R. MOORHEAD

Digital Equipment Corporation
Maynard, Massachusetts

Maintenance Manual
PDP-9/L
Volume I

DEC-9/L-HMAA-D

PDP-9/L Maintenance Manual Volume I

DIGITAL EQUIPMENT CORPORATION □ MAYNARD, MASSACHUSETTS

First Printing February 1969

Instruction times, operating speeds and the like are included in this manual for reference only; they are not to be taken as specifications.

CONTENTS

Page

CHAPTER 1
INTRODUCTION

CHAPTER 2
CONTROLS AND INDICATORS

CHAPTER 3
SYSTEM DESCRIPTION

CONTENTS (Cont)

CHAPTER 4
CONTROL MEMORY SYSTEM

CHAPTER 5
CENTRAL PROCESSOR

CHAPTER 6
CORE MEMORY SYSTEM

CONTENTS (Cont)

CHAPTER 7
CONSOLE POWER CONTROLS

CONTENTS (Cont)

Page

CHAPTER 8
INPUT/OUTPUT CONTROL

CHAPTER 9
ASR-33 TELETYPEWRITER

CONTENTS (Cont)

CHAPTER 10
MAINTENANCE

APPENDIX A
BASIC INSTRUCTION REPERTOIRE

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

ILLUSTRATIONS (Cont)

TABLES

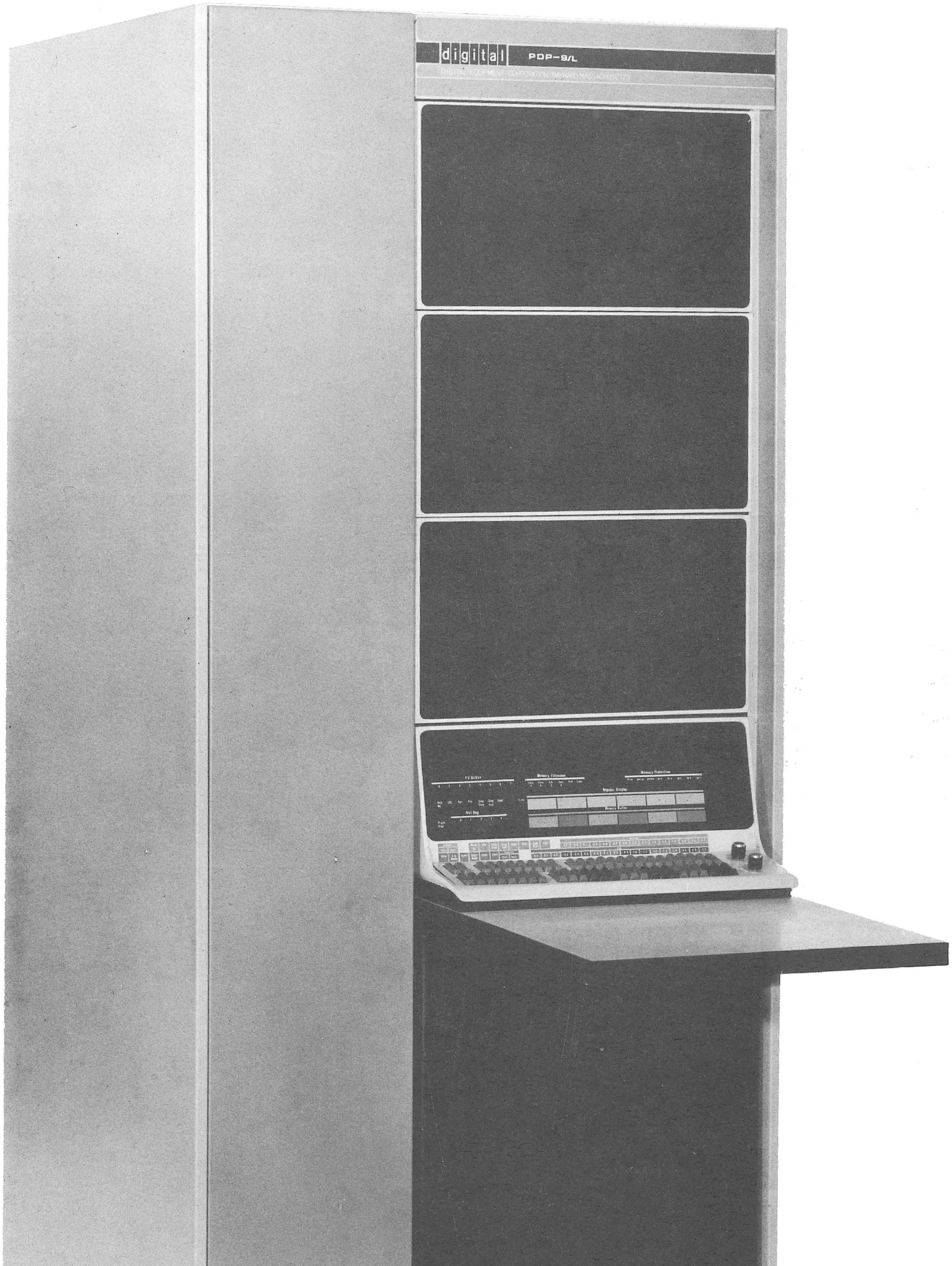


Figure 1-1   Programmed Data Processor PDP-9/L

CHAPTER 1
INTRODUCTION

This manual is one of several documents related to the PDP-9/L. It provides the user with a basic understanding of the system capabilities and assumes that the user is familiar with the technology of similar computer systems. For complete and comprehensive coverage in his area of interest, the user should refer to the documents listed in Table 1-1 at the end of this chapter.

Operation and maintenance information for the Programmed Data Processor PDP-9/L, manufactured by Digital Equipment Corporation, Maynard, Massachusetts are provided in this manual. It consists of two volumes. Volume I describes the basic computer system and discusses the logic circuits in terms of the computer's instruction repertoire. Manual operations and maintenance considerations are also included in this volume. Volume II contains a complete set of engineering drawings for the basic computer system.

1.1 PHYSICAL DESCRIPTION

With the exception of the ASR33 Teletype Unit, the basic PDP-9/L (Figure 1-1) is self-contained in a single DEC Type CAB-31 metal cabinet. Four casters permit cabinet mobility. No special power sources, air conditioning or floor bracing are required. The teletype unit is supplied with its own mounting stand. Figure 1-2 shows the front and rear dimensional views of the PDP-9/L layout.

Logic modules are mounted in three wings at the rear of the cabinet. These wings, each measuring approximately 22 in. by 30 in., swing out as a single door for module access. The wings also include self-contained cooling fans and marginal check switches. The top wing holds the 4096-word core memory system; the middle wing, the central processor; and the bottom wing, the I/O control section. Each wing has its own switched power-distribution system at the marginal check switches on the fan housings.

Several commonly purchased options are prewired into the wings. The central processor wing is prewired for the extended arithmetic element option. The I/O control wing is prewired for the DEC Type 34HL Oscilloscope Display Control, the automatic priority interrupt, and the power failure detection options. Control logic for memory extension, memory parity, and memory protection options is also wired in the two wings.

1.2 FUNCTIONAL DESCRIPTION

The PDP-9/L is a general purpose, solid-state digital computer designed for data handling in a scientific laboratory, a computation center, or in a real-time process control system. Figure 1-3 is a functional block diagram of the PDP-9/L GPC. The system is a single-address 18-bit computer using 1's complement arithmetic which is program-convertible to 2's complement notation to facilitate multiple precision operations. Indirect addressing to one level and autoindexing features afford programming flexibility. The 4096-word core memory provides random access to any word within 1.5 μs.

The I/O bus system accommodates up to 64 low-speed peripheral devices under program control, up to 8 high-speed devices optionally multiplexed in 8 device-controlled data channels (DCH), and up to 28 devices in an optional 32-channel automatic priority interrupt (API) system. The program-controlled transfer system includes the Teletype printer with a reader/punch. Program-controlled transfer operations include program interrupt (PI), I/O skip, and I/O status checking facilities.

The memory bus provides for memory expansion up to 32,768 words of memory.

A DEC Type 712 Power Supply provides the DC voltages required for the PDP-9/L system and the optional paper tape reader/punch. Voltages of +10V and -15V are supplied to the basic PDP-9/L system, -30V to core memory, and +10V and -15V to the tape reader/punch. A variable 0 to 20V output is available to check the operation of the system under marginal power supply limits. By substituting this output for the normal + 10V and -15V outputs, an existing fault or potential system failure can be isolated.

1.3 REFERENCE DOCUMENTS AND PROGRAMS

Tables 1-1 and 1-2 list the standard maintenance documents and program tapes supplied with the basic PDP-

9/L. Others may be furnished as appropriate to customer requirements.

1.4 REFERENCE CONVENTIONS

a. Numerical Notation - Unless otherwise indicated, all number representations are in octal notation.

b. Circuit References - All references to logic signals include the module type designation, module location code, and output pin designation; e.g., Pulse Amplifier B602-E26D means that module B602 is located in rack E, slot 26, and the output signal is taken from pin D. All racks are designated alphabetically from left to right as viewed from the module mounting side. All module mounting slots are numbered 01 through 40 from top to bottom. Modules are mounted horizontally in the slots.

Dual-width modules carry dual location designations; e.g., G219-AB16AF, where AF designates pin F in the A slot of dual slot location AB.

c. Signal Mnemonics - Uncommon mnemonics are explained parenthetically the first time that they are mentioned in the discussion; e.g., KDN (key deposit next). A glossary of all signal mnemonics and their logic drawing origins is provided in Volume II.

d. Illustrations - References to in-text illustrations include the chapter prefix number - Figure 3-10 is the tenth illustration in Chapter 3.

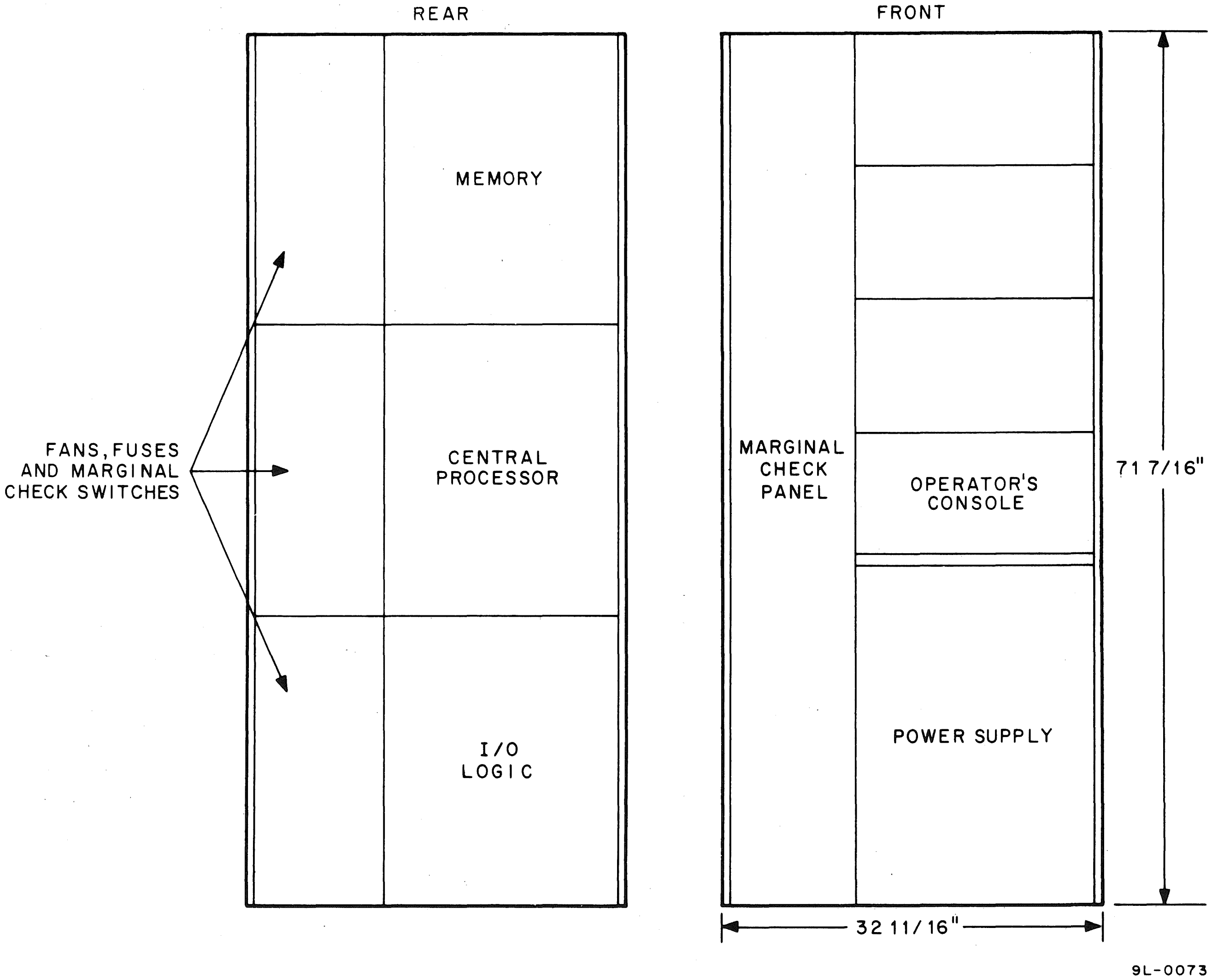
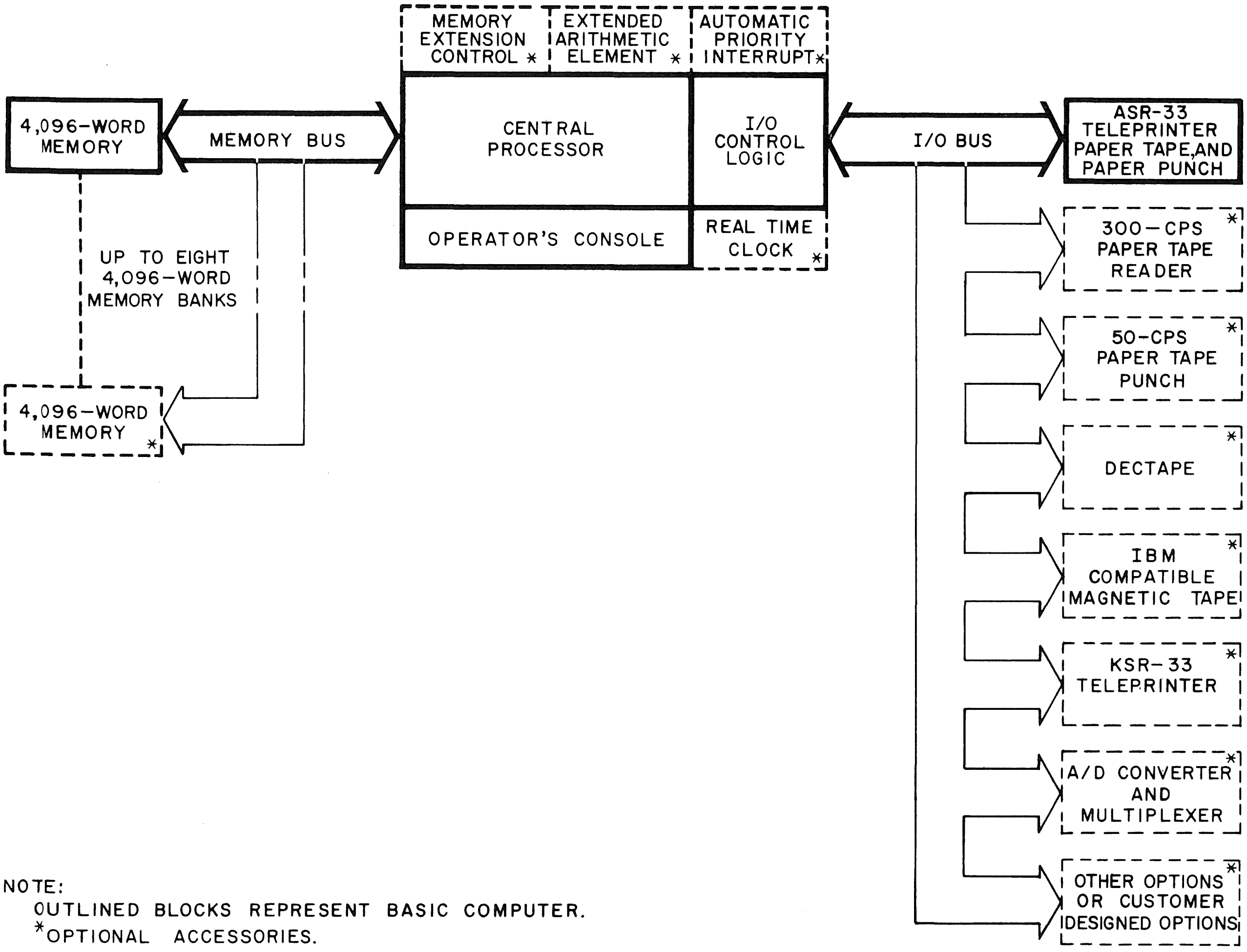


Figure 1-2 Basic PDP-9/L Layout, Front and Rear


MEMORY EXTENSION CONTROL *
EXTENDED ARITHMETIC ELEMENT *
AUTOMATIC PRIORITY INTERRUPT*
4,096-WORD MEMORY
MEMORY BUS
CENTRAL PROCESSOR
I/O CONTROL LOGIC
I/O BUS
ASR-33 TELEPRINTER PAPER TAPE,AND PAPER PUNCH
OPERATOR'S CONSOLE
REAL TIME CLOCK *
UP TO EIGHT 4,096-WORD MEMORY BANKS
4,096-WORD MEMORY *
300-CPS PAPER TAPE READER *
50-CPS PAPER TAPE PUNCH *
DECTAPE *
IBM COMPATIBLE MAGNETIC TAPE *
KSR-33 TELEPRINTER *
A/D CONVERTER AND MULTIPLEXER *
OTHER OPTIONS OR CUSTOMER DESIGNED OPTIONS *
NOTE:
OUTLINED BLOCKS REPRESENT BASIC COMPUTER.
*OPTIONAL ACCESSORIES.

9L-0072

Figure 1-3 PDP-9/L System Configuration and Optional Accessories Block Diagram

Table 1-1
Reference Documents

Document Number	Title	Publisher
C-105	Logic Handbook	DEC
DEC-9L-GRVA-D	PDP-9/L User Handbook	DEC
Bulletin 310B	ASR33 Technical Manual	Teletype Corp.
Bulletin 1184B	KSR33 Teletype, 33 Page Printer Set, Parts	Teletype Corp.

e. Drawings - Logic drawings are identified in the text by a literal prefix code and a single numeric; e.g., drawing KC20(2), where KC denotes a central processor drawing and the parenthetic portion denotes sheet 2 of a multiple-sheet drawing. Other literals are: MC, core memory drawings, and KD, I/O control drawings. Complete drawing codes appear on the drawings themselves.

1.5 TERMINOLOGY

Terms used frequently throughout the text are defined below. Others are defined within the discussions themselves.

a. Core Memory - The major storage device containing the computer program and the results of program execution. Sometimes referred to as the main memory, as opposed to control memory.

b. Core Memory Cycle - The 1.5 μs read/restore or read/modify/write cycle during which a word is extracted, then restored or modified, and written into core memory. The word may be an instruction word, an effective address word, or a data word (operand).

c. Memory Reference Instruction - An instruction word containing a direct address or an indirect address of an operand in core memory, as opposed to augmented instruction.

d. Direct Address - The effective address in a memory reference instruction word of a location in core memory which contains an operand.

e. Effective Address - The actual address of an operand in core memory. The effective address may be a direct address in a memory reference instruction word, or an address in core memory which is addressed by an indirect address in a memory reference instruction word.

f. Indirect Address - The address in a memory reference instruction word of a location in core memory which contains an effective address.

g. Augmented Instruction - An instruction word (OPR,IOT) that does not contain an address of an operand in core memory. The portion of the instruction usually reserved for an address contains microcoded computer commands. These commands are executed during either normal (OPR) or extended (IOT) computer fetch cycles, as opposed to memory reference instruction.

h. Computer Fetch Cycle - The 1.5 μs period during which a core memory cycle extracts and restores an instruction word. Instruction words are addressed sequentially by an incrementing program counter (PC) in the central processor, unless otherwise stipulated by program developments.

i. Computer Defer Cycle - The 1.5 μs period during which a core memory cycle extracts and restores an effective address word. A defer cycle follows a fetch cycle whenever the fetched instruction word contains an indirect address. If the indirect address refers to auto-index locations 10-17 in core memory, the effective address is incremented by 1 during defer; the operand is taken from the location designated by the incremented effective address during the computer execute cycle.

j. Computer Execute Cycle - The 1.5 μs period during which a core memory cycle extracts and restores an operand addressed by a memory reference instruction word or by an effective address word. The operand is manipulated in the computer in accordance with the op code of the instruction word. The restored word may be the original operand, or may be an operand modified by the manipulation process. In the latter case, the original operand is lost.

k. Computer IA0 Cycle - The 1.5 μs execute period for certain instructions which ignore and replace completely the operands that they address, as opposed to the modifying operations of the normal computer execute cycle.

l. Op Code - Operation code. A portion of a memory reference instruction word or an augmented instruction word that defines the operation to be executed.

m. Control Memory - The magnetic storage device in the central processor which issues timed, sequential gating levels to process or execute the instruction. The levels are strobed out of the read-only control memory in the form of 36-bit process words, each bit representing a gate-on or gate-off condition. Process word storage locations are addressed on the basis of the decoded instruction word op codes and previous processing results.

n. Program Break - A data channel transfer request that "break" program control at completion of the current instruction and suspends execution of the program in progress until the current word transfer is completed.

Table 1-2
Maintenance Program Documents

Name	Number
Instruction Test Part 1	9L-DOA1-PH
Instruction Test Part 1A	9L-DOA2-PH
Instruction Test Part 2	9L-DO2A-PH
ISZ Test	9L-DOBA-PH
Memory Address Test	9L-DOCA-PH
JMP Self Test	9L-DODA-PH
JMP-Y Interrupt Test	9L-DOEA-PH
JMS-Y Interrupt Test	9L-DOFA-PH
Basic Memory Checkerboard (Low)	9L-DIB1-PH
Basic Memory Checkerboard (High)	9L-DIB2-PH
Extended Memory Checkerboard	9L-DIBB-PH
9L ASR 33/35 Teletype Test Part 1	09-D2AB-PB
9L ASR 33/35 Teletype Test Part 2	09-D2BB-PB

o. Program Interrupt - An interruption in the program which is caused by a device service flag. Interrupt is granted at the completion of current instruction. The status of the program interrupted is stored in memory location 00000 at the grant of the interruption.

1.6 ENGINEERING DRAWINGS AND CIRCUIT SCHEMATICS

A complete set of engineering drawings and module circuit schematics is delivered with the PDP-9/L. Volume II contains a set of engineering drawings indexed by their drawing number codes; these drawings apply to the basic system only. Logic symbols used on the drawings are defined in the Logic Handbook, Document C-105.

1.7 SYSTEM SPECIFICATIONS

Functional Characteristics

Word Length	18 bits
Cycle Time	1.5 μs
Core Memory Operation	Read/restore or read/modify/write cycle
Core Memory Capacity	4096 words, expandable to 32,768 words
Core Memory Access	Single direct-address in any 4096-word memory bank; single indirect-addressing from one bank to another

Computation Rate	333,333 additions per second
ASR33 Teletype	10 char per second
Program-Controlled I/O Capacity	Up to 64 devices, 4 mode selections each device
Data Channel Capacity	Up to 8 devices

Operating Characteristics

Power Requirements	120V±15%, 60 cps±2%, single-phase, 17-30A or 230V±15%, 50 cps±2%, single-phase, 17-30A
Power Consumption	2KW
Power Supply Outputs	+10, -15, -30, ±20 Vdc
Logic Levels	0V=logic 0, -3V=logic 1
Test Temperature Range	55-122°F
Relative Humidity Range	10-95%
Heat Dissipation	6830 Btu/hr

Dimensions

Cabinet Height	71-3/16 in.
Cabinet Width	32-11/16 in.
Cabinet Depth	30 in.
Shelf Width	32-11/16 in.
Shelf Depth	22 in.
Door Clearance (Rear)	31 in.
Cabinet Weight	750 lb.
Teletype Height	8-3/8 in.
Teletype Width	22 in.
Teletype Depth	18-1/2 in.
Teletype Weight	44 lb.

