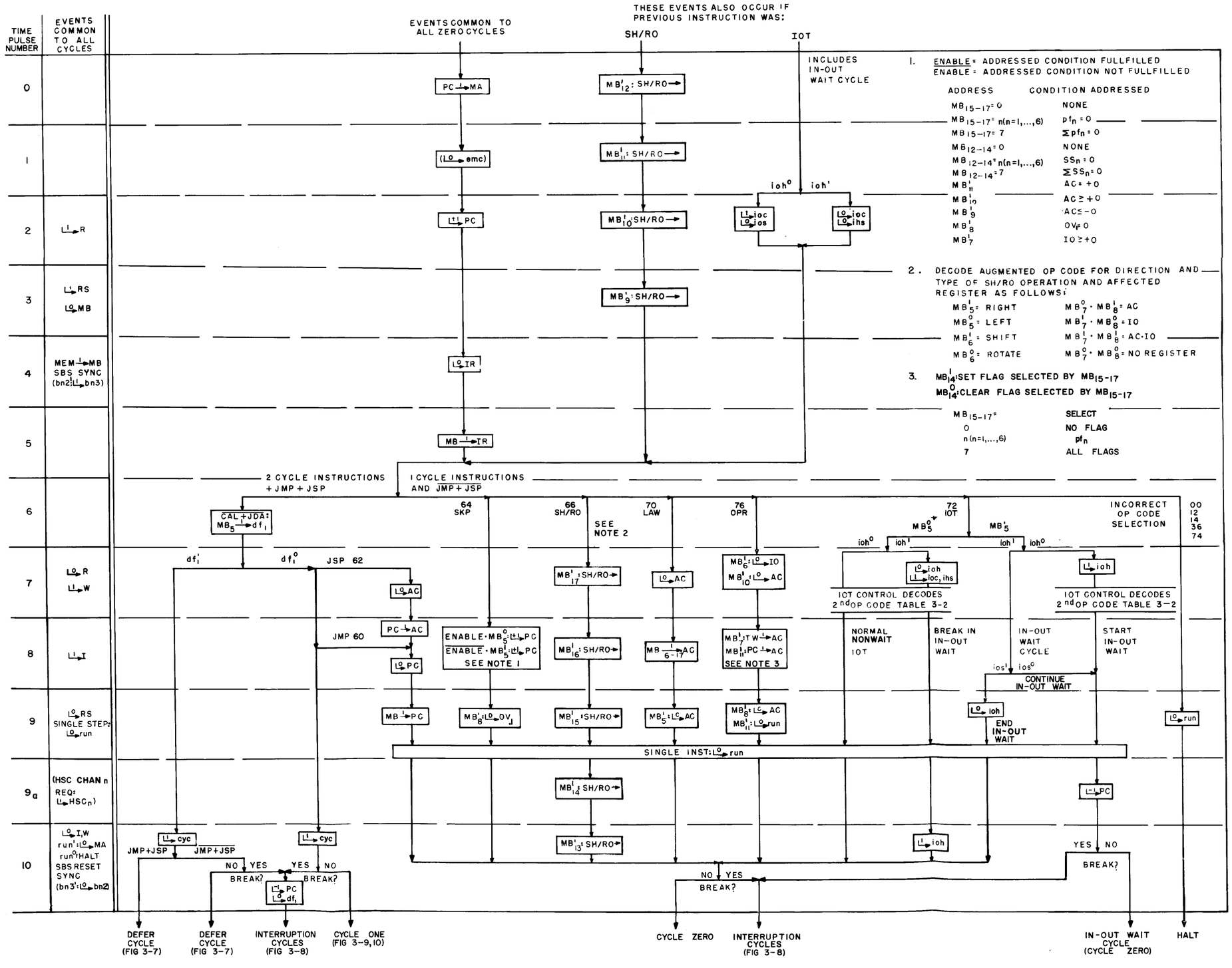


Figure 3-6 Flow Chart, Cycle Zero

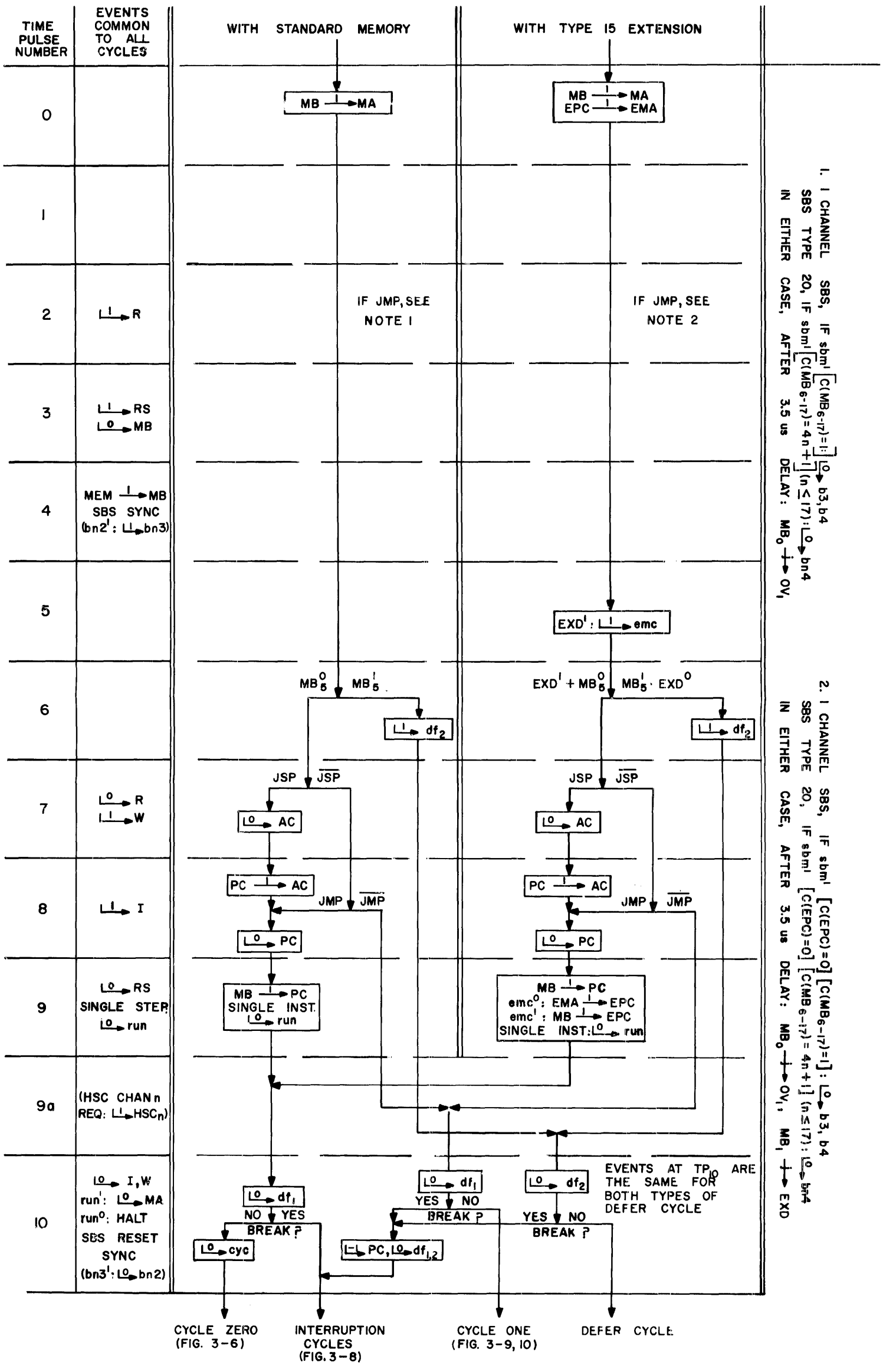


Figure 3-7 Flow Chart, Defer Cycle

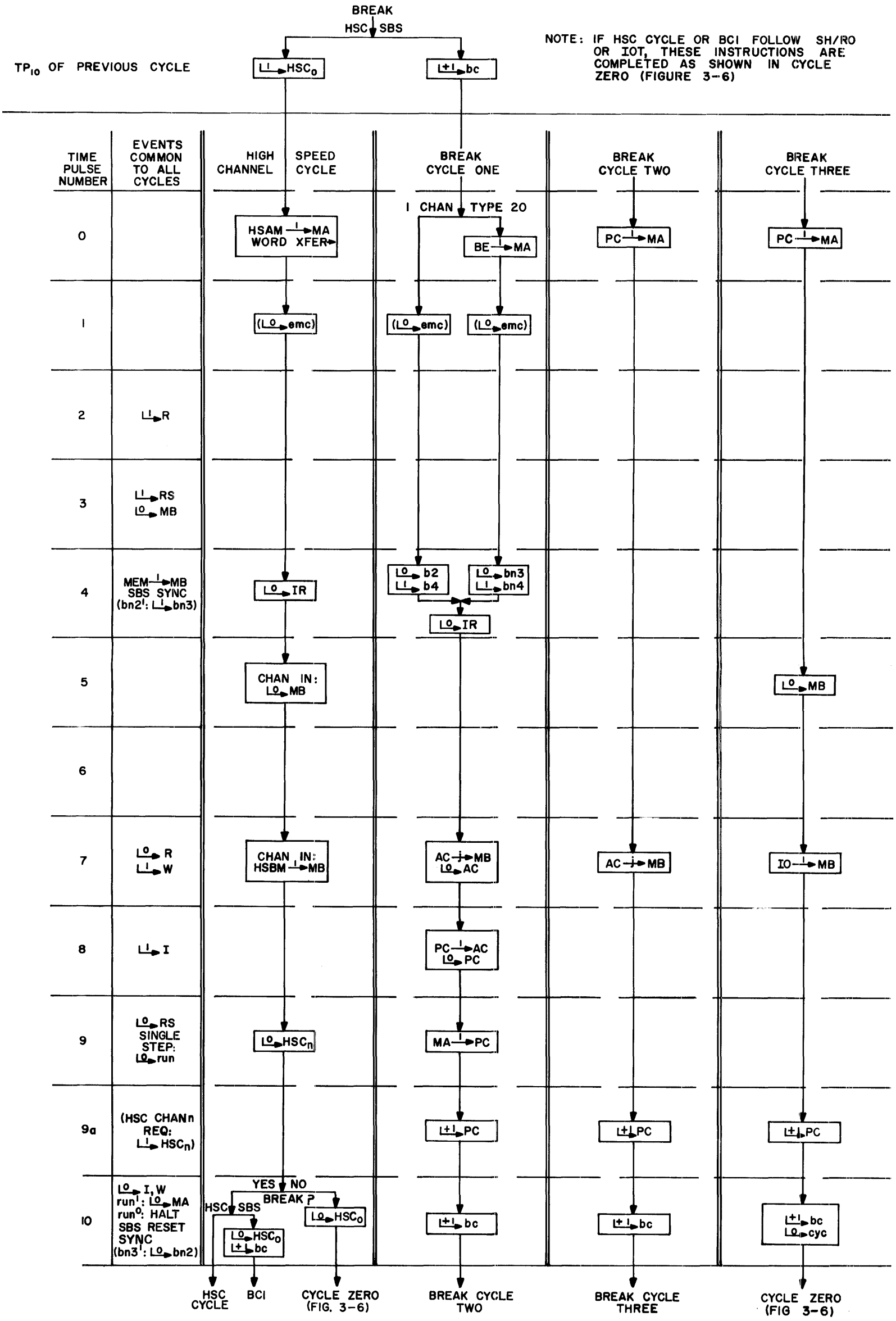


Figure 3-8 Flow Chart, Interruption Cycle

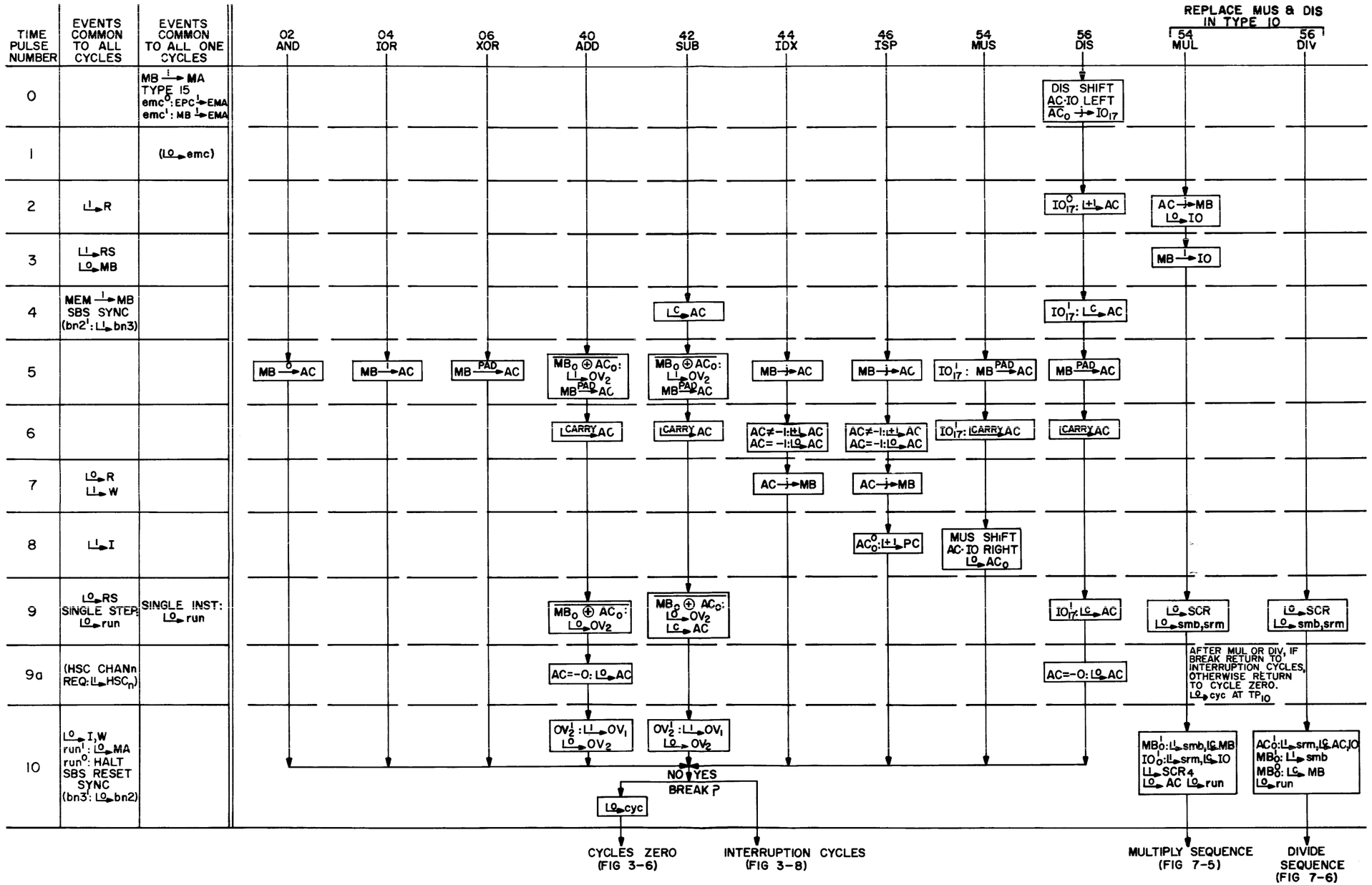


Figure 3-9 Flow Chart: Cycle One, Logical and Arithmetic Instructions

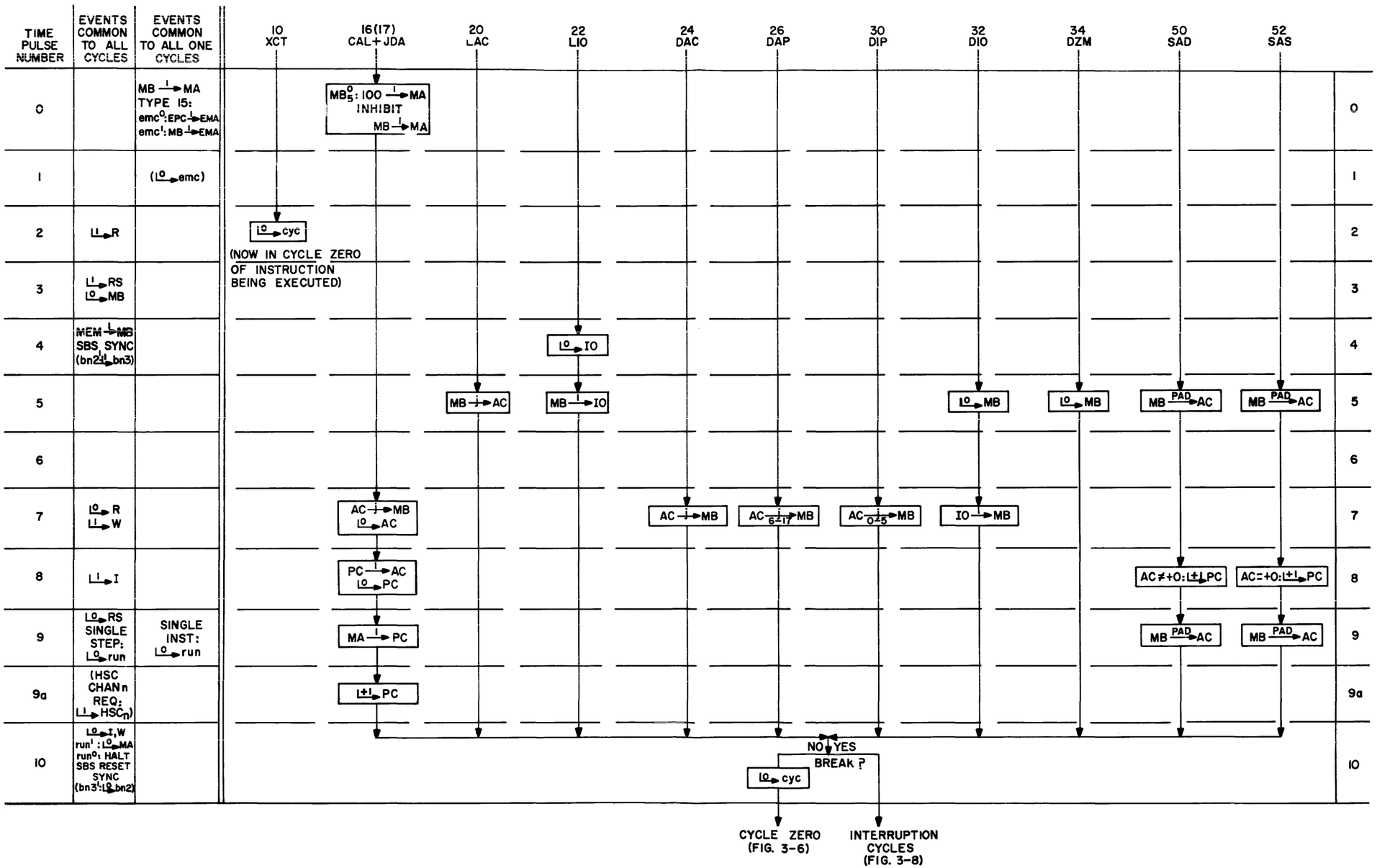


Figure 3-10 Flow Chart: Cycle One, Data Handling and Program Control Instructions