CHAPTER 2
CONTROLS AND INDICATORS

2.1 OPERATOR CONSOLE

The PDP-9/L operator console (Figure 2-1), an integral part of the main computer frame, includes a work shelf and a control console equipped with rocker switches, rotary switches, and indicators for operator control and monitoring of system operation. Typical console uses are:

a. Manual entry of instruction and/or data; start/stop/continue control of program execution.

b. Stepping through a program sequence by instruction or by machine cycle for debugging or maintenance purposes.

c. Visual examination of register contents and/or of system status.

Table 2-1 details the functional use of items on the control console. Indicators on the panel show the existing binary states of specific register bits and control flip-flops by being lighted for binary 1s and being extinguished for binary 0s. The operator console can be electrically locked by a control on the marginal check panel to prevent undesired alteration of the program in progress. With exception of data switches, switch settings will not affect the system when the console is locked.

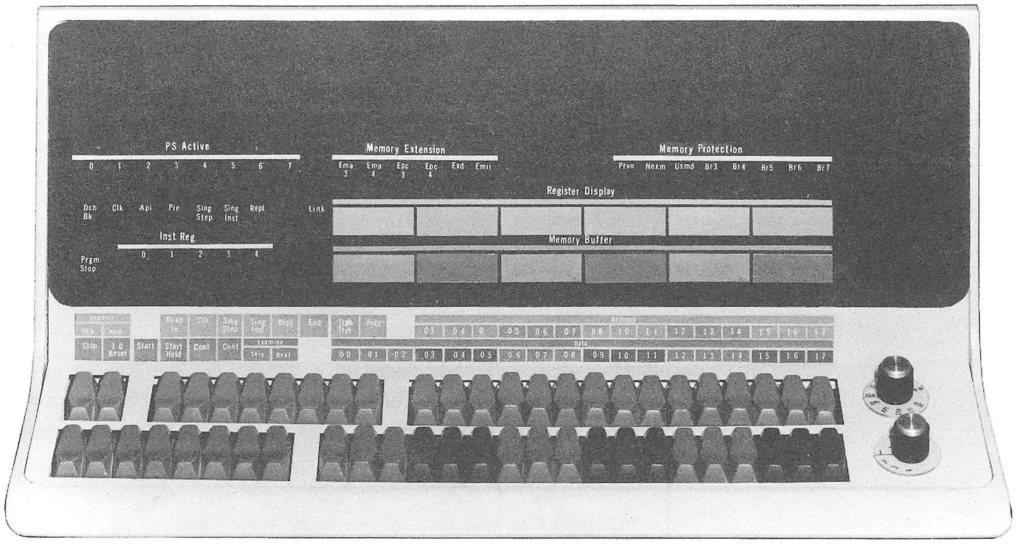


Figure 2-1 PDP-9/L Operator Console

Table 2-1
Operator Console Controls and Indicators

Controls and Indicators	Function
START and START HOLD switches	Depressing START starts program execution at the location specified by the ADDRESS switches. The START HOLD switch is used for maintenance purposes.

Table 2-1 (Cont)
Operator Console Controls and Indicators

Controls and Indicators	Function
IO RESET switch	Two positions: off (center) and operate (down, spring-loaded return). Depressing switch clears all I/O device flags, clears the AR, MA, AC, MQ, and the Link, turns off the real-time clock, program interrupt facility, and API system and disables the memory protection and extended memory modes.
STOP switch	Two positions: off (center) and operate (down, spring-loaded return). Operate halts program execution at completion of the current instruction.
CONT and CONT HOLD switches	Depressing CONT resumes program execution from the point at which it stopped. The CONT HOLD switch facilitates use of the REPT (repeat) function for the single instruction and single step provisions.
EXAMINE THIS and EXAMINE NEXT switches	Depressing the EXAMINE THIS switch transfers the contents of the memory location specified by the ADDRESS switches from memory to the MB. After the transfer, the contents of the ADDRESS switches appear in the AR as the address of the memory location examined. Depressing the EXAMINE NEXT switch increments the contents of the AR by one and transfers the contents of the newly addressed memory location from memory to the MB. EXAMINE NEXT facilitates monitoring of sequential memory locations as the ADDRESS switches need only be set to the lowest memory location. The use of EXAMINE THIS transfers the contents of this location to the MB and enters the lowest order address in the AR. Thereafter, use of EXAMINE NEXT step-advances the addresses through the sequential memory locations.
DEPOSIT THIS and DEPOSIT NEXT switches	Depressing DEPOSIT THIS switch deposits the contents of the DATA switches in the memory location specified by the ADDRESS switches. After the transfer, the contents of the ADDRESS switches appear in the AR as the address of the memory location in which the data was entered. Depressing the DEPOSIT NEXT switch increments by one the AR contents, and deposits the contents of the DATA switches in the memory location specified by the new address. DEPOSIT NEXT facilitates the entering of data and/or instruction words in sequential memory locations as the ADDRESS switches need only be set to the lowest order address.

Table 2-1 (Cont)
Operator Console Controls and Indicators

Controls and Indicators	Function
	The DEPOSIT THIS function deposits the DATA switch word in this location and transfers the address to the AR. Thereafter, the DEPOSIT NEXT function step-advances the addresses through the sequential memory locations.
READ IN switch	Two positions: off (center) and operate (down, spring-loaded return). Depress switch to initiate read-in of paper tape punched in hardware read-in format. The selected repeat speed switch should be set to ON position on the REPT switch (each set of three 6-bit lines read from tape forms one 18-bit computer word). Storage of words read in begins at the memory location specified by the ADDRESS switches. At the completion of tape read-in, the computer reads the last word from core memory and executes it. Read-in occurs at the selected repeat speed.
REPT (repeat) control and system ON-OFF switch	With REPT switch and CONT HOLD up, the control establishes one of five speeds at which single-step or single-instruction operations repeat without operator intervention. The repeating speeds range from approximately 2 µs (ON position) to 1s (position 1).
REGISTER DISPLAY control and display control and REGISTER DISPLAY indicators	Eleven-position switch: Each position interrogates a specific register and displays its contents in the REGISTER DISPLAY indicators. REGISTER DISPLAY indicators display the contents of selected register only when machine is stopped. Moving the selection switch while the program is running has no effect. The functions of the positions are as follows: RDR Display contents of the paper-tape reader information buffer. TTI Display contents of the teleprinter-keyboard information buffer. STA Display status of flags for I/O devices connected to status reading facility of I/O system. API Display activity of automatic priority interrupt system's four device-oriented priority levels.

Table 2-1 (Cont)
Operator Console Controls and Indicators

Controls and Indicators	Function
	DPY Display optional 34HL x-, y-buffers. The x-buffer is displayed in the nine most-significant REGISTER indicators; the y-buffer is displayed in the nine least-significant indicators. The least significant bit of each buffer is not displayed.
	IOA Display 15-bit address word present on address lines of I/O bus for data channel and API operation.
	IOB Display 18-bit data word present on data lines of I/O bus for program controlled and data channel data transfers.
	AC Display contents of the AC.
	AR Display contents of the AR.
	PC Display contents of the PC and status bits as stored during this instruction.
	MQ Display contents of the MQ.
PRTC switch and indicator	The up position causes the memory protection mode to be entered by operation of the START switch. In either position, the mode may be enabled or disabled by program control. While the console is locked, the switch is electrically in the down position, regardless of its actual position. The indicator is lit while the mode is in effect. (Memory protection is a system option.)
EXD switch and indicator	The up position causes the extend mode of addressing to be entered by operation of the START switch. In either position, the mode may be enabled or disabled by program control. While the console is locked, the switch is electrically in the down position, regardless of its actual position. The indicator remains lit while the mode is in effect. (Extend mode is a system option.)
CLK switch and indicator	The up position disables the optional real-time clock facility. The down position allows program control to enable or disable the clock. The indicator remains lit while the clock is enabled. While the console is locked, the switch is electrically in the down position, regardless of its actual position.

Table 2-1 (Cont)
Operator Console Controls and Indicators

Controls and Indicators	Function
SING STEP indicator and switch	The indicator lights when the associated switch is up. This enables the single-step mode which halts program execution at each machine cycle. Repetitive depressing of the CONT HOLD switch, while the mode is enabled, steps the program through the sequence one cycle at a time. When the console is locked, this switch is disabled.
SING INST indicator and switch	The indicator lights when the associated switch is up. This enables the single instruction mode which halts program execution at completion of each instruction. Repetitive depressing of the CONT HOLD switch, while the mode is enabled, steps the program through its sequence one instruction at a time. When the console is locked, this switch is disabled.
TTYH/TTYF switch	Determines whether Teletype operation is half or full duplex.
REPT indicator and switch	The indicator lights when the associated switch is up. This enables the repeat function. This function causes operations initiated by actuation of CONT HOLD, EXAMINE NEXT, or DEPOSIT NEXT switches to repeat while the key remains in an operator position. The repeat speed control establishes the rate of repetition.
ADDRESS switches (3-17)	Establish a 15-bit core memory address to be entered in the PC by operation of the START switch, or in the AR by operation of the EXAMINE THIS or DEPOSIT THIS switch. Switch is placed up for a 1 bit and down for a 0 bit. The 15 switches to the right (3-17) set up the address of a location within an 8192-word memory block. The two switches to the left (0 and 0) are for extended memory addressing of locations, in up to three other 8192-word memory blocks of the system.
DATA switches	Establish an 18-bit data or instruction word to be read into memory by DEPOSIT THIS or DEPOSIT NEXT operation, or to be entered in the AC by a programmed LAS (load DATA switches) instruction. Up position of the switch is a binary 1; down position is a binary 0.
PRGM STOP indicator	Lights when the RUN flip-flop has been cleared to stop program execution.
INST REG	The five indicators reveal the contents of the IR, being lit for 1 bits and extinguished for 0 bits, to

Table 2-1 (Cont)
Operator Console Controls and Indicators

Controls and Indicators	Function
	show the operation code of the instruction just executed or in progress, and indirect address occurrence.
DCH BK	Lights to indicate that data channel activity is in progress; i.e., data is being transferred between core memory and a data channel I/O device via the I/O bus.
PS ACTIVE indicators	Each indicator, relating to one of the API system's eight priority levels, individually lights to show the priority program interrupt request currently being serviced. Indicators 0, 1, 2, and 3 show activity resulting from device-initiated requests; indicators 4, 5, 6, and 7 show activity resulting from program-initiated requests. The priority levels for each set decrease in rank from left to right with any device request having higher priority than any program request.
PIE indicator	Lights when the PI system has been enabled by program control.
API indicator	Lights when the API system has been enabled by program control.
LINK indicator	Shows the content of the Link register.
MEMORY BUFFER indicators	Shows the contents of the MB register.

2.2 MARGINAL CHECK PANEL

The marginal check panel (Figure 2-2) is concealed behind the red hinged panel on the front of the central processor. Table 2-2 details the functions of the panel-mounted controls and indicators.

2.3 CORE MEMORY BANKS

Two selector switches, SW3 and SW4, are located in module slot B09 of each core memory bank. These must be preset to the particular memory bank assignment. The down positions designate 0, while up positions designate 1. Selections are as follows: 00, bank 0; 01, bank 1; 10, bank 2; 11, bank 3.

2.4 TELETYPE UNIT

The ASR33 Teletype Unit appears in Figure 2-3. Table 2-3 lists the teletype controls and their functions.

2.5 OPERATING PROCEDURES

Several methods are available for loading or unloading PDP-9/L information, as described in the PDP-9/L User Handbook. The method used depends upon the form of the information, time limitations, and the peripheral equipment connected to the computer. The following procedures are basic to any PDP-9/L configuration.

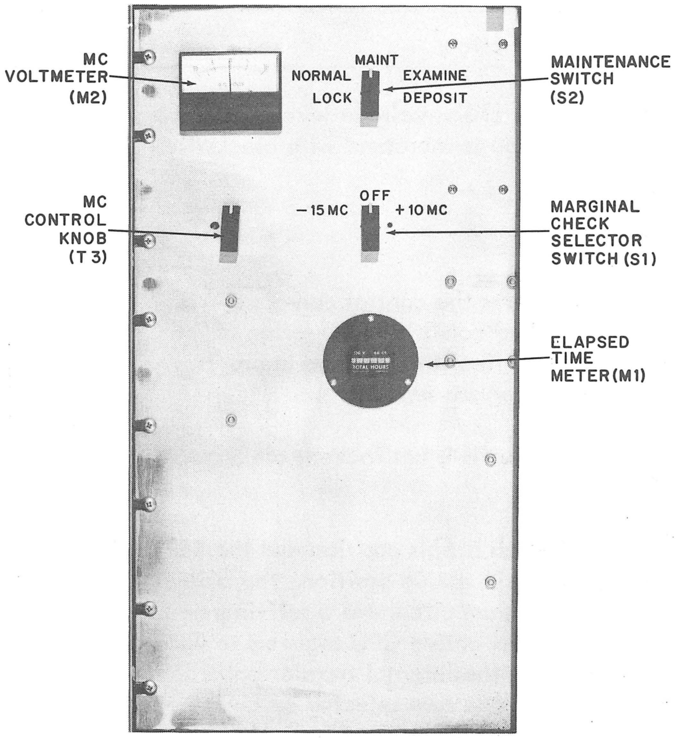


Figure 2-2 Marginal Check Panel

2.5.1 Manual Data Storage and Modification

The manual controls on the console permit storage or modification of programs and data. These facilities are used primarily in the manual storage of the Read-In Mode Loader (RIM) program and other programs in the read-in mode format.

The stored RIM Loader program automatically reads data into PDP-9/L memory from perforated paper tape in RIM format when the ADDRESS switches are set to the RIM Loader starting address and the START key is pressed. The RIM tape format is further described in the PDP-9/L User Handbook and in Digital Program Library descriptions. To store the RIM Loader manually in PDP-9/L core memory:

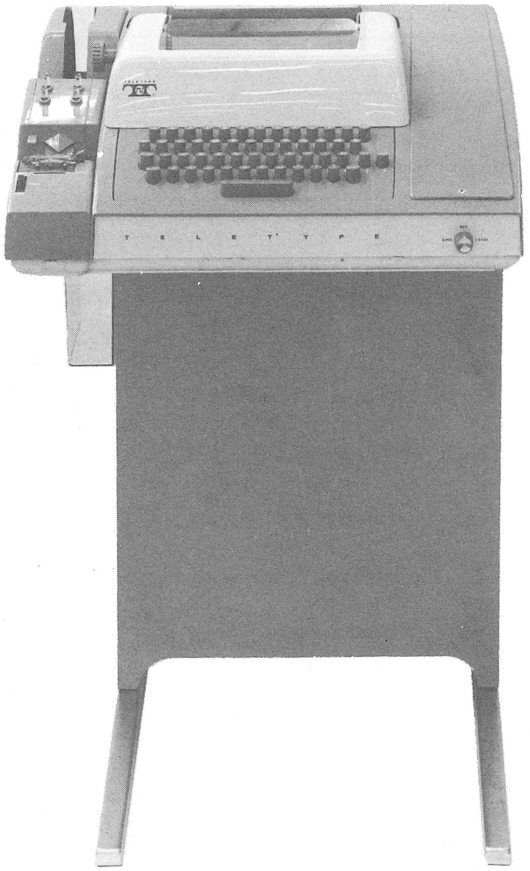


Figure 2-3 Teletype Model ASR33 Console

Table 2-2
Marginal Check Panel Controls and Indicators

Controls and Indicators	Function
Marginal check voltmeter (1, Figure 2-2)	Indicates the selected voltage output of the marginal check power supply. The center of the scale relates to the reference voltage selected, either +10 or -15V dc. Movement of the pointer to the right indicates an increase in magnitude for the marginal check voltage.

Table 2-2 (Cont)
Marginal Check Panel Controls and Indicators

Controls and Indicators	Function
Marginal-check voltage control (2, Figure 2-2)	Establishes the marginal check voltage level of the selected output. Voltage is increased with clockwise rotation.
Maintenance switch (3, Figure 2-2)	Five positions: LOCK - electrically locks the control console. With the console in the locked condition, operation of any console control cannot affect the program in progress. Data switches can be program sensed. NORMAL - Control console is not locked; all controls may be used. MAINT - With the switch in this position and the REPT switch (control console) in the up position, the built-in maintenance test program circulates a self-incrementing count through all active CPU registers to verify both their operation and the internal transfer paths. The program proceeds at the rate selected by the repeat speed control (control console). The START HOLD switch should be selected during MAINT. EXAMINE - simulates the "examine" function. With the switch in this position, the CPU responds as if the EXAMINE THIS switch (control console) was being actuated at the rate selected by the repeat speed control (control console). With the REPT switch in the down (inoperative) position, each movement of the selector switch to position EXAMINE simulates an actuation of the EXAMINE THIS switch. DEPOSIT - Simulates the "deposit" function. With the switch in this position and the REPT switch (control console) in the up position (activated), the CPU responds as if the DEPOSIT key were being actuated at the rate selected by the repeat speed control (control console). With the REPT switch in the down (inoperative) position, each movement of the selector switch to position DEPOSIT THIS simulates an actuation of the DEPOSIT THIS switch.
Marginal-check selector switch (4, Figure 2-2)	Three positions: OFF - marginal check disabled. +10 MC - selects the +10V output of the marginal check power supply.

Table 2-2 (Cont)
Marginal Check Panel Controls and Indicators

Controls and Indicators	Function
Marginal-check selector switch (4, Figure 2-2) (Cont)	-15 MC - selects the -15V output of the marginal check power supply.
Elapsed time meter (5, Figure 2-2)	Indicates, to the nearest tenth of an hour, the cumulative number of hours in which the system has been in the "power on" state. Meter counts from 00000.0 to 99999.9.

Table 2-3
Teletype Controls And Indicators

Control or Indicator	Function
REL. pushbutton	Disengages the tape in the punch to allow tape removal or tape loading.
B. SP. pushbutton	Backspaces the tape in the punch by one space, allowing manual correction or rubout of the character just punched.
OFF and ON pushbuttons	Control use of the tape punch with operation of the teletype keyboard/printer.
START/STOP/FREE switch	Controls use of the tape reader with operation of the Teletype. In the lower FREE position, the reader is disengaged and can be loaded or unloaded. In the center STOP position, the reader mechanism is engaged but de-energized. In the upper START position, the reader is engaged and operated under program control.
Keyboard	Provides a means of printing on paper in use as a typewriter and punching tape when the operator presses the punch ON pushbutton. The keyboard also supplies input data to the computer when the LINE/OFF/LOCAL switch is in the LINE position.
LINE/OFF/LOCAL switch	Controls application of primary power in the Teletype and controls data connection to the processor. In the LINE position, the Teletype is energized and connected as an I/O device of the computer. In the OFF position, the Teletype is de-energized. In the LOCAL position, the Teletype is energized for off-line operation, and signal connections to the processor are broken. Only line use of the Teletype requires that the computer be energized through the POWER switch if primary power for the Teletype is supplied from a source other than the outlet at the back of the computer.

a. Turn the maintenance panel switch to NORMAL and the console POWER switch ON.

b. Set the console ADDRESS switches to correspond with the address of the first word to be stored. (In the case of the RIM Loader program, this is 07762 for 4K, 17762 for 8K.)

c. Set the DATA switches to correspond with the binary equivalent of the first word. (In the case of the RIM Loader program, this is 0.)

d. Depress the DEPOSIT THIS switch to deposit the word in memory.

e. Display the contents of the AR in the REGISTER indicator by turning the REGISTER DISPLAY selector to the AR position. The contents of the MB are automatically displayed in the MEMORY BUFFER indicator. Observe that the MB contains the data word just deposited, and that the AR contains the address of the core memory cell in which the word was deposited.

f. Store all additional data words by pressing the DEPOSIT NEXT switch after each successive data word has been set into the DATA switches. The contents of the AR will increment by 1 during each DEPOSIT NEXT operation, thus setting up the address of the core memory cell to be used for the next operation.

g. To recheck the loaded program, set the ADDRESS switches to the starting address and press the EXAMINE switch. After the first core memory cell has been checked at the MEMORY BUFFER indicator, the remaining cells may be examined in sequence by repeatedly pressing the EXAMINE NEXT switch without regard to the ADDRESS switch settings. By repeating steps b through d using the address of the cell in question, it is possible to alter the contents of any cell.

2.5.2 Storing Binary Data Using READ IN Key

Hardware Read-in (HRI) tapes (including the RIM Loader tape) can be loaded directly into core memory without the need of a prestored program, by using the following procedure.

a. Turn the maintenance panel switch to NORMAL and the console POWER switch ON.

b. Load the tape in the Teletype reader. Proper positioning of the tape enables three channels to the left of the sprocket wheel to be sensed. Set Teletype reader to start position.

c. Set the ADDRESS switches to the starting address to be used for the program as it is stored in core memory.

d. Press the READ IN key. The tape will be read automatically.

e. Channel 8 on binary format tapes is always punched; channel 7 is punched only in the last line of the last word to stop tape motion and to conclude the read-in operation. The program just read can be made self-starting by making this last word a JMP instruction to the starting address of the program. When the JMP instruction is read, it is interpreted as the current instruction to be executed and thus starts the program. If the tape is not self-starting, the last instruction is a HLT. To initiate the program, set the starting address into the ADDRESS switches and press the START key.

2.5.3 Storing Data Under Program Control

Information can be automatically stored or modified in the computer by executing programs previously stored in memory. For example, having the RIM Loader stored in core memory allows the loading of RIM format tapes, as follows.

a. Turn the maintenance panel switch to NORMAL and the console POWER switch ON.

b. Insert the tape in the tape reader.

c. Using the ADDRESS switches, set the starting address of the RIM Loader program.

d. Press and release the console START key. The tape is read and stored automatically.

2.5.4 Assembling Programs

Programs prepared in binary format and written in symbolic language can be assembled into binary, machine-language program tapes as described in appropriate Digital Program Library documents, as follows:

a. Turn the maintenance panel switch to NORMAL and the console POWER switch ON.

b. Store the RIM Loader program, either manually or by use of the READ IN key, as previously described.

c. Load the assembler program into core memory by means of the assembler tape. (The assembler tape is in RIM format.) When the tape has been read, the AC should contain all 0s. If it does not, a checksum error has been detected, showing improper storage of the program. When this occurs, the tape must be rerun until the AC finally contains all 0s at the conclusion of the loading process. Repeated errors indicate defects in either the assembler tape or the PDP-9/L system.

d. Insert the symbolic language tape to be converted into machine-language, binary-format, into the tape reader.

e. Put the starting address of the assembly program into the ADDRESS switches on the console. (Set DATA switch 10 up to indicate ASCII, or down to indicate FIODEC code.)

f. Press and release the CONTINUE key.

g. When assembly is complete, the assembler will stop with all 1s in the AC.

2.5.5 Teletype Code

The 8-bit code used by the ASR33 Teletype is the American Standard Code for Information Interchange (ASCII), modified.

2.5.6 Maintenance Programs

Diagnostic programs are designed to test specific functions within the computer system. These routines are available as perforated paper tapes in hardware read-in mode (HRI) format. Each diagnostic routine is accompanied by a description of the program, procedures for using the program, and information on analyzing the results to locate specific failures. Applications of these routines are indicated in Chapter 10, as they apply to preventive or corrective maintenance of the PDP-9/L system. To exercise these routines the user should be familiar with the machine programming described in the PDP-9/L User Handbook.

CHAPTER 3
SYSTEM DESCRIPTION

3.1 COMPUTER ORGANIZATION

The PDP-9/L Programmed Data Processor System is a general purpose computer, incorporating FLIP CHIP hybrid circuits. The computer is a single address, fixed word length (18 bits), parallel binary computer. Minimum system configuration is 4096 words of memory, paper-tape input and output, and keyboard input and printer output.

Major functions of the PDP-9/L System are the central processor, core memory, main memory, and input/output facilities. The system has a bus transfer network and jam-transfers data between registers at dc levels to minimize timing problems.

Figure 3-1 illustrates the organization of the PDP-9/L General Purpose Computer.

3.2 CENTRAL PROCESSOR

3.2.1 Control Memory (CM)

The CM issues all sequences of internal processes required to fetch and execute a program's instructions, to effect operation of I/O channels, and to respond to operator commands from the console. It is a read-only, prewired magnetic core storage unit. The CM supplies new address information to the CP (Central Processor) based on the instruction to be executed and on the conditions sensed by the previous process levels.

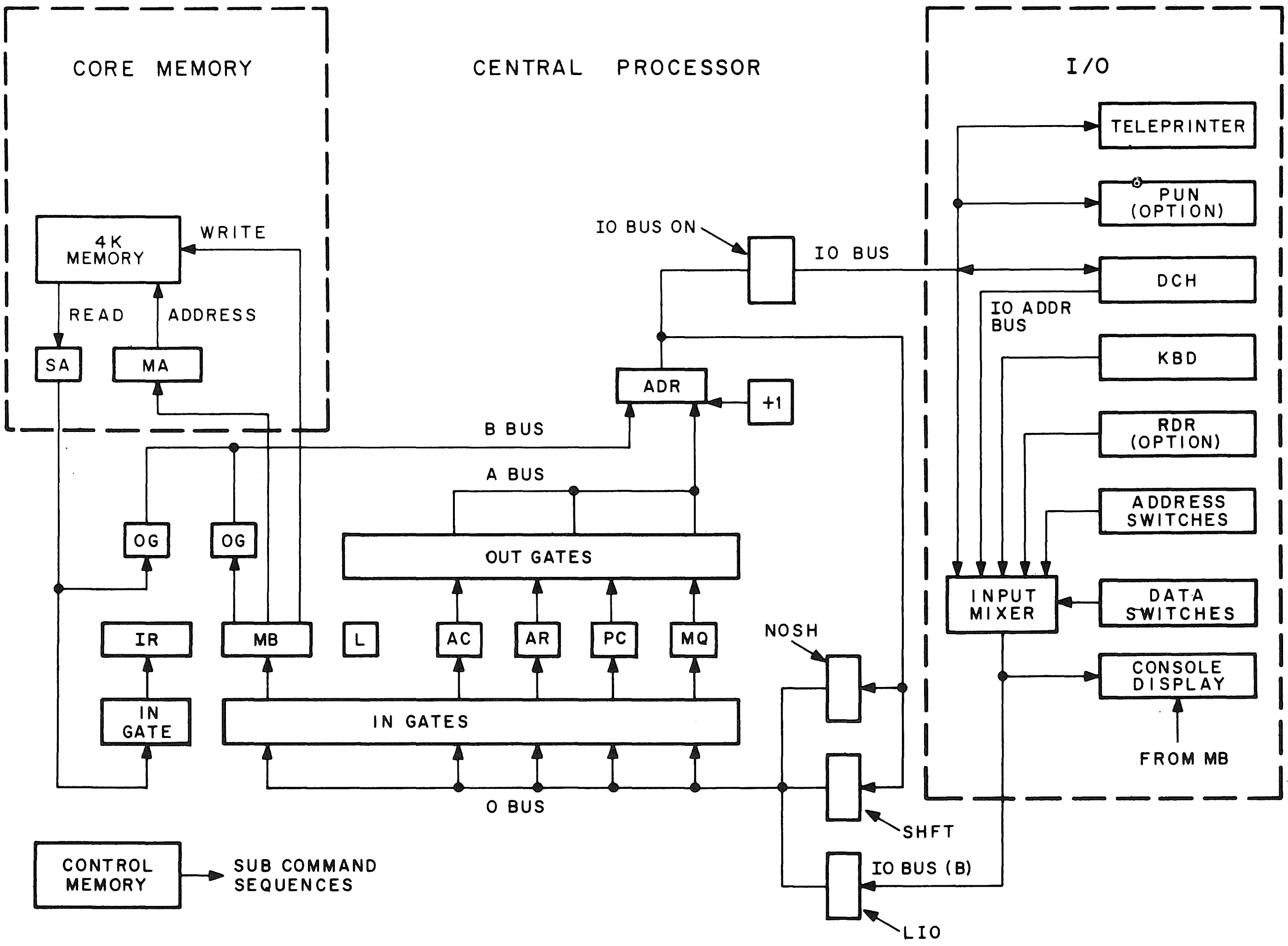


Figure 3-1 PDP-9/L Functional Diagram

9L-0048

3.2.2 Adder (ADR)

The 18-bit ADR functions as a nonstoring adder for arithmetic operations and as a common bus transfer for all inter-register transfers and shift operations. A 19th-bit adder link (ADRL) transfers the content of the arithmetic register link (LAR) to the accumulator register link (Link) and vice-versa when those registers are used for arithmetic operations. The ADR operates at a 5 Mc rate for a transfer time for 200 ns and a carry time of less than 5 ns per stage.

3.2.3 Accumulator (AC)

The 18-bit AC retains the result of arithmetic/logical operations. The AC can be cleared, complemented, rotated right, or rotated left. The contents of the memory buffer (MB) register can be added to the contents of the AC. The contents can also be combined, in the ADR, by logical AND or Exclusive-OR instructions and the result left in the AC. An Inclusive-OR can be formed in the ADR between the AC and the DATA switches on the console, and the result left in the AC. For program-controlled input data transfers, information is transferred from an external device to the AC via the I/O bus.

3.2.4 AC Link (Link)

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

3.2.5 Arithmetic Register (AR)

The AR functions with the AC to perform arithmetic and logical operations. It accepts and stores the contents of the AC for manipulation through the ADR; the results are then deposited in the AC. The AR is not accessible to the programmer. For program-controlled output data transfers, information is transferred from the AR to an external device via the I/O bus.

3.2.6 Optional Multiplier-Quotient Register (MQ)

The 18-bit MQ register is part of the optional extended arithmetic element (EAE). The MQ holds the multiplier during multiply operations and receives the low-order 18-bits of the resulting product. During divide operations, it holds the low-order 18-bits of the dividend, and at the completion of the divide operation, it contains the quotient. It can also be used as an extension of the AC for 36-bit shift operations.

3.2.7 Program Counter (PC)

The PC determines the program sequence in which instructions are executed. This 13-bit register contains the address of the core memory cell from which the next instruction is to be taken. Addition of the extended memory options adds two extended program counter (EPC) bits to the system addressing scheme.

3.2.8 Instruction Register (IR)

The IR accepts the five most-significant bits of each instruction fetched from core memory. Of these, the four most-significant bits constitute the instruction operation code (op code). For a memory reference instruction, the fifth bit indicates whether the instruction contains a direct (effective) address of an instruction or an indirect address of a location in core memory which contains the effective address. These bits are decoded during the fetch cycle to determine the CM sequence necessary to execute the instruction.

3.2.9 Memory Buffer Register (MB)

All information transferred into or out of core memory passes through the MB. During a fetch cycle, instructions are read from a memory location into the MB and are rewritten into the location as the instruction is retained in the MB for decoding and later execution. During an execute cycle, the operand addressed by the instruction is fetched from a memory location and is placed in the MB. The operand is then rewritten into the memory location, in accordance with the operation called for by the instruction. The MB also serves as a buffer for both information transfers between core memory and an external device, and address transfers between the PC and the MA.

3.3 CORE MEMORY SYSTEM

The PDP-9/L core memory used a 3D design concept for speed, compactness, and reliability; it operates with a complete cycle time of 1.5 μs. Each 4096-word core memory package contains a core stack, sense amplifiers, drivers, and a 13-bit memory address (MA) register. The MA addresses the memory location to be used for data retrieval or storage. System core memory can be expanded from the basic 4096 words up to 32,768 words in 4,096-word increments. Such expansion requires the implementation of the memory extension option to extend the PDP-9/L addressing capability.

Table 3-1
Reserved Core Memory Locations

Address	Purpose
00000	Stores the contents of the PC, Link, optional EPC, extend mode option status, and memory protection option status during a program interrupt.
00001	Stores the first instruction to be executed following a program interrupt, normally a JMP.
00002-00006	Currently unused
00007	Stores optional real-time clock-time
00010-00017	Autoindex registers
00020	Stores the contents of the PC, Link, optional EPC, extend mode option status, and memory protection option status upon execution of a CAL instruction.
00021	First instruction to be executed following a CAL instruction.
00022-00027	Currently unused
00030-00037	Four pairs of word counter/current address registers for use with data channels 0, 1, 2, and 3.
00040-00077	Store entry instructions for each of 32 optional automatic priority interrupt channels.

3.3.1 Direct Addressing

Directly addressed memory reference instructions (bit 04 = 0, Figure 3-2) take the 12-bit address (06 to 17) specified in their instruction words as the effective address of the memory register which contains the required operand. The 12-bit address allows direct addressing of up to 4096 locations in a currently addressed memory bank. Locations in memory banks other than that currently addressed must be accessed by indirect addressing.

3.3.2 Indirect Addressing

Indirectly addressed memory reference instructions (bit 04 = 1, Figure 3-2) take the 12-bit address (06 to 17) specified in their instruction words, not as the effective address of the memory register containing the operand, but as the address of the memory register which contains the effective address. For example, the instruction LAC 100 directs the computer to load the contents of memory register 100 into the AC. But the instruction LAC * 100 (see footnote 1) directs the computer to load the contents of the memory register addressed by the contents of memory register 100 into the AC. Indirect addressing adds one computer cycle (defer) to the instruction execution time, during which the effective address of the operand is fetched.

3.3.3 Autoindexing

When any one of core memory locations 00010 through 00017 is indirectly addressed, the contents of that location are automatically incremented by 1, and the result is taken as the effective address of the instruction. Incrementing is accomplished with no additional instruction execution time. Such autoindexing operations are effective only when the locations are indirectly addressed. When directly addressed, the locations contain operands just as any other memory locations.

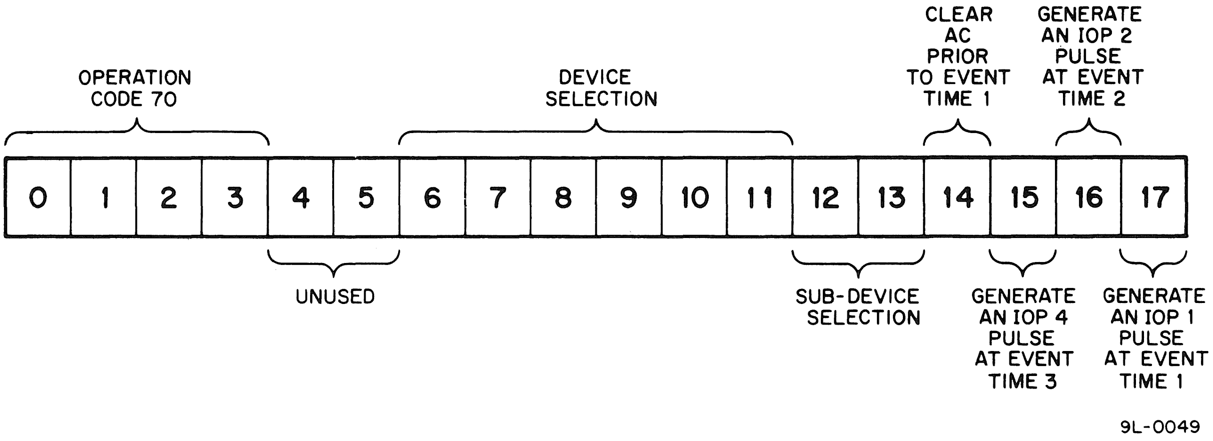
3.3.4 Extend Mode Addressing

Installation of additional memory banks requires the memory extension control option for addressing a memory location outside the currently addressed memory bank.

3.4 I/O CONTROL

The I/O control section includes logic for program-controlled transfers between the central processor and

[1] Indirect addressing in earlier software programs was represented with an I symbol. The symbol * represents indirect addressing in current programs.


OPERATION CODE 70
DEVICE SELECTION
CLEAR AC PRIOR TO EVENT TIME 1
GENERATE AN IOP 2 PULSE AT EVENT TIME 2
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
UNUSED
SUB-DEVICE SELECTION
GENERATE AN IOP 4 PULSE AT EVENT TIME 3
GENERATE AN IOP 1 PULSE AT EVENT TIME 1
9L-0049

a. Word Format

Operation	Mnemonic	Code	Cycles
Call subroutine	CAL	00	2
Deposit AC	DAC	04	2
Jump to subroutine	JMS	10	2
Deposit zero in memory	DZM	14	2
Load AC	LAC	20	2
Exclusive OR	XOR	24	2
Add, 1's complement	ADD	30	2
Add, 2's complement	TAD	34	2
Execute	XCT	40	1+
Increment and skip if zero	ISZ	44	2
AND	AND	50	2
Skip if AC different from memory	SAD	54	2
Unconditional jump	JMP	60	1

Add 1 cycle time for indirect addressing or auto indexing

b. Instruction Description

Figure 3-2 Memory Reference Instructions

as many as 64 devices. Program-controlled transfers make use of program interrupt, I/O skip, and I/O status checking facilities. The I/O control section also allows operation of up to 8 devices connected to 8 data channels (DCH), and up to 28 devices in 32 optional API channels multiplexed at 8 priority levels.

All of the above operate off the bidirectional I/O bus, which serially links the central processor to the peripheral devices. Devices with high transfer rates, such as DECtape and magnetic tape normally use DCH access to core memory to allow operation at maximum transfer rates. Slower asynchronous devices such as line printers, teletype keyboards and punched card equipment may operate at maximum speeds through the use of the API option under program control.

Some timed-transfer devices can operate independently of the central processor after they have been set in operation. These devices are normally connected to the DCH to transfer a block of data words at a time. Once the program has supplied information about the location and size of the data block, the DCH takes over the responsibility of effecting the actual transfer. Separate parallel buffers are provided in the device controls interfaced to the I/O bus.

3.5 INSTRUCTION WORD FORMATS

The PDP-9/L has two general instruction groups: memory reference (single-address) instructions, and augmented (no-address) instructions. The latter group has three subclasses: operate (OPR) instructions, input/output transfer (IOT) instructions, and optional extended arithmetic element (EAE) instructions.

3.5.1 Memory Reference Instructions

Memory reference instructions (Figure 3-2) consist of and op code, an indirect address bit, and an address. The op code, bits 00-03, specifies one of 13 memory reference instructions. The indirect address bit, 04, indicates whether the 12-bit address (06 to 17) is a direct address (bit 04 = 0), or an indirect address (bit 04 = 1). If direct addressing is indicated, the addressed memory location contains the required operand. If indirect addressing is indicated, the addressed memory location contains the address of the required operand. In either case, the address of the memory location containing the operand is the "effective address" for the instruction. Descriptions of the memory reference instructions are given in Appendix A.

3.5.2 Augmented Instructions

3.5.2.1 Operate - OPR instructions (op code 74) are used to sense and/or alter the contents of the AC and Link. Typical functions (Figure 3-3) are: conditional or unconditional skips; complementing, setting, clearing, or rotating the contents of the AC and Link.. A HLT instruction is included. OPR instructions are fetched and executed in one computer cycle, the actions being specified by the microprogramming of bits 04 to 17 in the instruction word. Each of the 14 bits can effect a unique response; hence, they are "microinstructions" to the computer. The important feature of the OPR class is its microprogramming capability where two or three microinstructions can be combined in one instruction word, and therefore be executed sequentially during one computer cycle.

3.5.2.2 Input/Output Transfer - IOT instructions (op code 70) initiate transmission of signals via the I/O bus to control peripheral equipment, sense their status, and effect information transfers between them and the central processor. Each instruction contains an 8-bit device selection code, bits 06 through 13, and a command code, bits 14 through 17 (Figure 3-4). Bits 06 through 11 of the device selection code perform the primary device selection among up to 64 devices while bits 12 and 13 select an operational mode or subdevice. Selection logic in a peripheral's interface responds only to its preassigned code. The command code, bits 14 through 17, is capable of being microprogrammed to clear the AC and to issue up to three sequential command pulses to the peripheral equipment via the I/O bus.

Execution of an IOT instruction requires an instruction fetch cycle and three execute cycles of 1.5 μs duration each, designated event times 1, 2, and 3. Only the fetch cycle contains a core memory read/write cycle. Thereafter, core memory is idle until completion of the IOT execute cycles. Bit 17 generates an IOP1 pulse during event time 1 while bits 16 and 15 generate IOP2 and IOP4 pulses during event times 2 and 3, respectively. IOT skip instructions are microprogrammed to produce an IOP1 pulse for testing a

								Bit 7 = 0				
CLA	CLL	Additional Rotate	0 = OR of	SNL	SZA	SMA	HLT	RAR	RAL	OAS	CML	CMA
			1 = AND of	SZL	SNA	SPZ		RTR	RTL			
								Bit 7 = 1				
5	6	7	8	9	10	11	12	13	14	15	16	17

								Event Time
	OPR NOP	7	4	0	0	0	0	-
	CMA	7	4	0	0	0	1	3
	CML	7	4	0	0	0	2	3
	OAS	7	4	0	0	0	4	3
	RAL	7	4	0	0	1	0	3
	RAR	7	4	0	0	2	0	3
	HLT XX	7	4	0	0	4	0	-
	SMA	7	4	0	1	0	0	1
	SZA	7	4	0	2	0	0	1
	SNL SML	7	4	0	4	0	0	1
	SKP	7	4	1	0	0	0	1
	SPA	7	4	1	1	0	0	1
	SNA	7	4	1	2	0	0	1
	SZL SPL	7	4	1	4	0	0	1
	RTL	7	4	2	0	1	0	2,3
	RTR	7	4	2	0	2	0	2,3
	CLL	7	4	4	0	0	0	2
(CLL-CML)	STL CCL	7	4	4	0	0	2	2,3
(CLL-RAL)	RCL	7	4	4	0	1	0	2,3
(CLL-RAR)	RCR	7	4	4	0	2	0	2,3
	CLA	7	5	0	0	0	0	2
(CLA-CMA)	CLC	7	5	0	0	0	1	2,3
(CLA-OAS)	LAS LAT	7	5	0	0	0	4	2,3
(CLA-RAL)	GLK	7	5	0	0	1	0	2,3

Figure 3-3 Operation Instruction Word Format

device status flag. IOP2 pulses are normally used to effect programmed transfers of information from a device to the central processor. Because the AC serves as the data register for input transfers, the "clear AC" microinstruction (bit 14) is usually microprogrammed with the IOP2 microinstruction; this combination clears the AC prior to the start of event time 1, then strobes in the new information with IOP2 during event time 2. The IOP pulses trigger IOP flip-flops which remain set for the event time duration. The IOP4 is usually used to transfer data from the computer to the device.

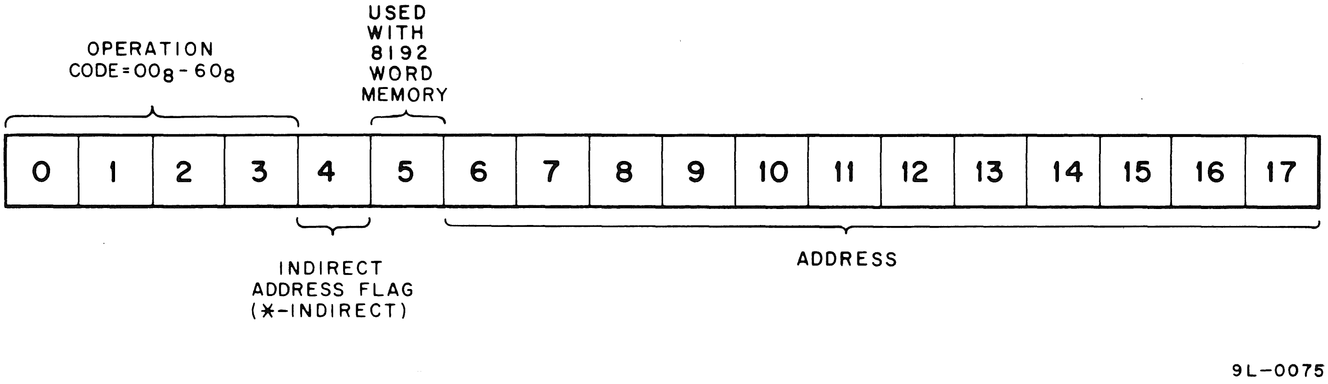


Figure 3-4 IOT Instruction Word Format

CHAPTER 4
CONTROL MEMORY SYSTEM

4.1 ORGANIZATION

The control memory (CM) system in the central processor is a read-only, linear-select magnetic core system which issues 36-bit "process words" to a control register (CR) composed of core-sensing flip-flops. A 6-bit control memory address (CMA) included in each process word addresses the next process word location in control memory. The remaining 30 bits comprise the data-path gating levels which implement the fetching and execution of instructions and also specify the timing of control memory readout. The CMA may be modified by conditions sensed during the processes, or by gating levels issued by the previous process word. The CM issues up to four such process words per computer cycle in a sequence which is largely determined by the previous processing results. A continue bit in the previous process word determines if another process word shall follow; an SM (start memory) bit determines if a main core memory cycle shall follow, concurrent with another process word.

Figure 4-1 is a functional block diagram of the control memory system. The system consists of CM timing logic (drawing KC16), two G210 Address Selectors (drawing KC17), Control Memory core array (drawing KC18) and 36 core-sensing control register flip-flops (drawing KC19).

The importance of maintaining synchronism of the control memory processes, and the main core memory cycles may readily be appreciated. Both the main memory cycle and the control memory timing are started by CLK POS pulses every 1.5 µs and a start memory (SM) level from the CM sense flip-flop that is always set during the last CM process word of a comp-

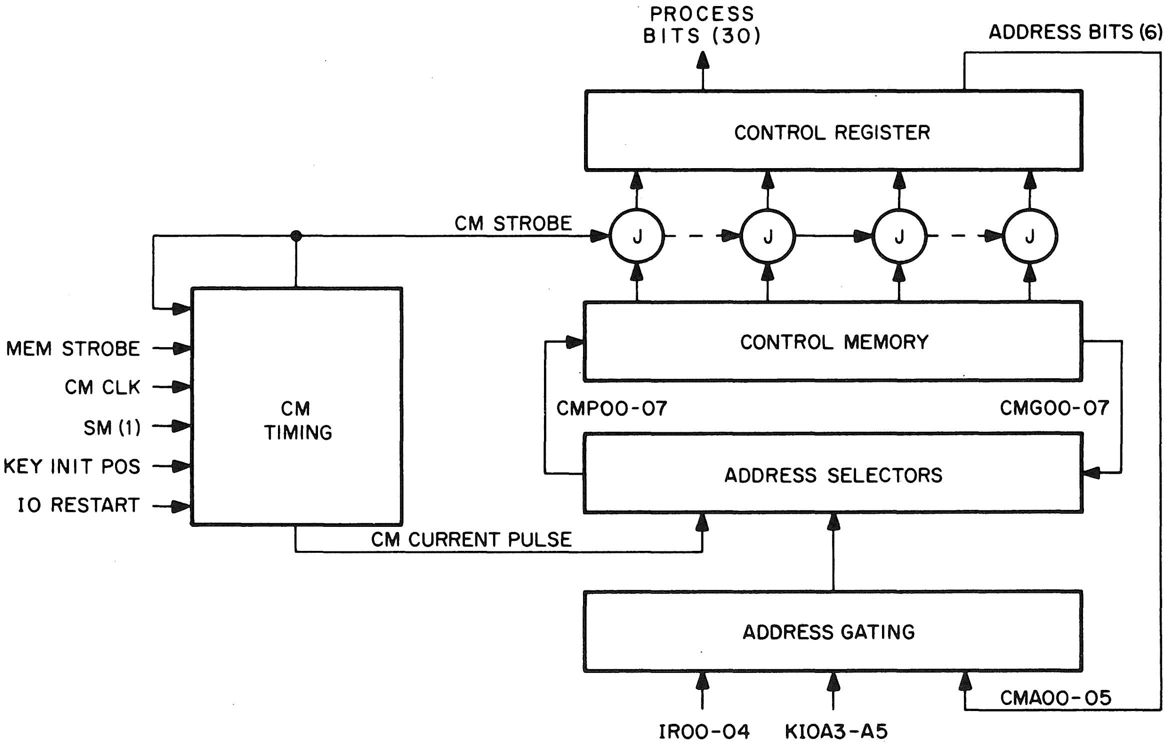


Figure 4-1 Control Memory System, Block Diagram

puter cycle (fetch, defer, execute, IA0). The last CM process word in an execute cycle, IA0 execute, and end of DCH break, always contains the SM level and address 21 in the CM sense flip-flops. The next computer cycle is a fetch cycle and the next CM process word will be extracted from address 21. The CM reads out the process word from address 21, which contains gating levels to load the PC with the address of the next sequential instruction, and also contains continue, a level which retriggers the CM timing chain. The address contained in process word 21 will have been sent to the G210 Address Selectors, and will always be address 12. The CM will now extract the process word at address 12.