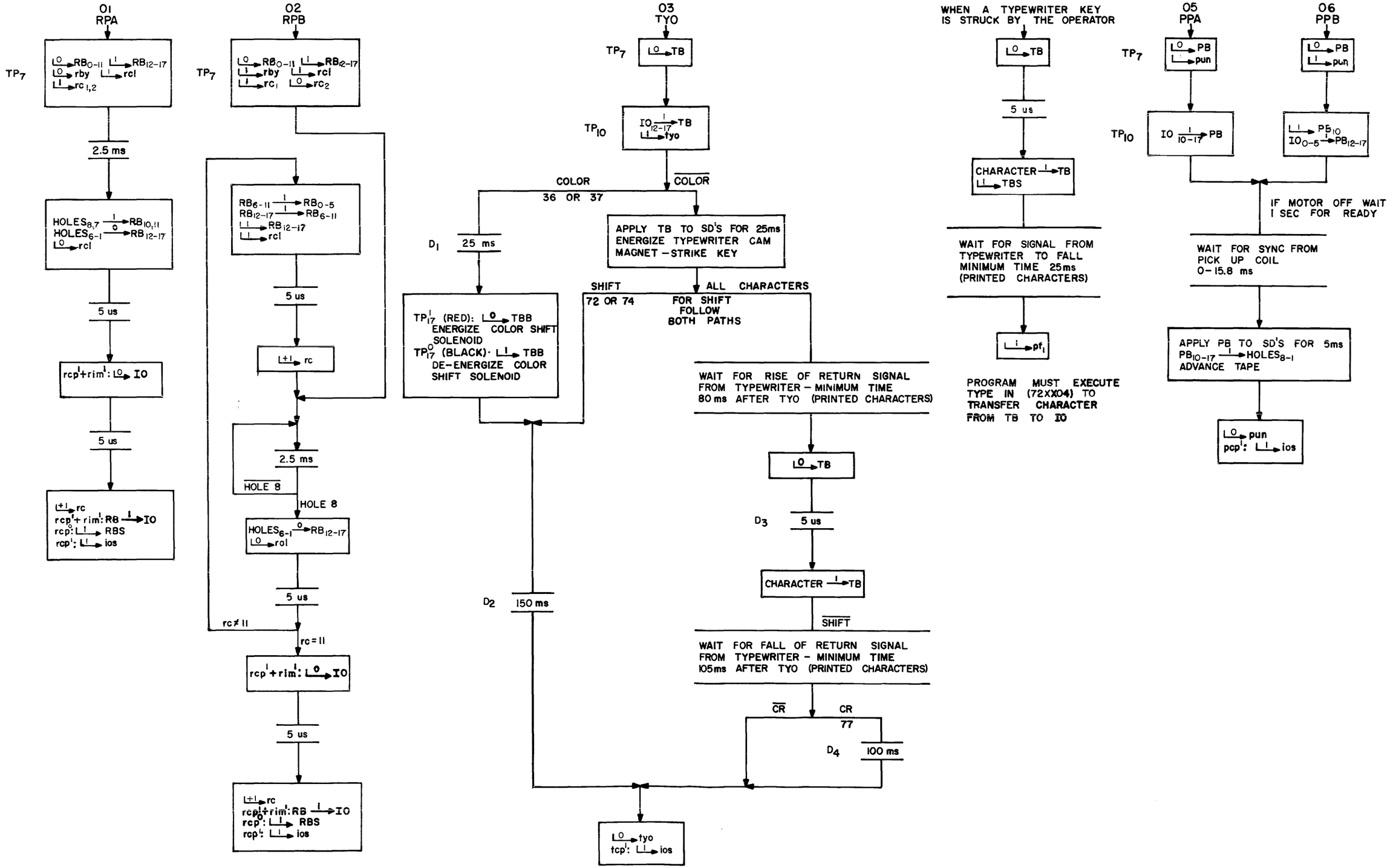


Figure 3-11 Flow Chart: Standard Input-Output Operations

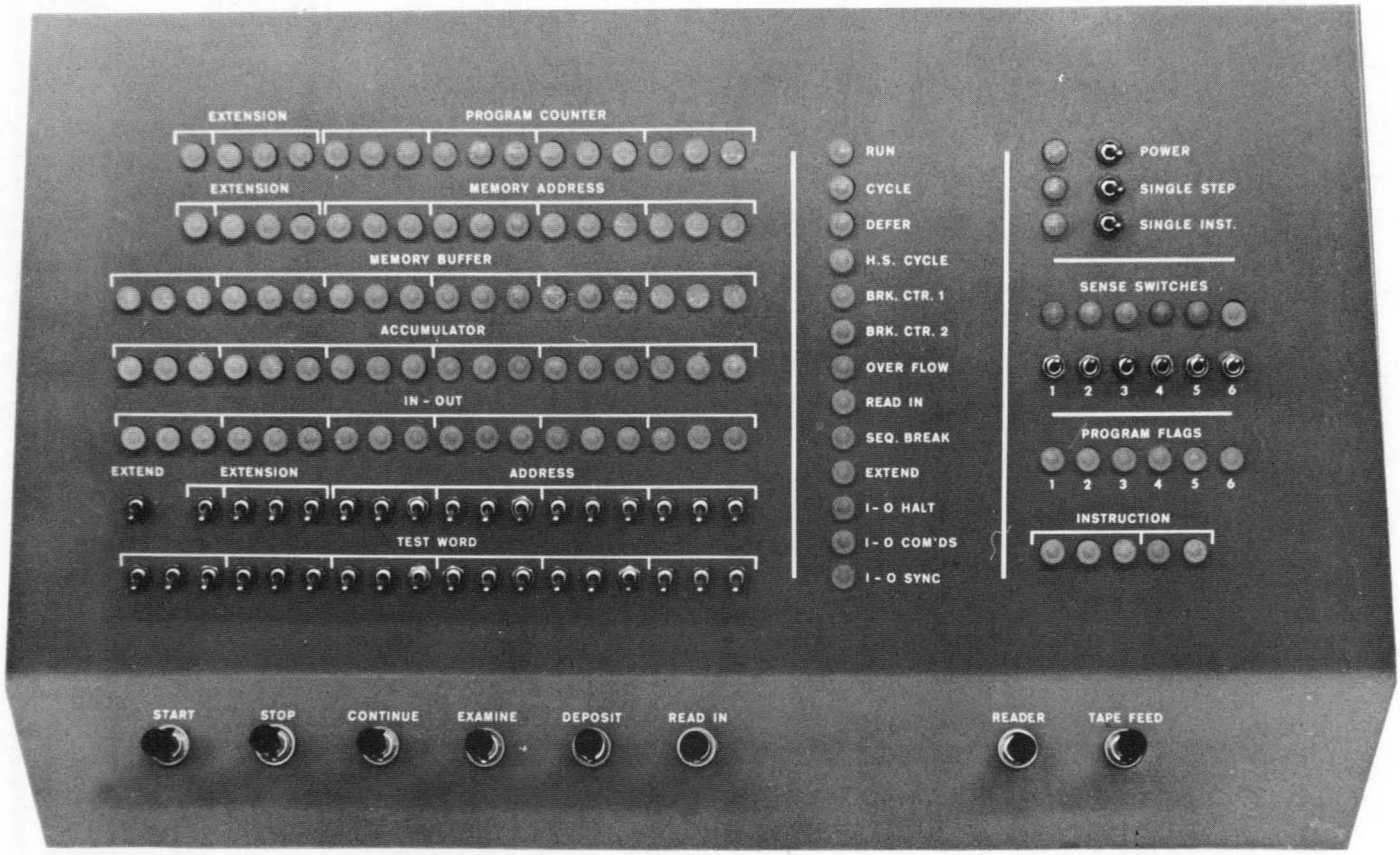


Figure 5-1 Operator Control Panel

Figure 5-1

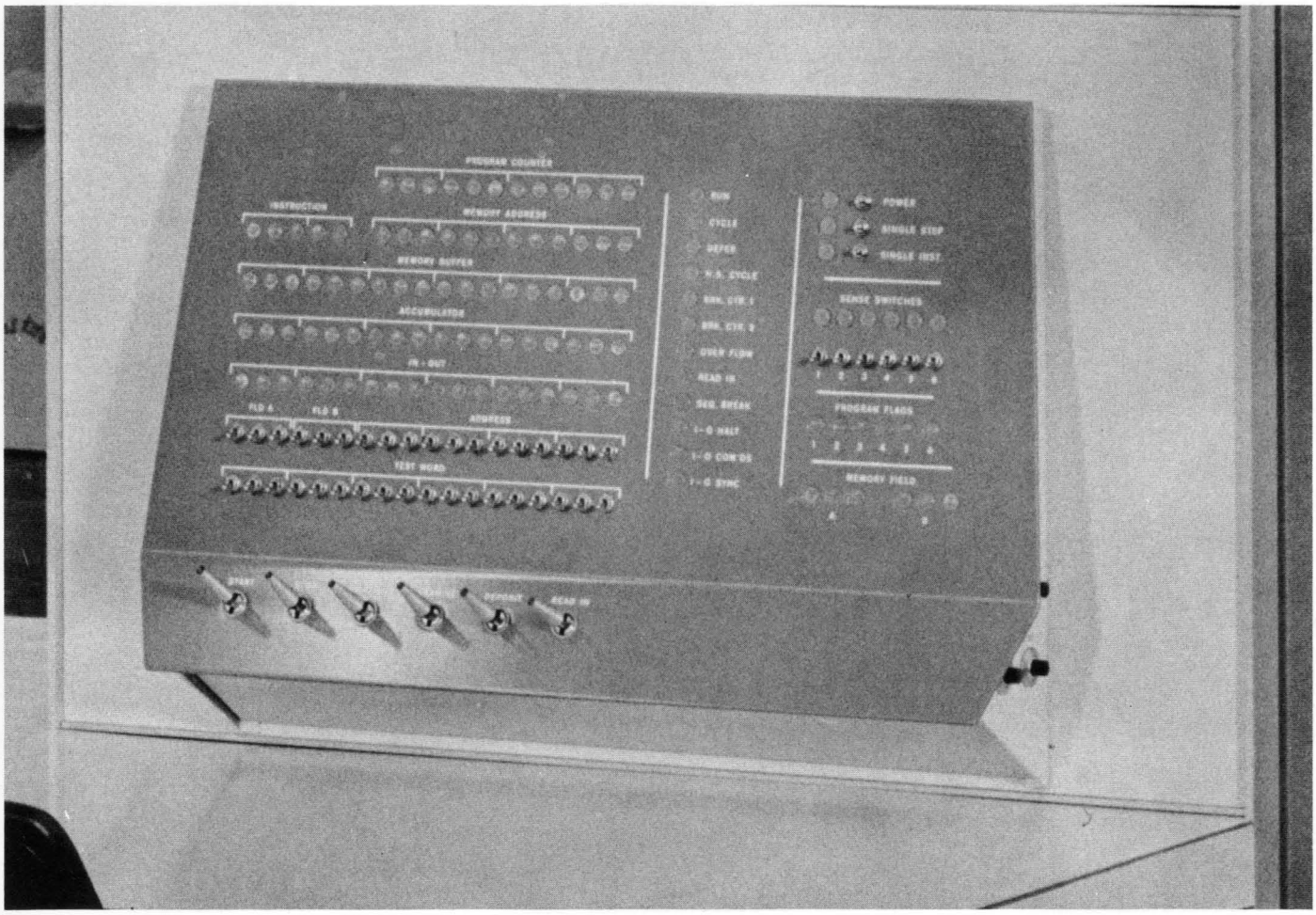


Figure 5-2 Operator Control Panel, Front and Side Views

Figure 5-2

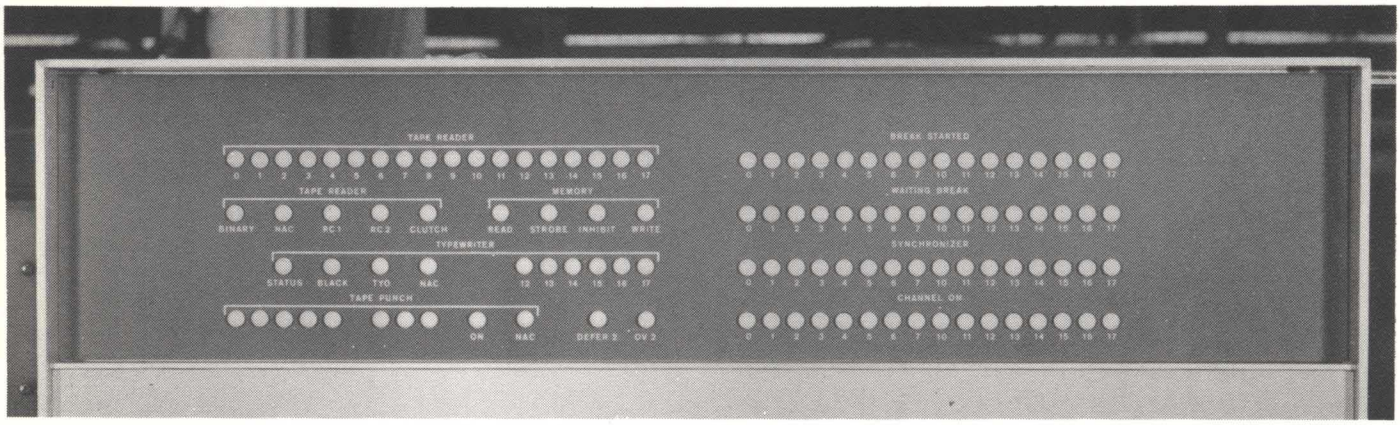


Figure 5-3 In-Out Sequence Break Indicator Panel

Figure 5-3

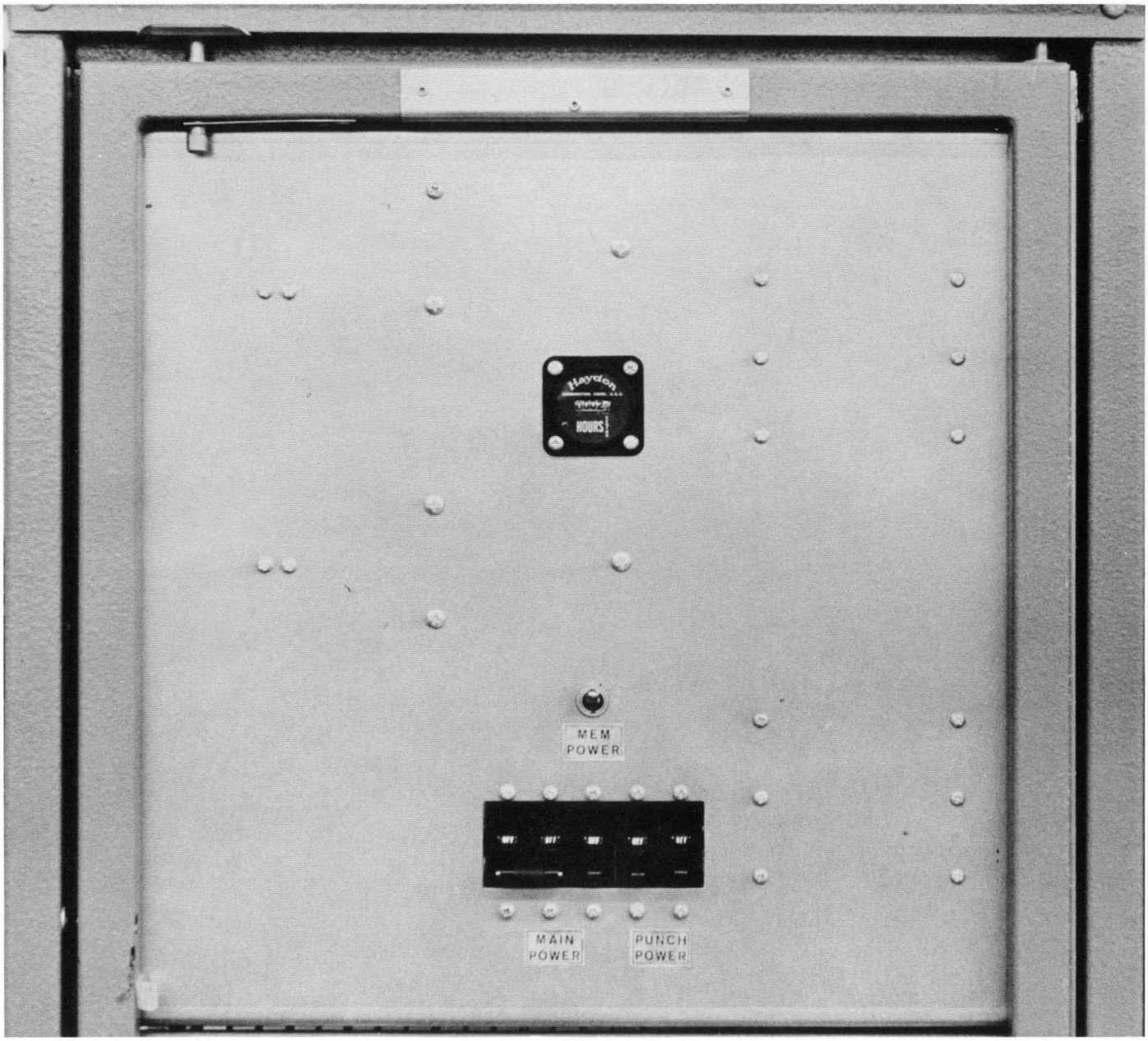


Figure 5-4 Power Control Panel Type 813

Figure 5-4

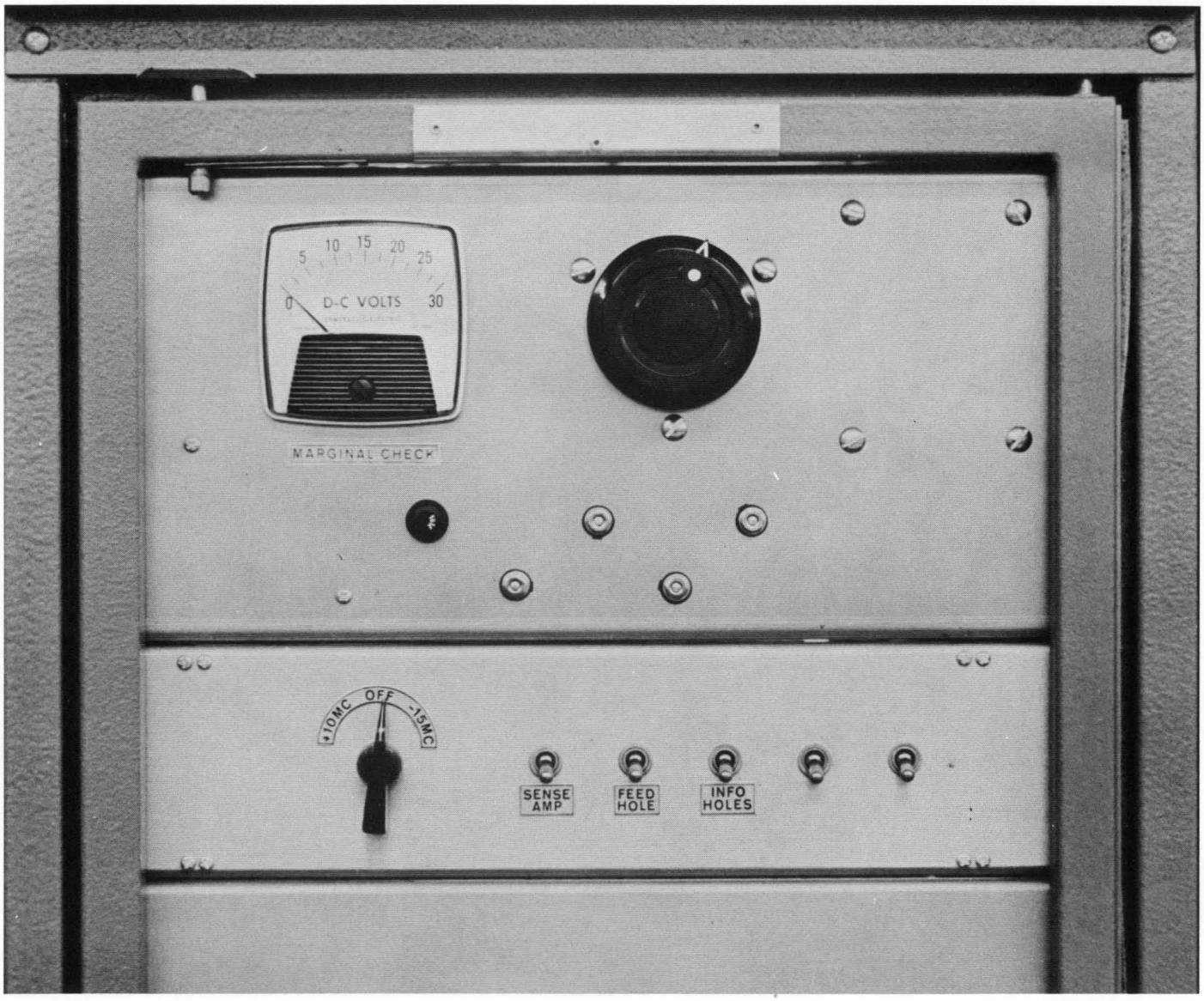


Figure 5-5 Variable Power Supply Type 734 and Marginal Check Switch Panel

Figure 5-5

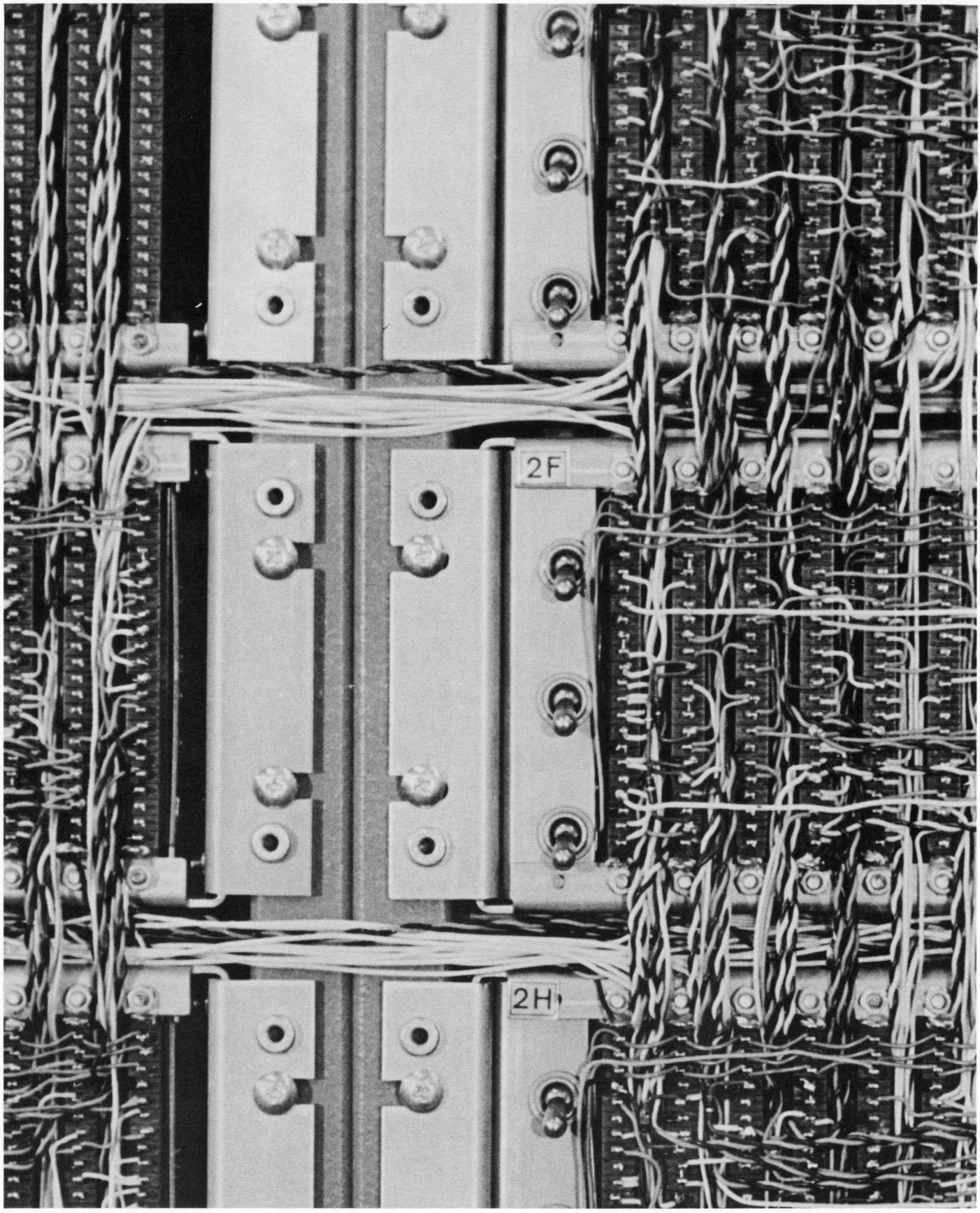


Figure 5-6 Marginal Check Toggle Switches on Mounting Panel

Figure 5-6

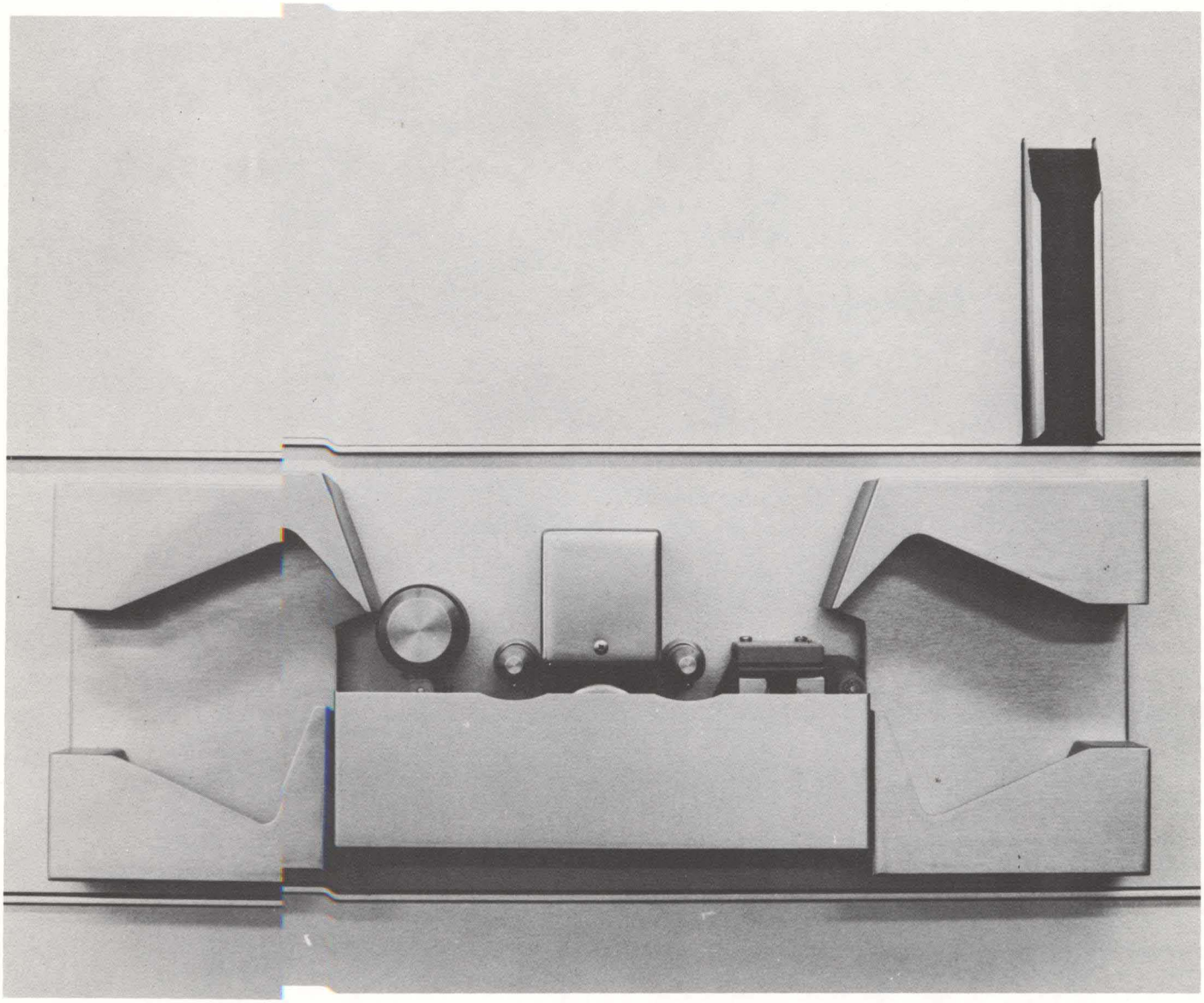


Figure 5-7 Photoelectric Paper Tape Reader

Figure 5-7

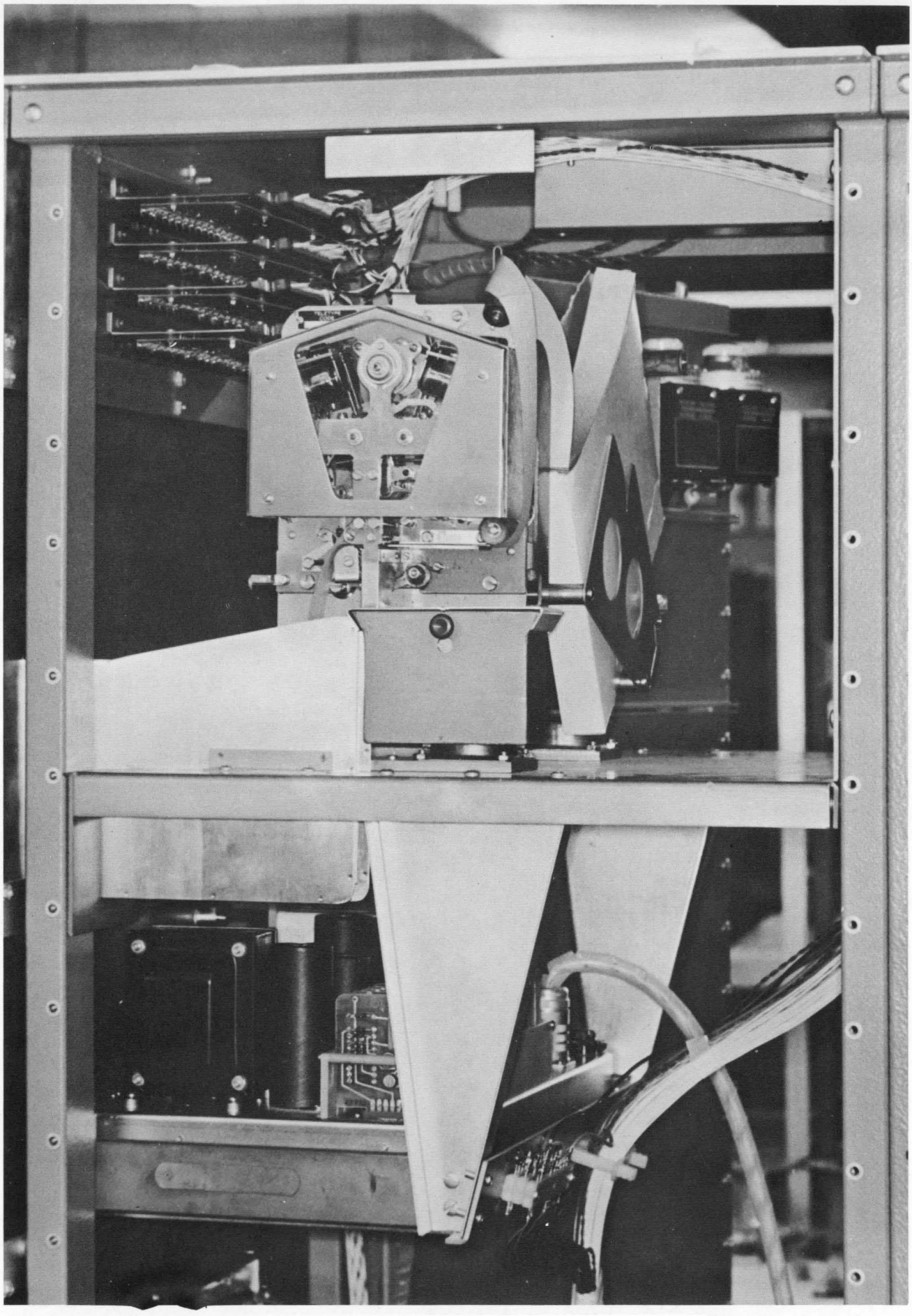


Figure 5-8 Paper Tape Punch, Front View

Figure 5-8

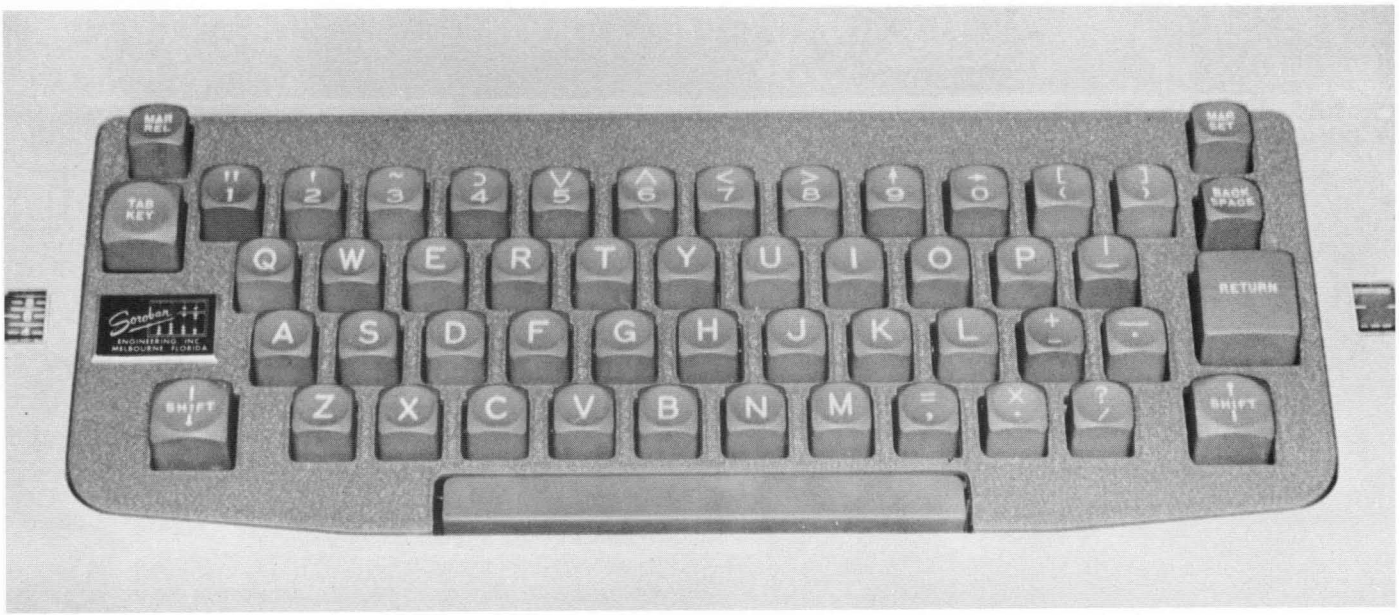
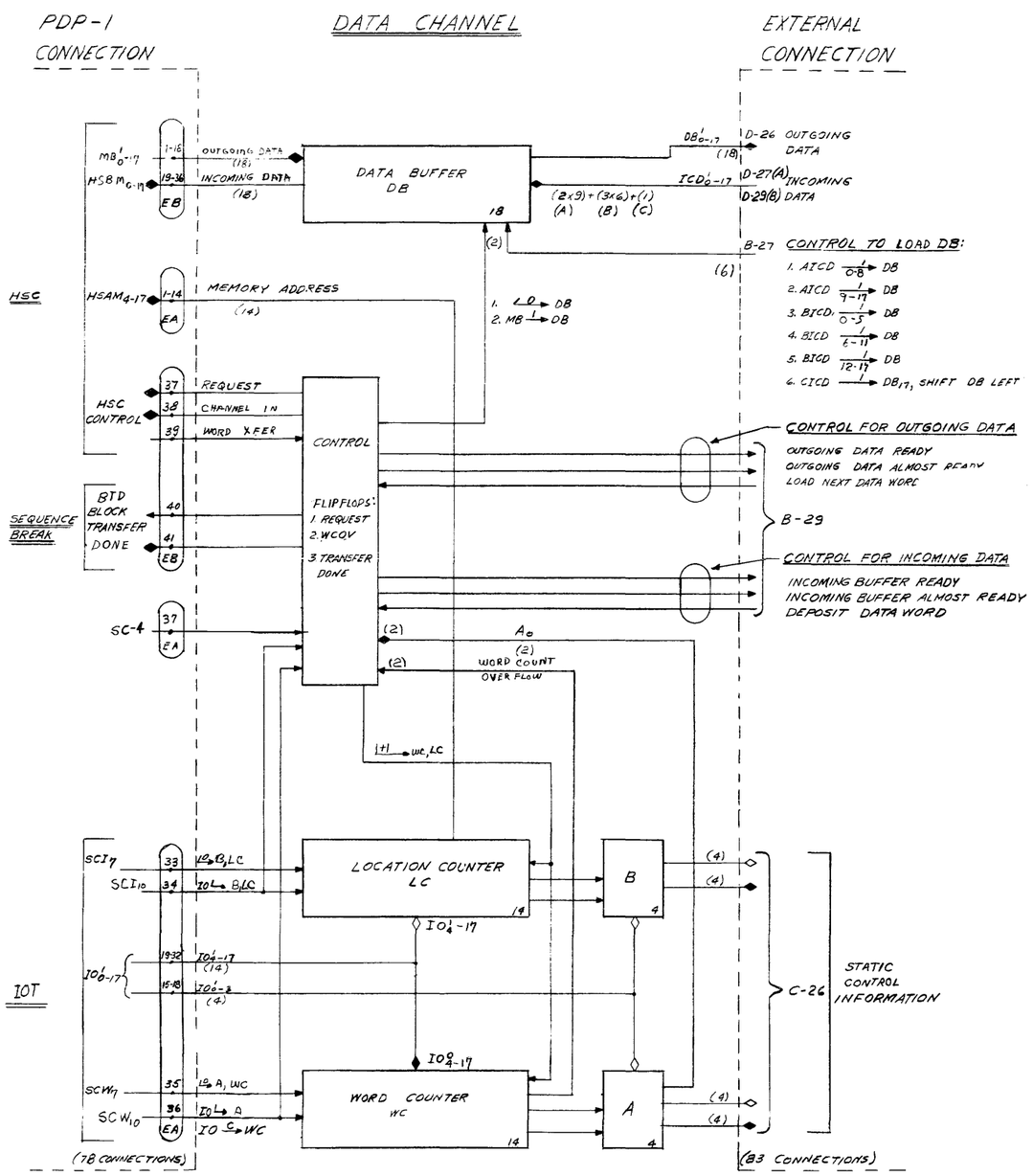


Figure 5-9 Automatic Typewriter Keyboard

Figure 5-9



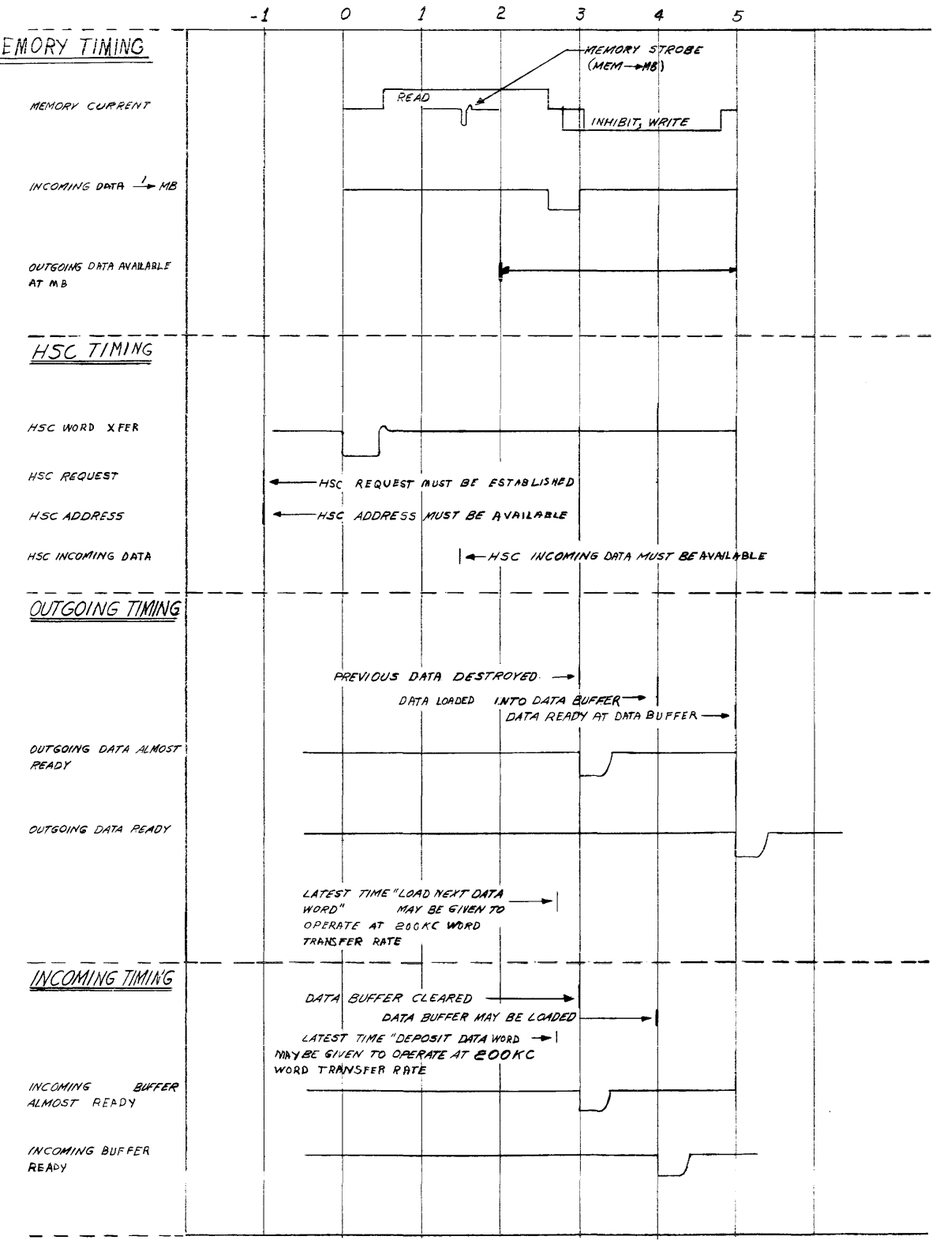


Figure 6-12 Data Channel Type 123, Logical Organization and Timing

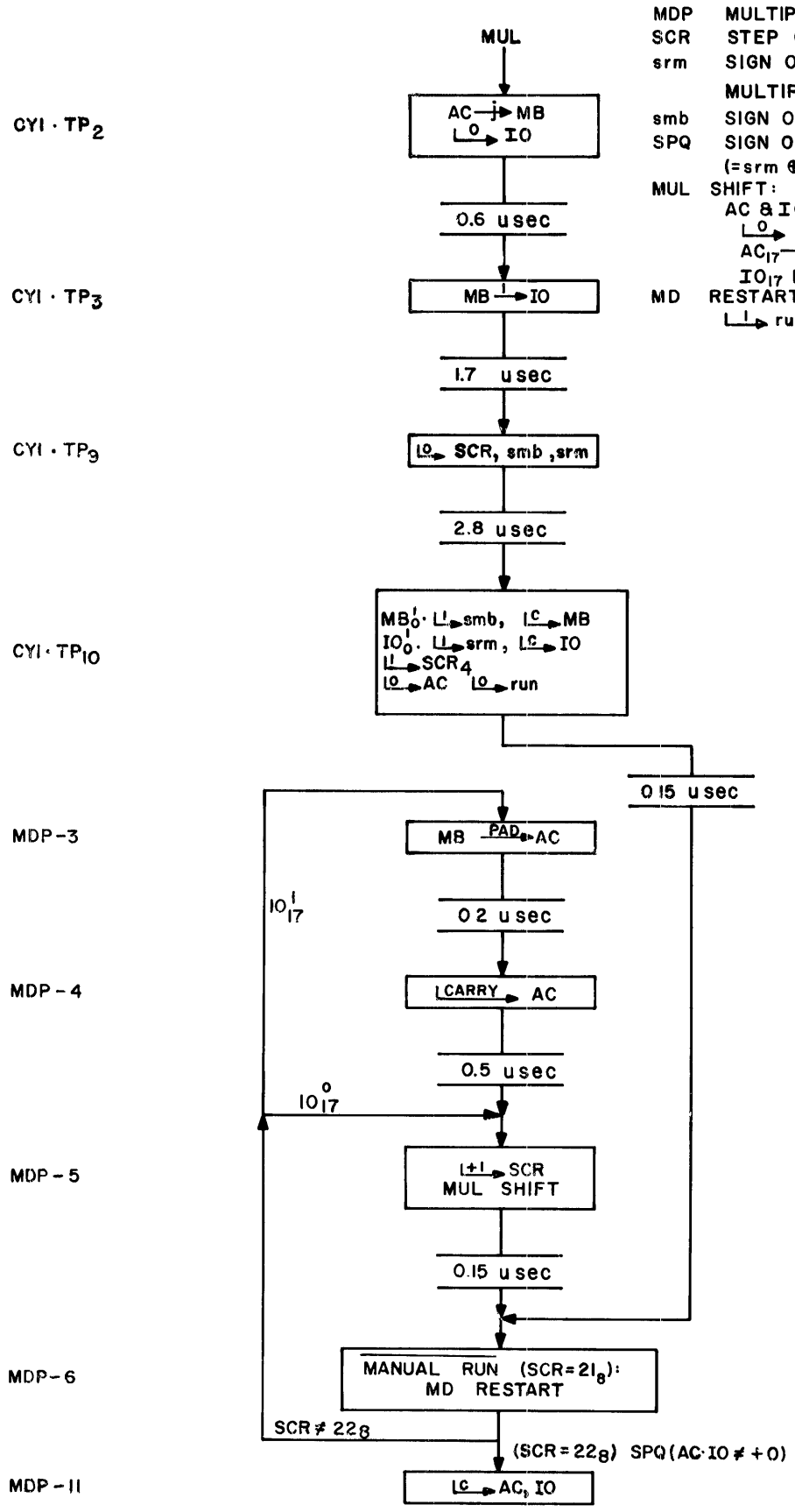


Figure 7-5 Flow Chart, Automatic Multiply

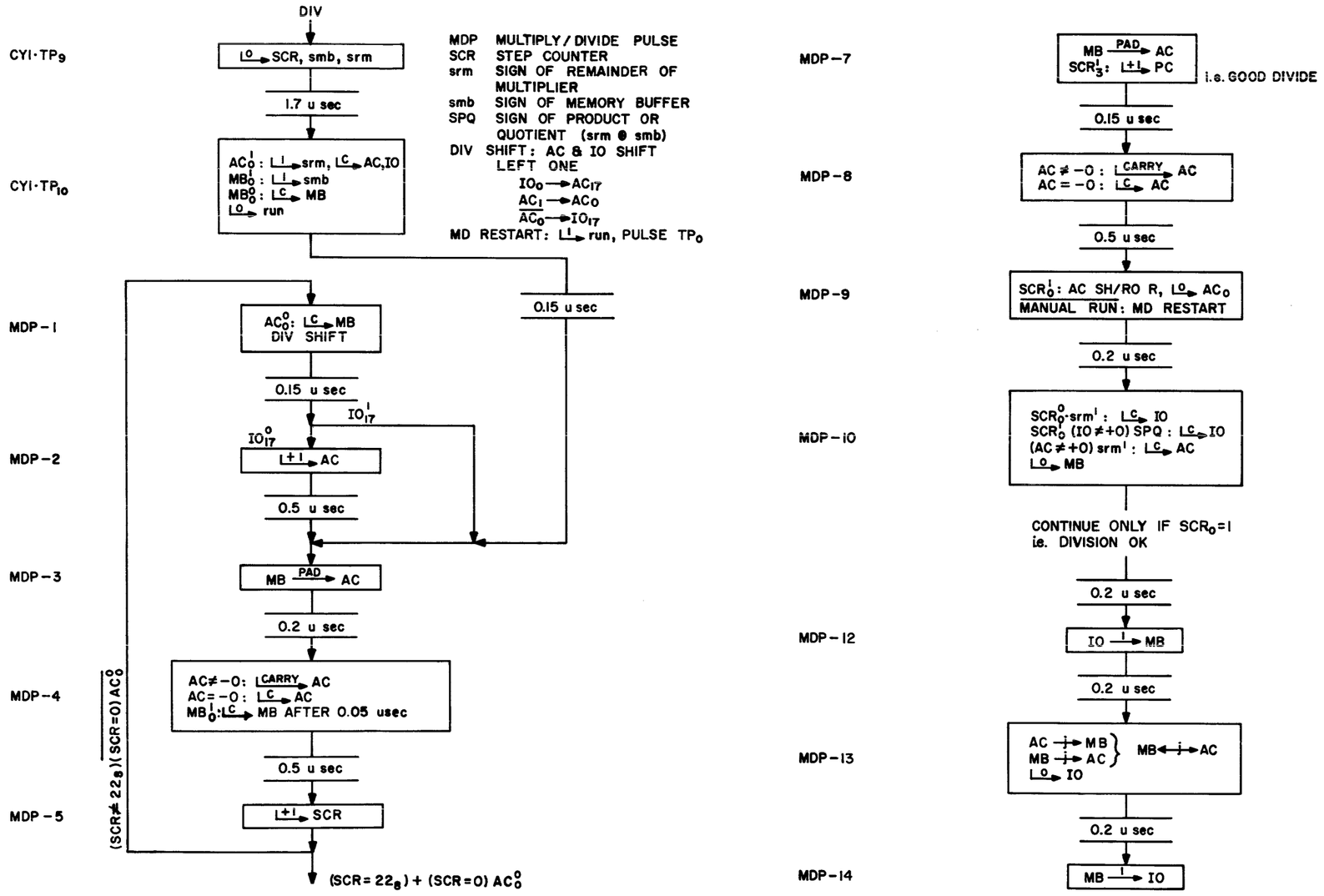
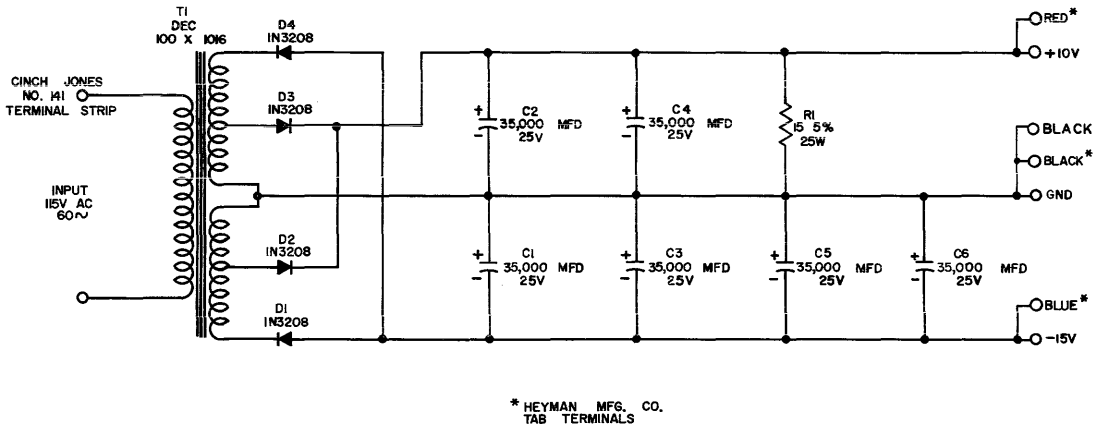