The process word in CMA12 does not contain a CONT (1) bit, so another CM extraction does not follow immediately. In this instance, the CM timing must wait for the main core memory's MEM STROBE for triggering.

The address selectors contain positive and negative transistor switches operation in complementary pairs to send drive current through 1 of 64 lines in the control memory array. Each line is threaded through all 36 cores in series; the side of the core the wire passes through determines the state a core will assume when the line is driven. Core windings on one side induce positive voltages in their respective sense lines, setting the sense flip-flops in the control register. Core windings on the other side induce negative voltages, resetting the flip-flops. The sense flip-flops set and reset on the CM STROBE from the CM timing chain.

The processes of the fetch cycle determine whether the next cycle shall be a defer, IA0, or execute. Certain instructions also demand that the next cycle be another fetch. Whatever the case, the address in the third process word commands the extraction of the appropriate cycle entry word.

As shown in Figure 4-1, the address in the control register may be preempted by manual operations from the console, where the levels KIOA3-A5 perform the CM addressing function. Drawing KCI8(1) is the Control Memory Program Chart in which one process word (LOC column) address the next (JMP column) in the main flow of execution, with conditional branching from the main flow as a result of sampling certain events. The BITS column lists those bits that are 1s in each word and the JMP column lists the location in control memory of the next sequential word. The SYNC column contains the bits that control the initiation of the next process (CONT) or cycle (SM).

Drawings KC3 through KC6 are the flow diagrams for the processes. Numbers within the process word blocks indicate the CM addresses from which the words are taken. Note that some bits of the process word can be operated on by hardware external to the control memory system as a result of conditional events.

4.2 TIMING AND CONTROL

CM timing logic is shown on drawing KC16. Any of the following five conditions can start the timing chain to produce CM CURRENT and CM STROBE.

a. KEY INIT POS

b. IO RESTART

c. CM CLK ∧ SM (1) ∧ AM SYNC BUS(0)

d. CM STROBE D∧CONT (1)

e. MEM STROBE ∧ IA0 (0)

The following is a general description of the functions that these conditions control. Detailed descriptions and timing diagrams are found in the referenced sections.

KEY INIT POS occurs during manual entry of address and/or data words from the console switches or after depressing IO RESET on console. This pulse starts the chain to extract a series of manual entry process words from locations 00 through 07 using the KIOA3 through 5 levels for CM addressing.

CM CLK ∧ SM (1) starts the timing chain to extract the entry word of any computer cycle concurrently with a core memory cycle.

CM STROBE ∧ CONT (1) starts the chain to extract the second word, and MEM STROBE ∧ IA0 (0) extracts the third, in any computer cycle. For execute cycles, CM STROBE ∧ CONT (1) is allowed to extract a fourth. The determining factor for the number of words extracted is the status of the CONT flip-flop in the current process word.

For IOT instructions, IO RESTART extracts a process word after an extended 4.5 μs execute period. This word prepares the computer for the next fetch cycle. For the manual read-in operations, IO RESTART extracts a process word when the tape reader reads a hole in channel 7. This word also prepares the computer for the next fetch cycle.

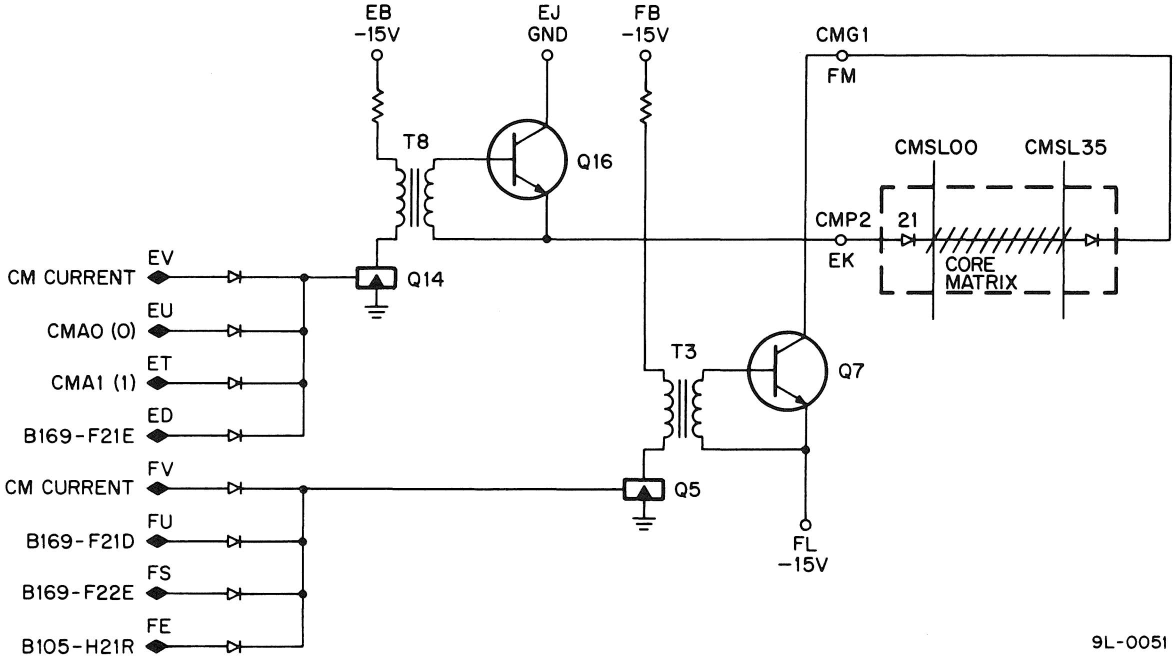


Figure 4-2 CM Line Selection, Line 21

4.3 CONTROL MEMORY G920

Control Memory G920 is a quadruple-height module containing the linear core array. The Control Memory Wiring Matrix, drawing KC18(2), is a practical representation of the 64 drive lines threading the cores that induce 1s into the sense lines. In reality, all drive lines thread all 36 cores serially in specific 1 and 0 winding patterns to produce 64 separate and distinct 36-bit words. At CM CURRENT time, the address selectors supply drive current through one selected line. At CM STROBE time, the core states are transferred to the sense flip-flops from the sense lines, CMSL00-35.

4.4 ADDRESS SELECTORS G210

Two double-height Address Selector Modules G210, drawing KC17, perform control memory line selection by decoding the CM address and turning on line drive-current in response to the CM CURRENT pulse from the CM timing logic, drawing KC16. Each module contains four positive-select and four negative-select switches, connected together to form an 8 by 8 coordinate matrix. Input address decoding gates turn on a pair of complementary switches to connect a ground to one end of the selected line and a negative source to the other. The module is similar to the G219 Address Selector Modules in core memory.

Figure 4-2 is a simplified schematic of the drive selection circuits for line 21, containing the fetch-entry process word. In this case, the active switch pair is located entirely within one module, EF20. Although selection of some lines makes use of a switch in each module, the logic is identical.

For the fetch entry word extraction, the address in control register bits CMA 0-5 is 21. In Figure 4-2 CMA 0 (0), CMA 1 (1), and the level from B169-F21E are all negative. On drawing KC17, the output at B169-F21E comes from the paralleled NAND gates controlling address bit 2. The output is negative because each of the four paralleled gates is disabled: the IR sampling is disabled by a grounded input from NOR gate R111-F24N, the DCH and API gate is disabled by the absence of EXT (1), etc. The negative levels are applied to the input decoding gates at transistor Q14. Likewise, the 03, 04, and 05 parallel gates are disabled and the consequent negative levels are applied to the input decoding gates at transistor Q5.

CM CURRENT enables the Q14 and Q5 gates to turn on these transistors. The resulting current-surges through transformers T8 and T3 turn on Q16 and Q7. The emitter of Q16 goes to ground and the collector of Q7 rises to the -15V supply voltage. The Q16 emitter output at EK is the CMP 2 connection to one end of core drive line 21; the Q7 collector output at

FM is the CMG 1 connection to the other end. Current flows through the line from CMG 1 to CMP 2.

4.5 CURRENT SOURCES

The control memory system is powered by the +10, -15V computer supply. The -15V output supplies the drive current via the address selector switches. Maximum current is limited by the inductance of the drive lines to 200 mA, inducing positive or negative signals of 2V (with 5V flyback) in the sense lines (reference, -2.2V bias).

CHAPTER 5
CENTRAL PROCESSOR

5.1 CENTRAL PROCESSOR LOGIC

The central processor performs the arithmetic logic, and system control operations of the PDP-9/L System. This chapter correlates the central processor logic with the execution of the program instructions. Functions common to most instructions, i.e., fetch cycle, defer cycle, autoindexing, are treated as an introduction to the individual execute descriptions.

The flow charts of drawing KC3 through KC6 and associated timing diagrams supplement the text. The system functional diagram in Chapter 3 will also be an aid to following the data transference described in this chapter.

5.1.1 Fetch Cycle

The fetch cycle description is based on the fetch flow drawing KC3 and the memory timing diagram MC71-13. The computer enters the fetch cycle from the BGN process word (10) in control memory. This is always the last word extracted from control memory during the current execute cycle of a running program. The BGN word contains PCO, SM, and the "next CM address," CMA21. The ground level PCO(1) is NORed at R111-E22HJ to produce a negative ΔMB level, drawing KC19(2). ΔMB is NANDed with SM(1) of the BGN word and CLR from drawing KC10. The NAND gate output at R111-E22N produces 1 → MBI at pulse amplifier B602-E23D when CLK at the end of the execute cycle occurs. The 1 → MBI pulse sets the MBI flip-flop. Under these conditions PCO(1) and MBI(1) transfer the address in the PC to the MB via the A bus, ADR, and O bus. At MA JAM time the MB is transferred to the MA in core memory. The BGN word remains in the CM sense flip-flops until SM(1) and the next CLK pulse (CM CLK) generate CM STROBE to extract the fetch entry process word at CM location 21.

At CLK time in the BGN process, CM CLK and SM(1) produce CM CURRENT (≈80 ns) and CM STROBE (≈40 ns) in the CM timing chain, drawing KC16, to extract the fetch entry word. This is the first of three process words to be extracted during the 1.5 μs fetch cycle period or interval between CLK pulses. CLK and SM(1) also start the core memory cycle.

The fetch entry word 21 contains MBO, +1, PCI, CONT, and CMA12. The address placed in the MB by the BGN process is still there; MBO(1) gates it onto the B bus, and the B bus contents go directly into the ADR on drawing KC21. Process +1(1) produces CI17, drawing KC14, which initiates a carry into ADR17 on KC21(3), in effect incrementing the address in the ADR by 1. NOSH (no shift) gates the incremented address onto the O bus, drawing KC20. NOSH from KC13 is present at all times except during rotating operations. PCI(1) places the incremented address in the PC and resets the SKIP and AUT INX flip-flops, drawing KC14. Unless otherwise modified by the execution of certain instructions, this is the address (PC+1) to be entered into the MB by the next BGN word of the impending execute cycle.

The CM STROBE that extracted the fetch entry word also restarts the CM timing chain with CM STROBE D on drawing KC16. This pulse triggers a 65-ns delay in B310-EF33. When the delay recovers at B310-EF33EU, its trailing edge grounds the emitter of inverter B104-F31R. The inverter turns on because CONT(1) of the fetch entry word is applied to the base. The collector goes to ground, thus triggering pulse amplifier B602-E32D. The PA output triggers delay EF33EL. When this delay recovers, it grounds the emitter of B104-F31F. $\overline{\text{TESTER}}$ is always negative, turning this inverter on. The collector passes NOR gate R111-F26U for CM CURRENT. The output of B104-F31 also pulses B602-F30D for CM STROBE after a 65-ns delay. The delay in EF29FL extends the CM CURRENT duration. CM CURRENT is turned off 80 ns later by CM STROBE D, via R111-E24U.

The second word extracted (from location 12) contains ACO, ARI, IRI, and CMA24. ACO(1) and ARI(1) gate the contents of the AC (placed there by a previous instruction) into the AR via the A bus, ADR, and O bus. These processes prepare for the execution of certain logical (AND,XOR), arithmetic (ADD,TAD), IOT, and OPR instructions.

IRI(1) of the second word turns on inverter B104-F31L (KC16) in conjunction with MBI(0) and CM STROBE DLYD. The inverter output goes to ground, triggering the 50-ns delay B310-EF29FU. Upon recovery, the delay produces an IN CLR pulse and a CLR pulse.

IN CLR produces 1 → MBI to set the MBI gate, and generate CLR I, drawing KC19(2). CLR I resets ACI, ARI, PCI, MQI, and MBO. CLR resets +1 and ACO, and sets SAO. MEM STROBE, STROBE 0-8 and STROBE 9-17 occur in core memory, drawing MC1. Strobes 0-8 and 9-17 strobe the sense amplifier contents SA00-17 out to the CP/memory interface. The sense amplifiers contain the instruction word read from core memory at the address specified by the contents of the MA.

SAO(1) gates the sense amplifier outputs onto the B bus and IRI(1) gates the op code portion SA00-04 into the IR. The B bus contents, SA00-17, go directly into the ADR, and NOSH places them on the O bus. MBI(1) then gates the contents into the MB. Thus, the entire instruction word reaches the MB for execution in accordance with the op code in the IR.

At this time, a request execution phase (REP) is determined (KC12). REP results if IRI(1) detects an IOT (R111-E14H), OPR (R111-E14H), XCT $\overline{*}$ (R111-E14U), JMP $\overline{*}$ R111-E14U), or an optional EAE instruction (R111-E14N). IRI(1) also sets the IOT flip-flop if a LAW, OPR, or IOT instruction is detected, or the CAL flip-flop if a CAL instruction is detected. The op code bits are also sampled for the ISZ instruction, independently of IRI(1).

MEM STROBE and IAO(0) start the CM timing, drawing KC16, for the third CM STROBE. MEM STROBE ∧ IAO(0) triggers pulse amplifier B602-F32D via R111-E31U. The PA pulse triggers the 50-ns delay B310-EF33FL, and the timing chain restarts.

REP allows the op code bits in the IR to change the address presented to the CM address selectors from 24 to the address appropriate to the detected instruction (CMA70, 74, 75, 76, 77). Note that, following the REP detection for OPR, LAW or JMP instructions, the instruction is executed within the fetch cycle period and the next computer cycle ensues on the fetch entry process word 21. For XCT, a quasi-fetch cycle tagged XCT entry ensues. For IOT instructions, the fetch cycle is extended by a 4.5-μs execute period during which the main core memory is idle, and the next fetch cycle ensues.

Assuming that there is no REP for the sake of convenience, the third word is extracted from CM location 24. This word contains TI, SM, and CMA30. TI(1) (test for indirect address) samples IR4 at R111-F23H on drawing KC17 for indirect addressing. If IR4 = 1, then TI(1) ∧ IR4(1) at R111-F23H boosts the CM address from 30 to 31 (defer entry). At the same time, TI(1) ∧ IR4(1) examines the main core memory address bits MB05-14 for autoindexing, drawing KC14. Bits MB05-14 are wired directly from the MB to the input gating structure at the AUT INX flip-flop.

If IR4 = 0, then TI(1) ∧ IR4(0) examines the IR bits at R111-F234 for CAL$\overline{*}$, JMS $\overline{*}$, DAC $\overline{*}$, or DZM $\overline{*}$, drawing KC17. If any of these are detected, the address gates boost the address from 30 to 32 (IAO entry).

The third word remains in the CM sense flip-flops until the next CLK pulse occurs. SM(1) waits for CM CLK to start the next computer cycle. During this waiting period, the core memory write-half-cycle restores the instruction word in the MB to the location specified by the MA. Neither the MB nor the MA has changed its contents up to this point, therefore, the instruction word is restored to the same location from which it was fetched. When the write-half-cycle ends, the next computer cycle begins and MA JAM enters the address portion of the instruction word from the MB into the MA. This sets up the core memory address of the operand which is referenced by the next computer cycle.

Depending on the conditions sensed by the IR samplings during fetch, the next computer cycle will be another fetch in the case of REP, a defer cycle in case of indirect addressing, an IAO cycle for JMS $\overline{*}$, CAL $\overline{*}$, DAC $\overline{*}$, or DZM $\overline{*}$, or an execute cycle.

5.1.2 Defer Cycle

During the third process of the fetch cycle, refer to flow drawing KC3, the address in the CMA for the next process word is 30 (execute entry). The TI(1) level samples the IR4 bit on drawing KC17. If IR4=1, the instruction word read out to the MB during fetch does not contain a direct address of an operand, but rather an indirect address; i.e., the address of the effective address. In this case, TI(1) ∧ IR4(1) on drawing KC17 enables the CMA5 gating at the address selectors in control memory. SM(1) is present in the process word to start the core memory cycle and the control memory timing on the next CLK pulse. The CMA5 gate boosts the existing address (30) to 31, from which the defer entry process word is taken at CM STROBE time.

Process word 31 contains the DEI (defer/execute initiate) process bit and CMA24. DEI(1) resets IR4 and CAL, on drawing KC12. CM STROBE DLYD, MBI(0) EXT(0), and DEI(1) produce an IN CLR and a CLR pulse after a recovered delay, drawing KC16. IN CLR produces 1 → MBI to set the MBI sense flip-flop, drawing KC19(2), and CLR sets SAO on drawing KC19(3), as for the fetch cycle. SM STROBE is prevented from restarting the CM timing chain for the next CM STROBE by the absence of a CONT(1) bit in the defer entry word. Therefore, the defer entry word remains in the sense flip-flops until MEM STROBE restarts the chain. MEM STROBE, STROBE 0-8, and STROBE 9-17 occur in core memory, as for the fetch cycle, to read the effective address word into the MB. Note that the op code placed in the IR during fetch remains unchanged throughout the defer cycle. This leaves bits MB00-04 of the effective address word available for addressing extended memory systems and for use as pointer bits. Because of this scheme, DEI(1) resets IR4 to limit indirect addressing to one level.

Before MEM STROBE produces CM STROBE as for fetch, DEI(1) samples the IR bits for REP, drawing KC12. Since DEI(1) has reset IR4, it can now sample the IR for JMP $\overline{*}$ and XCT $\overline{*}$.

MEM STROBE ∧ IA0(0) produces the CM STROBE which extracts the next process word (CMA 24) if not changed by REP as for fetch. Since DEI(1) has reset IR04 and CAL, TI(1) can neither enable CMA5 for another defer cycle nor process a true CAL instruction in a subsequent IA0 cycle. If a CAL * instruction is programmed, it will be treated in the defer cycle as a JMS * fetching an effective address from core memory location 00020. A subsequent IA0 cycle will store the conditions of the program exit point at the location reached by the effective address, and the following fetch cycle will take its instruction from that location +1 (see Sections 5.1.6.3 and 5.1.6.8). Similarly, an IA0 cycle follows defer if TI(1) ∧ IR4(0) detects a JMS *, DAC *, or DZM * op code on drawing KC17. Consequently, the CM address for the next process word can change from 30 (execute entry) to 32 (IA0 entry) or can remain at 30. SM(1) and the next CLK pulse will initiate the next computer cycle to fetch the operand (or instruction) located in core memory at the effective address.

5.1.3 Autoindexing

The use of the eight autoindex core memory locations 10-17 must be predetermined by program requirements. If an instruction word directly addresses an autoindex location, its contents comprise an operand which is treated like any other operand in executing the instruction. If the instruction word indirectly addresses an autoindex location, its contents comprise an effective address (Y) which is incremented by 1 during a defer cycle before the instruction is executed. During defer, the core memory write-half-cycle replaces Y with Y+1 in the autoindex location. During execute, therefore, the operand is fetched from location Y+1. Jumping repeatedly to an instruction which thus indirectly addresses the same autoindex location will repeatedly increment Y. This simplifies a program which performs the same arithmetic operation on sequentially located operands.

The instruction word is read out to the MB and the IR during fetch (Section 5.1.1) where IR4 is examined by TI(1) for indirect addressing. For indirect addressing, IR4 = 1; so, the TI(1) and IR4(1) levels sample address bits MB05-14 of the instruction word at the AUT INX flip-flop, drawing KC14. If bits MB05-14 designate address 0001X, they set the AUT INX flip-flop B213-D36. Examination of the least-significant bits MB15-17 is unnecessary.

During defer, the core memory read-half-cycle reads out the effective address from the autoindex location. IN CLR and CLR set SAO and MBI so that the effective address gets to the MB via the B bus, ADR, and O bus. Now, SAO(1) is gated with AUT INX(1) on drawing KC14, to produce the ground CI17 level. CI17 increments the effective address by 1 as the address passes through the ADR. NOSH takes the ADR contents to the O bus and MBI(1) places them in the MB. The core memory write-half-cycle writes the incremented address into the autoindex location. The incremented address also remains in the MB in preparation for the execute or IA0 cycle. The defer processes branch to process word 24 for most memory reference instructions to process word 70 for XCT or to 74 for JMP. TI(1) ∧ IR4(0) resets AUT INX during process word 24 or 70, and PCI(1) resets AUT INX during process word 74.

5.1.4 IA0 Cycle

The IA0 cycle replaces the normal execute cycle for CAL, JMS, DAC, and DZM instructions. These instructions neither see nor care about the contents of the core memory locations which they address. During their execution, SAO(1) is absent in the operand processing, so that the core memory read-half-cycle is ignored and the contents of the addressed location are, therefore, lost. The write-half-cycle stores new information in the addressed location in accordance with the particular instruction.

5.1.5 Execute Cycle

The computer enters the execute cycle from either the fetch cycle or the defer cycle (refer to flow diagrams KC3 and KC4). Process word 24 occurs in both cases and remains until the next CLK pulse arrives. SM(1) of process word 24 and the next CLK pulse start the control memory process and the core memory cycle. MA JAM occurs after CLK to gate the direct address (during fetch) or the effective address (during defer) into the MA from the MB. Now the core memory read-half-cycle will fetch the operand for execution of the instruction.

The CM address held in the control register during process word 24 is 30 (execute entry). SM(1) and CM CLK pulse generate the first CM STROBE in control memory. CM STROBE extracts the execute-entry process word at location 30, which contains CJIT, DEI, and CMA60.

The CM STROBE DLYD produces an IN CLR and a CLR pulse in conjunction with DEI(1), EXT(0), and MBI(0), drawing KC16. IN CLR produces 1 → MBI to set the MBI flip-flop, drawing KC19(2), and CLR sets SAO on drawing KC19(3). CM STROBE is prevented from restarting the CM timing chain for the second time because of the absence of the CONT(1) bit in the execute entry word 30. Therefore, the execute entry word remains in the sense flip-flops until MEM STROBE ∧ IA0(0) restarts the chain. STROBE 0-8 and STROBE 9-17 occur in core memory to place the operand in the MB. While the operand is on its way through the ADR to the MB, CJIT(1) will produce CI17 on drawing KC14, if the ISZ instruction op code was detected during fetch. CI17 then increments the operand by 1 in the ADR. If no ISZ, the operand remains unchanged.

MEM STROBE ∧ IA0(0) generates the next CM STROBE to extract the next process word from control memory. During the execute-entry process word 30, the address in the control register for the next word is 60 (CMA0 and CMA1 = 1). This is the starting point for the extraction of the execute word determined by the op code in the IR register. CMA0(1) and CMA1(1) on drawing KC17 allow the IR bits to address the control memory, so that address 60 is boosted to the address specified by the op code.

All memory reference instructions (except XCT) require a single 212-ns process word for execution. For these instructions, the execute process word contains CMA10 as the location of the next word. This is the BGN word which sets up the MB for the next instruction fetch cycle. CONT(1) in the execute word allows the generation of another CM STROBE to extract the BGN word.

The core memory write-half-cycle starts independently during the execute process word period. For some memory reference instructions (LAC, XOR, ADD, TAD, AND, SAD), the memory write-half-cycle restores the original operand to memory while it is being manipulated elsewhere in the CP. For ISZ the operand is incremented by 1 before being written back into memory. The remaining instructions (DAC, DZM, JMS, CAL) replace the operand entirely in the special IA0 cycle. In the last two cases, the original operand is lost.

IOT, OPR, and optional non-EAE multiply and divide instructions do not require operand access. Consequently, they do not use the computer execute cycle herein described. During the computer execute periods for these instructions, no core memory cycle occurs. Execution of these instructions takes place during normal (OPR) or extended (IOT, EAE) fetch cycles.

5.1.6 Memory Reference Instructions

The following paragraphs describe the logic functions for execution of memory reference instructions. Whereas the preceding paragraphs cover the functions that are common to all instructions, the following discussions cover those functions that are unique to the individual instructions. The instructions are arranged in alphabetical order.

5.1.6.1 1's Complement Add (ADD) - The ADD instruction (30) adds the contents of the addressed memory location (addend) to the contents of the AC (augend) in 1's complement arithmetic. The sum is deposited in the AC and the previous contents of the AC are lost. The contents of the addressed memory location remain unchanged. The Link must have been previously reset and remains reset unless an arithmetic overflow occurs.

Under the rules of 1's complement arithmetic, the sign bits 00 in both the addend and the augend are added as an integral part of the magnitude bits 01-17 during ADD. An end carry out of the sum-sign bit 00 is end-around carried into the sum-magnitude bit 17 to establish the final magnitude. Overflow occurs if the magnitude of the sum exceeds $\pm 2^{17} -1$, or ±377777. If so, the Link sets as an indication of an error in magnitude. The OPR-SNL or OPR-SZL instruction can be used to check the state of the Link following an ADD of questionable outcome.

Using a hypothetical 3-bit register and a Link bit, the examples below demonstrate all ADD possibilities. Since this is a modulo-8 register, the sum magnitude limit is $\pm 2^2 - 1$, $= 011_2$ or 100_2.

Positive Sign ADD

(1)

$$\begin{array}{rrcl} & 001 & \text{or} & +1 \\ & 001 & & +1 \\ 0 & \overline{010} & & \overline{+2}\text{ (valid)} \end{array}$$

SIGN

LINK

(2)

$$\begin{array}{rrcl} & 001 & \text{or} & +1 \\ & 011 & & +3 \\ 1 & \overline{100} & & \overline{-3}\text{(error)} \end{array}$$

(3)

$$\begin{array}{rrcl} & 011 & \text{or} & +3 \\ & 011 & & +3 \\ 1 & \overline{110} & & \overline{-1}\text{(error)} \end{array}$$

Negative Sign ADD

(4)

$$\begin{array}{rrcl} & 110 & \text{or} & -1 \\ & 110 & & -1 \\ & \overline{100} & & \overline{-3} \\ & 1 & & +1 \\ 0 & \overline{101} & & \overline{-2}\text{(valid)} \end{array}$$

(5)

$$\begin{array}{rrcl} & 110 & \text{or} & -1 \\ & 100 & & -3 \\ & \overline{010} & & \overline{+2} \\ & 1 & & +1 \\ 1 & \overline{011} & & \overline{+3}\text{(error)} \end{array}$$

Unlike Sign ADD

(6)

$$\begin{array}{rrcl} & 011 & \text{or} & +1 \\ & 110 & & -1 \\ 0 & \overline{111} & & \overline{0}\text{(valid)} \end{array}$$

(7)

$$\begin{array}{rrcl} & 011 & \text{or} & +3 \\ & 110 & & -1 \\ & \overline{001} & & \overline{+1} \\ & 1 & & +1 \\ 0 & \overline{010} & & \overline{+2}\text{(valid)} \end{array}$$

(8)

$$\begin{array}{rrcl} & 011 & \text{or} & +3 \\ & 111 & & 0 \\ & \overline{010} & & \overline{+2} \\ & 1 & & +1 \\ 0 & \overline{011} & & \overline{+3}\text{(valid)} \end{array}$$

Close examination reveals two basic rules of 1's complement addition:

a. In like-sign addition, overflow is possible and the sum sign differs if overflow occurs;

b. In unlike-sign addition, no overflow is possible and the sum sign may be plus or minus.

The fetch cycle places the ADD instruction in the MB, the op-code portion in the IR, and the contents of the AC (augend) in the AR. The op-code is sampled but does not alter the execute entry address(30).

During execute, the core memory read-half-cycle places the contents of the addressed memory location (addend) in the MB in conjunction with the execute entry word 30. The CMA in the execute entry word is 60. CMA0(1) and CMA1(1) allow the IR bits to address control memory, drawing KC17, so that the next process word is extracted from address 66. Process word 66 contains MBO, ARO, ACI, AXS, LI, DONE, CONT, and CMA10(BGN).

MBO(1) places the contents of the MB on the B bus, while ARO(1) places the contents of the AR on the A bus(refer to Figure 5-1). The contents of the buses are added in the ADR, with carries resulting if an adder stage has two or more inputs representing A1. A carry (CO00) out of ADR00 causes ADRL on drawing KC15 to go negative. ADRL gates on CI17 in conjunction with AXS(1), drawing KC14. CI17 initiates an end-around-carry into ADR17, which propagates as necessary.

The ADRL module B132-A03, drawing KC15, contains the overflow detection circuits, producing the negative ADOF level if overflow occurs. Overflow is detected by XOR of the carries CO00 and CO01 out of ADR00 and ADR01. The following functions apply to overflow detection (see ADD examples).

$\text{CO00(0)} \wedge \text{CO01(0)} = \overline{\text{ADOF}}$ (ex. 1,6)
$\text{CO00(0)} \wedge \text{CO01(1)} = \text{ADOF}$ (ex. 2,3)
$\text{CO00(1)} \wedge \text{CO01(1)} = \overline{\text{ADOF}}$ (ex. 4,7,8)
$\text{CO00(1)} \wedge \text{CO01(0)} = \text{ADOF}$ (ex. 5)

ADOF $\wedge$ AXS(1) $\wedge$ LI(1) produces the OFLO level at R111-D02HJ, drawing KC15. LI(1) and OFLO set the LAR.

DONE(1) goes to the clock and run logic, drawing KC10(1), for manual key control and to KC17 for break and PI while CONT(1) allows the generation of another CM STROBE. CM STROBE extracts the

BGN word (10) from control memory to set up the MB for the next fetch cycle. At this time, LI(1) goes to 0 and sets the Link in conjunction with LAR(1) as an indication of overflow.

LINK SAVE, gate R111-C03N (drawing KC15) sets the Link, if the Link was not reset previous to the ADD instruction, by gating LINK(1) and AXS(1). LI(1) sets the LAR under this condition, then LI(0) sets the Link with LAR(1).

5.1.6.2 Logical AND (AND) - The AND instruction (50) logically ANDs the contents of the addressed memory location with the contents of the AC on a bit-for-bit basis. If corresponding bits are 1s, the result is 1. If corresponding bits differ or are 0s, the result is 0. The results are stored in the AC and the previous AC contents are lost. The contents of the Link and the addressed memory location remain the same.

The fetch cycle places the AND instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR. The op-code is sampled but does not alter the execute entry address(30).

During execute entry, the core memory read-half-cycle places the contents of the addressed memory location in the MB, in conjunction with the execute entry process word (30). The CMA in the execute entry word is 60. CMA0(1) and CMA1(1) allow the IR bits to address control memory, drawing KC17, so that the next word is extracted from location 72. Process word 72 contains MBO, ARO, AND, ACI, DONE, CONT and CMA10 (BGN). MBO(1) gates the contents of the MB onto the B bus, while ARO(1) gates the contents of the AR onto the A bus. The contents of both buses go into the ADR. Additionally, the complement of the A bus is gated onto the B bus by the AND(1) level ($\overline{\text{A BUS}}$, drawing KC21). AND(1) on drawing KC13 produces CMPL, which is applied to each bit of the

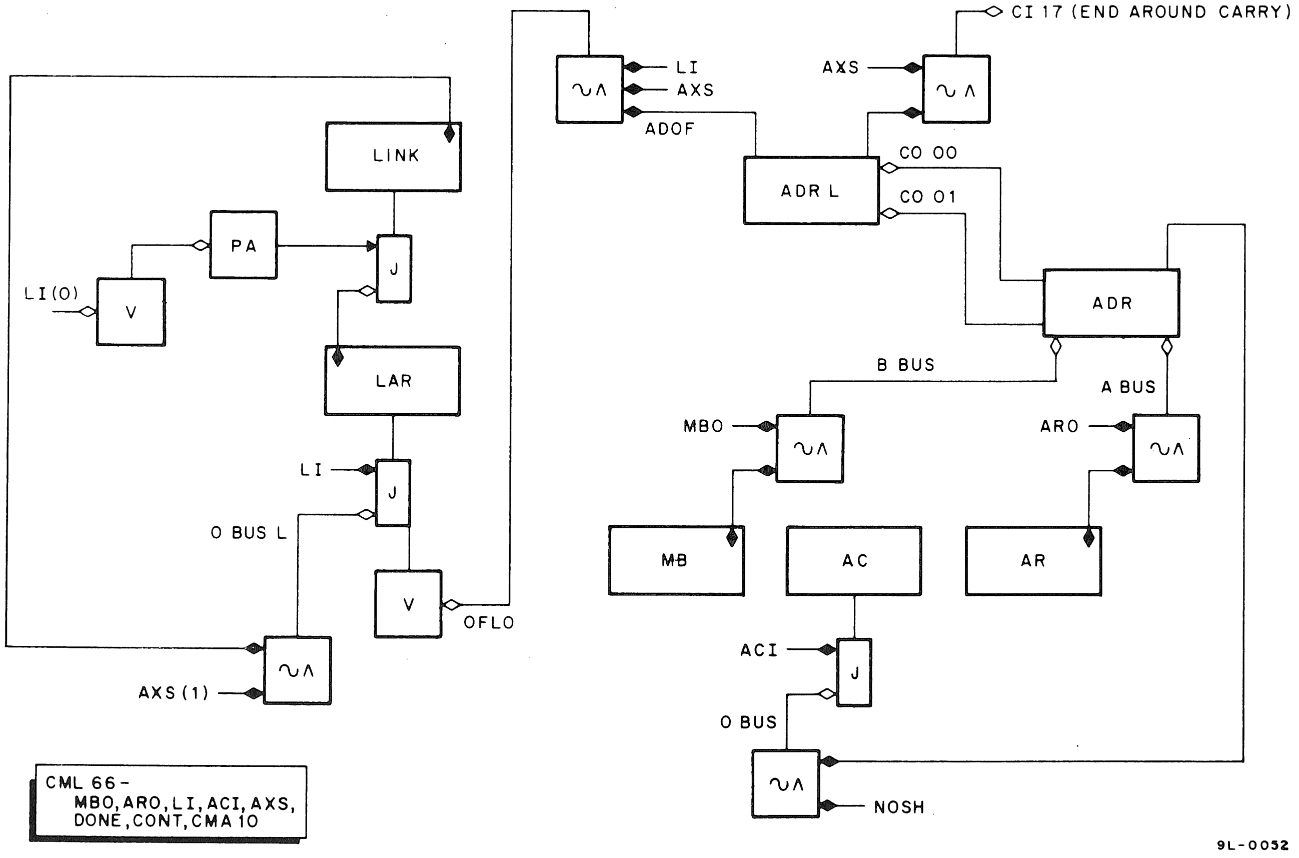


Figure 5-1 ADD Instruction Logic

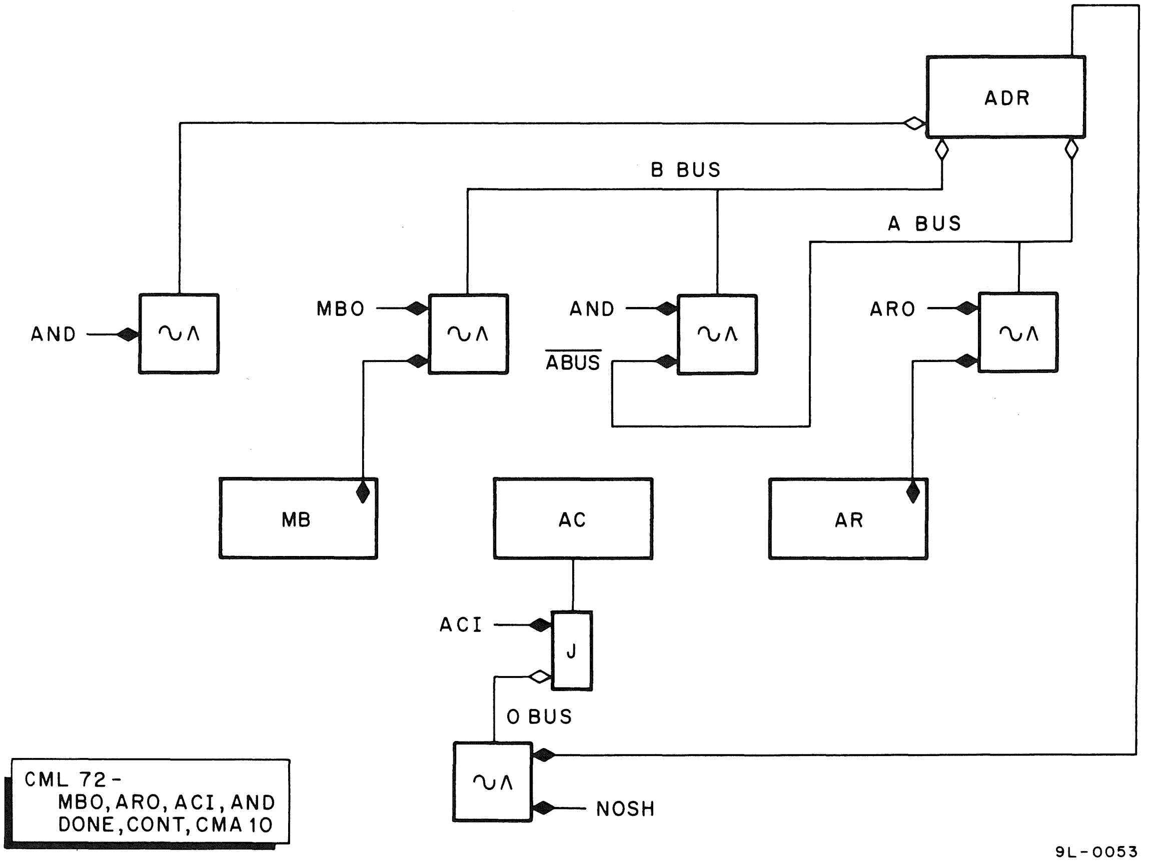


Figure 5-2 AND Logic

ADR to complement the half-add results. Figure 5-2 illustrates the AND logic for one bit position. If the respective MB and AR bits are 1s, they appear as ground levels on the A and B bus inputs to the ADR. This results in a ground level output which is then forced to -3V by the CMPL level. NOSH gates the negative level onto the O bus and ACI(1) jams a 1 into the AC.

If the respective bits differ, one of the inputs to the ADR is at ground and the ADR output goes negative. CMPL then forces it to ground. NOSH places a negative level on the O bus, and ACI(1) jams a 0 into the AC.

If the respective bits are 0s, both buses go negative, but $\overline{\text{A BUS}}$ makes the B bus go to ground to present the ADR with the "differ" conditions above. The result is a 0 in the AC.

DONE(1) goes to the clock and run logic (drawing KC10(1) for manual key control and KC17 for break and PI) while CONT(1) allows the generation of another CM STROBE. CM STROBE extracts the BGN word (10) from control memory, which sets up the MB for the next fetch cycle.

5.1.6.3 Call Subroutine (CAL) - The CAL instruction (00) is equivalent to instruction JMS 20, Section 5.1.6.8. The contents of the PC, the Link, and the status of the extended memory mode and memory protect mode (on or off) are deposited in memory location 00020. The previous contents of location 00020 are lost. The next instruction is read from memory location 00021, breaking the previous program sequence. The contents of the AC and Link remain unchanged.

The fetch cycle places the CAL instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR. The op-code is detected to set the CAL flip-flop, drawing KC12, and to extract the next process word from location 24 on the next CM STROBE. Process word 24 contains TI, SM, and CMA30. TI(1) on drawing KC17 samples bits IR0, IR1, and IR4 at the CM address gating. For a CAL $\overline{*}$ instruction (and DAC $\overline{*}$, DMZ $\overline{*}$, and JMS $\overline{*}$, these bits are all 0s, changing the presented CM address from 30 (execute entry) to 32 (IA0 entry). TI(1) and CAL(1) from the set CAL flip-flop place a 1 level (ground) on O BUS 13, drawing KC22. SM(1) waits for CLK on drawing KC19(2) at the 1 → MBI gate. CLK occurs in core memory just before the next CM CLK pulse starts the IA0 cycle. At the 1 → MBI gating, CLA(1) ∧ SM(1) ∧ CLK produces the 1 → MBI level, setting the MBI flip-flop. MBI(1) gates address 00020 from the O bus to the MB.

SM(1) and CM CLK generate CM STROBE in the CM timing, drawing KC16, to extract the IA0 entry word from location 32. Process word 32 contains PCO, ARI, CONT, and CMA23. PCO(1) gates the current contents of the PC, EPC, the Link, memory EXD mode, and memory protect mode status onto the A bus.

The contents on the A bus go directly through the ADR and NOSH places them on the O bus. ARI(1) transfers the contents of the O bus to the AR.

For any memory capacity up to fully-extended 32K systems, the PC register uses only 13 bits of the 18 available in the normal computer word, PC05-17. Of the five vacant bits in the address, bits 00-02 are used for gating the Link, EXD mode and memory protect mode status onto A BUS 00, A BUS 01, and A BUS 02, respectively, drawing KC20(1). The remaining bits EPC03, EPC04 come from the extended memory control option. Then ARI(1) of process word 32 gates these status bits into the AR, along with the contents of the PC above.

CONT(1) allows CM STROBE to restart the CM timing for the extraction of the next process word from location 23. Process word 23 contains MBO, +1, CJIT, CONT, and CMA60. MBO(1) gates the contents of the MB (00020) to the B bus, and then directly to the ADR. Process +1 produces CI17 on drawing KC14, which increments the address as it passes through the ADR. NOSH places the incremented address (00021) on the O bus. CJIT(1) generates 1 → PCI on drawing KC12 in conjunction with the IR00, IR01, and IR03 bits (all 0s). The 1 → PCI level sets the PCI flip-flop, drawing KC19(2). PCI(1) then gates address 00021 from the O bus into the PC.

Note that during this second process word (23) of the IA0 cycle, MEM STROBE, STROBE 0-8, and STROBE 9-17 occur in core memory to read out the contents of memory location 00020. However, the SAO and MBI gates are disabled in the absence of CLR on drawing KC16, so that the contents cannot reach the MB and are therefore lost.

CONT(1) in process word 23 allows CM STROBE to restart the CM timing to extract the third process word from location 60. Process word 60 contains ARO, MBI, DONE, CONT and CMA10 (BGN). ARO(1) gates the contents of the AR onto the A bus. (The AR contains the disrupted address from the PC, EPC, and the status bits discussed earlier.) The contents on the A bus go directly into the ADR and NOSH places them on the O bus. MBI(1) gates the contents from the O bus to the MB. At this time the core memory write-half-cycle stores this PC and status information in location 00020.

DONE(1) goes to the clock and run logic, drawing KC10(1) for manual key control and KC17 for break and PI while CONT(1) allows the generation of the fourth CM STROBE to extract the BGN word from location 10. The BGN word gates the new address held in the PC (00021) into the MB for the next fetch cycle. Thus, a new sequence of instructions starts from address 00021. A JMP * instruction can be used to return to the sequence stored at 00020. JMP * should be preceded by an IOT DBR instruction in order to restore the status bits to the system.

5.1.6.4 Deposit Accumulator (DAC) - The DAC instruction (04) deposits the contents of the AC in the addressed memory location. The previous contents of the addressed memory location are lost and the contents of the AC and Link remain unchanged.

The fetch cycle places the DAC instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR.

The op code is detected to extract the next process word from location 24 on the next CM STROBE. Process word 24 contains TI, SM, and CMA30. TI(1) on drawing KC17 samples bits IR0, IR1, and IR4 at the CM address gating. For a DAC $\overline{*}$ instruction (and CAL $\overline{*}$, JMS $\overline{*}$, DZM $\overline{*}$) these bits are all 0s, changing the presented CM address from 30 (execute entry) to 32 (IA0 entry).

SM(1) and the next CM CLK pulse generate CM STROBE in the CM timing, drawing KC16, to extract the IA0 entry word from location 32. Process word

32 contains PCO, ARI, CONT, and CMA23. PCO(1) gates the current contents of the PC, EPC, Link, memory EXD mode, and memory protect mode status onto the A bus. The contents on the A bus go directly through the ADR, and NOSH places them on the O bus. ARI(1) transfers the contents of the O bus to the AR. This process is common and useful only to the CAL and JMS instructions (and to program interrupt operations) where the current contents of the PC, EPC, and the status bits are to be stored in core memory. For DAC (and DZM) instructions, the contents do not get past the AR.

CONT(1) allows CM STROBE to restart the CM timing chain to extract the next process word from location 23. Process word 23 contains MBO, +1, CJIT, CONT, and CMA60. MBO(1) gates the contents of the MB (DAC instruction) onto the B bus, and the B bus contents go directly into the ADR. NOSH places the ADR contents on the O bus. For DAC, the contents do not get beyond the O bus.

This process is useful only to the CAL and JMS instructions (and to program interrupt operations) where CJIT(1) ∧ IR3(0) produce 1 → PCI on drawing KC12 for program count and status storage. For DAC, IR3 = 1, so that the PCI gate does not set.

Note that MEM STROBE, STROBE 0-8, and STROBE 9-17 occur in core memory to read out the contents of the addressed memory location. However, the SAO and MBI gates are disabled in the absence of CLR on drawing KC16, so that the contents cannot reach the MB and are therefore lost. Process word 61 will replace these contents with the contents of the AC.

CONT(1) allows CM STROBE to extract the third process word. The CMA in process word 23 is 60. CMA0(1) and CMA1(1) allow the IR bits to address the control memory, in which case IR3(1) changes the presented address from 60 to 61. Process word 61 contains ACO, MBI, DONE, CONT, and CMA10 (BGN). ACO(1) gates the contents of the AC onto the A bus (refer to Figure 5-3), the contents go from the A bus to the ADR, and NOSH places them on the O bus. MBI(1) gates them from the O bus to the MB.