Figure 7-6 Flow Chart, Automatic Divide



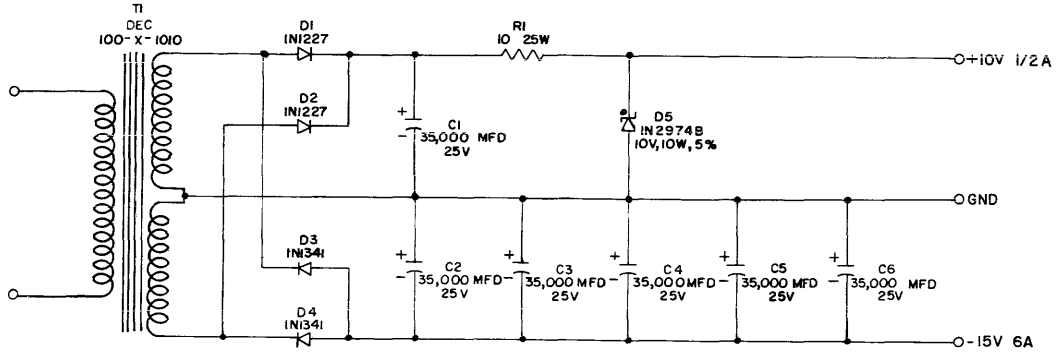
* HEYMAN MFG. CO.
TAB TERMINALS

NOTE
IN ORDER TO KEEP OUTPUT VOLTAGE WITHIN THE FOLLOWING LIMITS
+10V: +9.5 TO +11V
-15 V: -14.5 TO -16V
THE LOADING SHOULD BE WITHIN THE FOLLOWING LIMITS:

BOTH SIDES LOADED	+10V	.0 TO 7.0 AMPS	
	-15V	1.0 TO 8.0 AMPS	
ONE SIDE LOADED	+10V	.0 TO 7.5 AMPS	
	-15V	1.0 TO 8.5 AMPS	

SUM OF THE OUTPUT CURRENTS ARE LIMITED BY THE FOLLOWING EQUATION $5I_{10} + 6I_{15} \leq 53$

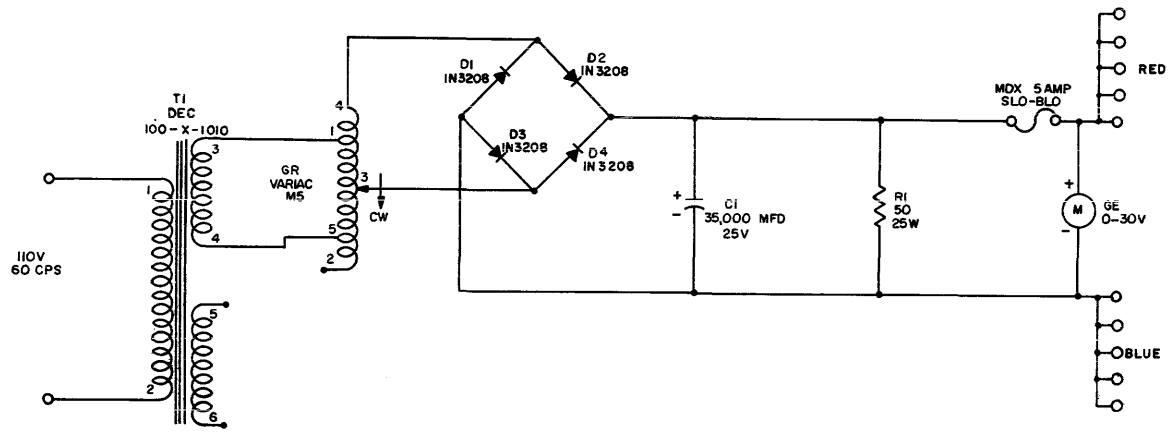
POWER SUPPLY 728



POWER SUPPLY 729

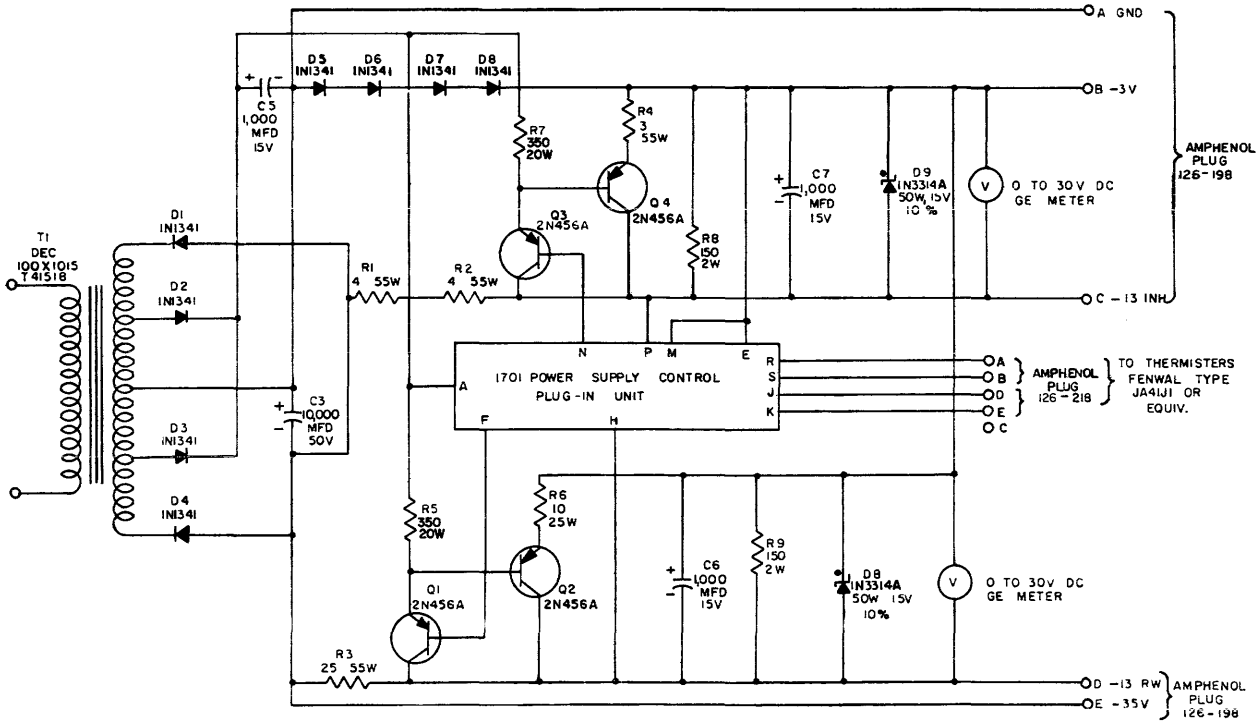
POWER SUPPLY 728

POWER SUPPLY 729



NOTE:
INPUT IS CONNECTED THROUGH JONES NO. 141 TERMINAL STRIP
OUTPUT IS CONNECTED THROUGH HEYMAN TAB TERMINALS

VARIABLE POWER SUPPLY 734

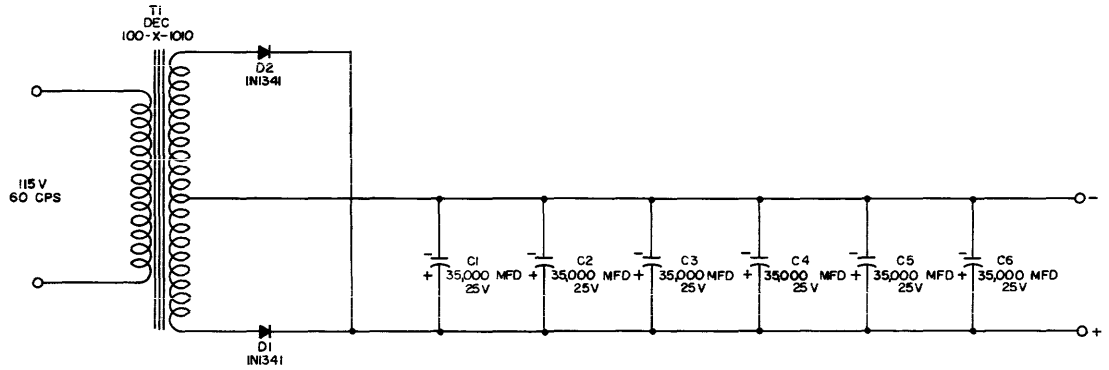


UNLESS OTHERWISE INDICATED:
RESISTORS ARE 10 %

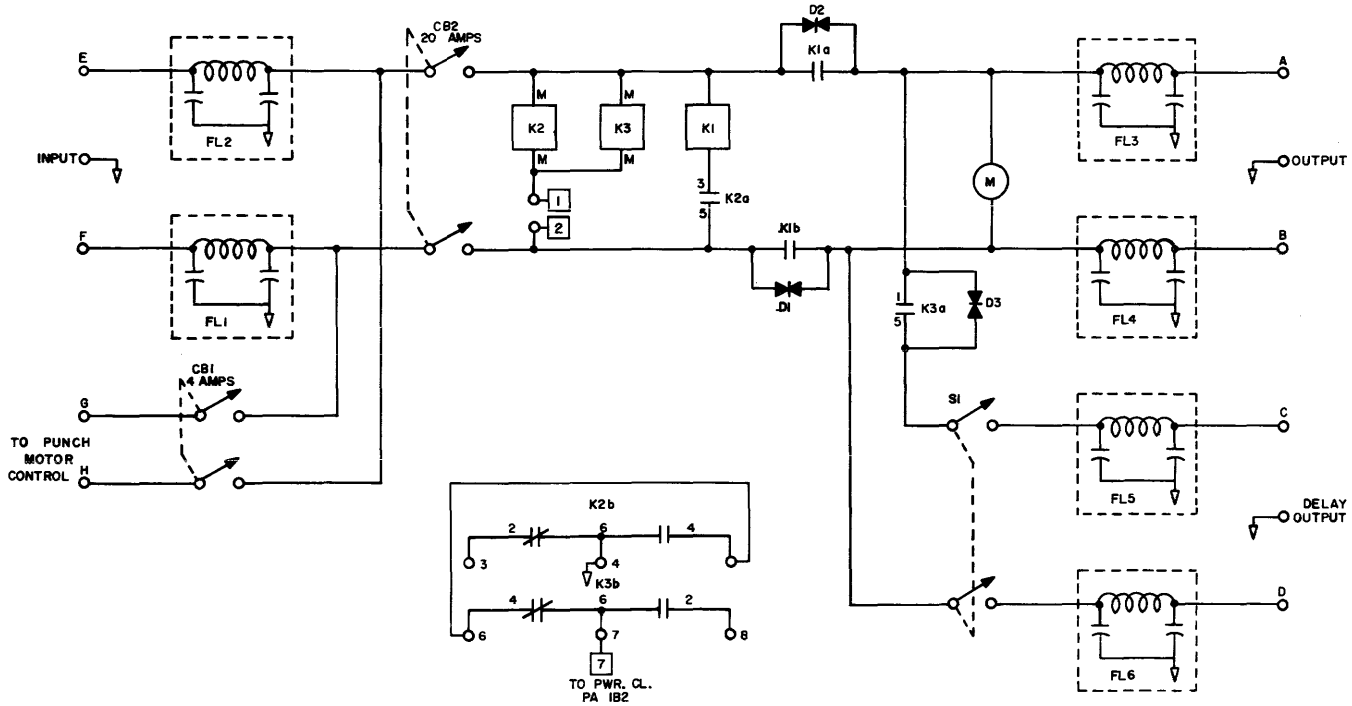
MEMORY POWER SUPPLY 735

VARIABLE POWER SUPPLY 734

MEMORY POWER SUPPLY 735



POWER SUPPLY 742

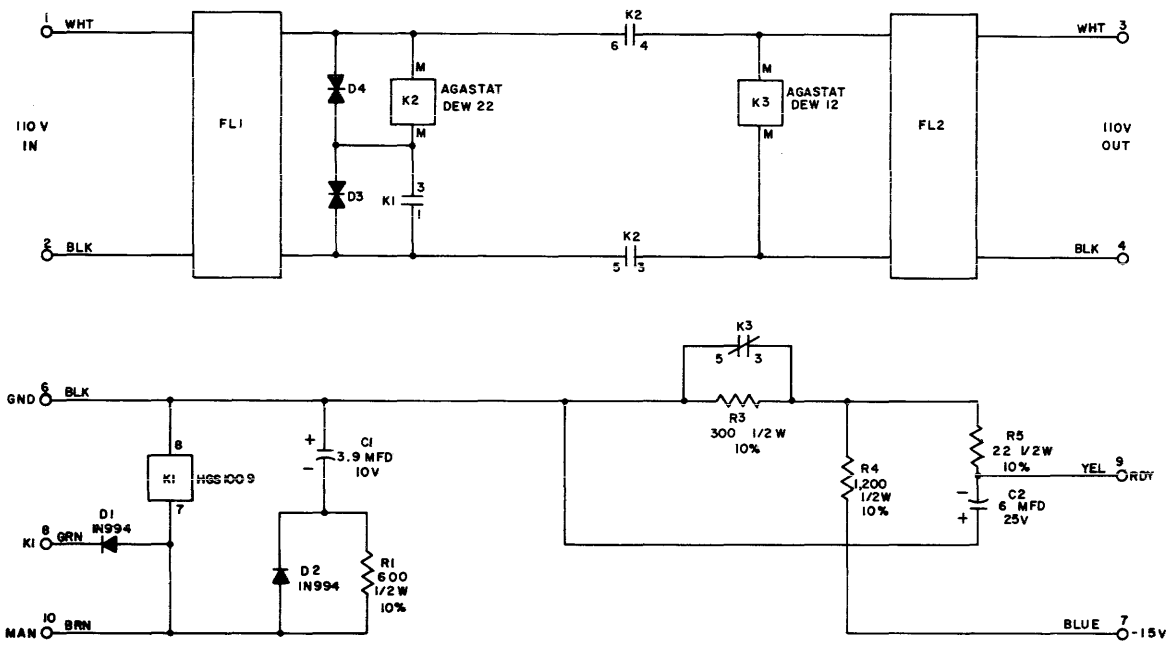


UNLESS OTHERWISE INDICATED
D1,D2 AND D3 ARE GE THYRECTOR DIODE 6RS20SP4B4
S1 IS A DPST CUTLER HAMMER
CB1 AND CB2 ARE XAM-33 HEINEMAN ELECT. CO. CURVE 4
FL1-FL6 SPRAGUE FILTROL #3
K1 IS AN GUARDIAN RELAY 115V AC 60~ 25AMPS
K3 IS AN AGASTAT DELAY TYPE DE-W-12
K2 IS AN AGASTAT DELAY TYPE DE-W-22

POWER CONTROL PANEL 810

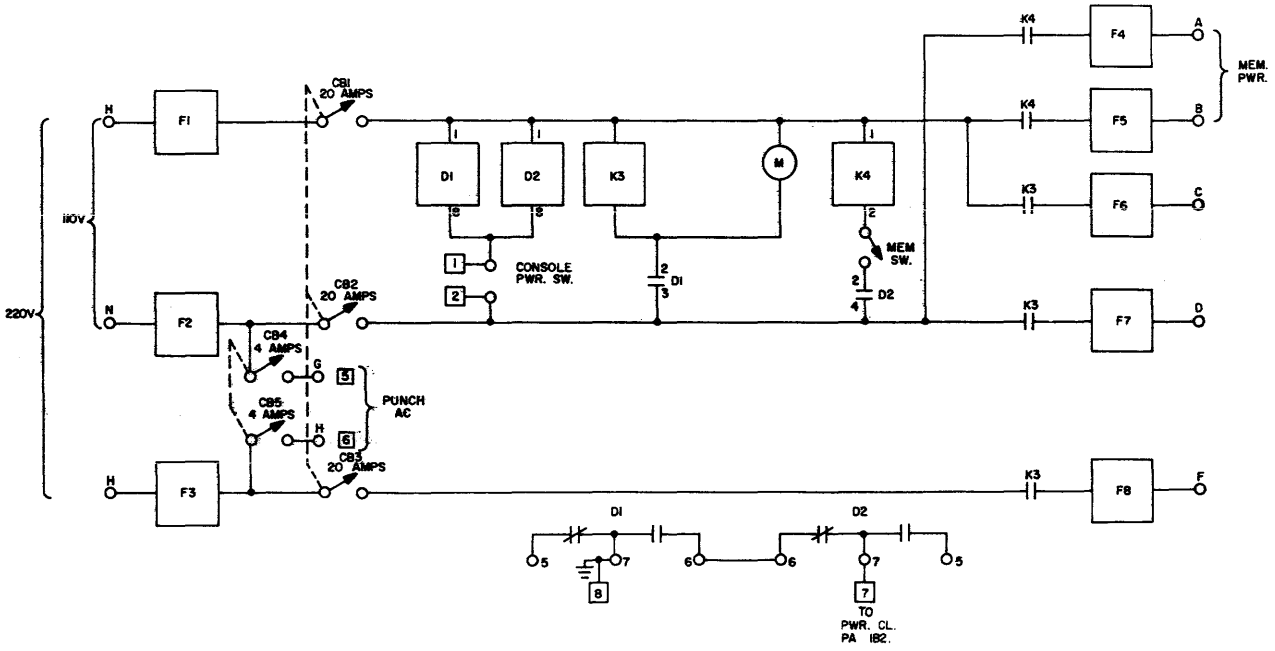
POWER SUPPLY 742

POWER CONTROL PANEL 810



D3 AND D4 ARE GE THYRECTOR
GR82OSP-484
FL1, FL2 CD NF 10280-1

POWER CONTROL PANEL 812

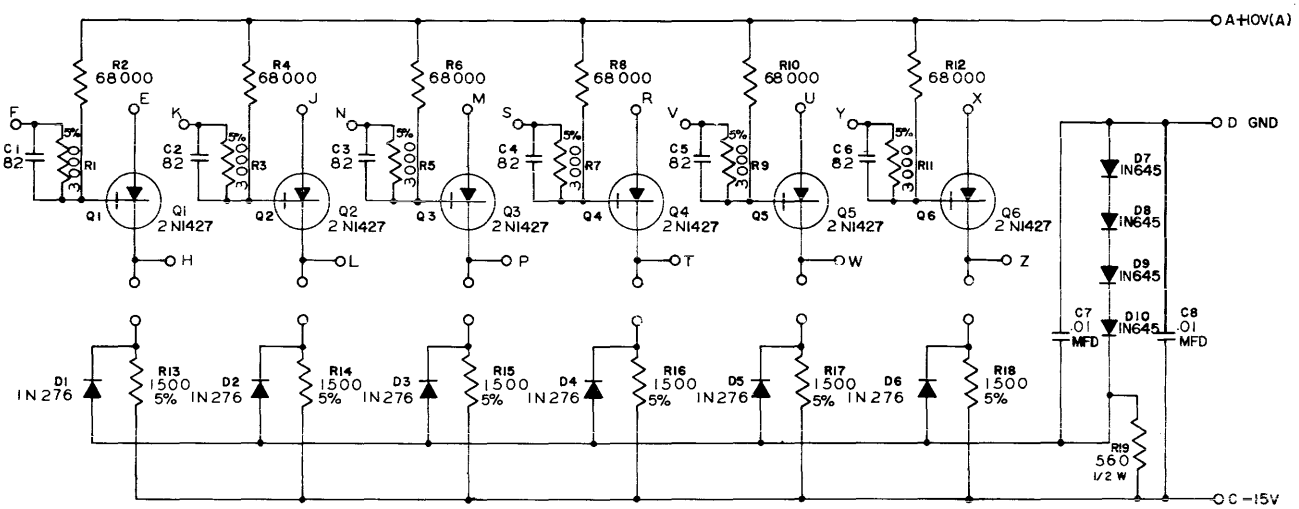


F1-F8 ARE SPRAGUE FILTROL #3
D1 IS AN AGASTAT TYPE 2122-A-5
D2 IS AN AGASTAT TYPE 2112-A-5
K3 AND K4 ARE ARROW HART #34321-U, 110V 60~
S1 IS A CUTLER HAMMER SPST
CB1-CB3 ARE HEINEMANN AM333 CURVE 4
CB4 AND CB5 ARE HEINEMANN XAM33 CURVE 4
M1 IS A HAYDON ED 71-001

POWER CONTROL PANEL 813

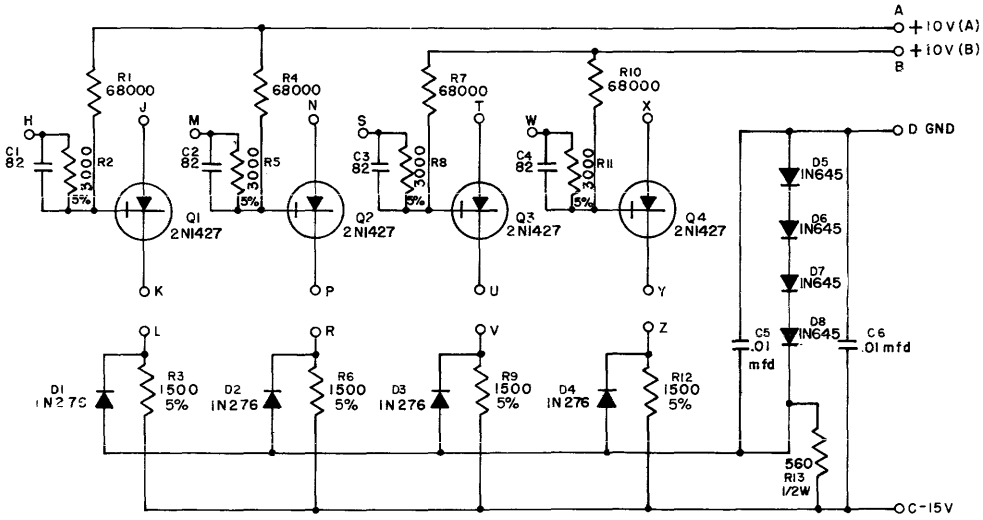
POWER CONTROL PANEL 812

POWER CONTROL PANEL 813



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD

INVERTER 1103

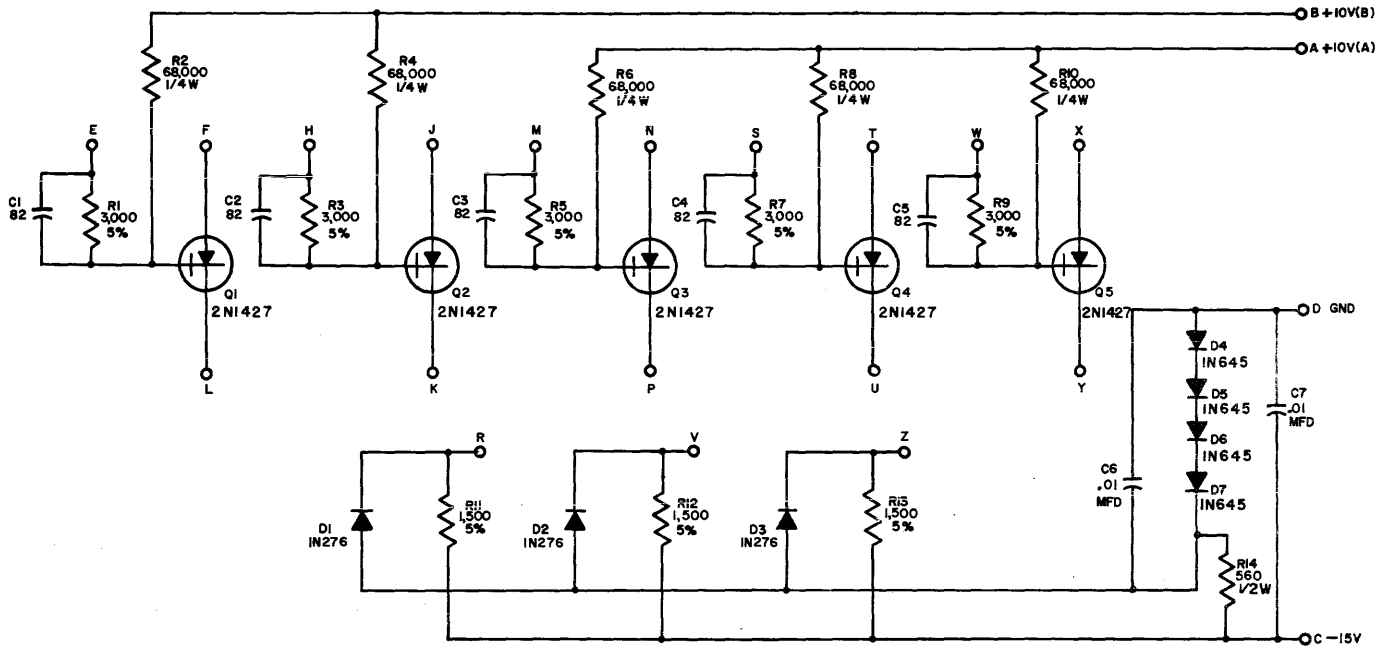


UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD

INVERTER 1104

INVERTER 1103

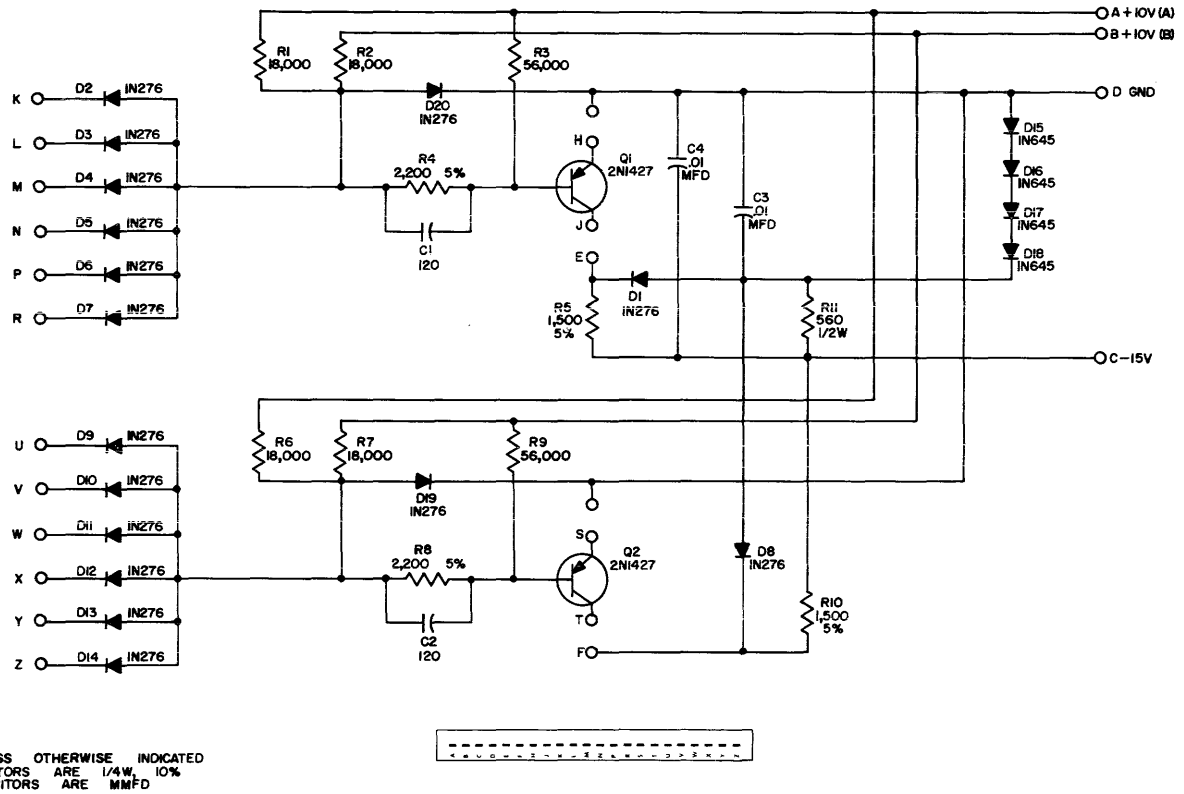
INVERTER 1104



UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD

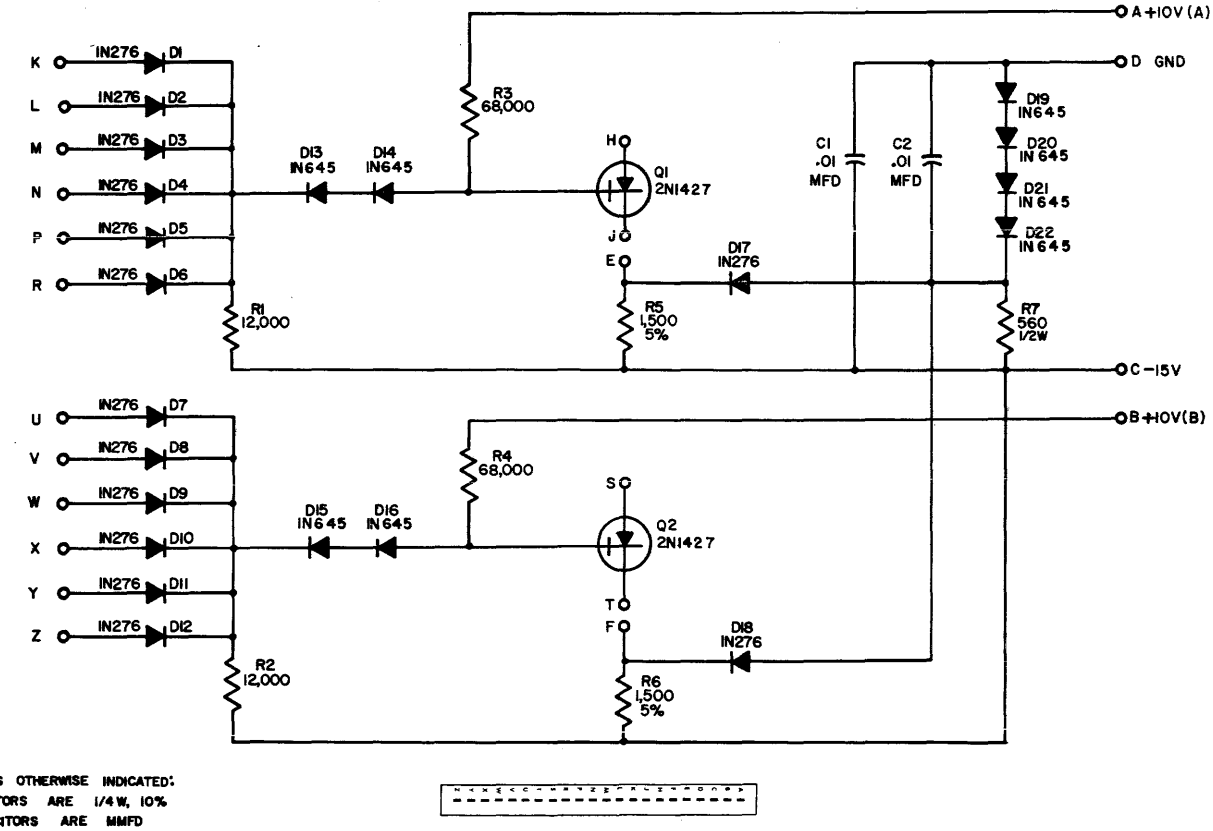
INVERTER 1105

INVERTER 1105



UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/4W, 10%
CAPACITORS ARE MMFD