DONE(1) goes to the clock and run logic, drawing KC10(1) for manual key control and KC17 for break and PI, while CONT(1) allows the generation of a

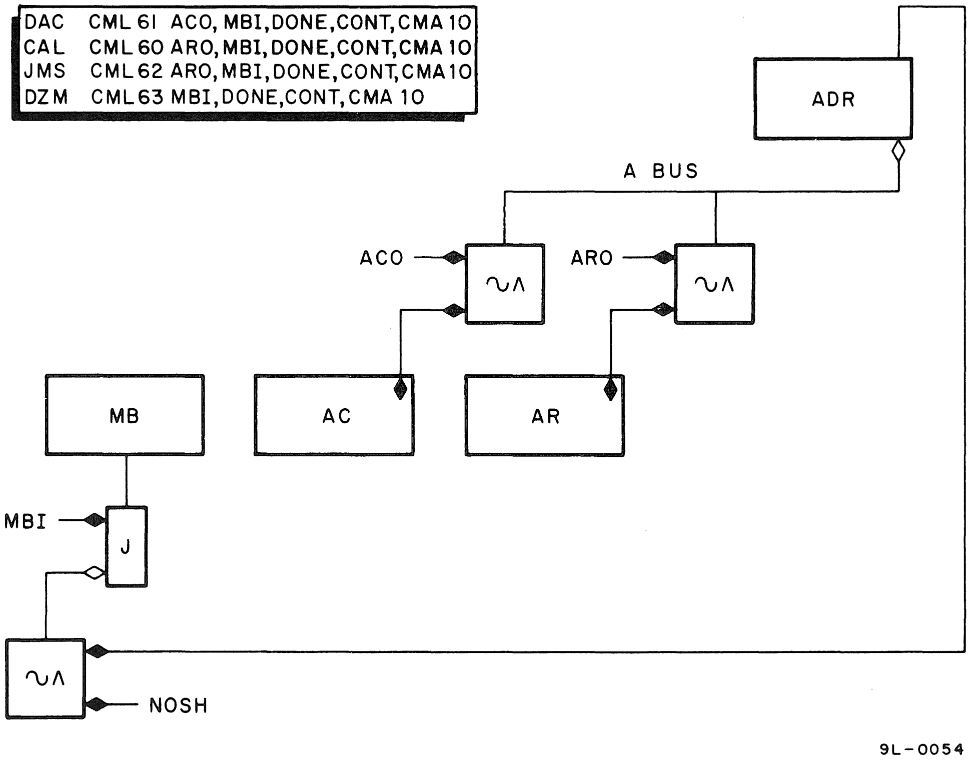

DAC CML 61 ACO, MBI, DONE, CONT, CMA 10
CAL CML 60 ARO, MBI, DONE, CONT, CMA 10
JMS CML 62 ARO, MBI, DONE, CONT, CMA 10
DZM CML 63 MBI, DONE, CONT, CMA 10
ADR
A BUS
ACO
∿∧
ARO
∿∧
MB
AC
AR
MBI
J
∿∧
NOSH
9L-0054

Figure 5-3 IA0 Instruction Logic For: DAC, CAL, JMS, and DZM

fourth CM STROBE to extract the BGN word(10). The BGN word sets up the MB for the next fetch cycle.

5.1.6.5 Deposit Zero in Memory (DZM) - the DZM instruction (14) deposits all 0s in the addressed memory location. The previous contents of the addressed location are lost and the contents of the AC and Link remain unchanged.

The fetch cycle places the DZM instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR.

The op code is detected to extract the next process word from location 24 on the next CM STROBE. Process word 24 contains TI, SM, and CMA30. TI(1) on drawing KC17 samples bits IR0, IR1, and IR4 at the CM address gating. For a DZM $\overline{*}$ instruction (and CAL $\overline{*}$, DAC $\overline{*}$, JMS $\overline{*}$) these bits are all 0s, changing the next CM address from 30 (execute entry) to 32 (IA0 entry).

SM(1) and the next CM CLK pulse generate CM STROBE in the CM timing, drawing KC16, to extract the IA0 entry word from location 32. Process word 32 contains PCO, ARI, CONT, and CMA23. PCO(1) gates the current contents of the PC, EPC, Link, memory extend mode, and memory protect mode status onto the A bus. The contents on the A bus go directly through the ADR and NOSH places them on the O bus. ARI(1) transfers the contents of the O bus to the AR. This process is common and useful only to the CAL and JMS instructions (and program interrupt operations) where the current contents of the PC are stored in core memory along with the status bits. For DZM and DAC instructions the contents do not get past the AR.

CONT(1) allows CM STROBE to restart the CM timing chain to extract the next process word from location 23. Process word 23 contains MBO, +1, CJIT, CONT, and CMA60. MBO(1) gates the contents of the MB (DZM instruction) onto the B bus, and the B bus contents go directly into the ADR. NOSH places the ADR contents on the O bus. For DZM, the contents do not get beyond the O bus. This process is useful only to the CAL and JMS instructions (and to program interrupt operations) where CJIT $\wedge$ IR3(0) on drawing KC12 produces 1 → PCI for program count and status storage. For DZM, IR3 = 1, so that the PCI gate does not set.

Note that MEM STROBE, STROBE 0-8, and STROBE 9-17 occur in core memory to read out the contents of the addressed memory location. However, the SAO and MBI gates are disabled in the absence of CLR on drawing KC16, so that the contents cannot reach the MB, and are therefore lost. Process word 63 will replace these contents with 0s.

CONT(1) allows CM STROBE to extract the third process word. The CMA in process word 23 is 60. CMA0(1) and CMA1(1) allow the IR bits to address the control memory, in which case the presented address is changed from 60 to 63. Process word 63 contains MBI, DONE, CONT, and CMA10 (BGN). MBI(1) gates the contents of the O bus into the MB. Since there is nothing on the O bus at this time, the MB is loaded with 0s.

DONE(1) goes to the clock and run logic, drawing KC10(1) for manual key control and KC17 for break and PI, while CONT(1) allows the generation of a fourth CM STROBE to extract the BGN word (10). The BGN word sets up the MB for the next fetch cycle.

5.1.6.6 Increment and Skip if Zero (ISZ) - The ISZ instruction (44) increments the contents of the addressed memory location by 1 and tests the result. If the result is 0, the contents of the PC are incremented by 1, so that the computer skips the next instruction. If the result is other than 0, the computer executes the next instruction. The contents of the AC and Link remain unchanged.

The fetch cycle places the ISZ instruction in the MB, the op-code portion in the IR and the contents of the AC in the AR.

The op code is detected to produce the ISZ level, drawing KC12, and to extract the next process word from location 24 on the next CM STROBE. TI(1) of process word 24 tests for indirect addressing and SM(1) waits for the next CM CLK pulse to restart the CM timing and the core memory cycle. The CM address in process word 24 is 30 (execute entry). SM(1) and CM CLK generate CM STROBE to extract the execute entry word. This word contains CJIT, DEI, and CMA60. During the execute entry process, the core memory read-half-cycle places the contents of the addressed memory location in the MB via the B bus, ADR, and O bus. The absence of the CONT(1) bit in process word 30 means that the process remains active throughout the period normally allotted to a second process word.

Therefore, as the contents of the addressed memory location pass through the ADR, CJIT(1) is present, on drawing KC14, to produce CI17 in conjunction with the ISZ level. CI17 is applied to ADR17 to increment the contents.

MEM STROBE in core memory and IA0(0) restart the CM timing on drawing KC16. Since the CM address in process word 30 is 60, CMA0(1) and CMA1(1) allow the IR bits to address the control memory, drawing KC17. The IR bits boost the address to 71 so that CM STROBE, initiated by MEM STROBE, extracts the next process word from that location. Process word 71 contains MBO, ARI, SKPI, DONE, CONT, and CMA10 (BGN). MBO(1) gates the incremented contents from the MB to the B bus, the B bus contents go through the ADR directly, and NOSH places them on the O bus. (Refer to Figure 5-4.) As the contents pass through the ADR, an output bus (ADRA = 0, ADRB = 0) goes negative if the ADR goes to all 0s. The negative ADRA = 0, ADRB = 0 levels are applied to the jam input gate of the ADR = 0 SAVE flip-flop, drawing KC14. ARI(1) of the process word sets the flip-flop and gates the contents of the O bus into the AR.

The core memory write-half-cycle writes the incremented contents of the MB into the addressed memory location. ADR = 0 SAVE(1) and ISZ are gated on drawing KC14 to place a ground level at the jam input gate of the SKIP flip-flop. SKPI(1) of process word 71 sets the flip-flop. (The flip-flop was reset by PCI(1) during the fetch entry process word 21.)

CONT(1) of process word 71 allows the generation of a third CM STROBE to extract the BGN word from location 10. With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17, drawing KC14, so that the current contents of the PC are incremented by 1 as they pass through the ADR to the O bus and MB.

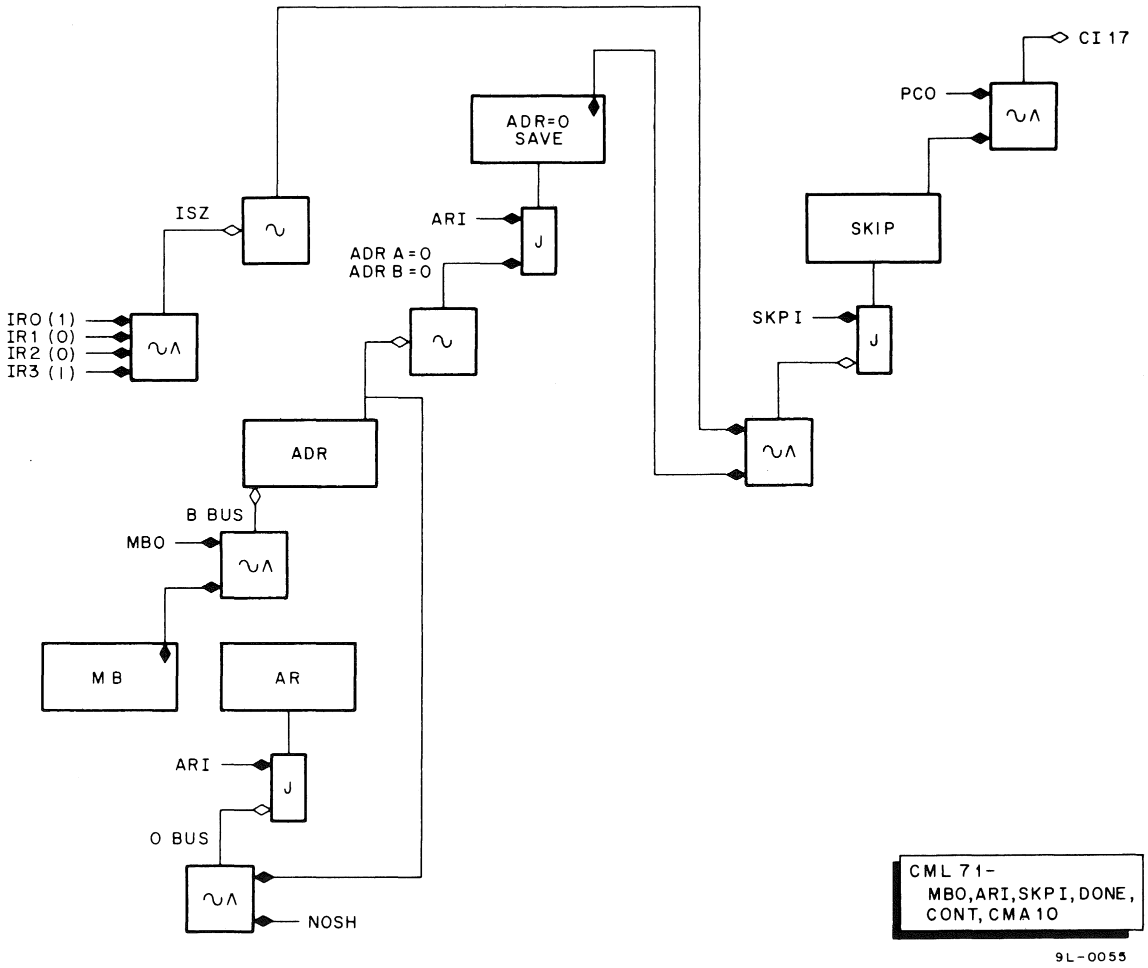


Figure 5-4 ISZ Instruction Logic

5.1.6.7 Jump (JMP) - The JMP instruction (60) transfers the program sequence to the address specified in the instruction. If the JMP instruction contains a direct address, this address is transferred during fetch to the PC (refer to fetch flow diagram KC3). If the JMP instruction contains an indirect address, the computer enters a defer cycle to fetch the effective address. REP then transfers the effective address to the PC during defer. In both cases, the previous contents of the PC are lost. The computer enters another fetch cycle from the new address in the PC. The contents of the AC and Link remain unchanged.

During fetch, IRI(1) detects the JMP $\overline{*}$ op code in the IR bits to provide the REP ground level, drawing KC12. During defer, DEI(1) resets IR4, and thus also detects a JMP op code for REP. REP goes to the CM addressing logic, drawing KC17, to gate the IR bits into the address selectors. The CM STROBE derived from MEM STROBE causes the third process word to be extracted from location 74 (JMP). The word in 74 contains MBO, PCI, LI, DONE, CONT, and CMA10 (BGN).

MBO(1) gates the address in the JMP instruction from the MB to the B bus, where it is fed directly into the ADR (refer to Figure 5-5). NOSH gates the address onto the O bus and PCI(1) puts it in the PC. LI(1) gates the Link content into the LAR via the ADRL. DONE(1) goes to the clock and run logic, drawing KC10(1), for manual key control and KC17 for break and PI, while CONT(1) allows the generation of a fourth CM STROBE. The fourth process word to be extracted is the BGN word (10) which sets up the MB for the coming fetch cycle. LI(1) of process word 74 goes to 0 at BGN time, strobing the content of the LAR into the Link. This recirculation of the Link is done mainly to restore the Link status when JMP * is preceded by DBR, Section 8.1.7.

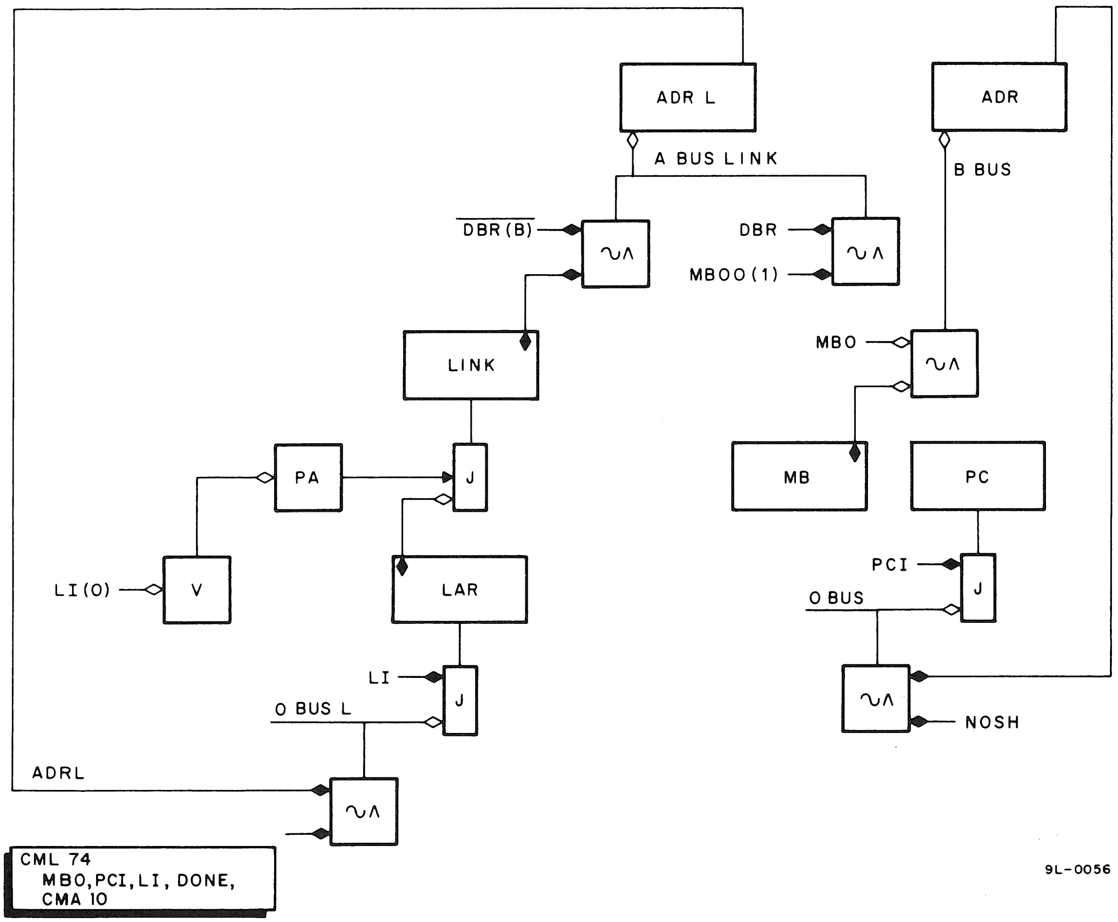


Figure 5-5 JMP Instruction Logic

5.1.6.8 Jump to Subroutine (JMS) - The JMS instruction (10) permits exit from the main program into a subroutine. The contents of the PC, the Link, and the status of the EPC, extended memory mode, and memory protect mode are deposited in the addressed memory location Y. The next instruction is taken from location Y+1, breaking the main program sequence and starting a new sequence from Y+1. The previous contents of Y are lost, and the contents of the AC and Link remain unchanged.

The fetch cycle places the JMS instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR.

The op code is detected to extract the next process word from location 24 on the next CM STROBE. Process word 24 contains TI, SM, and CMA30. TI(1) on drawing KC17 samples bits IR0, IR1, and IR4 at the CM address gating. For a JMS $\overline{*}$ instruction (and DAC $\overline{*}$, DZM $\overline{*}$, CAL $\overline{*}$) these bits are all 0s, changing the CM address from 30 (execute entry) to 32 (IA0 entry).

SM(1) and the next CM CLK pulse generate CM STROBE in the CM timing, drawing KC16, to extract the IA0 entry word from location 32. Process word 32 contains PCO, ARI, CONT, and CMA23. PCO(1) gates the current contents of the PC onto the A bus. The contents on the A bus go directly through the ADR, and NOSH places them on the O bus. ARI(1) transfers the contents of the O bus to the AR.

For any memory capacity up to fully extended 32K system, the PC register uses only 13 bits of the 18 available in the normal computer word, PC05-17. Of the five vacant bits in the address, bits 00-02 are used to gate the Link, EXD mode and memory protect mode status onto A BUS 00, A BUS 01, and A BUS 02, respectively, drawing KC20(1). The remaining bits EPC03, EPC04, come from the extended memory control option. ARI(1) of process word 32 then gates these status bits into the AR along with the contents of the PC above.

CONT(1) allows CM STROBE to restart the CM timing for the extraction of the next process word from location 23. Process word 23 contains MBO, +1, CJIT, CONT, and CMA60. MBO(1) gates the contents of the MB (address Y) to the B bus, and the contents go from the B bus directly to the ADR. Process +1 (1) produces CI17, on drawing KC14, thus incrementing the address as it passes through the ADR.

NOSH places the incremented address (Y+1) on the O bus. CJIT(1) of process word 23 generates 1 → PCI on drawing KC12 in conjunction with the IR0, IR1, and IR3 bits (all 0s). The 1 → PCI level sets the PCI flip-flop, drawing KC19(2). PCI(1) then gates address Y+1 from the O bus into the PC.

Note that during this second process word (23) of the IA0 cycle, MEM STROBE, STROBE 0-8, STROBE 9-17 occur in core memory to read out the contents of memory location Y. However, the SAO and MBI gates are disabled in the absence of CLR on drawing KC16, so that the contents cannot reach the MB and are therefore lost.

CONT(1) allows CM STROBE to restart the CM timing for the extraction of the third process word. The CM address in process word 23 is 60 (CMA0, CMA1 = 1). CMA0(1) and CMA1(1) allow the IR bits to address the control memory, in which case IR2(1) changes the address from 60 to 62. The process word at location 62 is extracted at the third CM STROBE. Process word 62 contains ARO, MBI, DONE, CONT, and CMA10 (BGN). ARO(1) gates the contents of the AR onto the A bus. (The AR contains the disrupted contents of the PC, EPC, and the Link, EXD mode, and memory protect mode status discussed earlier.) The contents on the A bus go directly into the ADR, and NOSH places them on the O bus. MBI(1) gates the contents from the O bus to the MB. At this time, the core memory write-half-cycle stores this PC and status information in location Y.

DONE(1) goes to the clock and run logic, drawing KC10(1), for manual key control and KC17 for break and PI, while CONT(1) allows the generation of the fourth CM STROBE to extract the BGN word (10). The BGN word gates the new address held in the PC (Y+1) into the MB for the next fetch cycle. Thus a new sequence of instructions starts from address Y+1. A JMP * instruction can be used to return to the stored sequence at Y. JMP * should be preceded by an IOT DBR to restore the stored status bits to the system (Section 8.1.7).

5.1.6.9 Load the Accumulator (LAC) - The LAC instruction (20) loads the contents of the addressed memory location into the AC. The previous contents of the AC are lost and the Link remains unchanged.

The fetch cycle places the LAC instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR. The op code is sampled but does not alter the execute entry address (30). During execute, the core memory read-half-cycle places the contents of the addressed memory location in the MB in conjunction with the execute entry word 30. The CMA in the execute entry process word is 60. CMA0(1) and CMA1(1) allow the IR bits to address the control memory, drawing KC17. The CM STROBE initiated by MEM STROBE then extracts the next process word from location 64. Process word 64 contains MBO, ACI, DONE, CONT, and CMA10. MBO(1) gates the contents of the MB onto the B bus, the B bus contents go directly through the ADR, and NOSH places them on the O bus (refer to Figure 5-6). ACI(1) gates the contents on the O bus into the AC.

DONE(1) goes to the clock and run logic, drawing KC10(1) for manual key control and KC17 for break and PI, while CONT(1) allows the generation of a third CM STROBE to extract the BGN process word from location 10. The BGN word sets up the MB for the next fetch cycle.

5.1.6.10 Skip if AC Differs (SAD) - The SAD instruction (54) compares the contents of the addressed memory location with the contents of the AC. If the contents differ, the PC is incremented by 1 and the computer skips the next instruction. If the contents are the same, the computer executes the next instruction. The contents of the addressed memory location and the contents of the AC and Link remain unchanged.

The fetch cycle places the SAD instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR. The op code is sampled but does not alter the execute entry address 30. During execute the core memory read-half-cycle places the contents of the addressed memory location in the MB in conjunction with the execute entry word. The CMA in the execute entry word is 60. CMA0(1) and CMA1(1) allow the IR bits to address the control memory drawing KC17. The CM STROBE initiated by MEM STROBE then extracts the next process word from location 73. Process word 73 contains SUB, ACO, AXS, SKPI, ARI, DONE, CONT, and CMA10 (BGN).