DIODE 1110

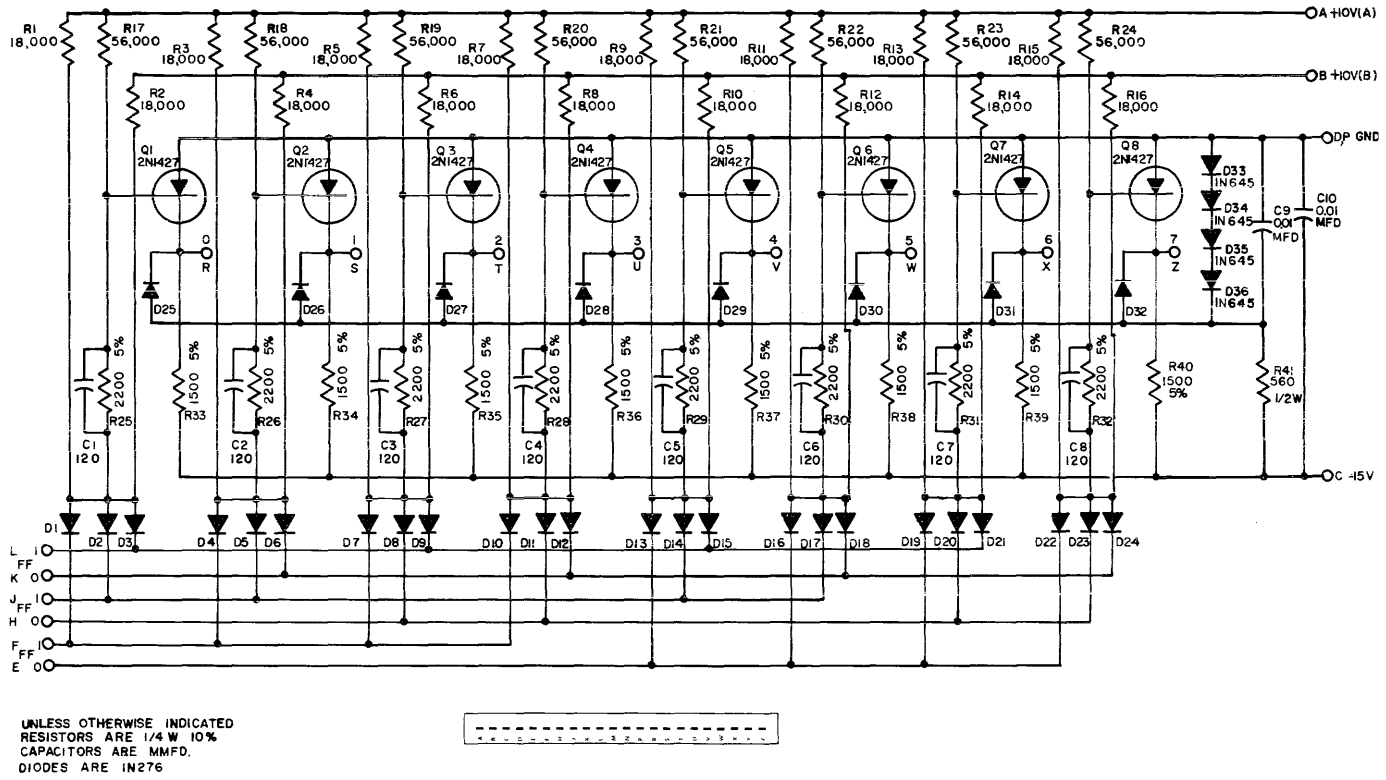


UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD

DIODE 1111

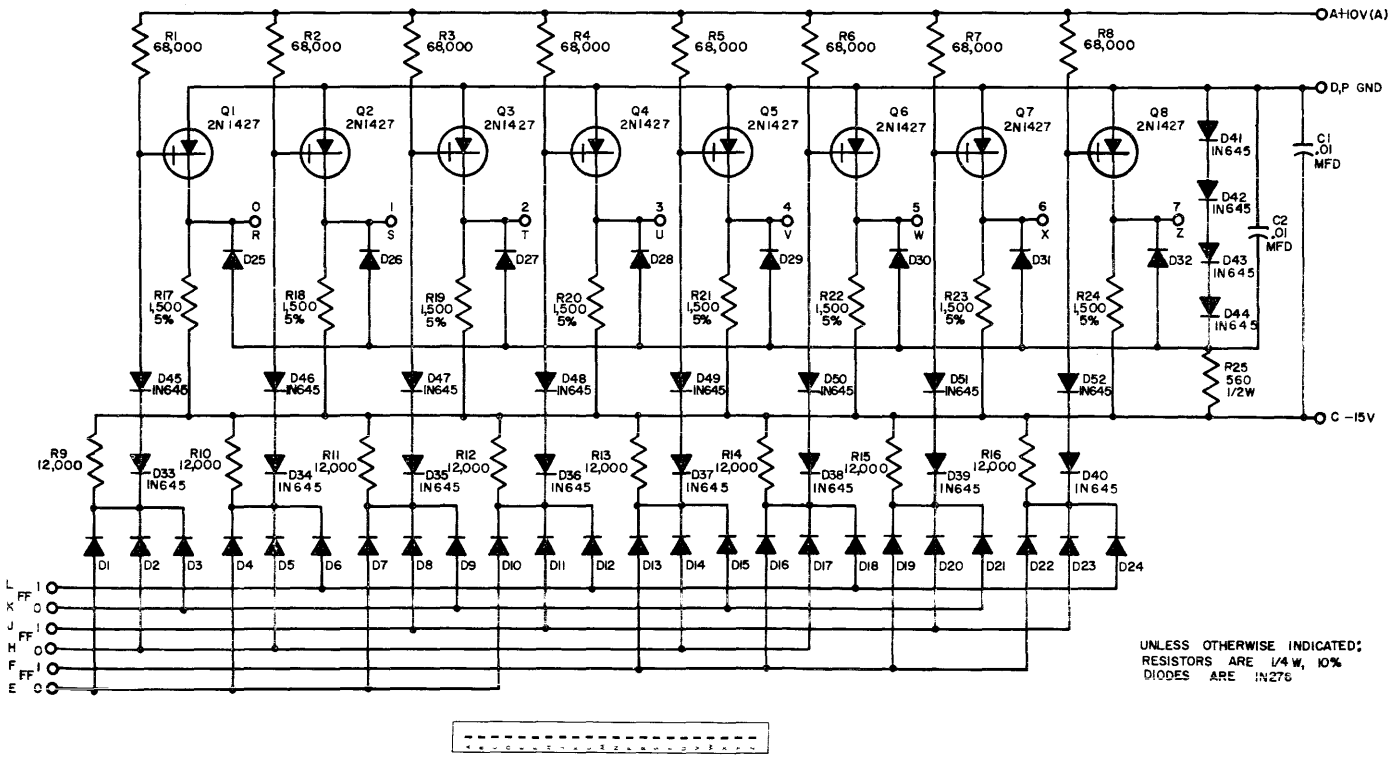
DIODE 1110

DIODE 1111



UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/4 W 10%
CAPACITORS ARE MMFD.
DIODES ARE IN276

BINARY-TO-OCTAL DECODER 1150

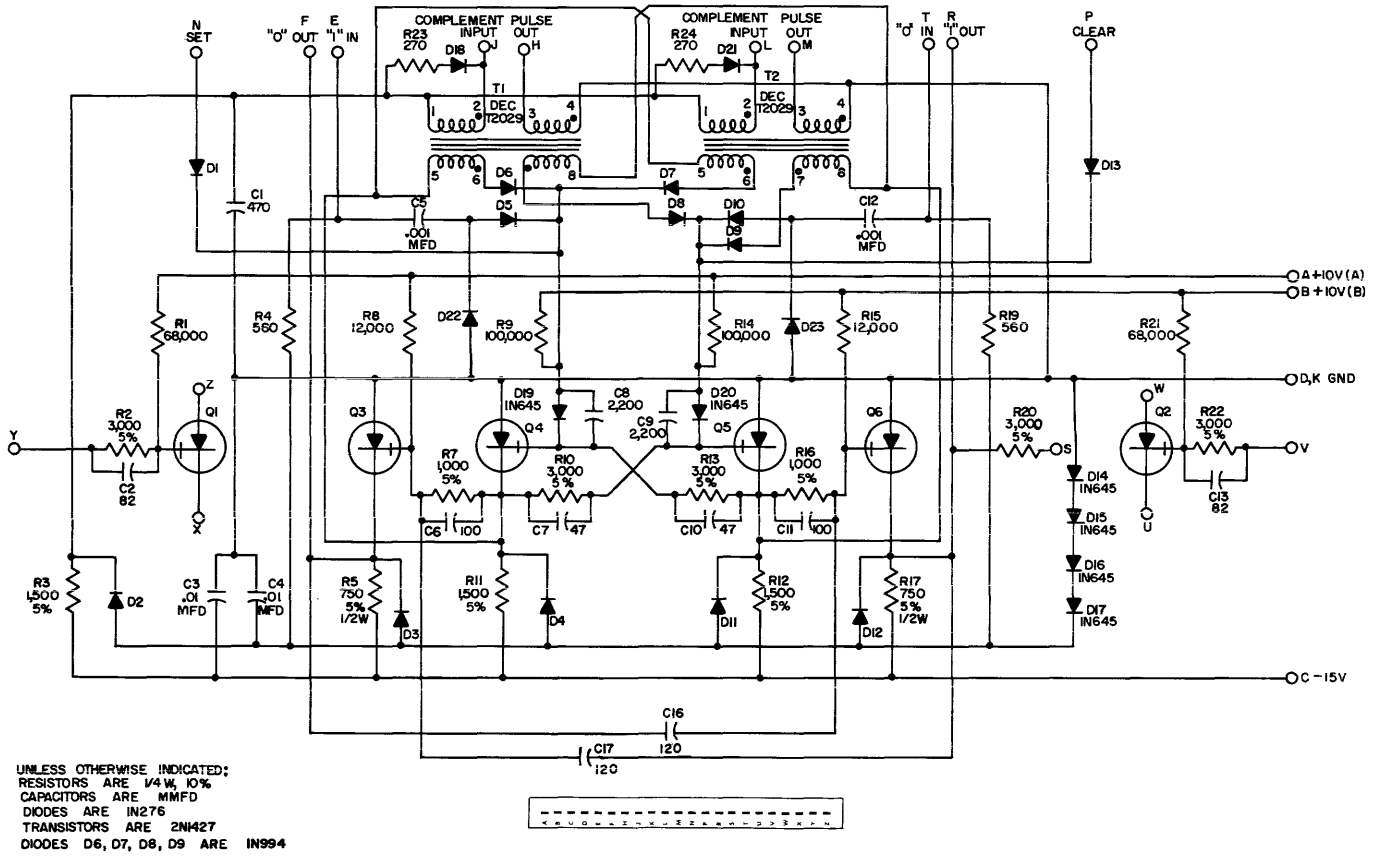


UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 10%
DIODES ARE IN276

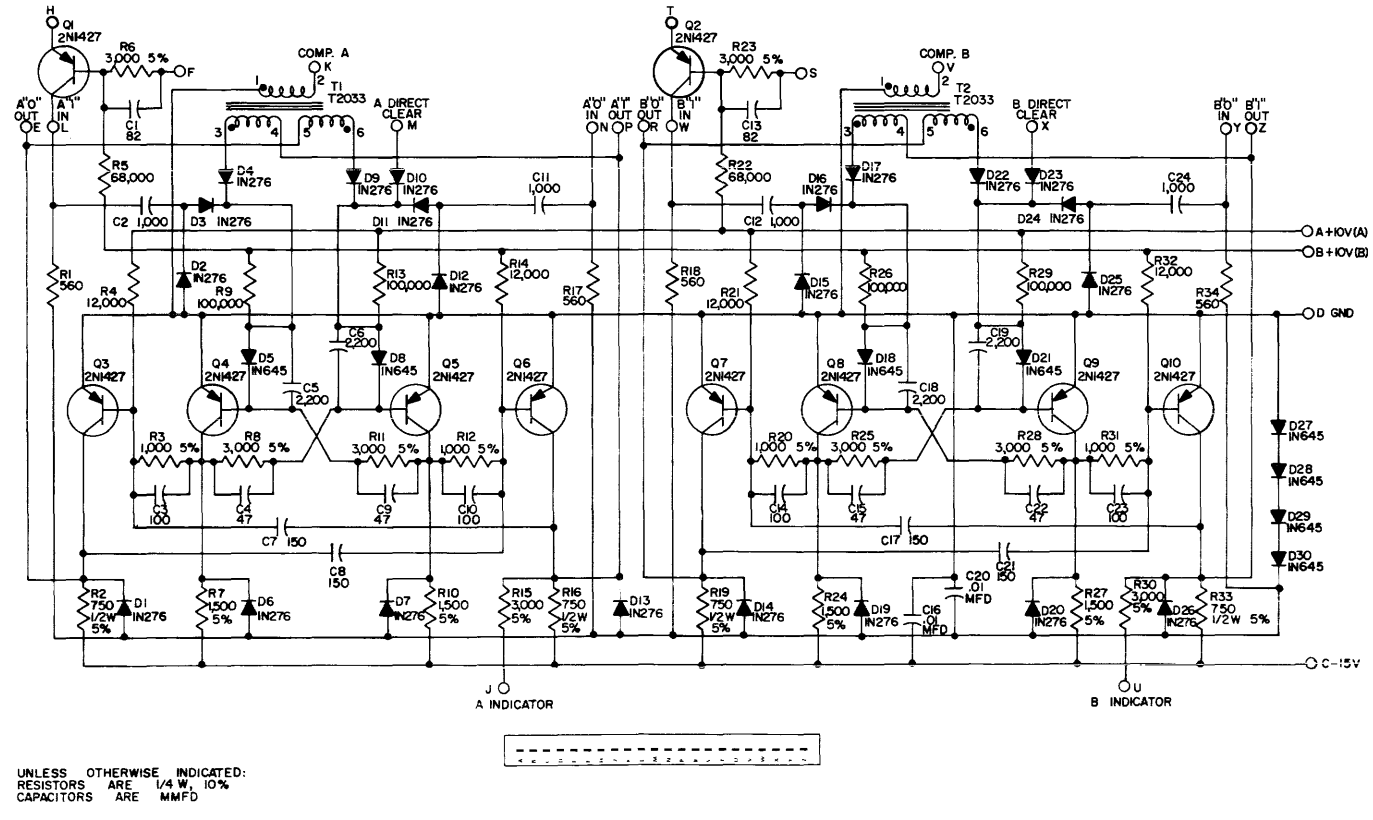
BINARY-TO-OCTAL DECODER 1151

BINARY-TO-OCTAL DECODER 1150

BINARY-TO-OCTAL DECODER 1151



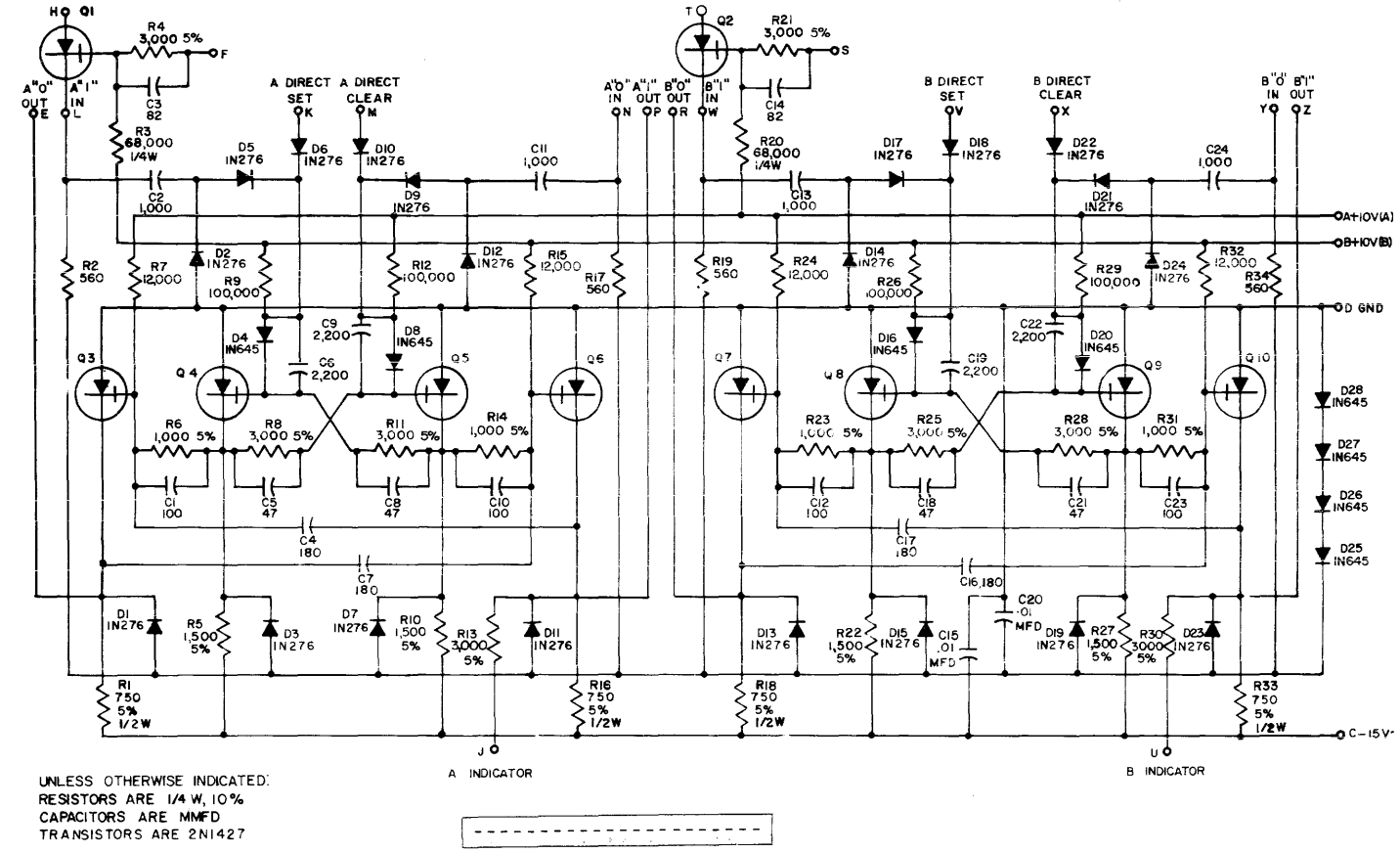
FLIP-FLOP 1201



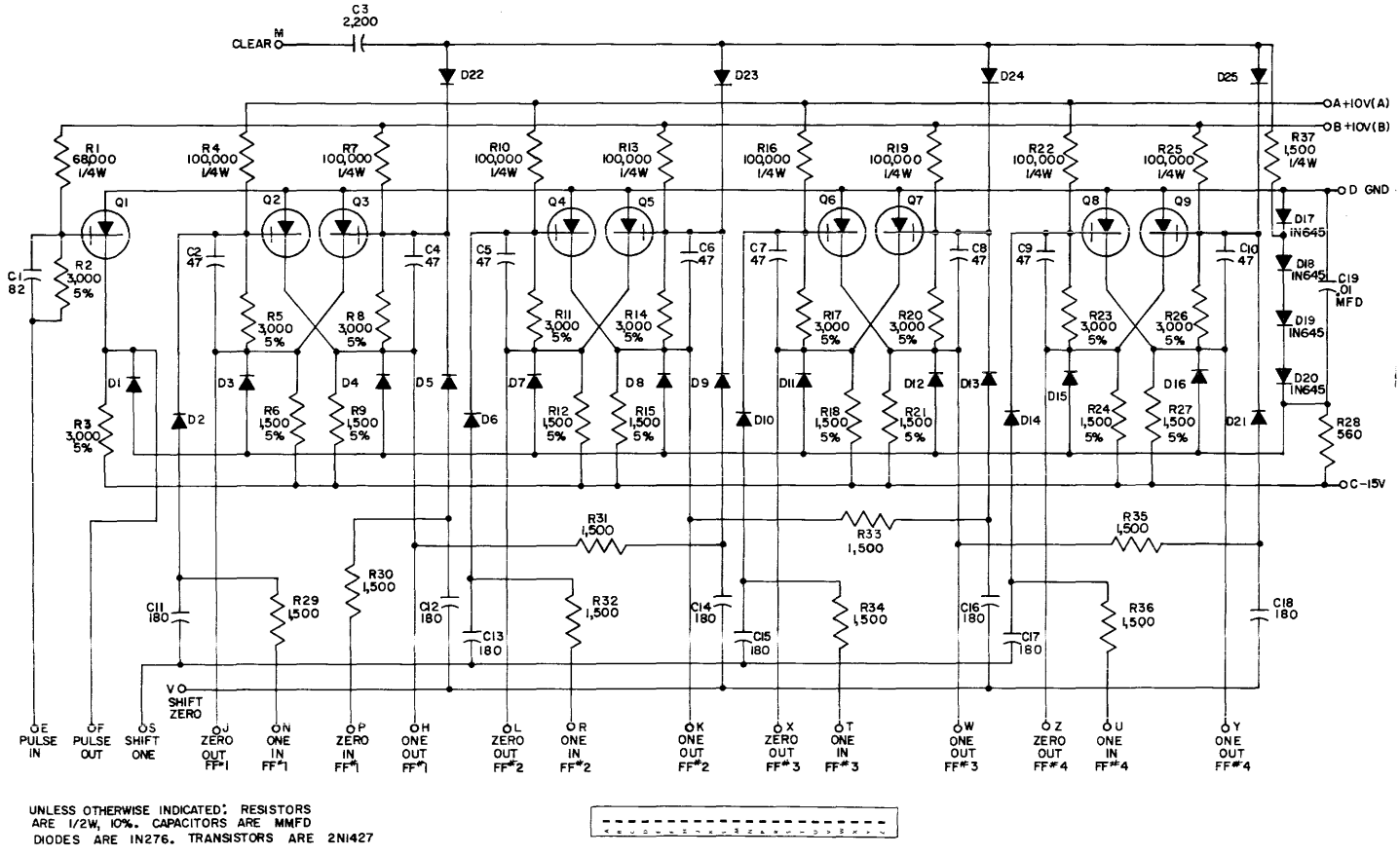
DUAL FLIP-FLOP 1204

FLIP-FLOP 1201

DUAL FLIP-FLOP 1204



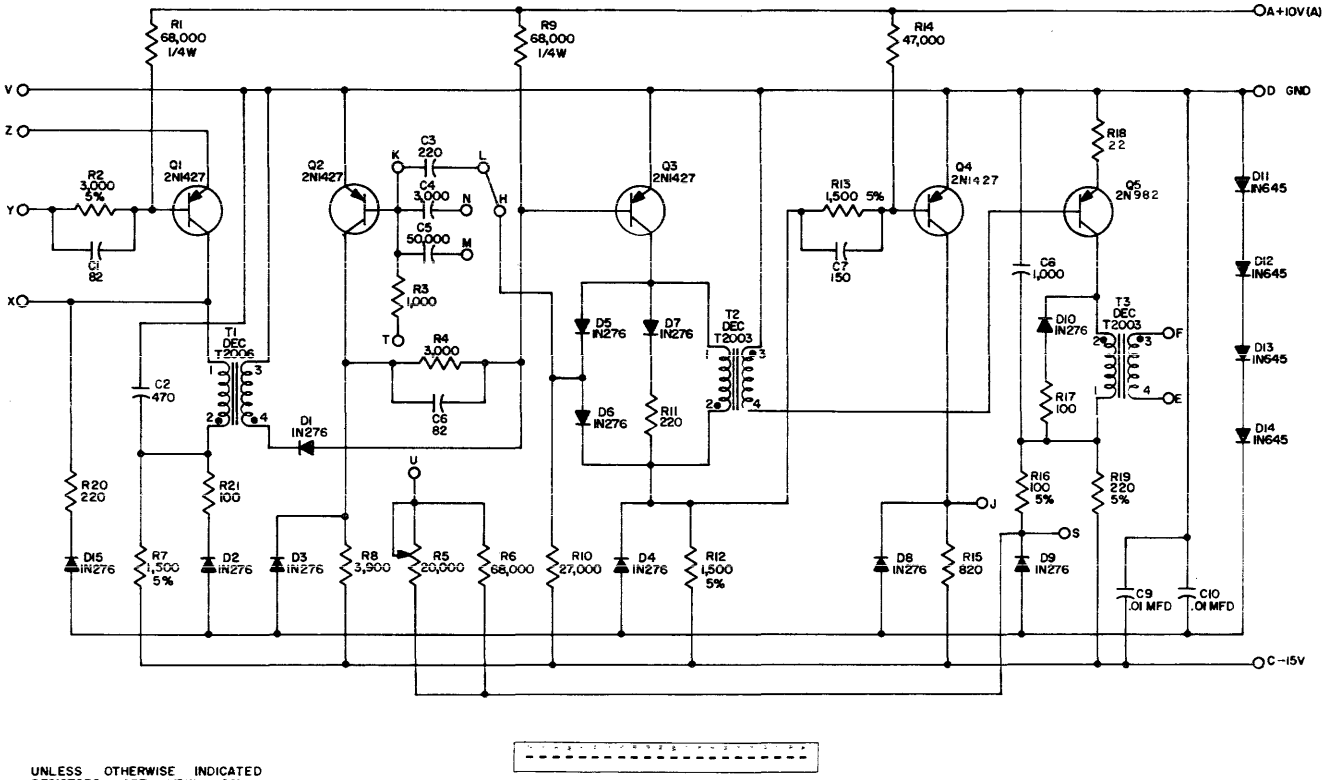
DUAL FLIP-FLOP 1209



QUADRUPLE FLIP-FLOP 1213

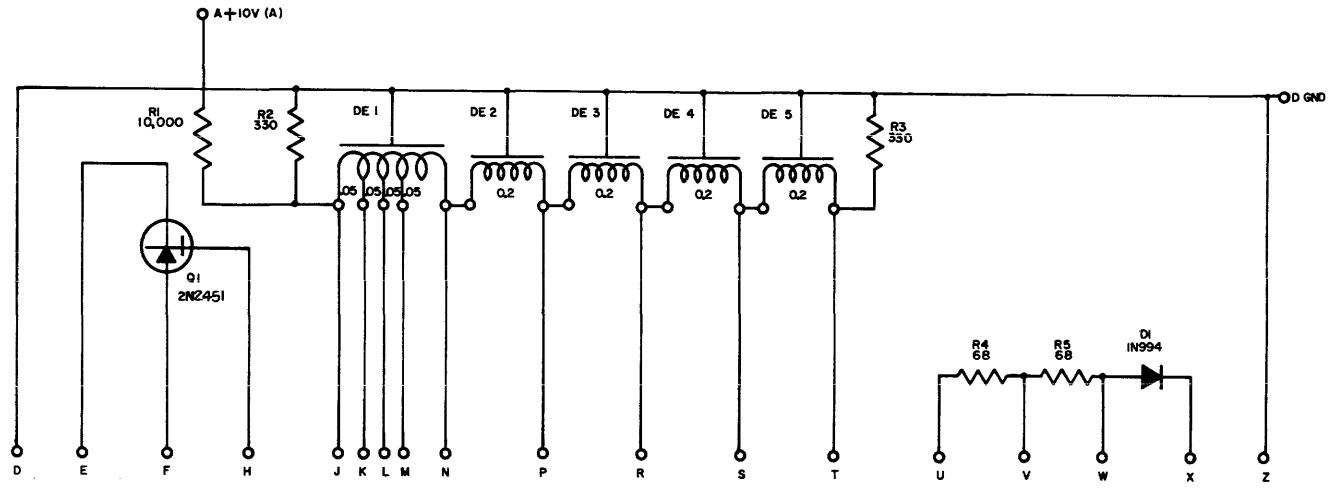
DUAL FLIP-FLOP 1209

QUADRUPLE FLIP-FLOP 1213



UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD.

DELAY 1304

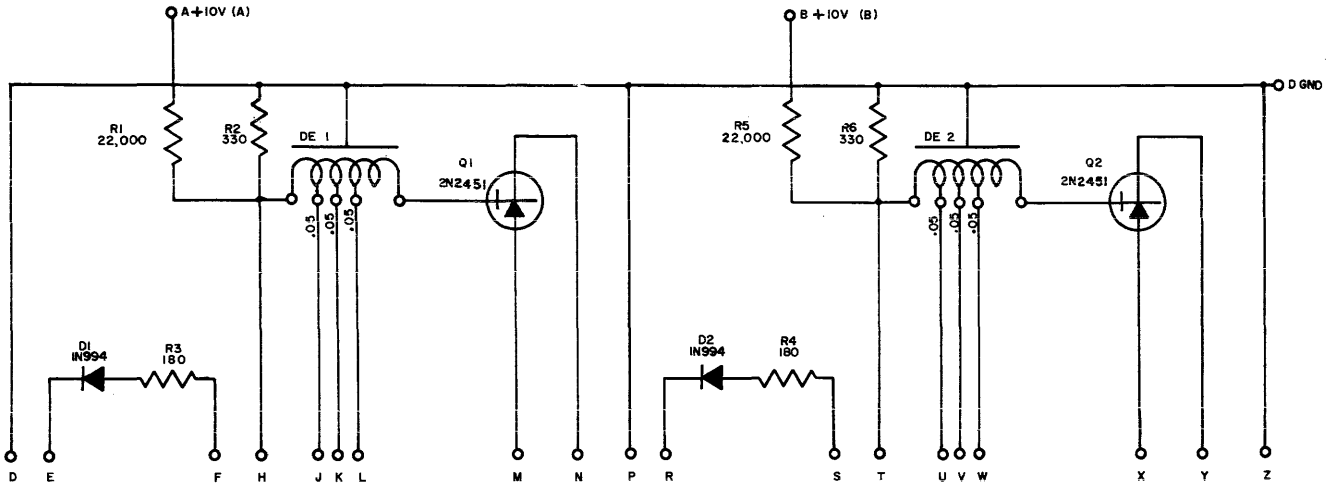


UNLESS OTHERWISE INDICATED: RESISTORS ARE 1/2W, 10%
DE 1= 0.2 μ SEC. DELAY LINE 330 OHMS. TAPPED AT 0.05 μ SEC. INTERVAL. DEC # 330-25E-6
DE 2-DE 5 = 0.2 μ SEC. DELAY LINE. DEC # 330-25E-3

DELAY 1310

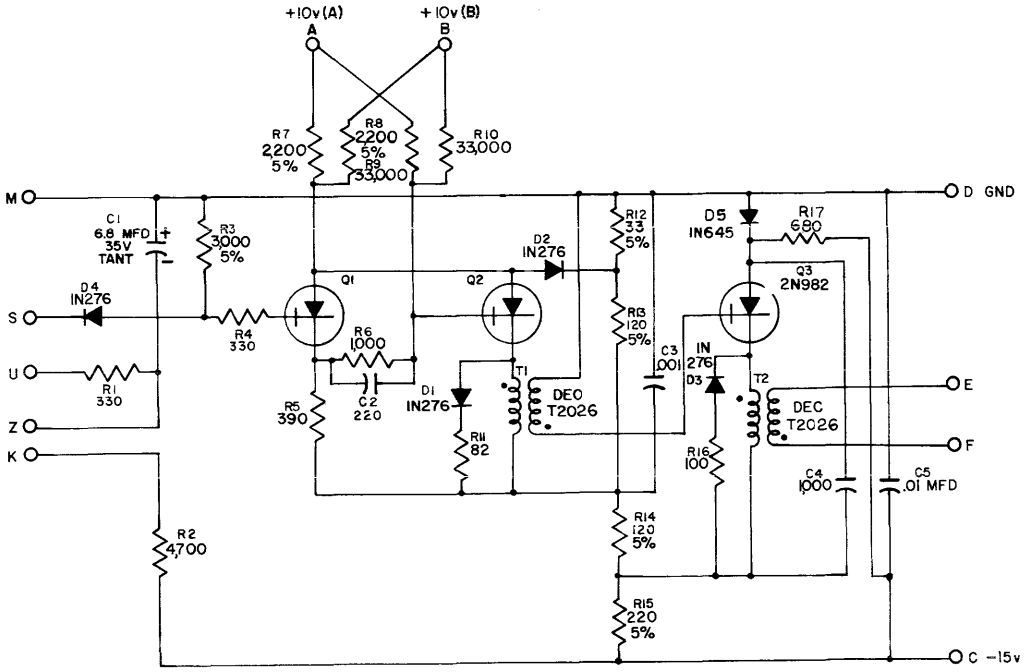
DELAY 1304

DELAY 1310



UNLESS OTHERWISE INDICATED: RESISTORS ARE 1/2W, 10%
DE 1 & DE 2 ARE 0.2 μ SEC. DELAY LINE, 330 OHMS
TAPPED AT 0.05 μ SEC. DEC #330-25E-6

DELAY 1311

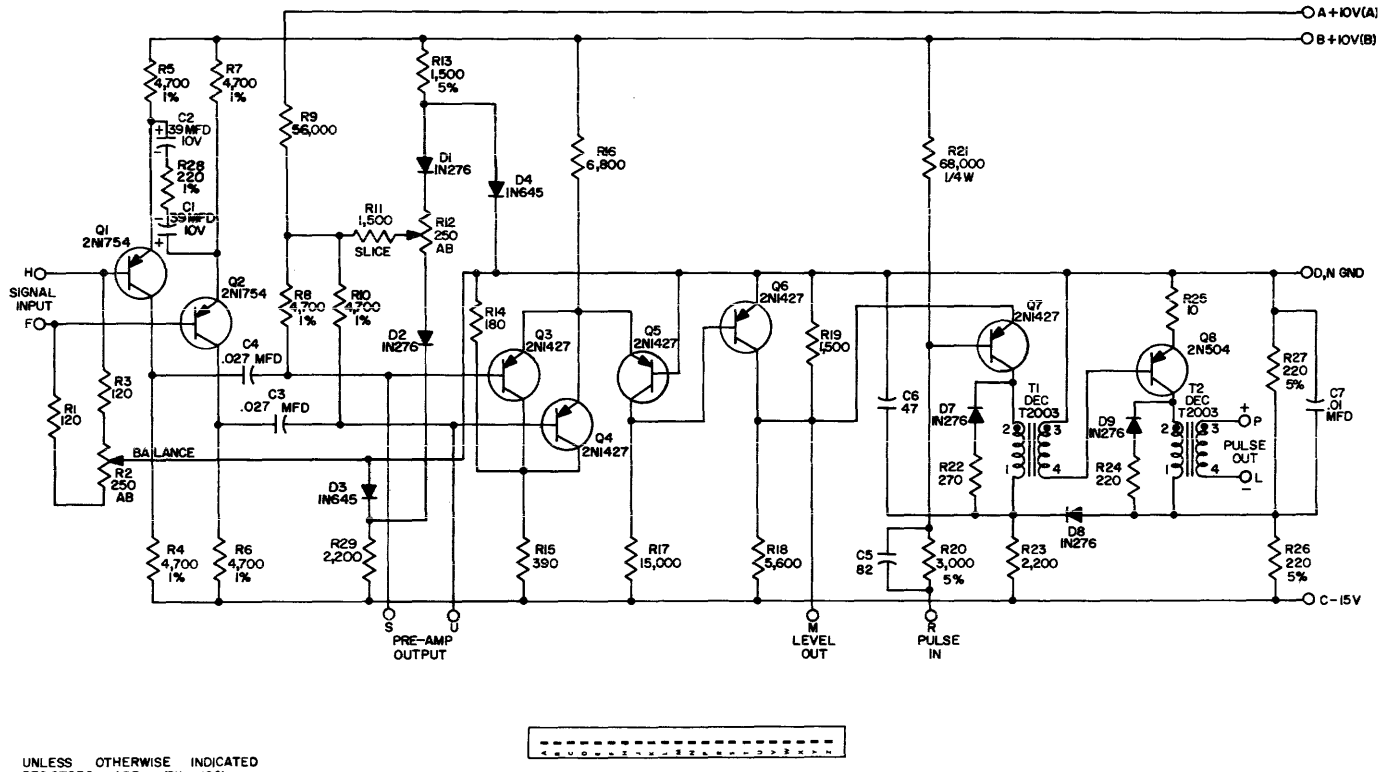


UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD
Q1 AND Q2 ARE 2N982 SELECTED
FOR IEBO ≤ 100 uA 4V

PULSE GENERATOR 1410

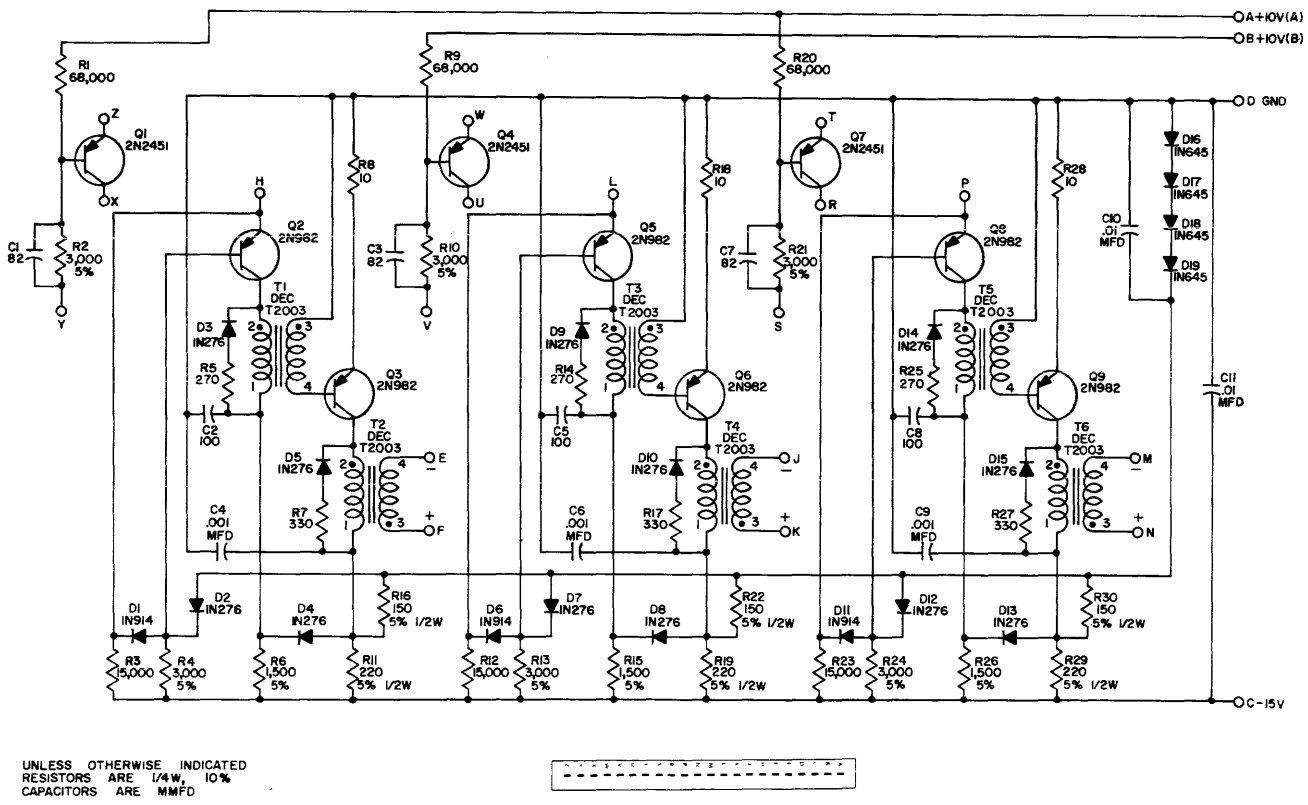
DELAY 1311

PULSE GENERATOR 1410



UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD.

SENSE AMPLIFIER 1540

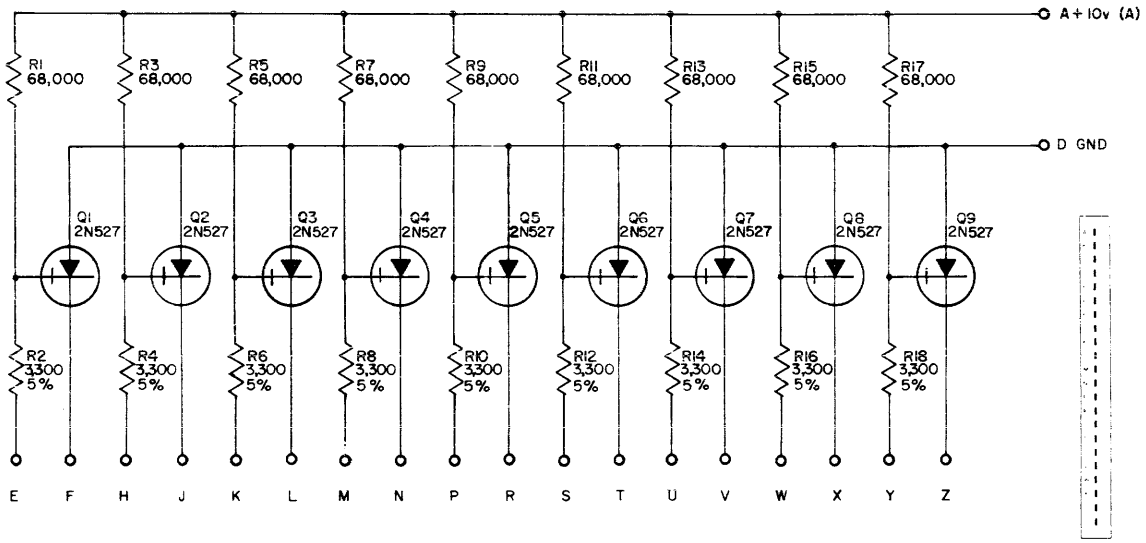


UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/4W, 10%
CAPACITORS ARE MMFD

PULSE AMPLIFIER 1607

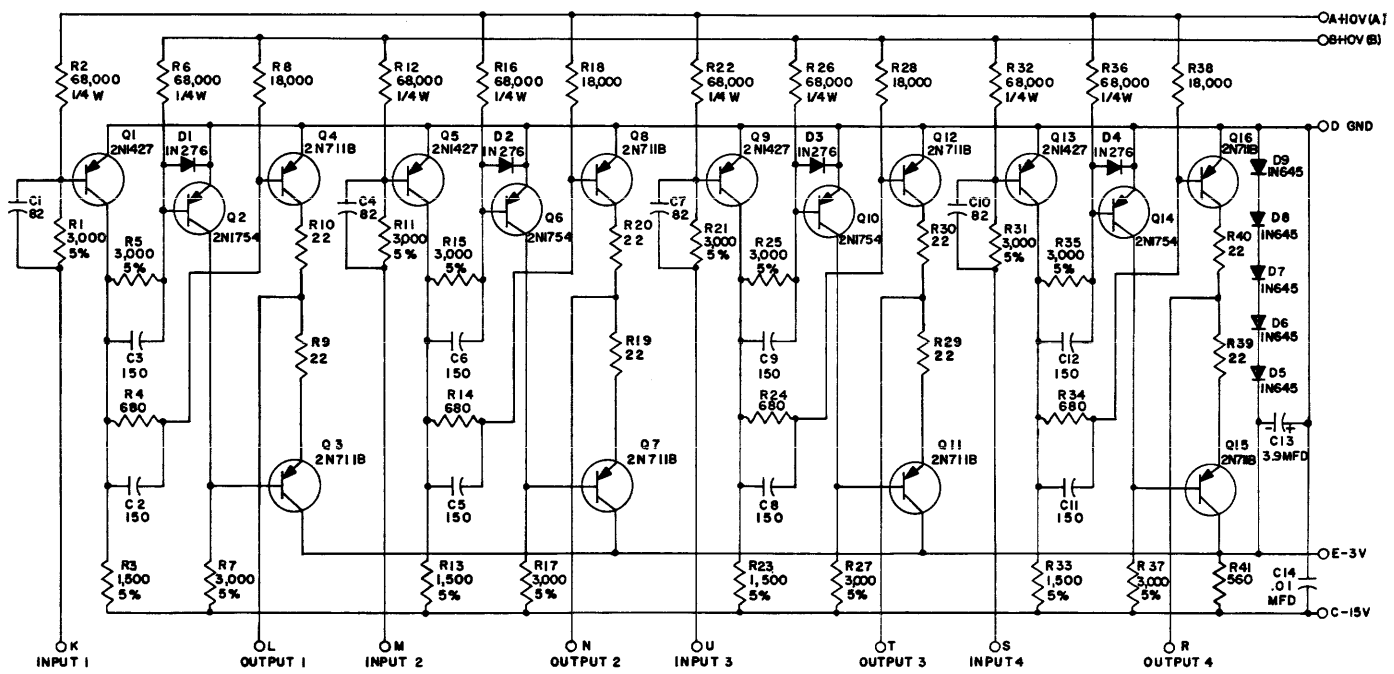
SENSE AMPLIFIER 1540

PULSE AMPLIFIER 1607



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%

INDICATOR DRIVER 1669

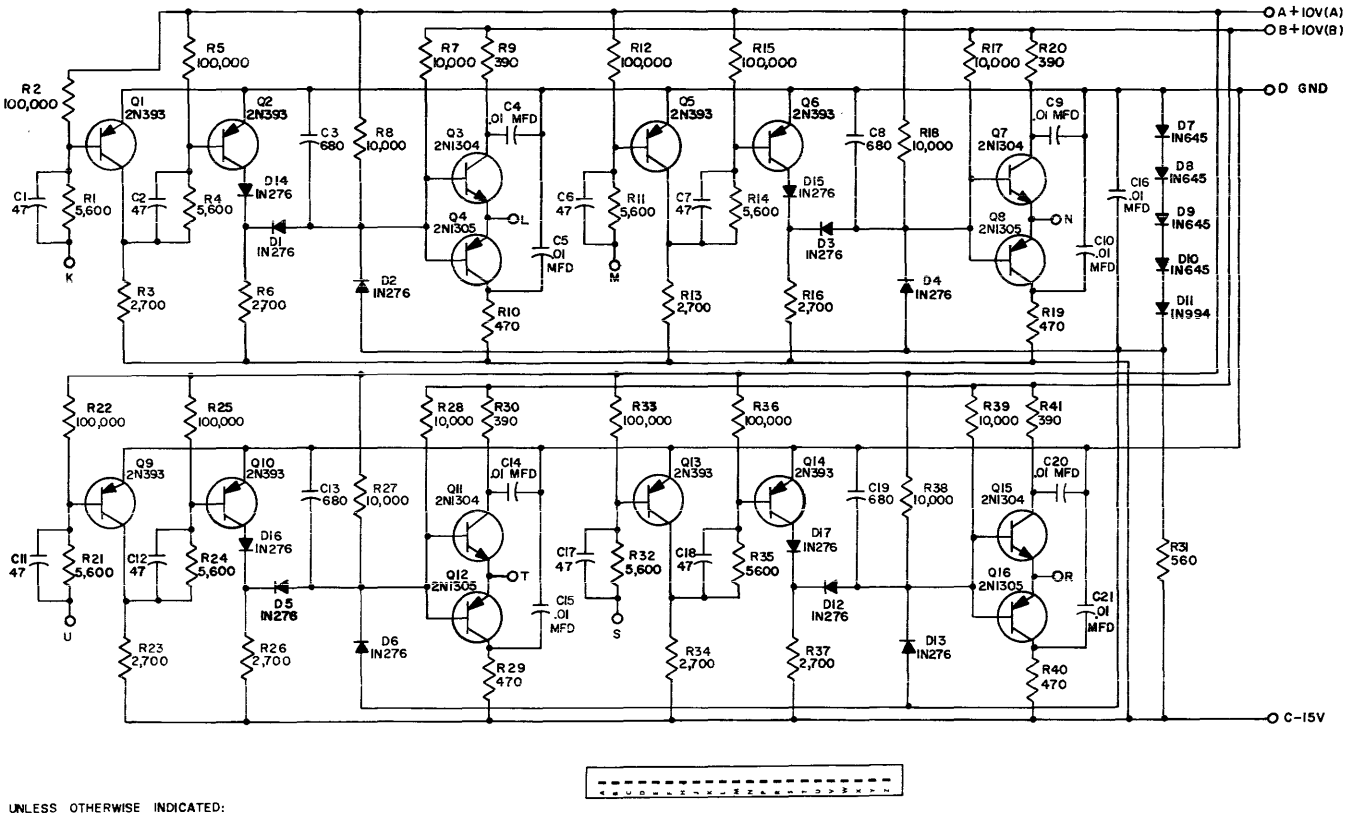


UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/2 W 10%
CAPACITORS ARE MMFD

BUS DRIVER 1684

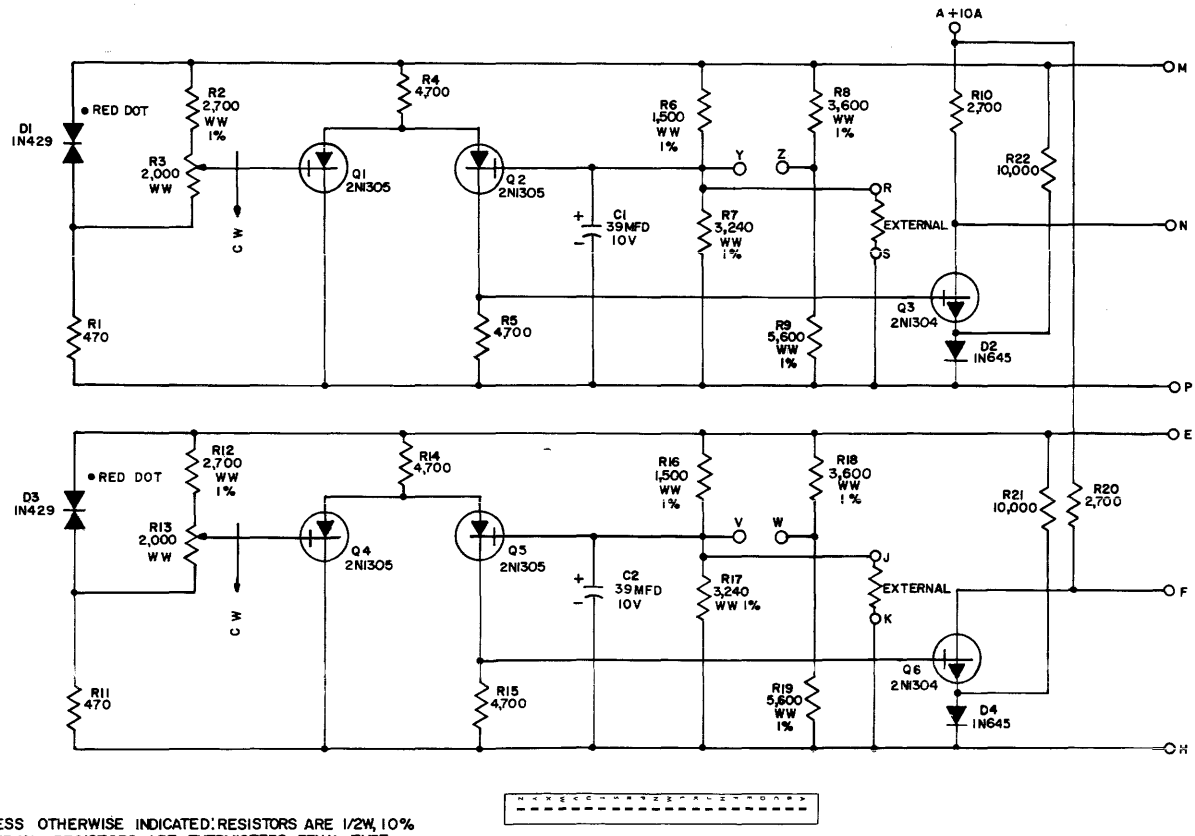
INDICATOR DRIVER 1669

BUS DRIVER 1684



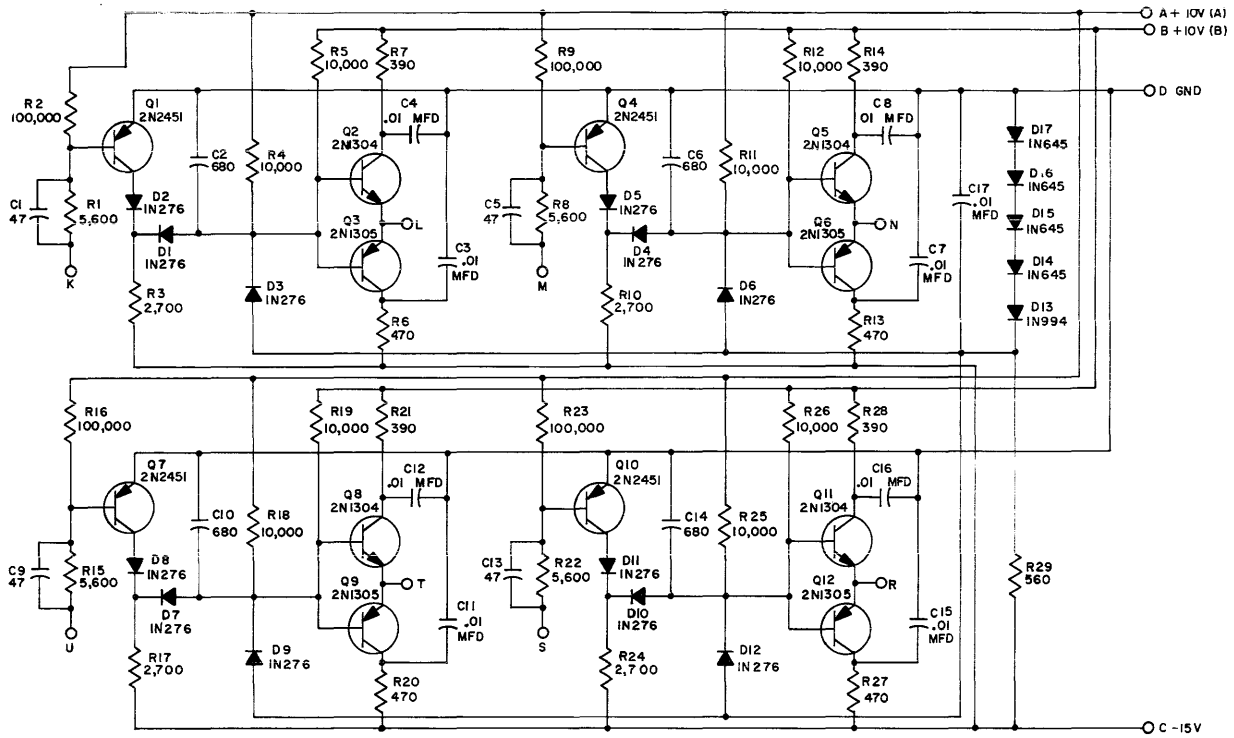
UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2 W, 10%
CAPACITORS ARE MMFD

BUS DRIVER 1685



UNLESS OTHERWISE INDICATED: RESISTORS ARE 1/2W, 10%
EXTERNAL RESISTORS ARE THERMISTERS FENAL TYPE
JA41J1 OR EQUIV. 10K Ω 25°C. 4.4% /°C

POWER SUPPLY CONTROL 1701



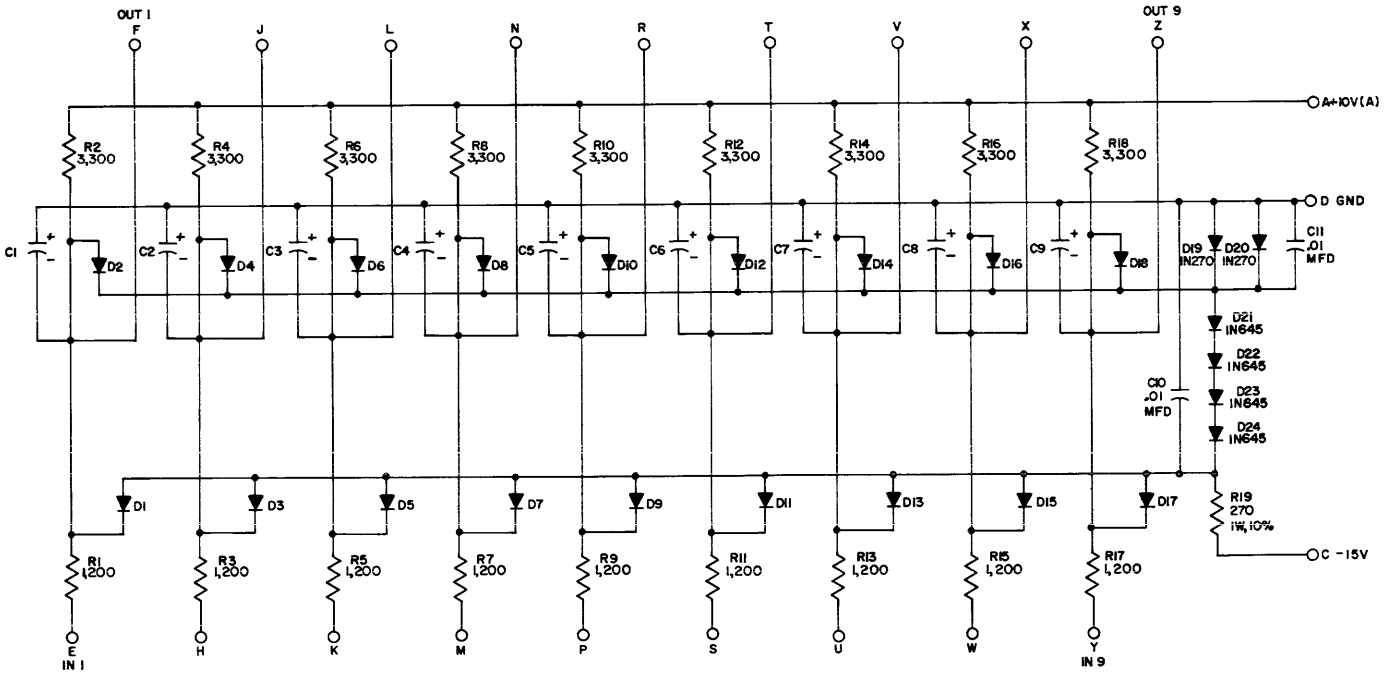
UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/2 W, 10%
CAPACITORS ARE MMFD

THIS SCHEMATIC IS FURNISHED ONLY FOR TEST AND MAINTENANCE PURPOSES. THE CIRCUITS ARE PROPRIETARY IN NATURE AND SHOULD BE TREATED ACCORDINGLY.

BUS DRIVER 1690

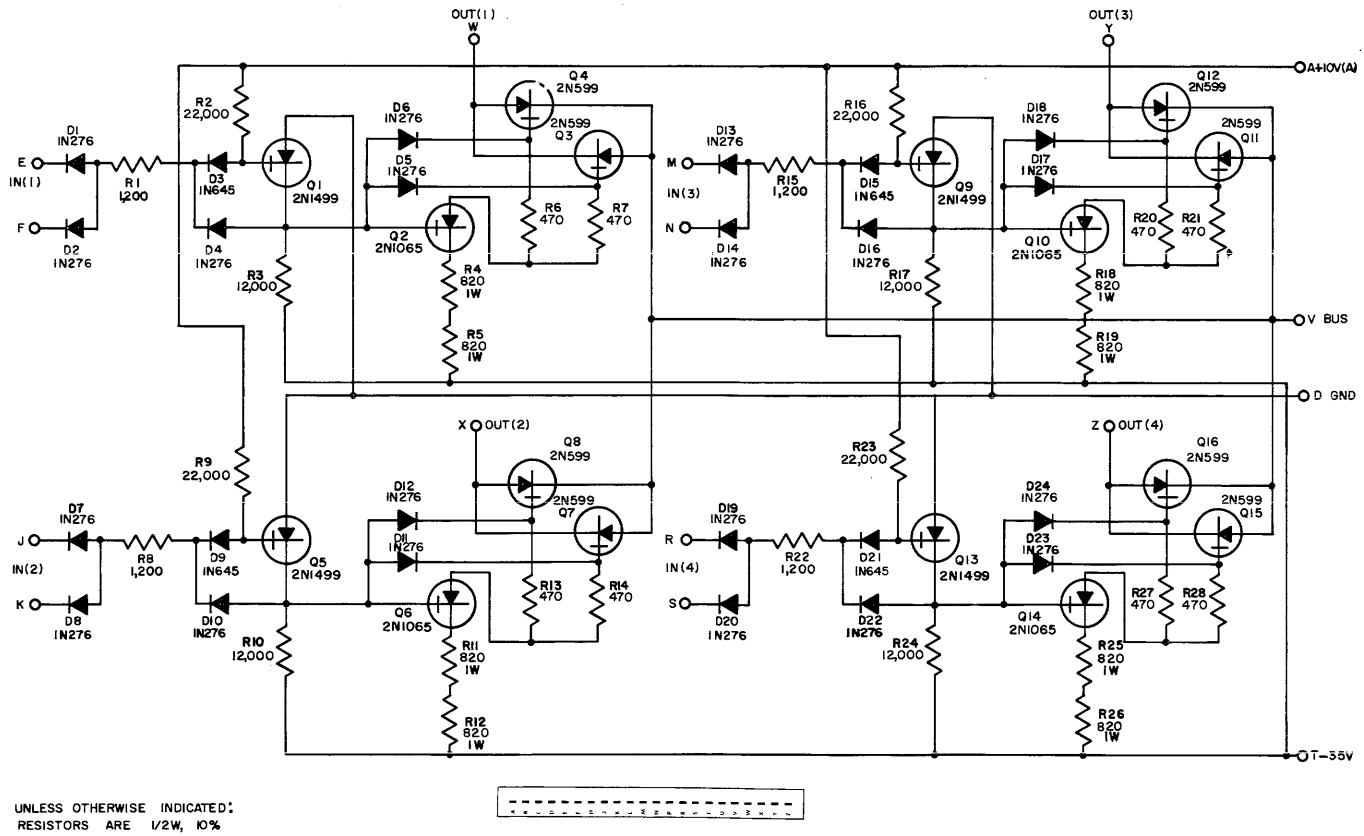
BUS DRIVER 1685

POWER SUPPLY CONTROL 1701



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%
DIODES ARE IN276
C1-C9 ARE 6.8 MFD, 35V, TANT.

SWITCH FILTER 1703

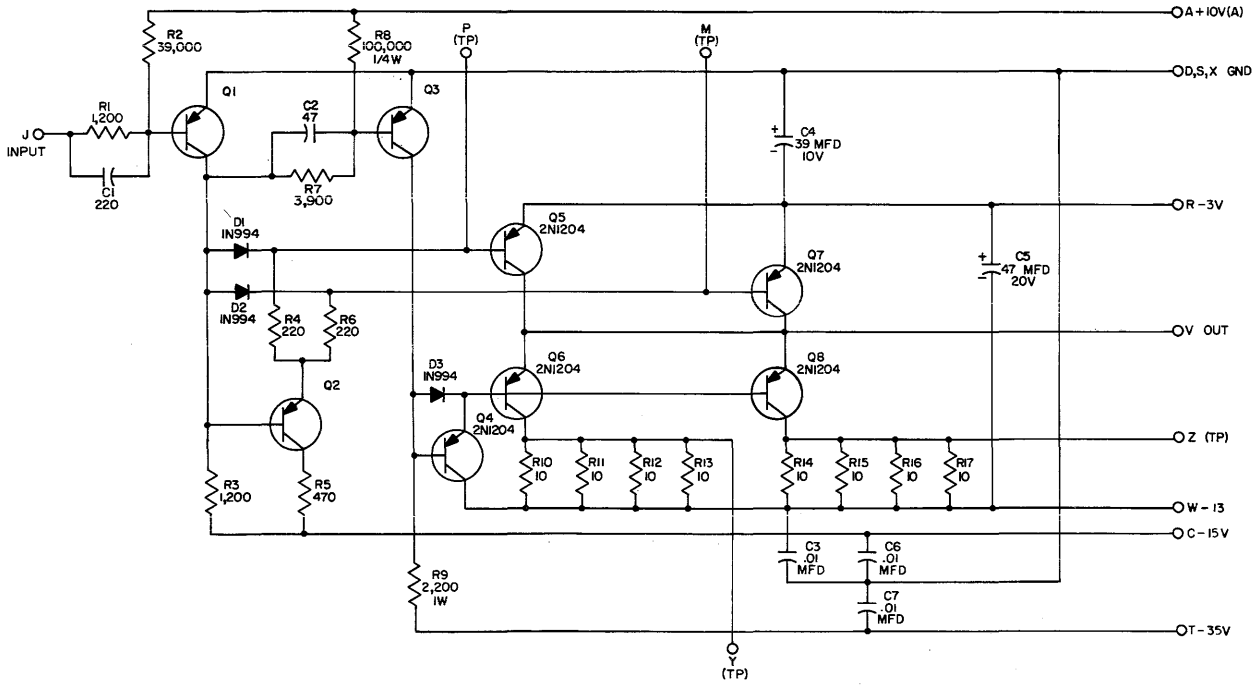


UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD

READ/WRITE SWITCH 1972

SWITCH FILTER 1703

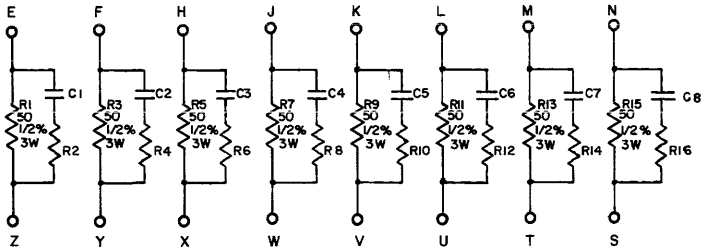
READ/WRITE SWITCH 1972



UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD
TRANSISTORS ARE 2N982 SELECTED FOR
$I_{EBO} \leq 100$ μA AMPS AT 4V

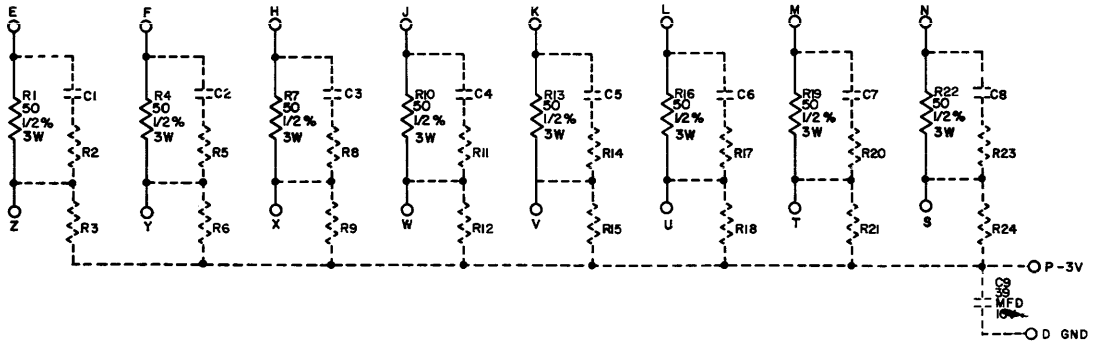
MEMORY DRIVER 1973

MEMORY DRIVER 1973



UNLESS OTHERWISE SPECIFIED:
ALL RESISTORS ARE 47Ω,J%
ALL CAPACITORS ARE 4700pf, 1%

RESISTOR BOARD 1976

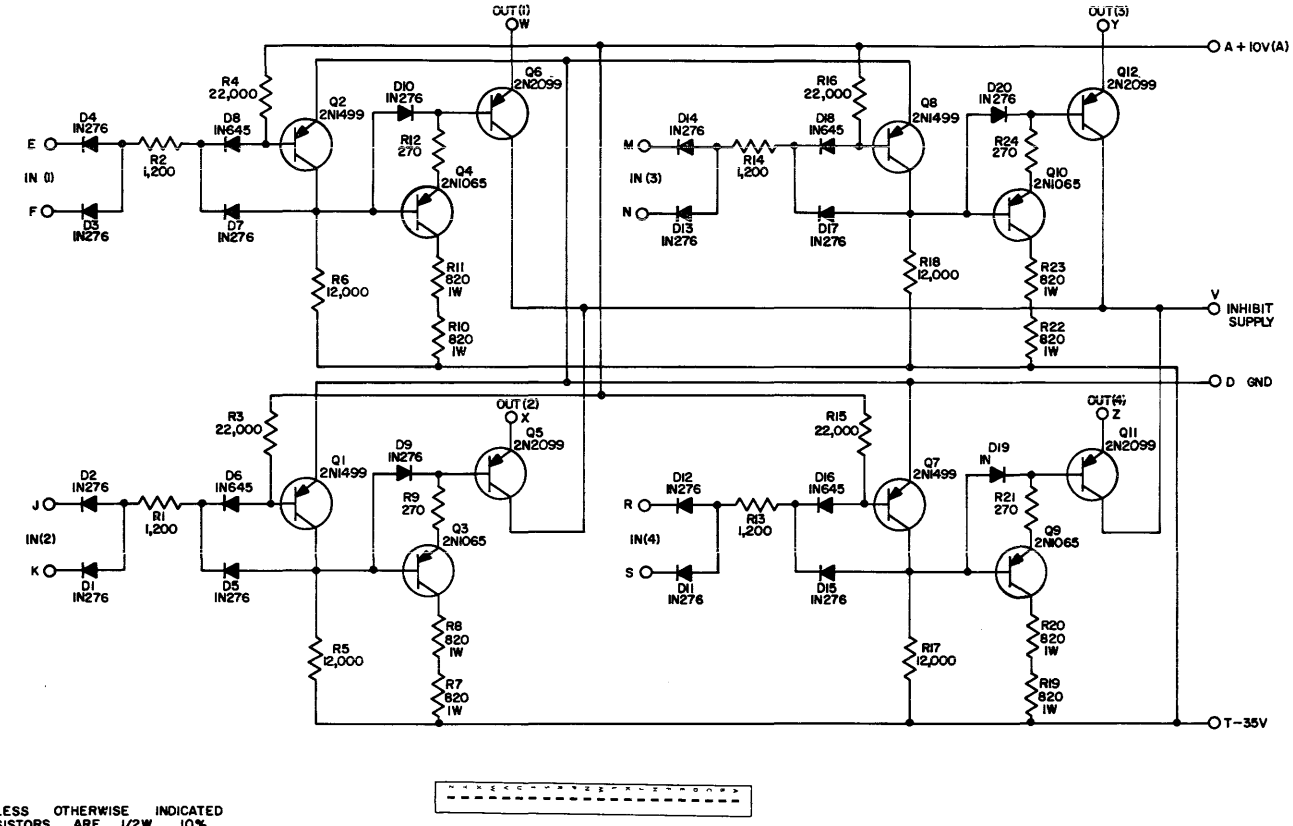


SPECIFICATIONS DEPEND UPON UNLESS CIRCUIT INDICATED APPLICATION

RESISTOR BOARD 1978

RESISTOR BOARD 1976

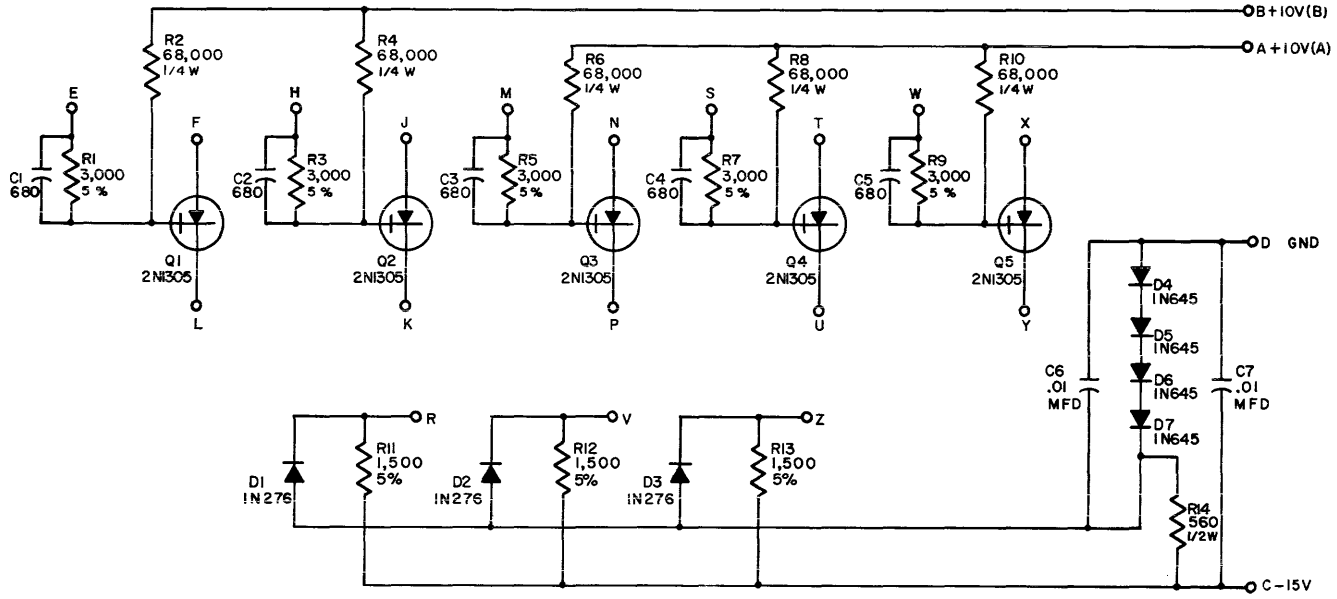
RESISTOR BOARD 1978



UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD.