SUB(1) takes the complement of the MB to the B bus and ACO(1) takes the direct outputs of the AC to the A bus (see Figure 5-7). Both A bus and B bus contents go directly to the ADR. If the output bus levels are both at ground or are both negative for any bit position, their corresponding MB and AC inputs differ and the half-add result out of the ADR bit is at ground. In the ADR, CMPL complements the half-add result, forcing the ADR bit output to go negative. CMPL is derived from AXS(1) of the process word and IR3(1) on drawing KC13. All ADR bit outputs are inverted

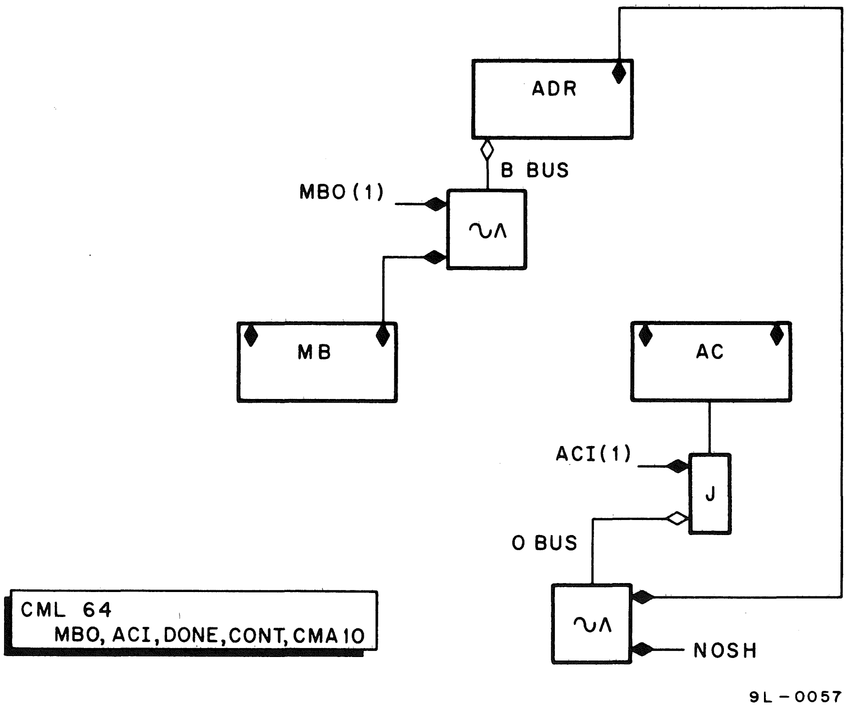


Figure 5-6 LAC Instruction Logic

and are placed on one of two common buses (ADRA =0, ADRB =0). Thus, a difference in any bit position forces a bus to ground.

On drawing KC14, AXS(1) is NAND-gated with the NORed ADRA = 0 or ADRB = 0 level at ground. This gate (R111-E35U) places a ground level at the jam input gate of the SKIP flip-flop. SKPI(1) of the process word sets the SKIP flip-flop. (The SKIP flip-flop was reset by PCI(1) of the fetch entry process word 21.)

DONE(1) goes to the clock and run logic, drawing KC10(1) for manual key control and KC17 for break and PI, while CONT(1) allows the generation of a third CM STROBE to extract the BGN process word from location 10. With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17, drawing KC14, so that the contents of the PC are incremented by 1 as they pass through the ADR to the O bus and MB.

5.1.6.11 2's Complement Add (TAD) - The TAD instruction (34) adds the contents of the addressed memory location to the contents of the AC in 2's complement arithmetic. The sum is deposited in the AC and the previous contents of the AC are lost. The contents of the addressed memory location remain unchanged. A carry out of the sum sign bit 00 complements the Link.

The fetch cycle places the TAD instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR.

The op code is sampled but does not alter the execute entry address 30. During execute entry, the core memory read-half-cycle places the contents of the addressed memory location in the MB in conjunction with the execute entry word. The CM address in the execute entry word is 60 (CMA0, CMA1 = 1). CMA0(1) and CMA1(1) allow the IR bits to address the control

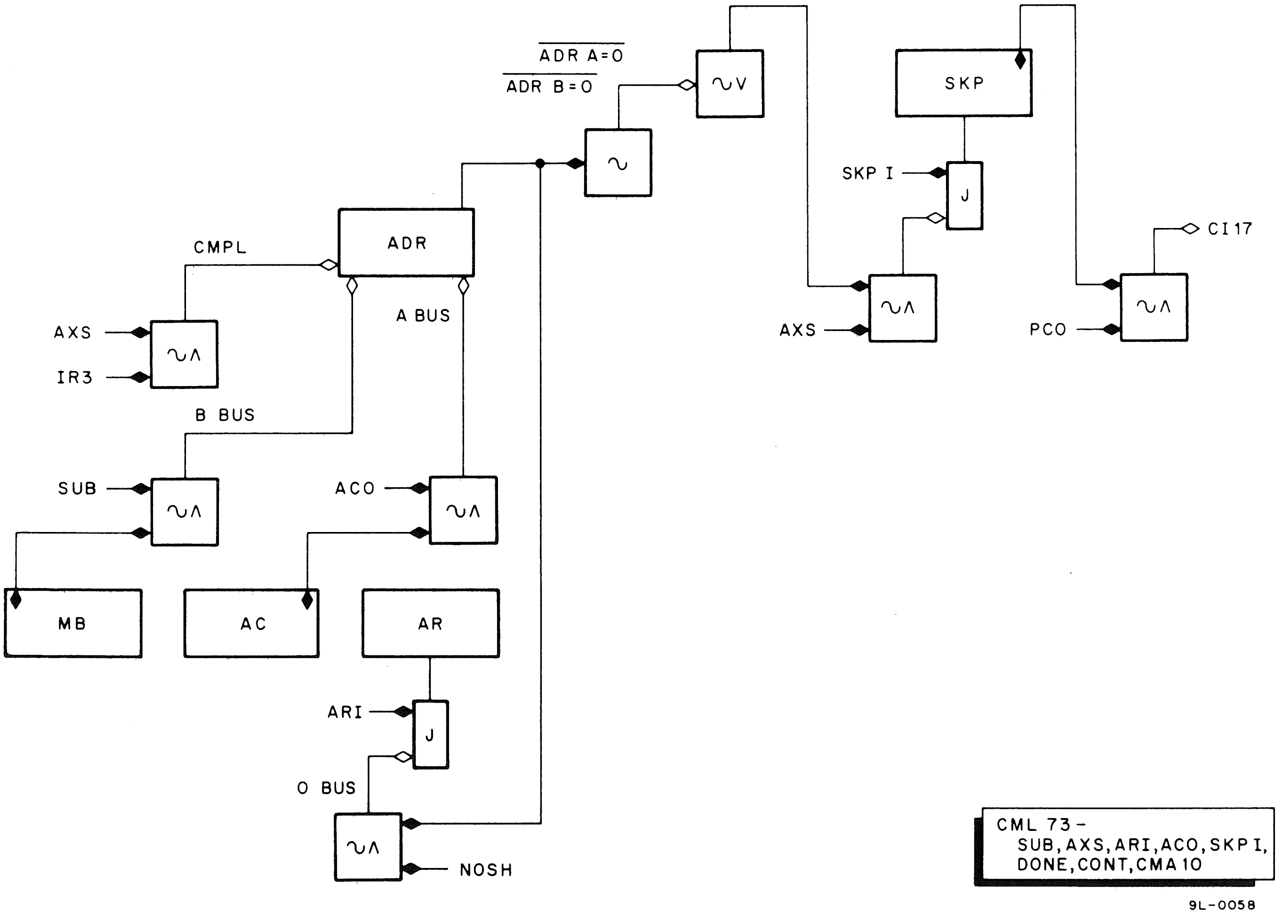


Figure 5-7 SAD Instruction Logic

memory, drawing KC17, so that the next word is extracted from location 67. Process word 67 contains MBO, ARO, ACI, LI, DONE, CONT, and CMA10 (BGN).

MBO(1) places the contents of the MB on the B bus, while ARO(1) places the contents of the AR on the A bus (refer to Figure 5-8). The contents of the buses are added in the ADR, with carries resulting where two or more input bits are 1s.

NOSH gates the contents of the ADR to the O bus, and ACI(1) gates them into the AC. LI(1) on KC15 samples the state of ADRL at the LAR. ADRL represents the state of the Link. If the Link was set and no carry resulted from ADR00, ADRL is negative. If the Link was set and a carry resulted from ADR00, CO00 forces the ADRL to ground. LI(1) gates the status of ADRL into the LAR via the "normal" gating on drawing KC15, in conjunction with the negation levels AXS(0), SHIFT, etc.

DONE(1) goes to the clock and run logic drawing KC10(1) for manual key control and KC17 for break and PI, while CONT(1) allows the generation of a third CM STROBE to extract the BGN word (10) from control memory. The BGN processes set up the MB for the next fetch cycle. The LI process upon going to 0 strobes the status of the LAR into the Link.

5.1.6.12 Execute (XCT) - The XCT instruction (40) causes the computer to execute the instruction contained in the addressed memory location. If the XCT instruction contains a direct address, REP is detected during fetch. The computer waits for the next CLK pulse, then enters a quasi-fetch cycle which fetches the instruction to be executed. If the XCT instruction contains an indirect address, the computer goes into a defer cycle to fetch the effective address, then enters the quasi-fetch cycle.

During fetch IRI(1) in process word 12 detects XCT $\overline{*}$ in the IR bits to provide a ground REP level, drawing KC12 (IR4 = 0 at R002-E13M). During defer DEI(1) resets IR4 and thus also detects an XCT op code for REP. In either case, REP and CMA1(1) at the CM addressing logic, drawing KC17, gate the IR bits into the address selectors. CMA1(1) is present because the CMA is 24. The IR gating changes the address from 24 to 70 (XCT).

The CM STROBE derived from MEM STROBE and IA0(0) causes the next (third) fetch cycle process word

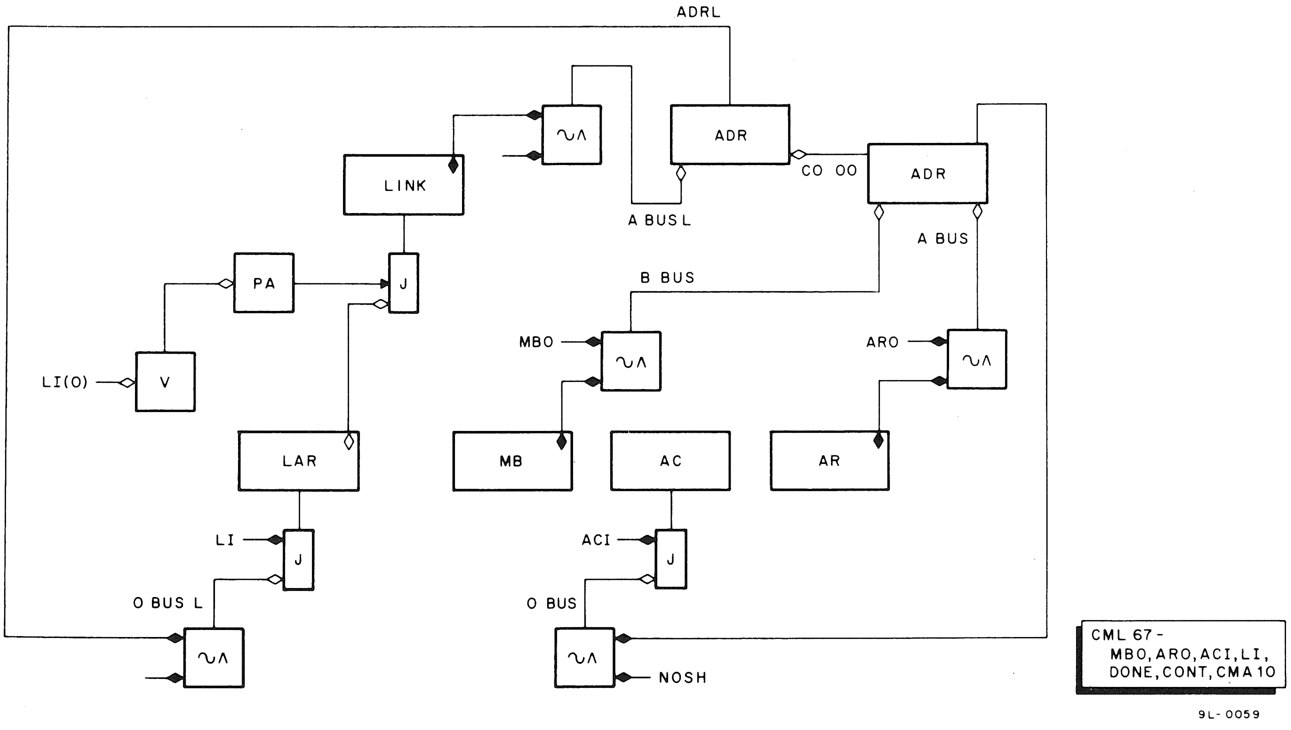


Figure 5-8 TAD Instruction Logic

to be extracted from location 70. The word in 70 contains SM, TI, and CMA33 (XCT entry). SM(1) waits for the next CLK pulse to start the quasi-fetch cycle from XCT entry. TI(1) allows XCT * to be used in the optional extend mode addressing scheme.

For XCT entry, the CM STROBE initiated by SM(1) and CM CLK extracts the XCT entry word in 33 which contains IRI and CMA24. The CM STROBE that extracted the XCT entry word restarts the CM timing, drawing KC16, but the absence of CONT(1) prevents the extraction of the normally timed second process word. CM STROBE, however, produces the IN CLR and CLR pulses in conjunction with MBI(0), EXT(0), and IRI(1). IN CLR produces 1 → MBI to set the MBI flip-flop, drawing KC19(2), and CLR sets SAO on drawing KC19(3). STROBE 0-7 and 8-17 occur in core memory to strobe the sense amplifier contents out to the CP/memory interface. The sense amplifiers contain the instruction word addressed by the XCT instruction. SAO(1) and MBI(1) gate the instruction word to the MB via the B bus, ADR, and O bus. IRI(1) gates the op-code portion into the IR.

From here the instruction is sampled and treated like any other instruction in a normal fetch or defer cycle. MEM STROBE and IA0(0) on drawing KC16 allow the generation of the next normally timed CM STROBE, which extracts the next process word (from location 24) if not changed by REP.

5.1.6.13 Exclusive OR (XOR) - The XOR instruction (24) performs the exclusive OR function between the contents of the addressed memory location and the contents of the AC on a bit-for-bit basis. If corresponding bits are the same, the AC bit is set to 0. If corresponding bits differ, the AC bit is set to 1. The previous contents of the AC are lost and the contents of the addressed memory location and the Link remain unchanged.

The fetch cycle places the XOR instruction in the MB, the op-code portion in the IR, and the contents of the AC in the AR.

The op code is sampled but does not alter the execute entry address 30. During execute entry the core memory read-half-cycle places the contents of the addressed memory location in the MB in conjunction with the execute entry word. The CMA in the execute entry word is 60 (CMA0, CMA1 = 1). CMA0(1) and CMA1(1) allow the IR bits to address the control memory, drawing KC17, so that the next process word is extracted from location 65. Process word 65 contains SUB, ARO, AXS, ACI, DONE, CONT, and CMA10 (BGN).

Figure 5-9 illustrates the XOR logic for one bit position. SUB(1) takes the complement of the MB to the B bus and ARO(1) takes the direct contents of the AC to the A bus. Both A bus and B bus contents go to the ADR. If the output bus levels are both at ground or are both negative for any bit position, their corresponding MB and AC inputs differ, and the half-add result out of the ADR is at ground. In the ADR, CMPL complements the result, forcing the ADR bit output to go negative. CMPL is derived from AXS(1) of the process word and IR3(1) on drawing KC13. NOSH gates the ADR bits to the O bus, and ACI(1) gates them into the AC.

DONE(1) goes to the clock and run logic drawing KC10(1) for manual key control and KC17 for break and PI, while CONT(1) allows the generation of a third CM STROBE to extract the BGN process word from location 10. The BGN word sets up the MB for the next fetch cycle.

5.1.7 Operate (OPR) Instructions

OPR instructions (op codes 74, 75, 76) need no reference to an operand in core memory and are executed during the computer fetch cycle. These instructions are used to perform certain operations on the current contents of the AC and/or Link. The operations to be performed are encoded in bits 05-17 in the instruction word as described in Section 3.5.1.1.

At strobe SA time (see Figure 5-10), the op code is sampled by IRI(1) of the process word to produce REP, drawing KC12. REP allows the IR bits to address the control memory, drawing KC17, for the extraction of the third process word from location 77 on the third CM STROBE.

Simultaneously with op code sampling, IRI(1) samples bits SA00-02 of the instruction at the LOT (LAW, OPR, IOT) flip-flop, drawing KC12. Since the bits contain octal code 7, the LOT flip-flop sets.

LOT(1) further samples other bits at the OR MBO, OR ACI, etc., gates. If IR3-4 are 1s, LOT(1) and these bits indicate a LAW instruction (code 76), producing OR MBO. This level sets the MBO gate, drawing KC19(3), on the third CM STROBE (which extracts process word 77). MBO(1) gates the LAW instruction from the MB to the B bus, and the instruction goes

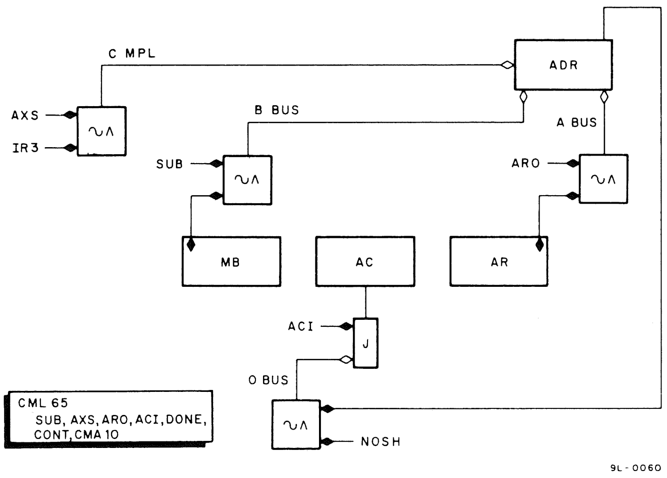
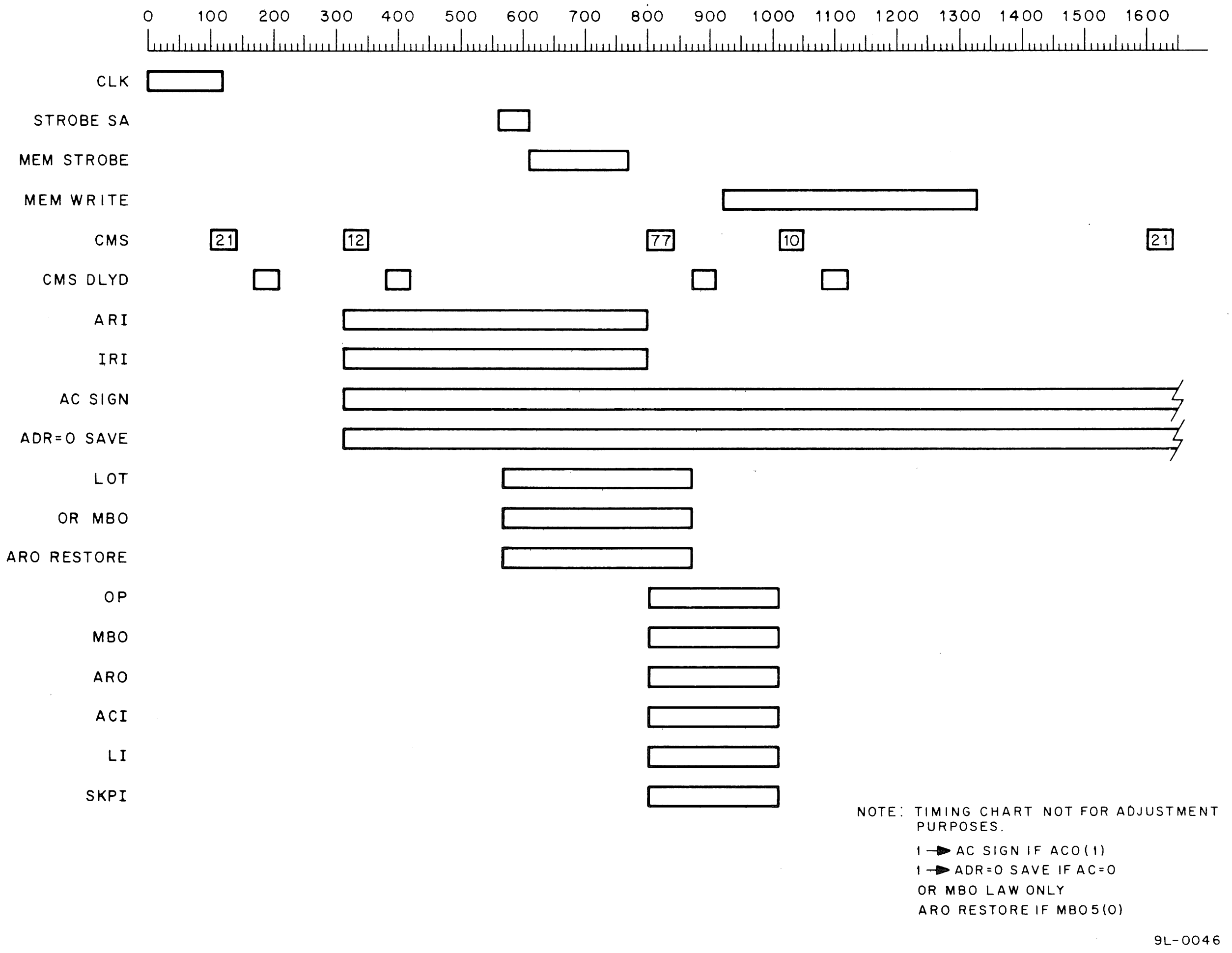


Figure 5-9 XOR Instruction Logic



NOTE: TIMING CHART NOT FOR ADJUSTMENT PURPOSES.

1 → AC SIGN IF ACO(1)
1 → ADR=O SAVE IF AC=O
OR MBO LAW ONLY
ARO RESTORE IF MBO5(O)

Figure 5-10 OPR Timing

into the ADR. NOSH places the ADR contents on the O bus. ACI(1) of process word 77 gates the instruction into the AC. With the LAW instruction, negative numbers can be loaded into the AC.

If IR3 is 1 and IR4-MB05 are 0s, LOT(1) and these bits indicate OPR instructions (code 74) other than LAW, producing ARO RESTORE. For OPR instructions, ARO RESTORE sets the ARO gate, drawing KC19(3), on the third CM STROBE (which extracts process word 77). ARO(1) gates the data in the AR onto the A bus and ADR. ACI(1) of process word 77 gates the data from the O bus into the AC, and LI(1) gates the content of the ADRL into the LAR. During these processes, other command bits in the OPR instruction operate on the data word as it passes through the ADR onto the O bus. The operation on the data word may also affect the Link content.

If MB05 is 1 at the ARO RESTORE gate, drawing KC12, it denotes a CLA instruction (code 75), inhibiting the gate. ACI(1) of process word 77 will transfer 0s to the AC from the O bus, since nothing appears on the bus in this case.

At the third CM STROBE, LOT(1), IR3(1), and IR4(0) also set the OP flip-flop, drawing KC12. OP(1) is the sampling gate which detects the command bits (MB06-17) in the OPR instructions, drawing KC13. Some of these command bits cause one-or two-place rotation of the contents of the AR and LAR, others are used for conditional skips in conjunction with SKPI(1) of process word 77, and still others for clearing, complementing, or setting the Link in conjunction with LI(1). Details are given in the instruction descriptions that follow.

The CM STROBE that extracts process word 77 returns to the CM timing chain to generate CM STROBE DLYD, after 80 ns CM STROBE DLYD resets LOT. The CONT(1) bit in the process word allows the timing chain to extract a fourth process word. The CMA in process word 77 is 10, from which the BGN word is extracted to enter the new core memory address (PC → MB) for the next fetch cycle. LI(1) of process word 77, going to 0 at BGN time, strobes the LAR status into the Link.

Some OPR instructions may be combined (microcoded) with others to perform two types of operations within one instruction period. Care must be taken in programming to avoid microcoding two conflicting operations. The more commonly used combinations of microcoded instructions are described below.