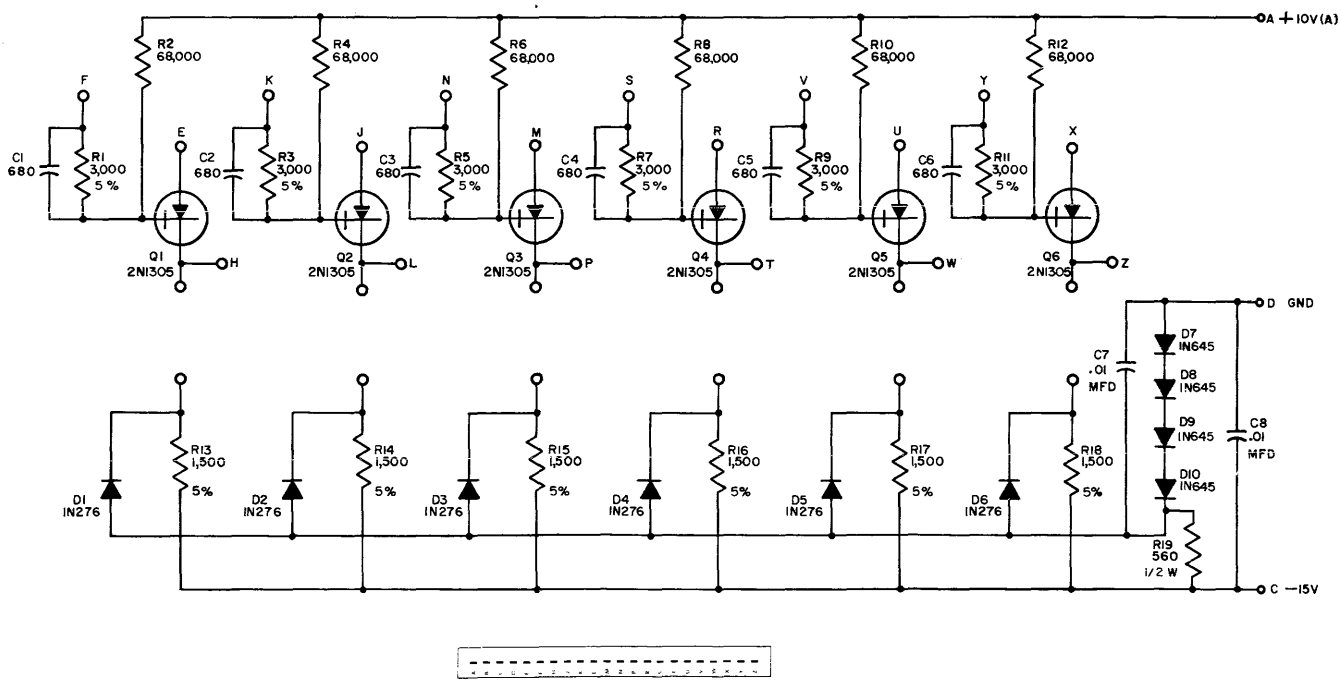
INHIBIT DRIVER 1982

INHIBIT DRIVER 1982



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4W, 10%
CAPACITORS ARE MMFD.

INVERTER 4105

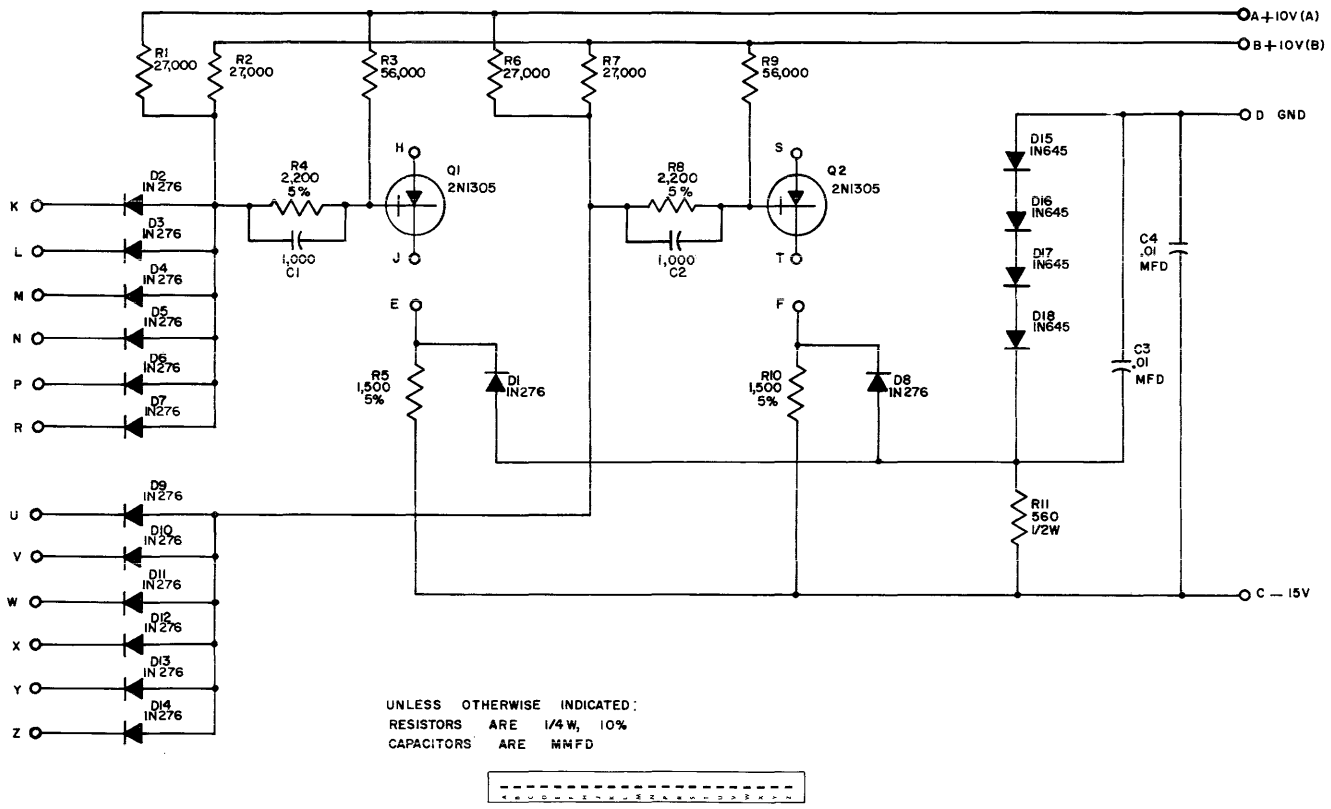


UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD

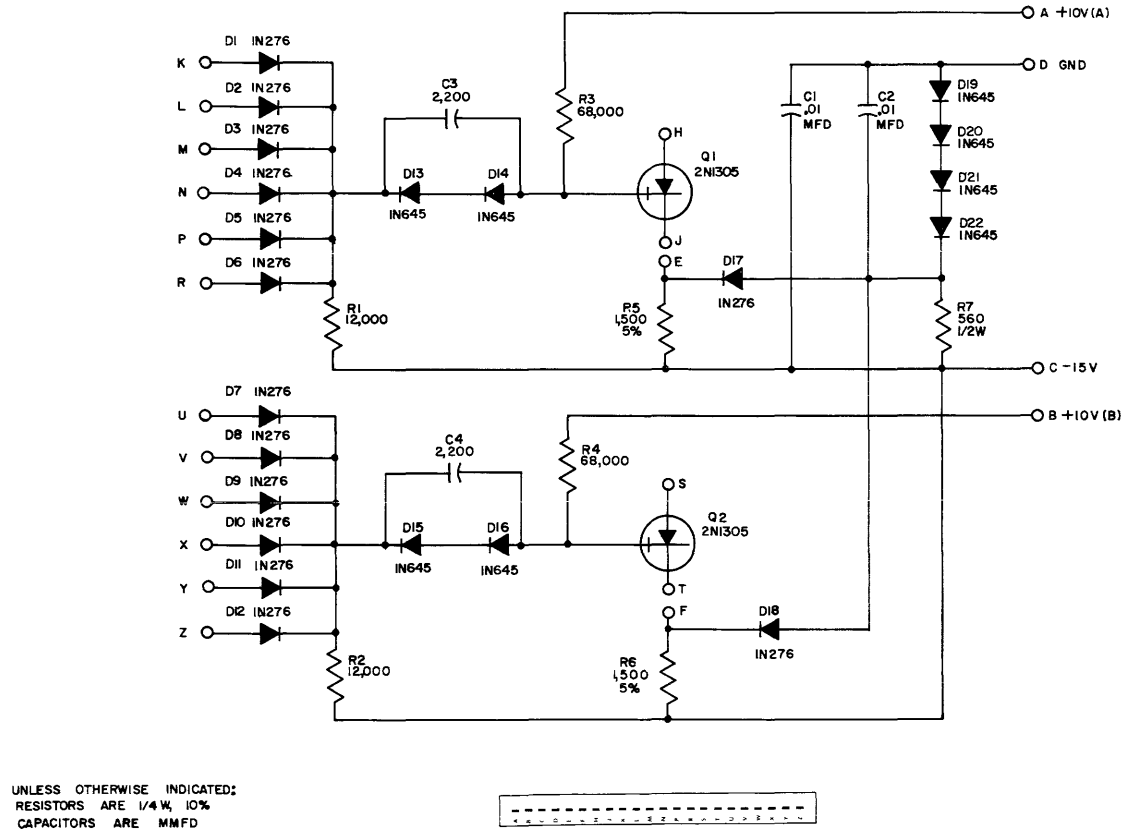
INVERTER 4106

INVERTER 4105

INVERTER 4106



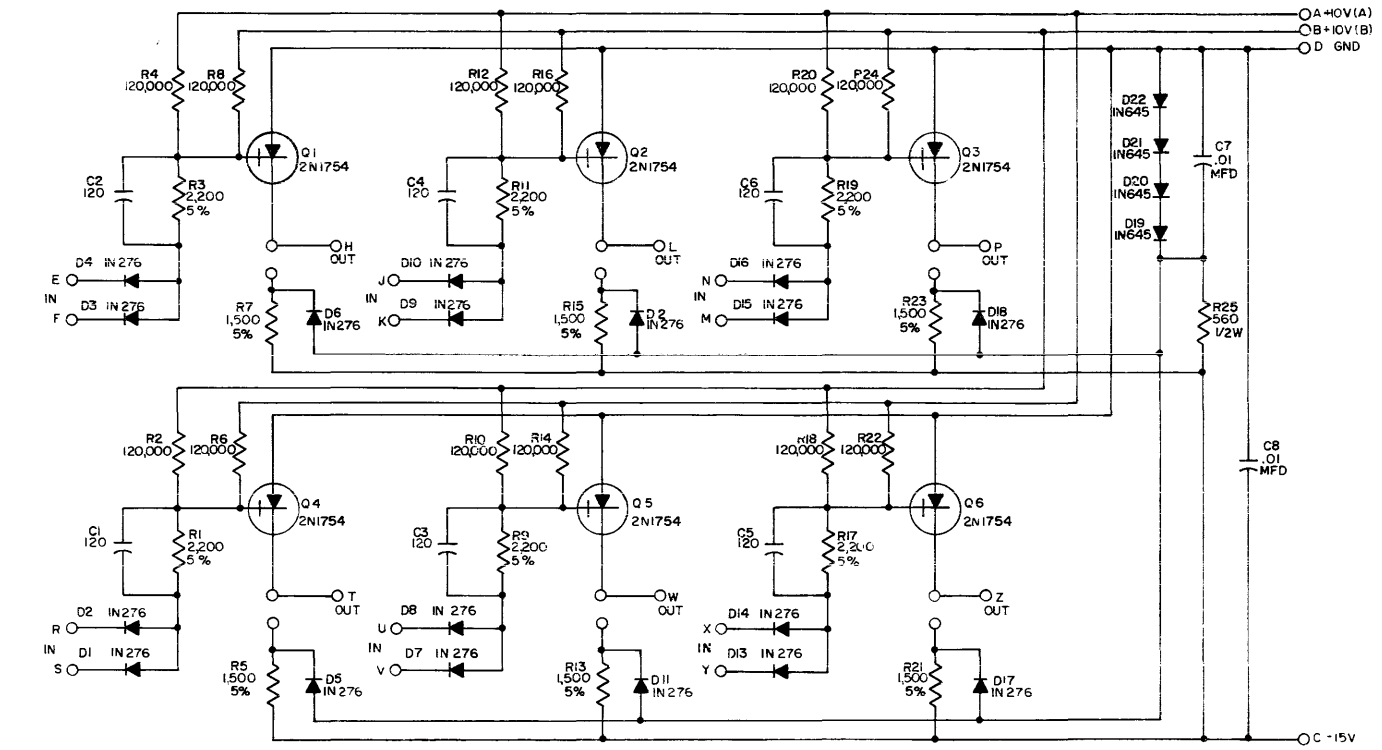
DIODE 4110



DIODE 4111

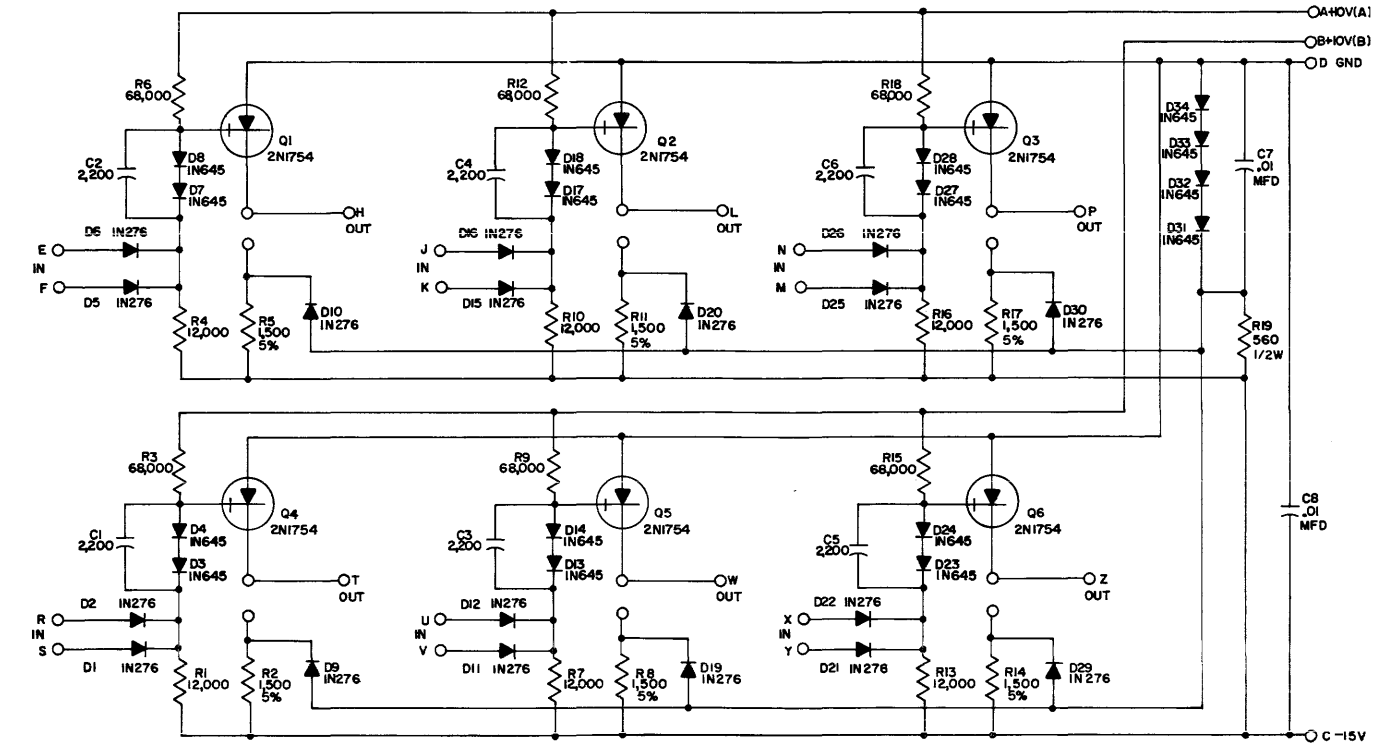
DIODE 4110

DIODE 4111



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W 10%
CAPACITORS ARE MMFD

DIODE 4112

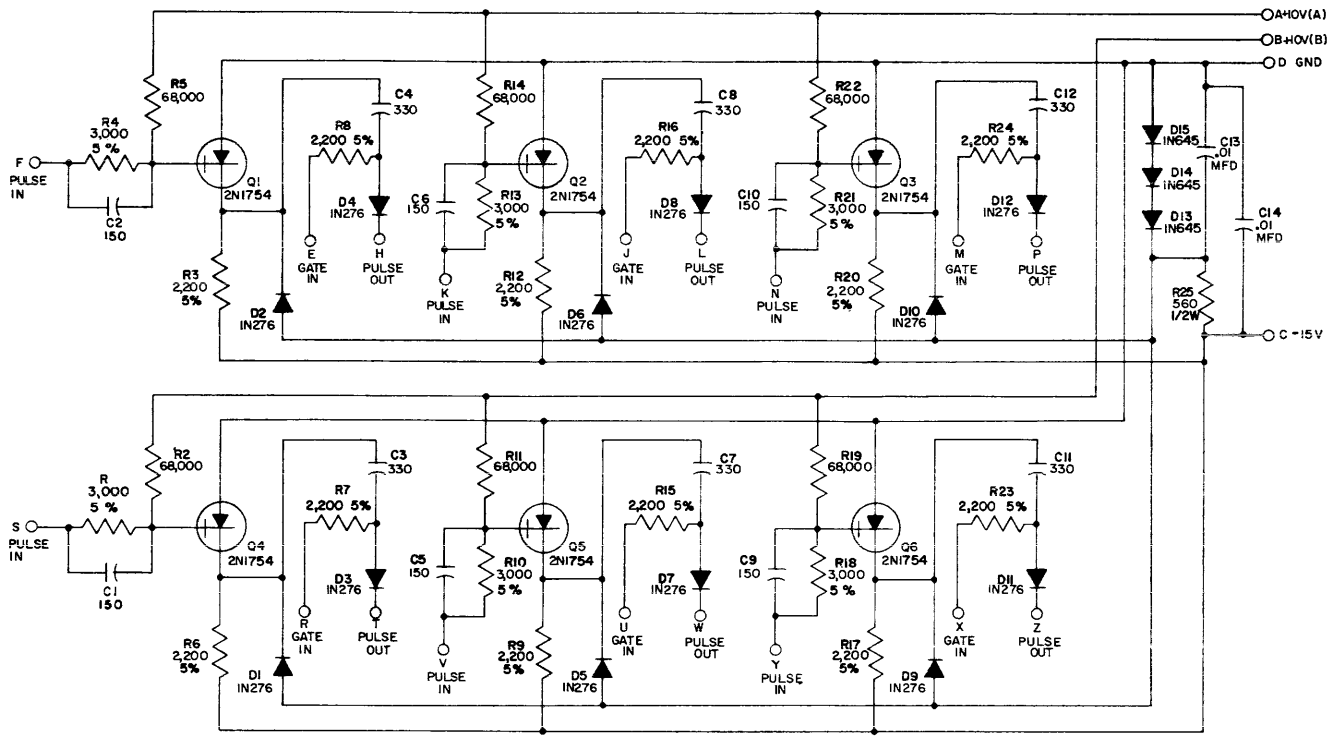


UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD

DIODE 4113

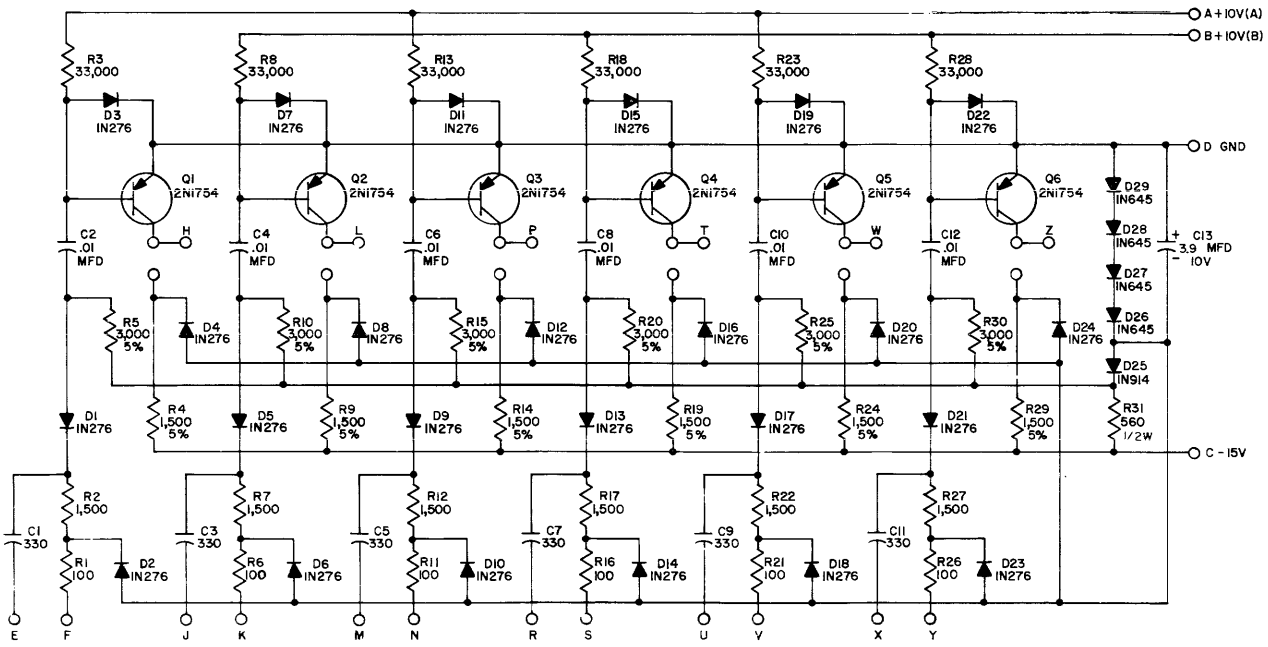
DIODE 4112

DIODE 4113



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD

CAPACITOR-DIODE 4126

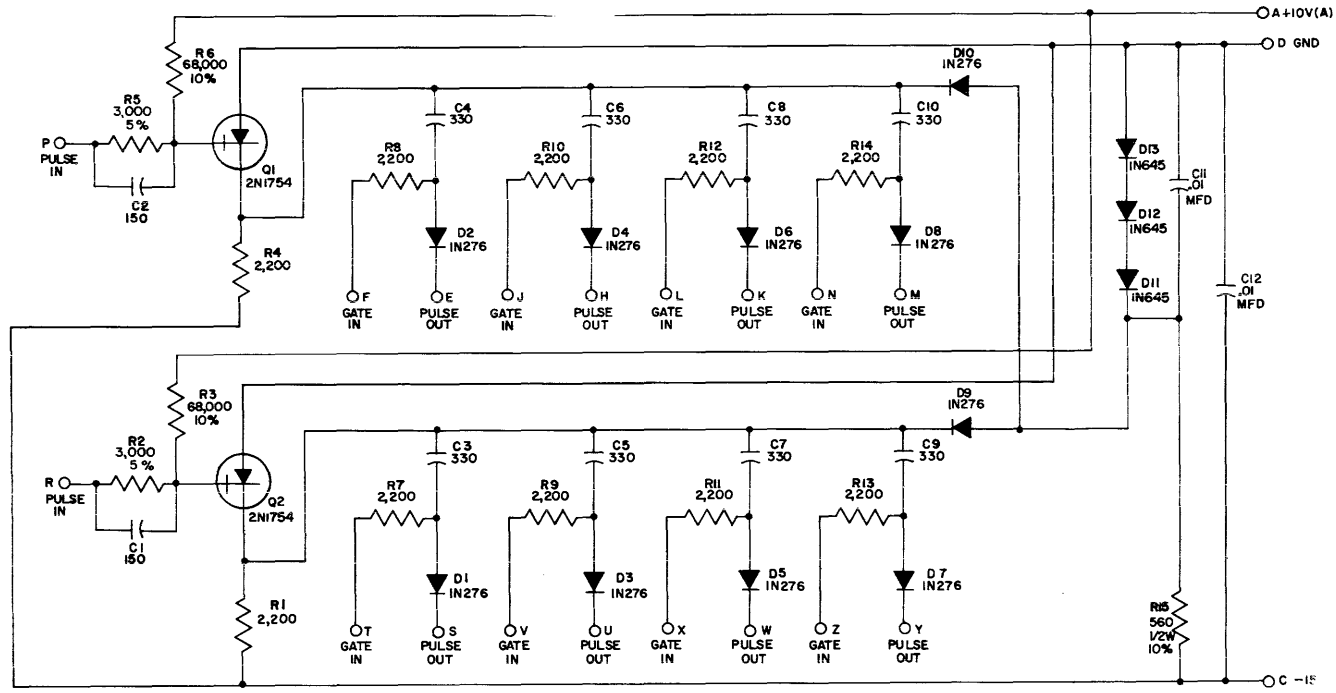


UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD

CAPACITOR-DIODE 4127

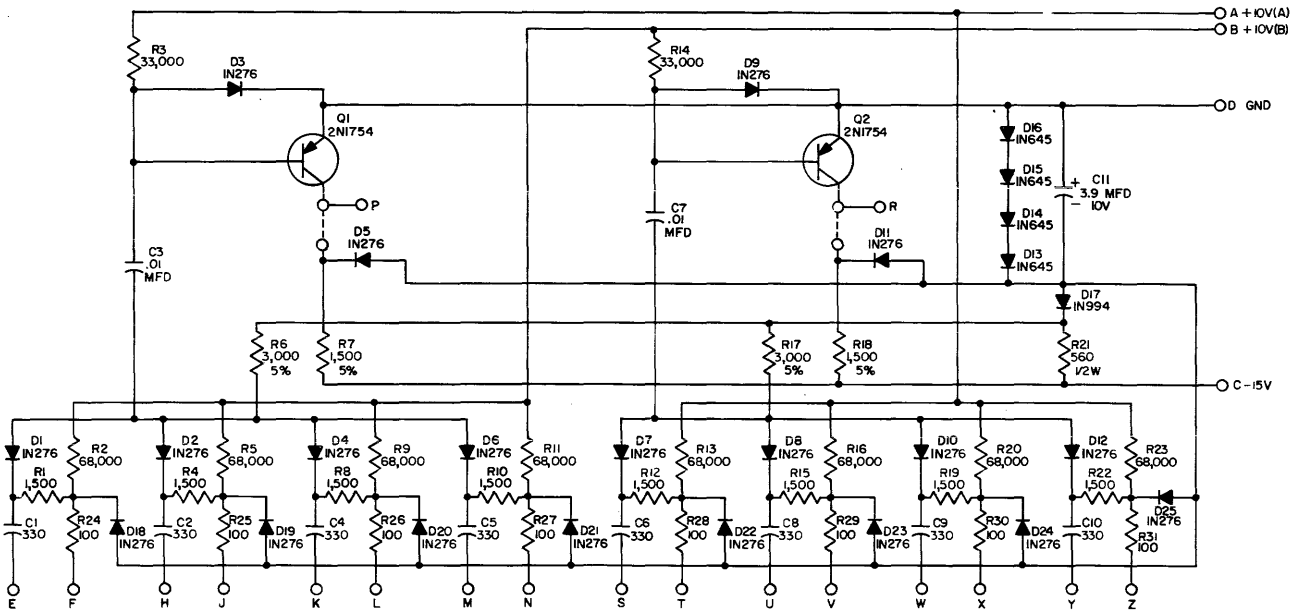
CAPACITOR-DIODE 4126

CAPACITOR-DIODE 4127



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 5%
CAPACITORS ARE MMFD.

CAPACITOR-DIODE 4128

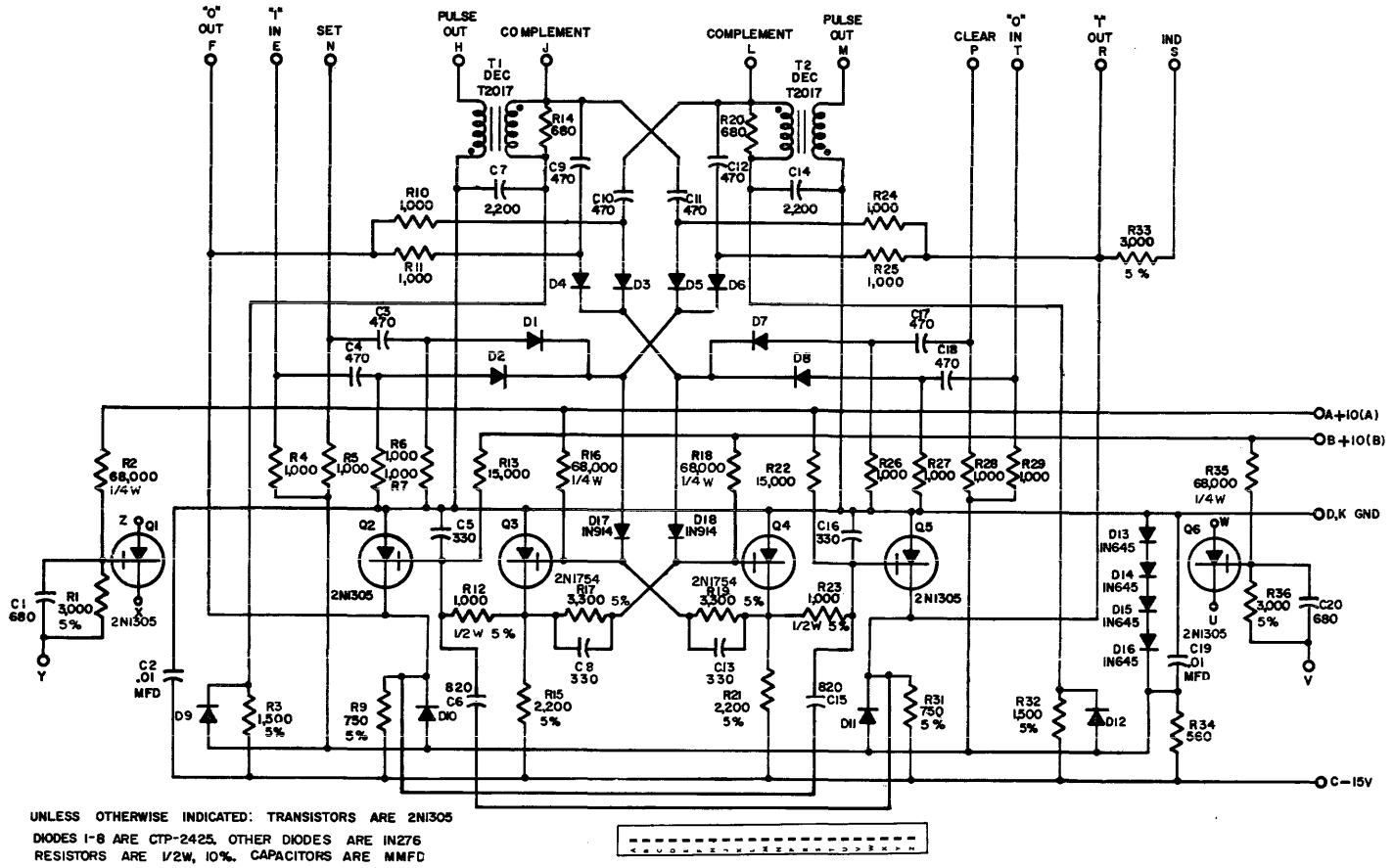


UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/4W, 10%
CAPACITORS ARE MMFD

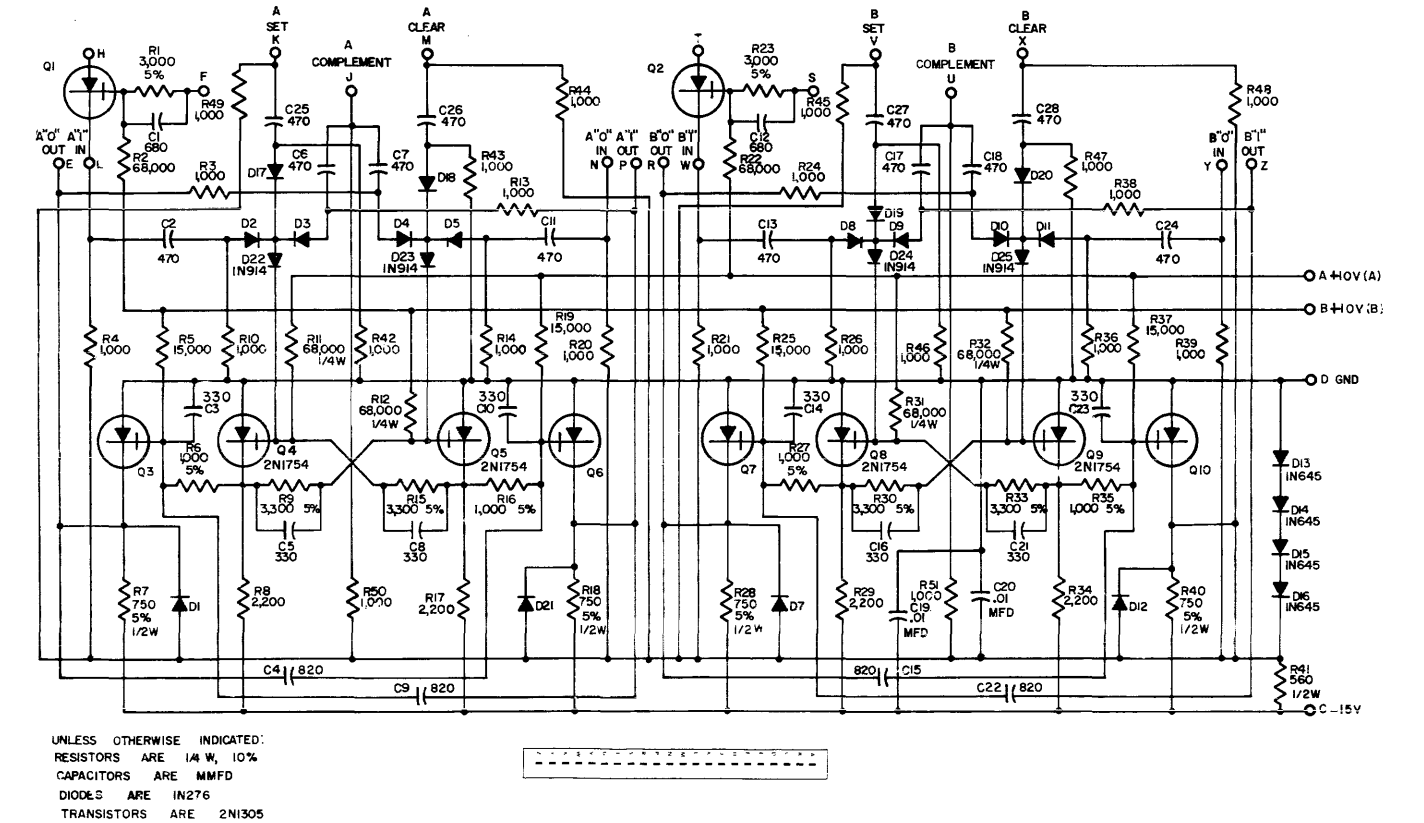
CAPACITOR-DIODE 4129

CAPACITOR-DIODE 4128

CAPACITOR-DIODE 4129



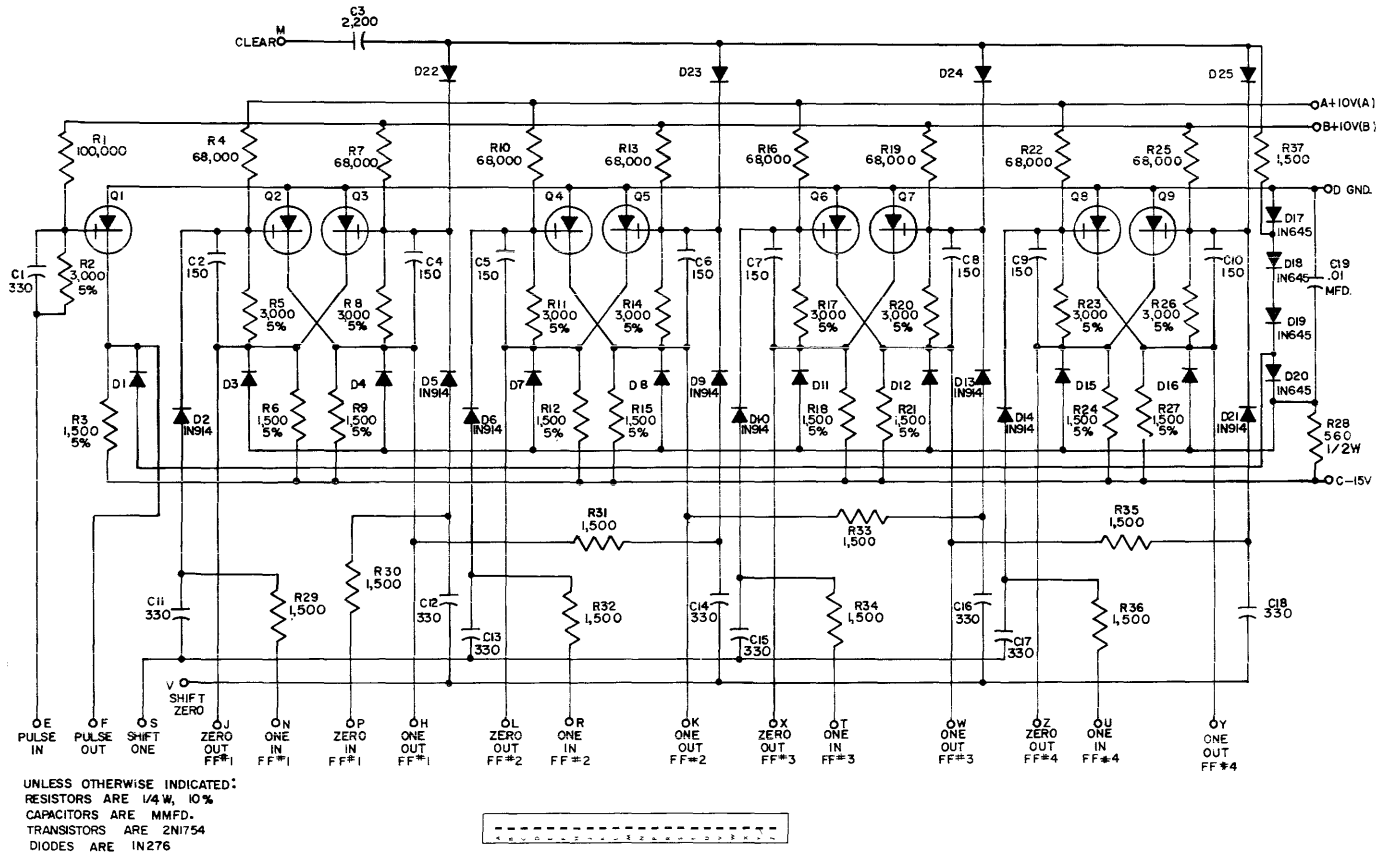
FLIP-FLOP 4201



DUAL FLIP-FLOP 4209

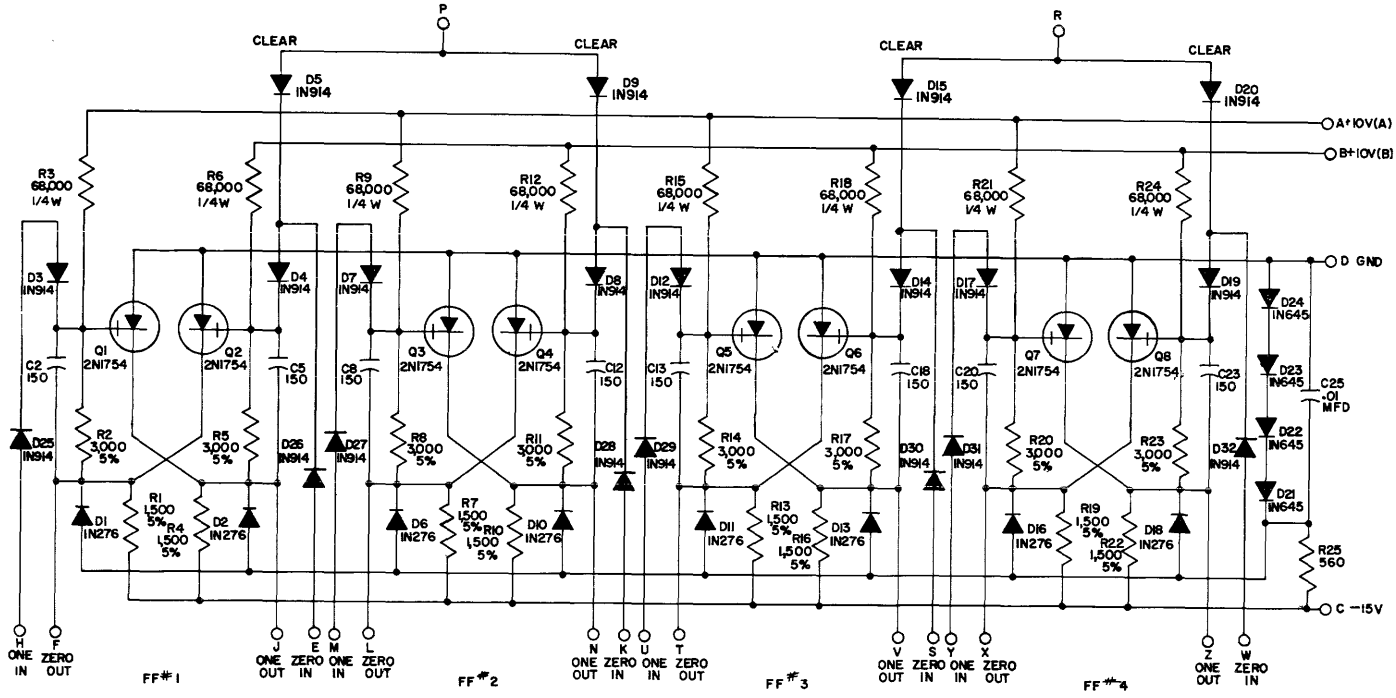
FLIP-FLOP 4201

DUAL FLIP-FLOP 4209



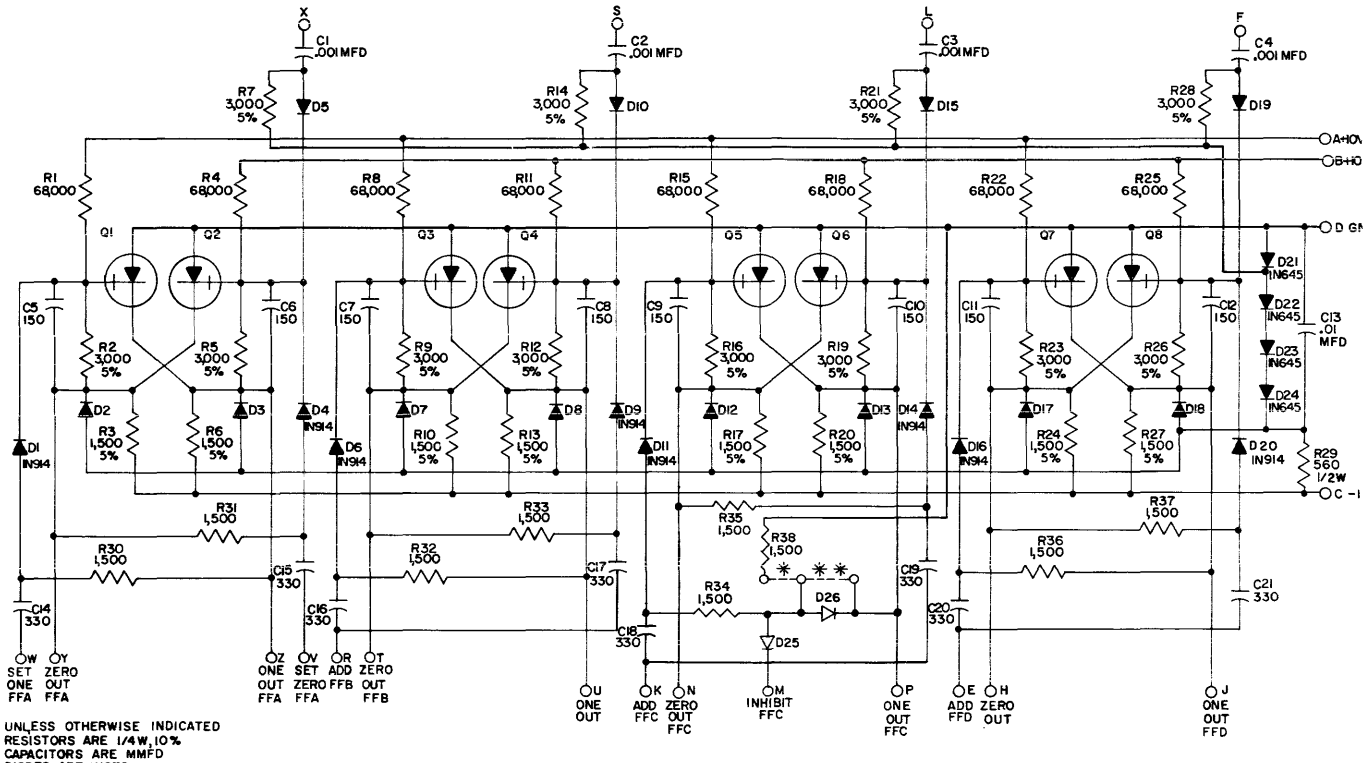
UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/4 W, 10%
CAPACITORS ARE MMFD.
TRANSISTORS ARE 2N1754
DIODES ARE IN276

QUADRUPLE FLIP-FLOP 4213



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2 W, 10%
CAPACITORS ARE MMFD

QUADRUPLE FLIP-FLOP 4214



UNLESS OTHERWISE INDICATED
RESISTORS ARE 1/4W, 10%
CAPACITORS ARE MMFD
DIODES ARE IN276
* THIS JUMPER IS INSTALLED DURING MANUFACTURING. IT IS NEEDED WHEN INHIBIT FEATURE IS DESIRED.
** THIS JUMPER IS NOT INSTALLED DURING MANUFACTURING. IT IS USED WHEN INHIBIT FEATURE IS NOT NEEDED AND STANDARD LOADING OF FFC IS DESIRED.
TRANSISTORS ARE 2N1499A

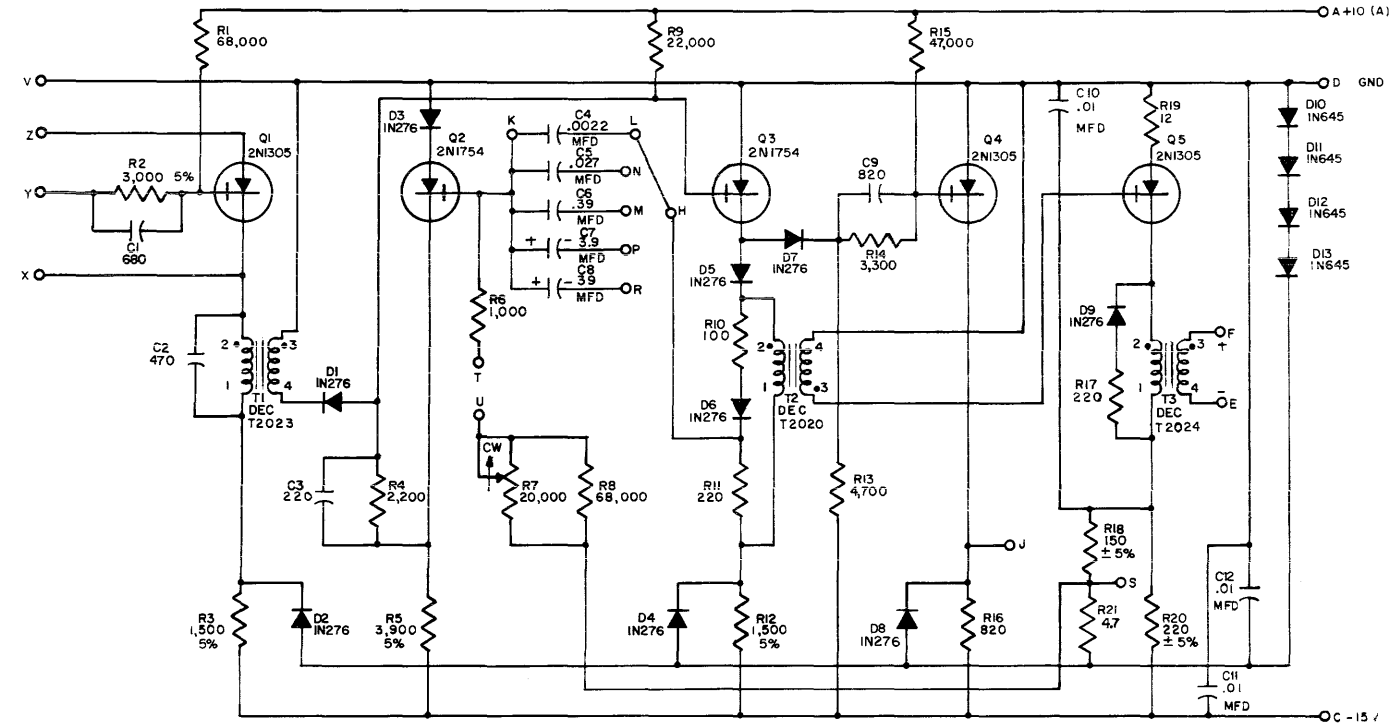
FOUR-BIT COUNTER 4215

QUADRUPLE FLIP-FLOP 4213

QUADRUPLE FLIP-FLOP 4214

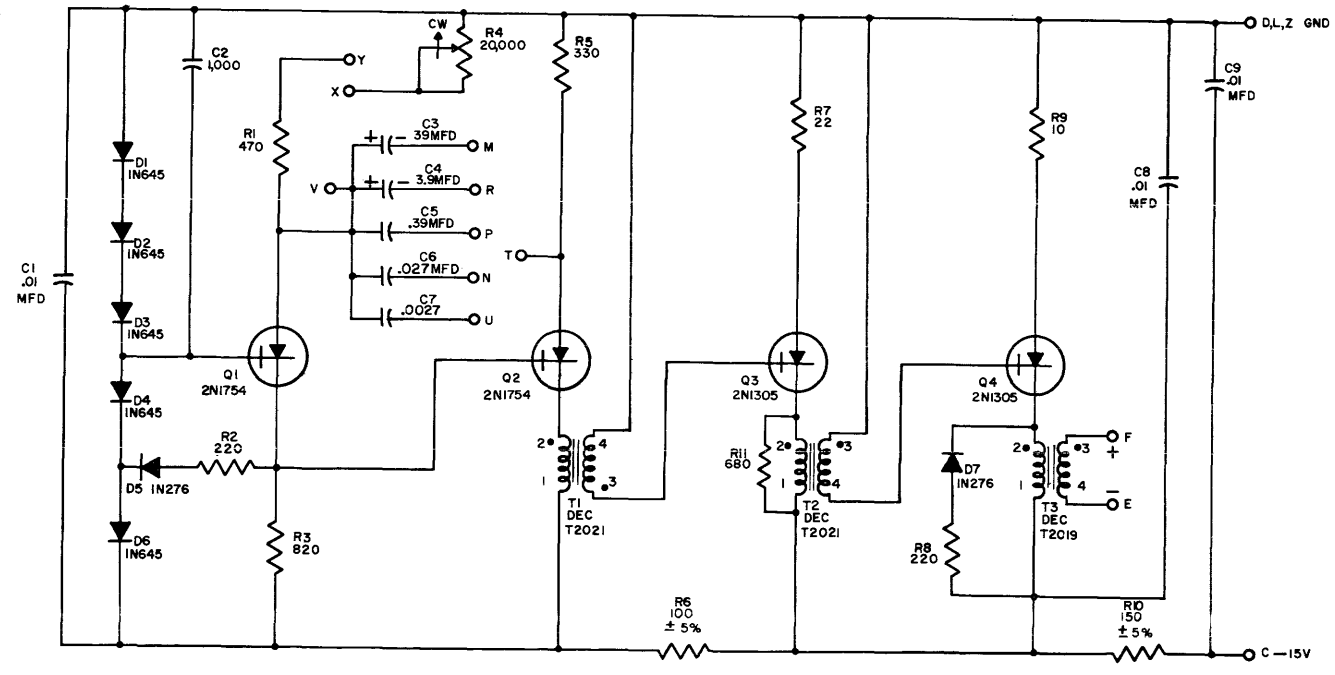
FOUR-BIT COUNTER 4215

A-142a



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD

DELAY 4301

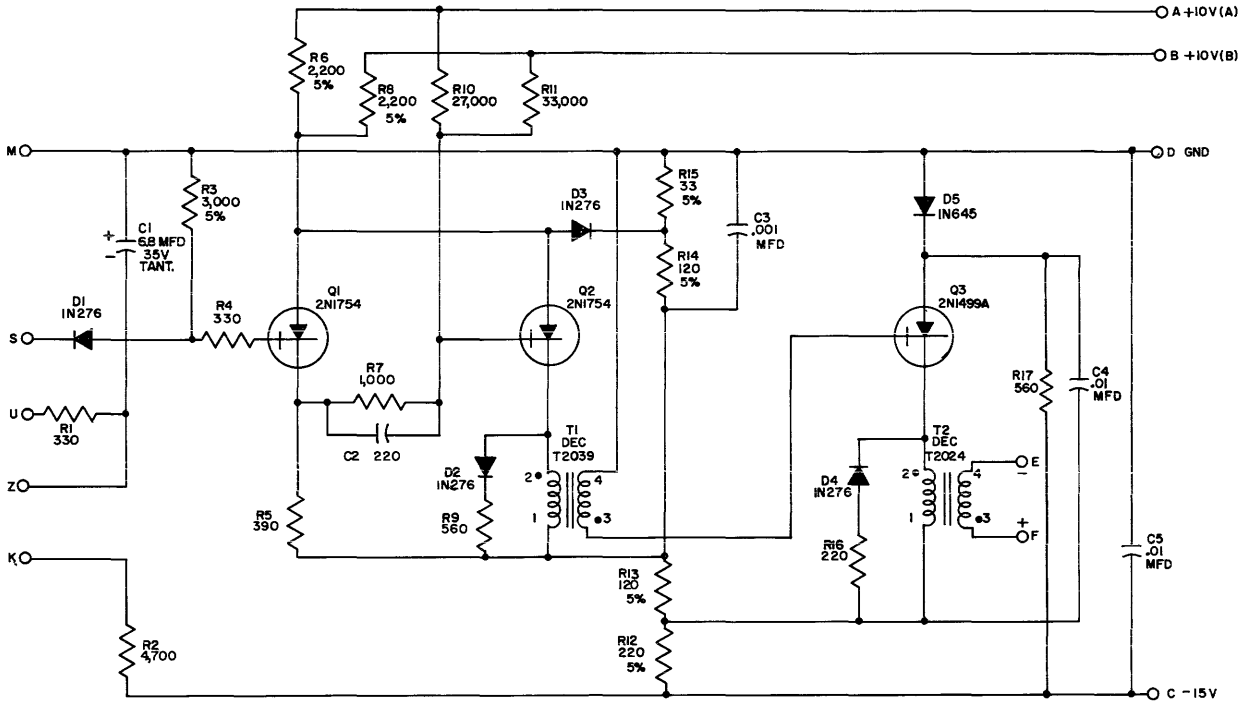


UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD

CLOCK 4401

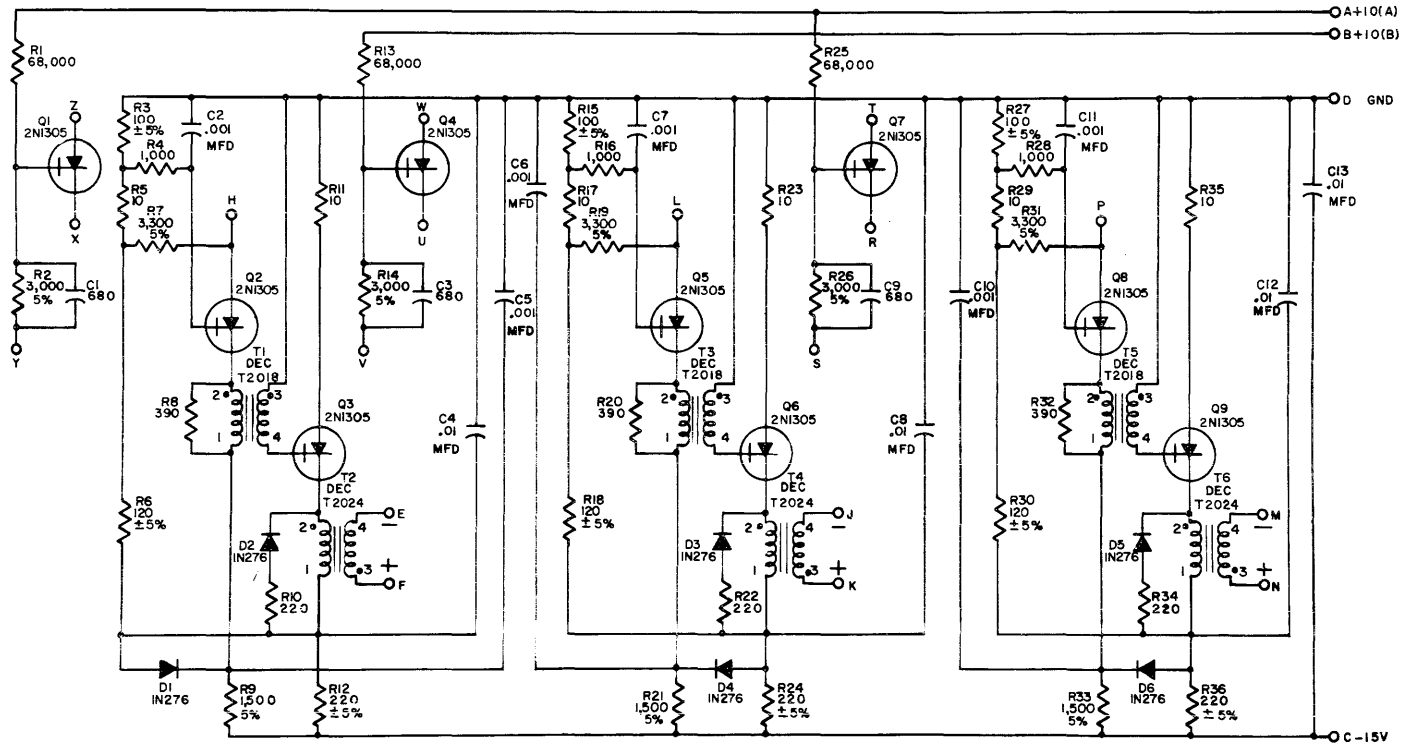
DELAY 4301

CLOCK 4401



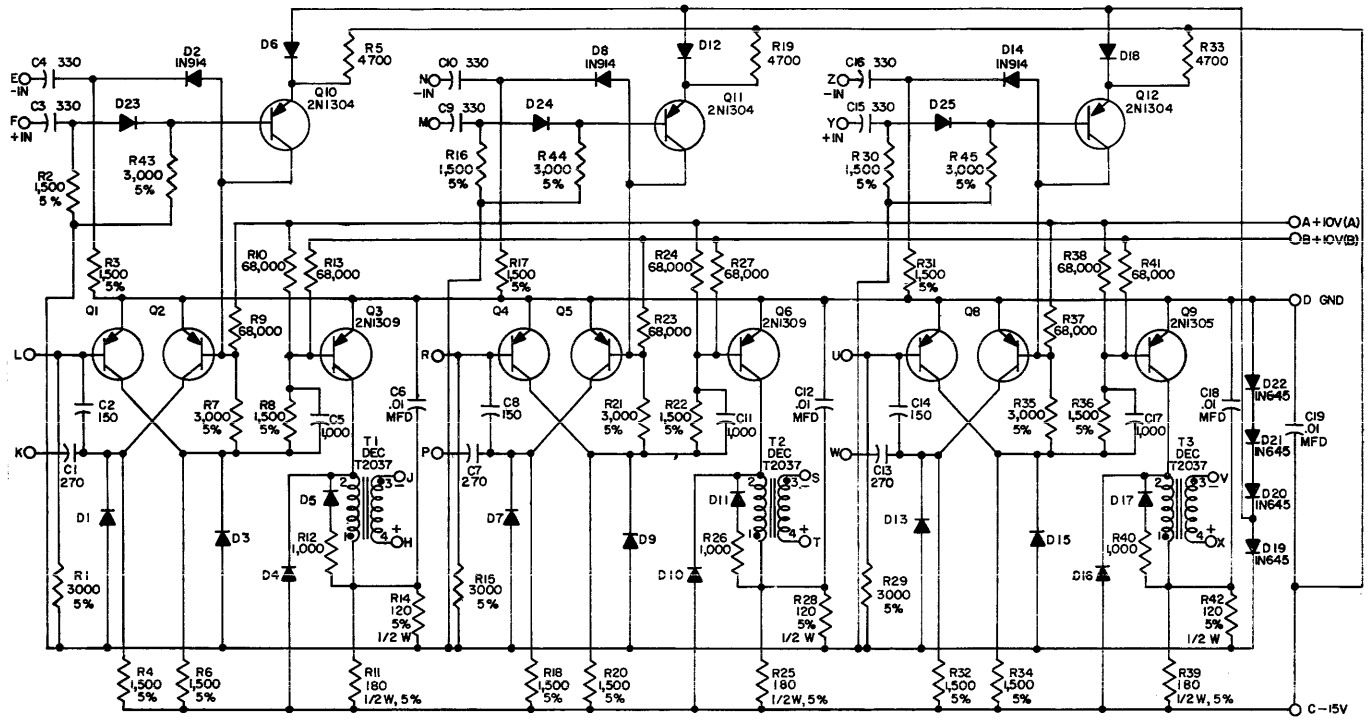
UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD

PULSE GENERATOR 4410



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD,

PULSE AMPLIFIER 4603



UNLESS OTHERWISE INDICATED
RESISTOR ARE 1/4 W 10%
CAPACITORS ARE MMFD
DIODES ARE IN276
TRANSISTORS ARE 2N1499A

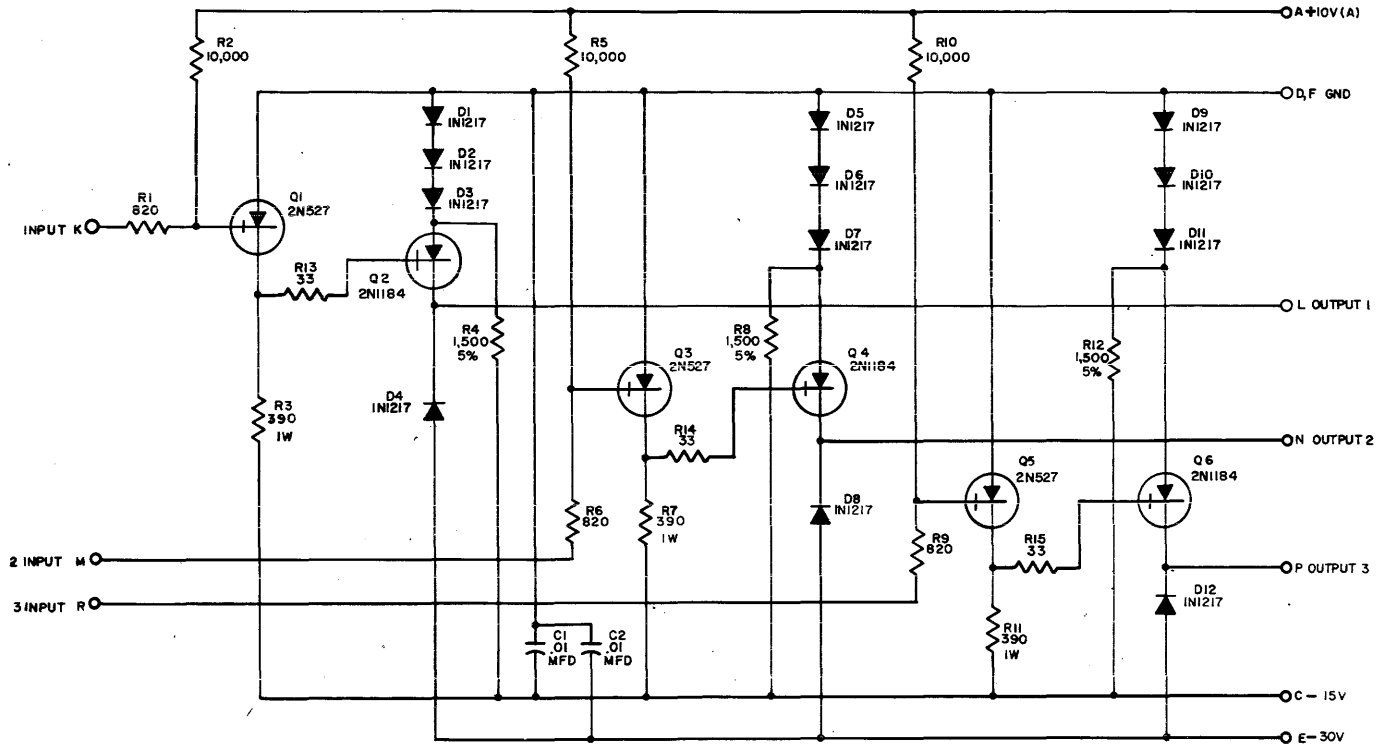
THIS SCHEMATIC IS FURNISHED ONLY FOR TEST AND MAINTENANCE PURPOSES. THE CIRCUITS ARE PROPRIETARY IN NATURE AND SHOULD BE TREATED ACCORDINGLY.

PULSE AMPLIFIER 4604

PULSE GENERATOR 4410

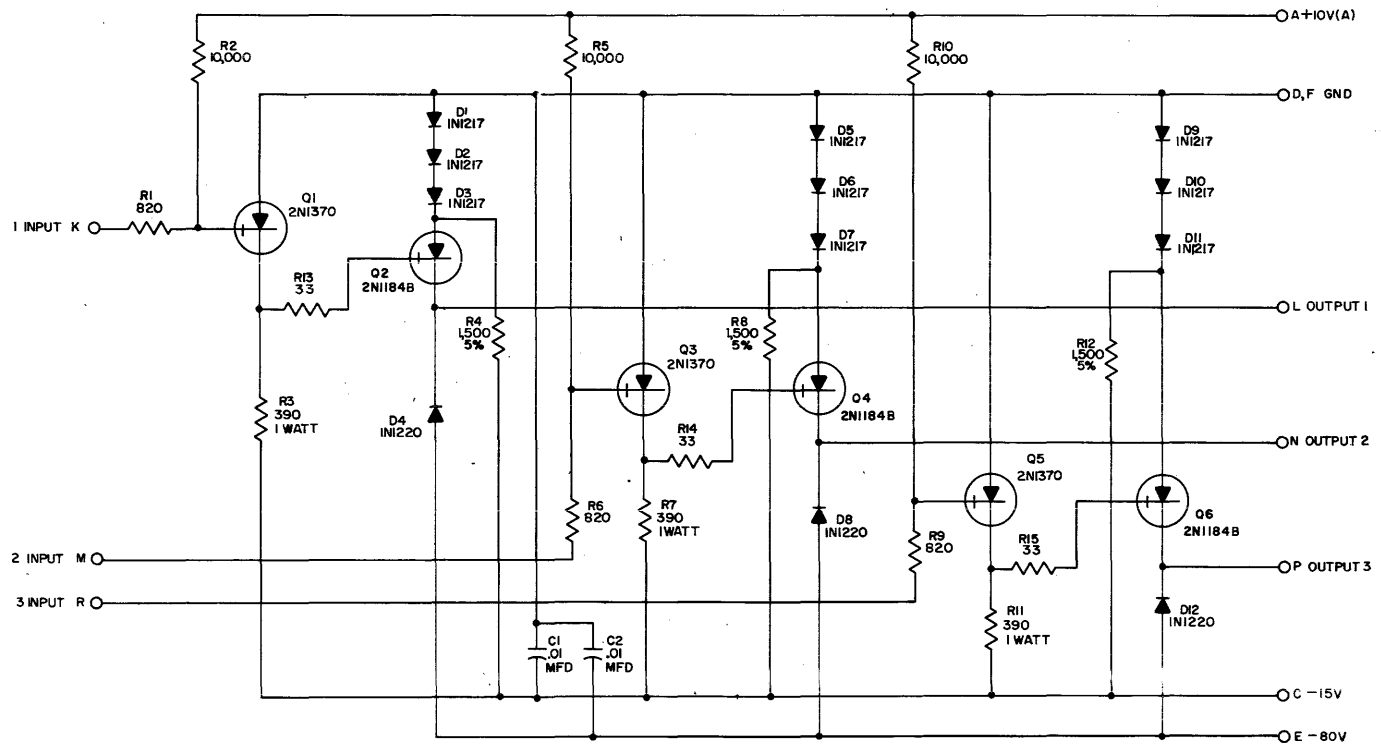
PULSE AMPLIFIER 4603

PULSE AMPLIFIER



UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD

SOLENOID DRIVER 4680

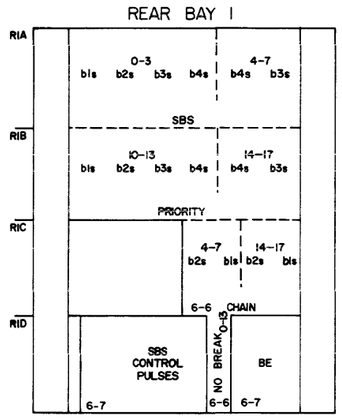


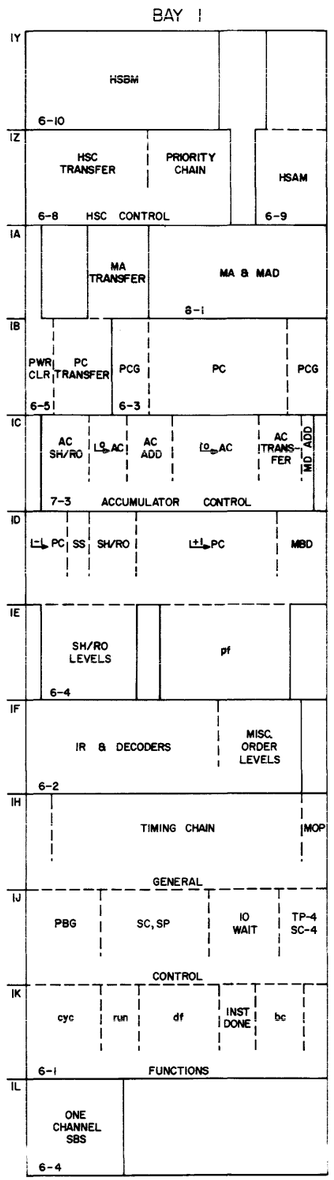
UNLESS OTHERWISE INDICATED:
RESISTORS ARE 1/2W, 10%
CAPACITORS ARE MMFD.

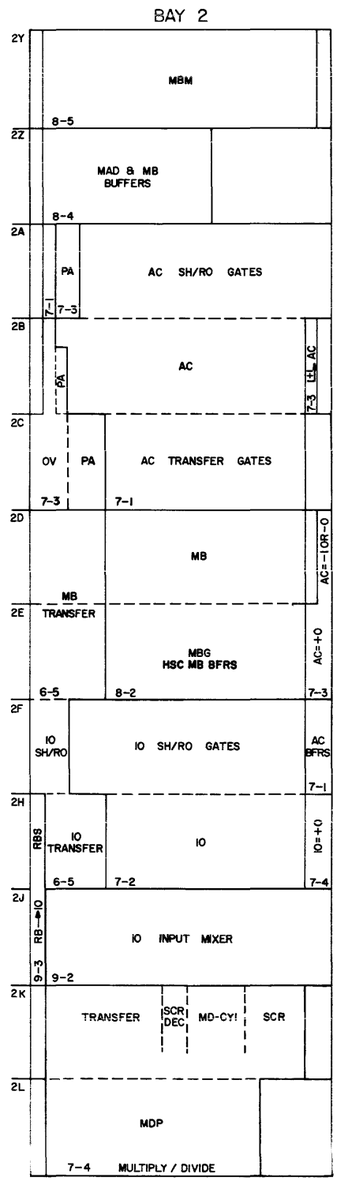
SOLENOID DRIVER 4681

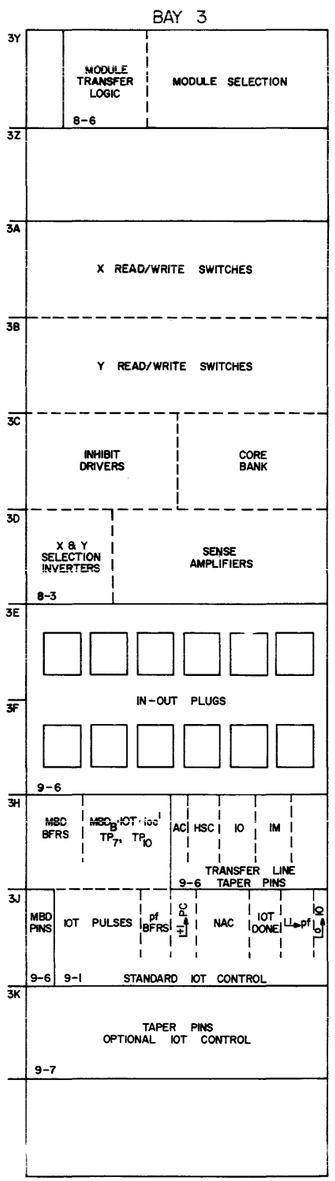
SOLENOID DRIVER 4680

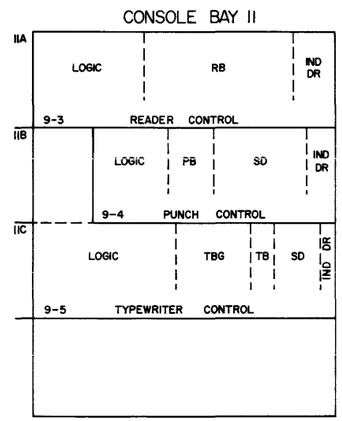
SOLENOID DRIVER 4681











NOTE
ALL FIGURE REFERENCES TO D-SIZE LOGIC DRAWINGS

Figure 11-1 Detailed Logic Layout, Central Frame

NOTES:

1. ■ REMOVE INTERNAL JUMPERS FROM LOAD RESISTORS ON MODULES SO DESIGNATED
2. (IL) ONE CHANNEL SBS NOT USED WITH SEQ BREAK TYPE 20
3. STAMP MODULES WITH R AS SHOWN

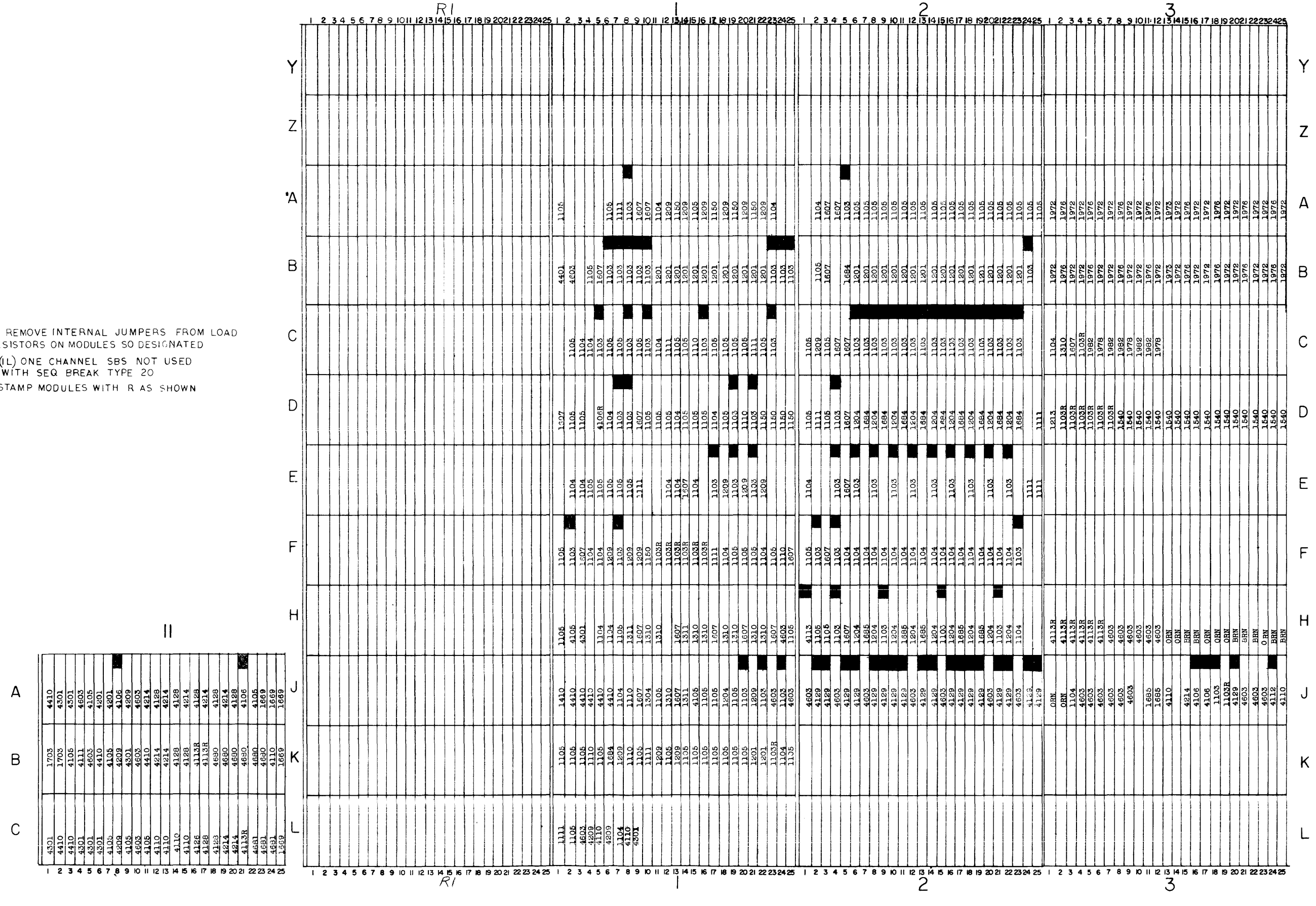


Figure 11-2 Central Frame Module Layout, Standard Computer

NOTES:

1. RI(ABCD)
   SEQUENCE BREAK SYSTEM
   TYPE 20
2. 1Y, 1Z & 2E
   HIGH SPEED CHANNEL OPTION
   TYPE 19
3. 2Y, 2Z & 3Y
   MEMORY EXTENSION OPTION
   TYPE 15
4. ■ REMOVE INTERNAL JUMPERS FROM LOAD RESISTORS ON MODULES SO DESIGNATED
5. 2K & 2L
   AUTOMATIC MULTIPLY/DIVIDE OPTION TYPE 20
6. OPTION TYPE SHOWN IN PARENTHESES FOR SINGLE MODULES
7. STAMP MODULES WITH R AS SHOWN

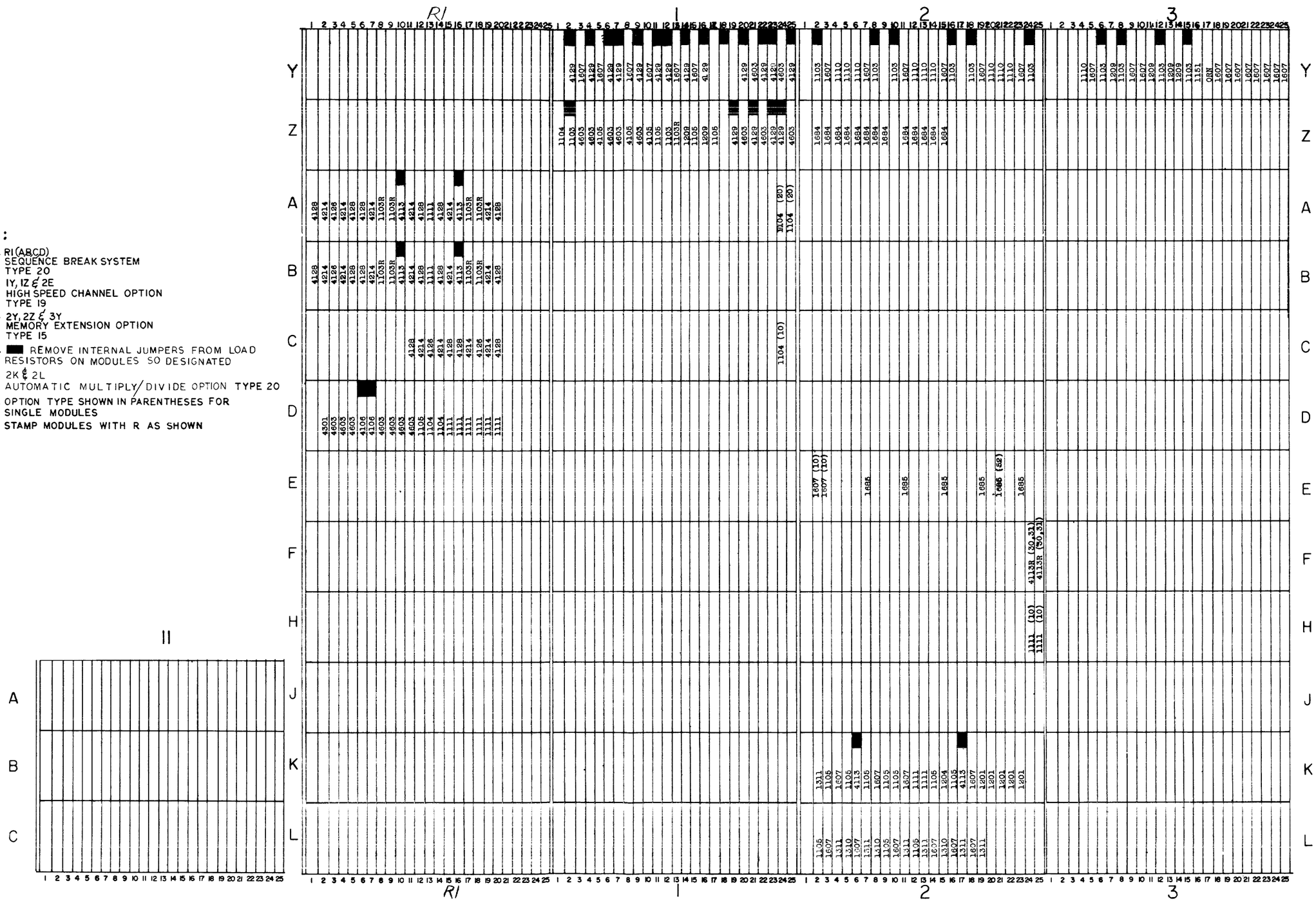


Figure 11-3 Central Frame Module Layout, Optional Equipment

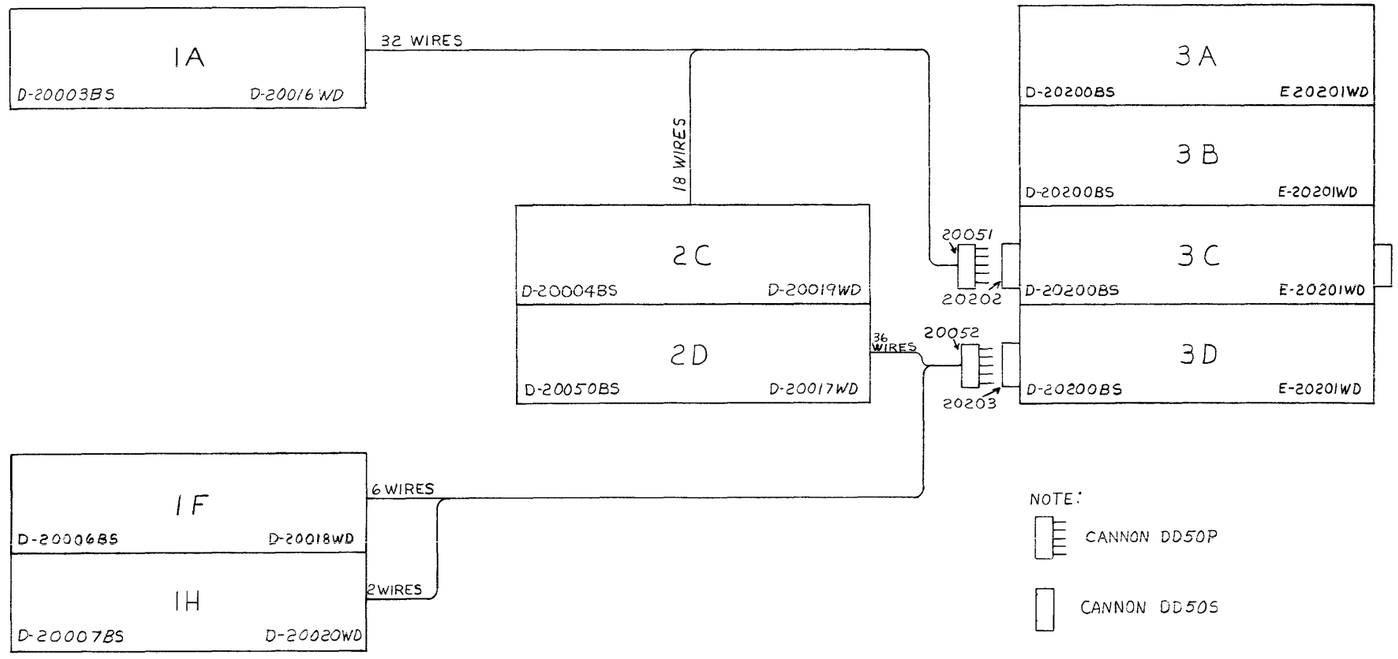


Figure 11-4 Standard Wiring Configuration for One Memory

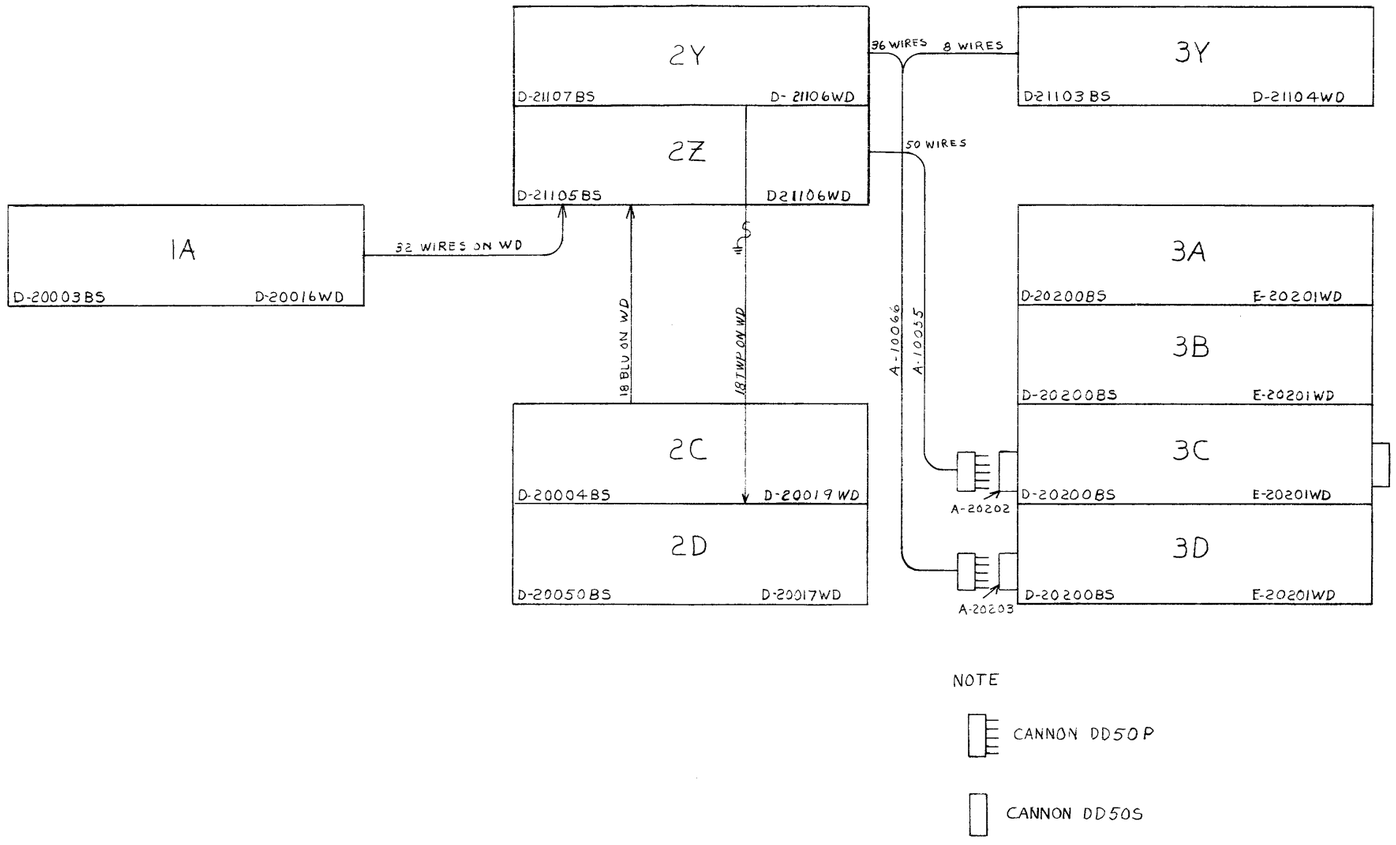


Figure 11-5 Wiring Configuration for Memory Extension Type 15

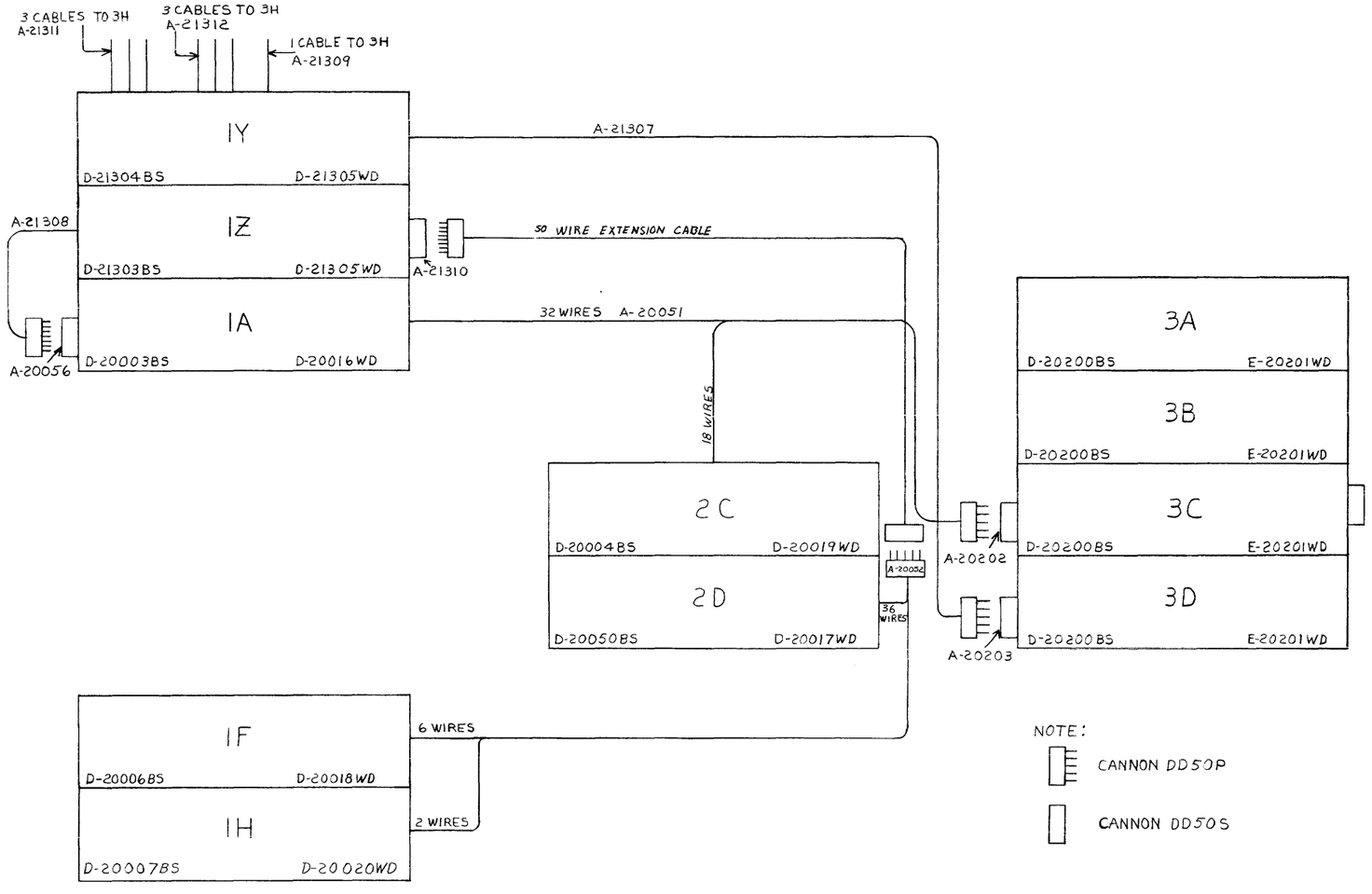


Figure 11-6 Wiring Configuration for High-speed Channel Type 19

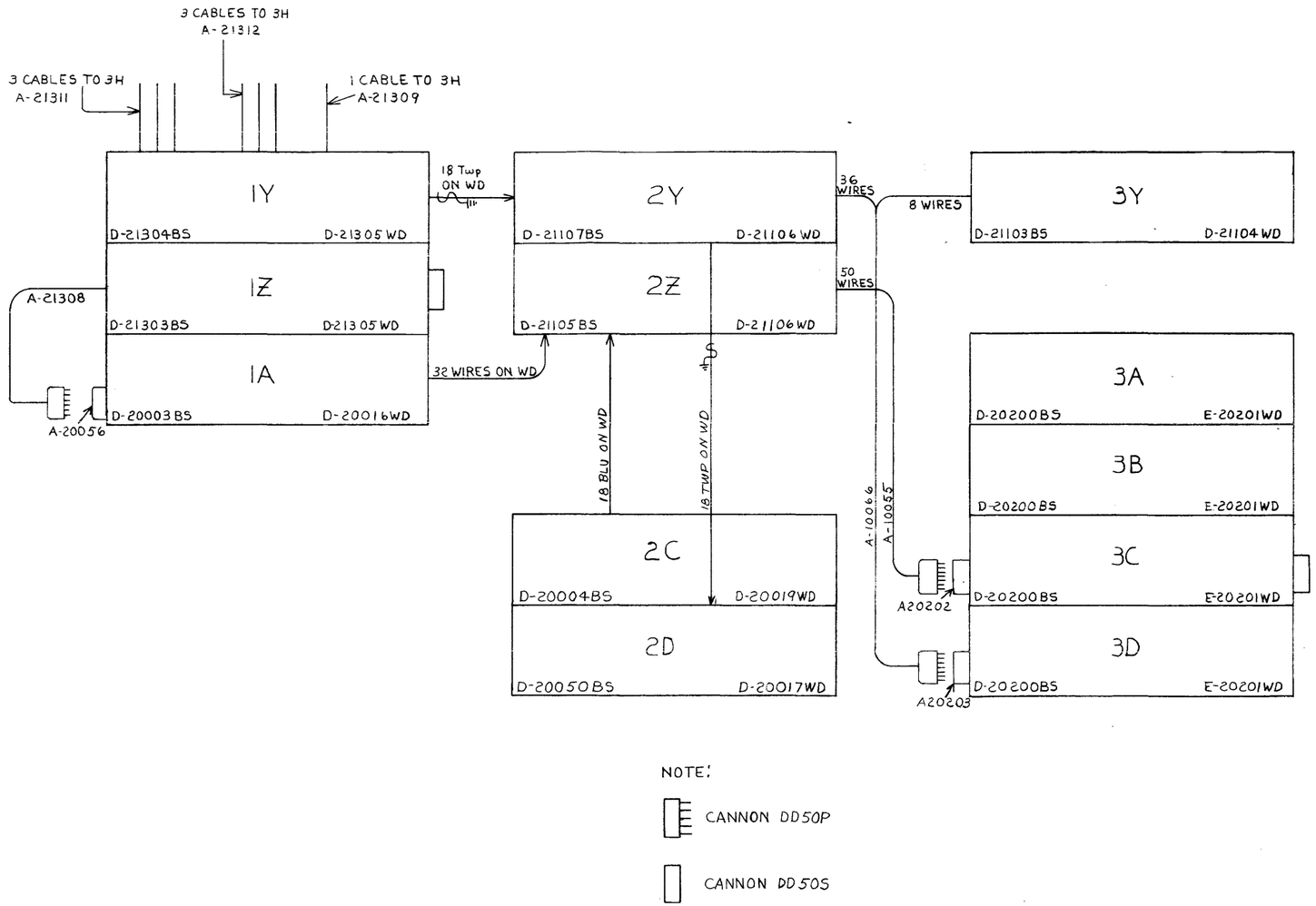


Figure 11-7 Wiring Configuration for High-speed Channel Type 19 and Memory Extension Type 15

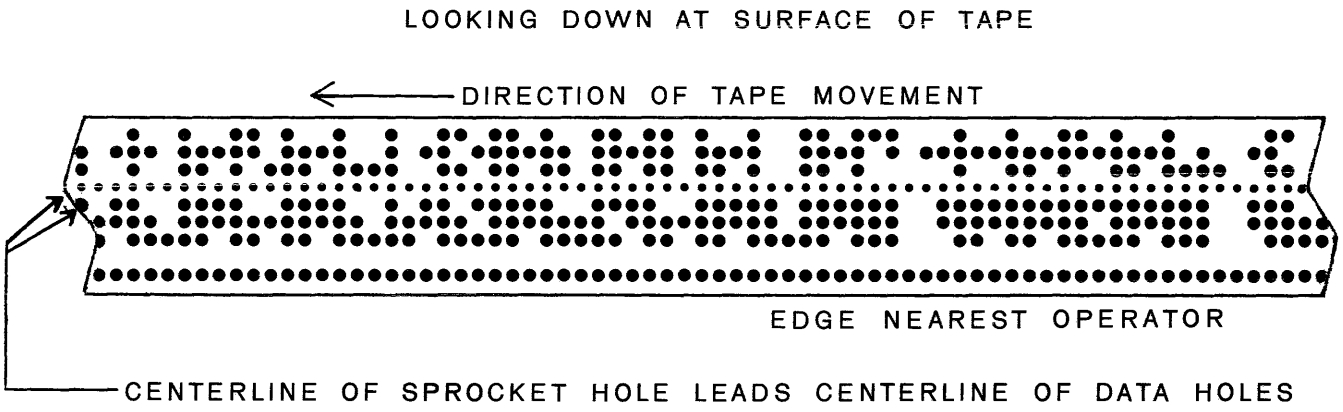


Figure 11-8 Correct Orientation for Loading Paper Tape

Figure 11-8