The foregoing OPR instruction descriptions serve as a foundation for the unique instruction particulars that follow. In most instances, the instruction execution starts with process word 77.

5.1.7.1 No Operation (NOP) - The NOP instruction (740000) is a "do nothing" instruction which delays the computer program for the duration of one cycle. At strobe time the NOP instruction is placed in the MB and the op-code portion in the IR. The op code is detected to extract process word 77 on the next CM STROBE. CM STROBE extracts the process word and sets the OP flip-flop, drawing KC12. Process word 77 merely recirculates the contents of the AR and ADRL into the AC and LAR, since OP(1) does not detect any operations to be performed on the contents in transit (MB05-17 are all 0s at the operate logic, drawing KC13).

5.1.7.2 Complement the Accumulator (CMA) - The CMA instruction (740001) complements each bit of the AC. The previous contents of the AC are lost and the Link remains the same. The op code is detected to extract process word 77 on the next CM STROBE. CM STROBE extracts the process word and sets the OP flip-flop, drawing KC12.

OP(1) ∧ MB17(1) generates CMPL, drawing KC13. ARO(1), derived at CM STROBE time from ARO RESTORE, on drawing KC12, gates the contents of the AR onto the A bus. The contents of the A bus go directly into the ADR, and NOSH places them on the O bus. As the contents pass through the ADR, CMPL complements all bits individually.

5.1.7.3 Complement The Link (CML) - The CML instruction (740002) complements the Link. The previous state of the Link is lost. The contents of the AC remain the same.

The CM STROBE extracts the OPR process word 77 and sets the OP flip-flop, drawing KC12. OP(1) and MB16(1) then enable CML to be generated (drawing KC13). CML, in conjunction with A BUS LINK, will either set or reset the LINK flip-flop.

If LINK is originally in a set condition, then A BUS LINK is at ground. Two grounds to an exclusive OR gate cause a grounded ADRL to be applied to the "normal" gate. The LAR is thus set. LAR(1) and LI(0) at CMA10 time resets the LINK. LINK(1) is thus complemented to LINK(0) during this process. By this same process a LINK(0) can be complemented to LINK(1).

5.1.7.4 Inclusive OR the AC/DATA Switches (OAS) - The OAS instruction (740004) inclusively ORs the contents of the AC with the manual settings of the DATA switches (switch levels DATA SW00-17) on a bit-for-bit basis. The results are left in the AC. The previous contents of the AC are lost and the Link remains the same. If corresponding bits are both 0s, the AC bit is set to 0. If corresponding bits differ or are both 1s, the AC bit is set to 1.

At strobe time the op code is detected to generate ARO RESTORE, drawing KC12, and to extract the OPR process word 77 on the next CM STROBE. The CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). OP(1) ∧ MB15(1) produces DASO and LIO on drawing KC13. DASO gates the DATA switch contents onto the I/O bus (B) via the CP/console interface and the input mixer, drawing KD7. The data on the I/O bus (B) is then gated onto the O bus by LIO, drawing KC20. Meanwhile, ARO(1) gates the contents of the AR onto the A bus, the contents on the A bus go into the ADR, and NOSH places them on the O bus. At each bit position, the O bus will go to ground where either or both I/O bus (B) and ADR bits are at 1 levels, or will go negative if both I/O bus (B) and ADR bits are at 0 levels. ACI(1) of the OPR process word gates the O bus results into the corresponding AC bit positions.

5.1.7.5 Rotate One Position Left (RAL) - The RAL instruction (740010) rotates the contents of the AC and the Link one bit position to the left. The Link enters AC17 and AC00 enters the Link.

At strobe time, the op code is detected to generate ARO RESTORE, drawing KC12, and to extract the OPR process word 77 on the next CM STROBE. The CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). OP(1) ∧ MB07(0) ∧ MB14(1) produces SHL1, drawing KC13. ARO(1) gates the contents of the AR onto the A bus. As the contents of the A bus pass into the ADR directly, SHL1 gates ADR bits XX onto O-bus bit positions XX+1, drawing KC20. This includes END BIT17 onto O bus 17 and ADR00 into the Link. END BIT17 is derived from ADRL(B) on drawing KC15. (When EAE is not used in the system, ADRL is connected to TEMP 3(1) B133-B03N.) The state of the ADRL represents the state of the Link. SHL1 gates the ADR00 bit to the jam input gate of the LAR. LI(1) of the OPR process word jam transfers the state of ADR00 into the LAR and ACI(1) transfers the shifted contents of the O bus into the AC. On drawing KC13, the positive IN SHL1 level inhibits the NOSH gate, so that only SHL1 controls the set enable input gating to the O bus. At BGN time LI(0) strobes the state of the LAR into the Link.

5.1.7.6 Rotate One Position Right (RAR) - The RAR instruction (740020) rotates the contents of the AC and the Link one position to the right. The Link enters AC00, and AC17 enters the Link.

At strobe time the op code is detected to generate ARO RESTORE, drawing KC12, and to extract the OPR process word 77 on the next CM STROBE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). OP(1) ∧ MB07(0) ∧ MB13(1) produces SHR1, drawing KC13. ARO(1) gates the contents of the AR onto the A bus. As the contents of the A bus pass into the ADR directly, SHR1 gates ADR bits XX onto O bus bit positions XX-1, drawing KC20. This includes END BIT00 onto O BUS 00. END BIT00 is derived from ADRL (B), drawing KC15. SHR1 also gates ADR17 into the LAR jam input gate. LI(1) of the OPR process word jam transfers the state of ADR17 into the LAR. ADR17 represents the state of AC17. ACI(1) of the OPR process word transfers the shifted contents of the O bus into the AC. On drawing KC13, the positive IN SHL1 level inhibits NOSH gate, so that only SHR1 controls the set enable input gating to the O bus. At BGN time LI(0) strobes the state of the LAR into the Link.

5.1.7.7 Halt Program (HLT) - The HLT instruction (740040) stops program execution. The op code is detected to generate ARO RESTORE, drawing KC12, and to extract the OPR process word 77 on the next CM STROBE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). OP(1) ∧ MB12(1) results in RUN(0) on drawing KC10. RUN(0) is applied to the collector at the set side of the RUN flip-flop, pulling it to ground (reset). Reset RUN inhibits CLK POS pulses, stopping the program.

5.1.7.8 Skip on Minus Accumulator (SMA) - The SMA instruction (740100) tests the sign (AC00) of a data word previously entered in the AC. If the sign is minus (AC00 = 1) the computer skips the next instruction. If the sign is plus (AC00 = 0) the computer executes the next instruction. The contents of the AC and Link remain unchanged.

During process word 12 the contents of the AC pass through the ADR, ARI(1) of the process word samples the ADR00 bit at the AC SIGN flip-flop, drawing KC14. ADR00 represents the sign bit, AC00. If ADR00 = 1, ARI(1) sets the AC SIGN flip-flop.

The op code 74 qualifies REP and allows IR to set up address 77. MB05(0) gives ARO RESTORE (see Section 5.1.8). CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). AC SIGN(1) ∧ MB11(1), on drawing KC14, applies a ground level to one of three B105 Inverters connected as a positive NAND gate. The other inverters receive ground OP(1) and MB08(0) levels.

This places a ground set level at the jam input gate to the SKIP flip-flop. SKPI(1) of the OPR process word sets the flip-flop in conjunction with the ground level. (The flip-flop was previously reset by PCI(1) during the fetch-entry process word 21.)

ACI(1) of the OPR process word and ARO(1) recirculate the contents of the AR into the AC. CONT(1) allows the next process word to be extracted from control memory (BGN, address 10). With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17, drawing KC14, so that the current address in the PC is incremented by 1 as it goes through the ADR to the O bus and MB. LI(0) recirculates the content of the LAR into the Link.

5.1.7.9 Skip on Zero Accumulator (SZA) - The SZA instruction (740200) tests the value of a data word previously entered in the AC. If the value in the AC is 0, the computer skips the next instruction. If the value is other than 0, the computer executes the next instruction. The contents of the AC and Link remain unchanged.

As the contents of the AC pass through the ADR during process word 12, an output bus on the ADR (ADRA = 0, ADRB = 0) goes negative if the ADR goes to all 0s. The negative ADR = 0 levels are applied to the jam input gate to the ADR = 0 SAVE flip-flop, drawing KC15. ARI(1) of process word 12 sets the flip-flop in conjunction with the negative levels.

The op code 74 qualifies REP and allows the IR to set up address 77. MB05(0) gives ARO RESTORE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). ADR = 0 SAVE(1) ∧ MB10(1), on drawing KC14, applies a ground level to one of three B105 inverters connected as a positive NAND gate. The other inverters receive ground OP(1) and MB08(0) levels. This places a ground set level at the input to the SKIP flip-flop. (The flip-flop was previously reset by PCI(1) of the fetch entry process word 21.) SKPI(1) of the OPR process word sets the flip-flop.

ACI(1) of the OPR process word and ARO(1) recirculate the contents of the AR to the AC. CONT(1) allows the next process word to be extracted from control memory (BGN, address 10). With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17, drawing KC14 so that the current address in the PC is incremented by 1 as it passes through the ADR to the O bus and MB. LI(0) recirculates the content of the LAR into the Link.

5.1.7.10 Skip on Non-Zero Link (SNL) - The SNL instruction (740400) tests the status of the Link; if set, the computer skips the next instruction; if reset, the computer executes the next instruction. The contents of the AC and the Link remain unchanged.

The op code 74 qualifies REP and allows the IR to set up address 77. MB05(0) gives ARO RESTORE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3).

If the Link is set, LINK(1) ∧ MB09(1) on drawing KC14 applies a ground level to one of three B105 inverters connected as a positive NAND gate. The other inverters receive OP(1) and MB08(0) levels. This places a ground set level at the input of the SKIP flip-flop. (The flip-flop was previously reset by PCI(1) of the fetch entry process word 21.)

SKPI(1) of the OPR process word sets the SKIP flip-flop. CONT(1) allows the next process word to be extracted from control memory (BGN, address 10). With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17, drawing KC14, so that the current address in the PC is incremented by 1 as it passes through the ADR to the O bus and MB. LI(0) recirculates the content of the LAR into the Link.

5.1.7.11 Unconditional Skip (SKP) - The SKP instruction (741000) causes the computer to skip the next instruction.

The op code 74 qualifies REP and allows the IR to set up address 77. MB05(0) gives ARO RESTORE. CM STROBE extracts the process word, sets the OP flip-flop (drawing KC12) and the ARO flip-flop in conjunction with ARO RESTORE (drawing KC19(3)). On drawing KC14, OP(0), MB09(1), MB11(1), and MB10(1) are all at ground, disabling their respective parallel NAND gates in B169-D38. This results in a negative level at D38E, which is in turn NANDed with MB08(1) at D38D. The result is a ground set level to the input gate of the SKIP flip-flop. SKPI(1) of the OPR process word sets the flip-flop.

CONT(1) allows the next process word to be extracted from control memory (BGN, address 10). With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17, drawing KC14, so that the current address in the PC is incremented by 1 as it passes through the ADR to the O bus and MB. LI(0) recirculates the content of the LAR into the Link.

5.1.7.12 Skip on Positive Accumulator (SPA) - The SPA instruction (741100) tests the sign (AC00) of a data word previously entered in the AC; if the sign is plus (AC00 = 0), the computer skips the next instruction; if the sign is minus (AC00 = 1), the computer executes the next instruction. The contents of the AC and Link remain unchanged.

During process word 12 the SPA instruction is placed in the MB, the op-code portion in the IR, and the contents of the AC in the AR. As the contents of the AC pass through the ADR, ARI(1) of the process word samplès ADR00 at the AC SIGN flip-flop, drawing KC15. ADR00 represents the sign bit, AC00. If ADR00 = 0, ARI(1) resets the AC SIGN flip-flop. AC SIGN(1) is at ground, therefore, at the respective NAND gate in B169-D38, drawing KC14.

The op code 74 qualifies REP and allows the IR to set up address 77. MB05(0) gives ARO RESTORE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). With none of the AND gates of the left section of NAND gate B169-D38 enabled, its output at pin D becomes a negative level. The negative output is further NAND gated with MB08(1) at B169-D38D, resulting in a ground set level to the input of the SKIP flip-flop.

SKPI(1) of the OPR process word sets the SKIP flip-flop. (The flip-flop was previously reset by PCI(1) of the fetch entry process word 21.) ACI(1) of the OPR process word and ARO(1) recirculate the contents of the AR into the AC. CONT(1) allows the next process word to be extracted from control memory (BGN, address 10). With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17, drawing KC14, so that the current address in the PC is incremented by 1 as it passes through the ADR to the O bus and MB. LI(0) recirculates the content of the LAR into the Link.

5.1.7.13 Skip on Non-Zero Accumulator (SNA) - The SNA instruction (741200) tests the value of a data word previously entered in the AC. If the value of the data word is other than 0, the computer skips the next instruction. If the value is 0, the computer executes the next instruction. The contents of the AC and Link remain unchanged.

During process word 12 the SNA contents of the AC pass through the ADR, and ADRA and ADRB goes negative only if the ADR contains all 0s. If any bit contains a 1, the bus levels go to ground. The ground ADRA = 0, ADRB = 0 level is applied to the jam input gate to the ADR = 0 SAVE flip-flop, drawing KC14. ARI(1) of process word 12 resets the flip-flop in conjunction with the ground level(s).

The op code 74 qualifies REP and allows the IR to set up address 77. MB05(0) gives ARO RESTORE. CM STROBE extracts the process word, sets the OP flip-flop drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). With none of the AND gates of the left section of NAND gate B169-D38 enabled, its output at pin E becomes a negative level. The negative output level is further gated with MB08(1) at D38D, resulting in a ground set level to the input of the SKIP flip-flop.

SKPI(1) of the OPR process word sets the SKIP flip-flop. (The flip-flop was reset by PCI(1) of the fetch entry process word 21.) ACI(1) of the OPR process word and ARO(1) merely recirculate the contents of the AR into the AC. CONT(1) allows the next process word to be extracted from control memory (BGN, address 10). With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17 on drawing KC14, so that the current address in the PC is incremented by 1 as it passes through the ADR to the O bus and the MB. LI(0) recirculates the LAR content into the Link.

5.1.7.14 Skip on Zero Link (SZL) - The SZL instruction (741400) tests the status of the Link. If the Link is reset, the computer skips the next instruction. If the Link is set, the computer executes the next instruction. The status of the Link and the contents of the AC remain unchanged.

Following REP and ARO RESTORE, the CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). The output of NAND gate B169-D38 E is negative as was the condition for the SPA and SNA instructions. The negative output level is further gated with MB08(1) to apply a ground set level to the input gate of the SKIP flip-flop. SKPI(1) of the OPR process word sets the flip-flop.

ACI(1) of the OPR process word and ARO(1) merely recirculate the contents of the AR into the AC. CONT(1) allows the next process word to be extracted from control memory (BGN, address 10). With the SKIP flip-flop set, PCO(1) of the BGN word produces CI17, drawing KC14, so that the current address in the PC is incremented by 1 as it passes through the ADR to the O bus and MB. LI(0) recirculates the LAR content into the Link.

5.1.7.15 Rotate Two Positions Left (RTL) - The RTL instruction (742010) rotates the contents of the AC and the Link two positions to the left. AC00 enters AC17 and AC01 enters the Link.

At strobe time, the op code is detected to generate ARO RESTORE, drawing KC12, and to extract the OPR process word 77 on the next CM STROBE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). OP(1) ∧ MB07(1) ∧ MB14(1) produces SHL2, drawing KC13. ARO(1) gates the contents of the AR onto the A bus.

As the contents of the A bus pass into the ADR directly, SHL2 gates ADR bits XX into O bus positions XX+2, drawing KC22. This includes ADRL onto O BUS 16 and ADR00 onto O BUS 17. ADRL represents the state of the Link, and ADR00 represents AC00. On drawing KC15, SHL2 gates ADR01 to the jam input gate of the LAR, and also generates the SHIFT level. The SHIFT level at ground makes $\overline{\text{SHIFT}}$ go to ground at the "normal" gate, so that only SHL2 ∧ ADR01 controls a set enable input to the LAR. Also, the ground $\overline{\text{SHIFT}}$ level on drawing KC13 inhibits the NOSH gate, so that only SHL2 controls the set enable input gating to the O bus.

LI(1) of the OPR process word jam transfers the state of ADR01 into the LAR, and ACI(1) transfers the shifted contents of the O bus into the AC. At BGN time LI(0) strobes the state of the LAR into the Link.

5.1.7.16 Rotate Two Positions Right (RTR) - The RTR instruction (742020) rotates the contents of the AC and the Link two positions to the right. AC16 enters the Link, and the Link enters AC01.

At strobe time, the op code is detected to generate ARO RESTORE, drawing KC12, and to extract the OPR process word 77 on the next CM STROBE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). OP(1) ∧ MB07(1) ∧ MB13(1) produces SHR2, drawing KC13. ARO(1) gates the contents of the AR onto the A bus. As the contents of the A bus pass into the ADR directly, SHR2 gates ADR bits XX onto O bus positions XX-2, drawing KC22. This includes ADR17 onto O BUS 00 and ADRL into O BUS 01. ADR17 represents AC17 and ADRL represents the state of the Link. On drawing KC15, SHR2 gates ADR16 to the jam input gate of the LAR, and also generates SHIFT. The SHIFT level at ground makes $\overline{\text{SHIFT}}$ go to ground at the "normal" gate, so that only SHR2 ∧ ADR16 controls a set enable input to the LAR. Also, the ground $\overline{\text{SHIFT}}$ level on drawing KC13 inhibits the NOSH gate, so that only SHR2 controls the set enable input gating to the O bus.

LI(1) of the OPR process word jam transfers the state of ADR16 into the LAR, and ACI(1) transfers the shifted contents of the O bus into the AC. At BGN time LI(0) strobes the LAR content into the Link.

5.1.7.17 Clear the Link (CLL) - The CLL instruction (744000) clears the Link. The AC contents remain the same. At strobe time the op code is detected to generate ARO RESTORE, drawing KC12 and to extract the OPR process word 77 on the next CM STROBE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC12, and the ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). OP(1) ∧ MB06(1) produces CLL, drawing KC13. ARO(1) transfers the contents of the AR to the A bus, the contents on the A bus go through the ADR directly, and NOSH places them on the O bus. The CLL level makes $\overline{\text{CLL}}$ go to ground at the ADRL input gate, drawing KC15, so that the ADRL goes to ground. ADRL at ground disables the "normal" gate to the LAR, and LI(1) of the OPR process word thus resets the LAR.

ACI(1) of the OPR process word recirculates the contents of the O bus into the AC. LI(0) at BGN resets the Link in conjunction with the reset LAR.

5.1.7.18 Set the Link (STL) - The STL instruction (744002) is a combined CLL and CML instruction where the presence of a ground $\overline{\text{CLL}}$ level at the ADRL input gate attempts to make the ADRL go to ground, but CML forces it to go negative, thus enabling the "normal" gate to the LAR.

5.1.7.19 Clear the Link and Rotate One Position Left (RCL) - The RCL instruction (744010) is a combined CLL and RAL instruction where the presence of a ground $\overline{\text{CLL}}$ level at the ADRL input gate makes the ADRL go to ground, thus disabling the "normal" gate to the LAR; however, the presence of SHL1 can enable the "shifting" gate input to the LAR if ADR00 is 1, thereby setting the LAR on LI(1). ACI(1) of the OPR process word 77 transfers the shifted contents on the O bus to the AC.

5.1.7.20 Clear the Link and Rotate One Position Right (RCR) - The RCR instruction (744020) is a combined CLL and RAR instruction where the presence of a ground CLL level at the ADRL input gate makes the ADRL go to ground, thus disabling the "normal" gate to the LAR; however, the presence of SHR1 can enable the "shifting" gate to the LAR, if ADR17 is a 1, thereby setting the LAR on LI(1). ACI(1) of the OPR process word transfers the shifted contents of the O bus into the AC.

5.1.7.21 Clear the Accumulator (CLA) - The CLA instruction (750000) resets all bits of the AC. The previous contents of the AC are lost and the Link remains the same.

At strobe time, the op code is detected to extract the OPR process word 77 on the next CM STROBE. CM STROBE extracts the process word and sets the OP flip-flop, drawing KC12. MB05(1) at the ARO RESTORE gate, drawing KC12, inhibits the ARO RESTORE level so that the CM STROBE cannot set the ARO flip-flop on drawing KC19(3). Therefore, ACI(1) transfers all 0s to the AC from the O bus because the contents of the AR cannot get to the bus. LI(1) transfers the state of the ADRL to the LAR and LI(0) at BGN strobes the LAR content into the Link.

5.1.7.22 Clear and Complement the Accumulator (CLC) - The CLC instruction (750001) is a combined CLA and CMA instruction where MB05(1) of CLA inhibits ARO RESTORE on drawing KC12 and consequently ARO(1) on drawing KC19(3). Ordinarily, ARO(1) takes the contents of the AR to the O bus via the ADR, and ACI(1) places them in the AC. Since the ADR is closed to the AR, CMPL of the CMA process word 77 complements the ADR to all 1s, and ACI(1) places all 1s in the AC.

5.1.7.23 Load AC from DATA Switches (LAS) - The LAS instruction (750004) is a combined CLA and OAS instruction where the inhibited ARO RESTORE and consequently inhibited ARO(1) levels prevent the AR contents from reaching the O bus. Therefore, DASO and LIO produced by the OAS instruction, drawing KC13, inclusively ORs the data in the DATA switches with 0s at the O bus. This in effect provides a direct transfer of data from the switches. ACI(1) gates the O bus data into the AC.

DASO is applied to the input mixer, drawing KD7, via the CP-I/O cable interface, drawing KC25. DASO and the data switches are ANDed in the input mixer and applied to the I/O bus(B). LIO and I/O bus are then ANDed and applied to the O bus.

5.1.7.24 Get the Link Content (GLK) - The GLK instruction (750010) is a combined CLA and RAL instruction where the inhibited ARO RESTORE and consequently inhibited ARO(1) levels prevent the AR contents from reaching the O bus, and SHL1 of the RAL instruction transfers the content of the Link into AC17 via the ADRL and END BIT17. LI(1) of the OPR process word transfers a 0 into the LAR from the cleared ADR00 and ACI(1) transfers all 0s into the AC, except when the Link going into AC17 is a 1. LI(0) then transfers a 0 from the LAR to the Link at BGN time. (See Sections 5.1.7.5 and 5.1.7.21.)

5.1.7.25 Load the Accumulator with 76XXXX (LAW)- The LAW instruction (76XXXX) loads itself into the accumulator, e.g., 76 + n, where n is any number from 0000 → 07777. The previous contents of the AC are lost and the Link remains the same.

At strobe time the op code is detected to generate OR MBO, drawing KC12, and to extract the OPR process word 77 on the next CM STROBE. CM STROBE extracts the process word, sets the OP flip-flop, drawing KC13 while OR MBO sets the MBO flip-flop, drawing KC19(3). MBO(1) gates the entire LAW instruction from the MB to the B bus and ACI(1) of the OPR process word gates it into the AC. LI(1) gates the content of the ADRL into the LAR. LI(0) at BGN strobes the LAR content into the Link.

5.1.8 Input/Output Transfer (IOT) Instructions

IOT instructions (op code 70) need no reference to an operand in core memory and are executed during an extended (6.0 µs) fetch cycle. Bits 6-11 and 12-13 in the instruction select the I/O device and device operating mode for the program-controlled data transfer. The remaining bits (14-17) command the IOT operations.

During fetch, STROBE 0-8, 9-17 in core memory read out the IOT instruction to the CP/memory interface. The strobe occurs at 500 ns, or ≈ 200 ns after the second process word (12) is extracted from control memory, Figure 5-11. IRI(1), SAO(1), and MBI(1) are present to place the instruction in the MB and the op-code portion in the IR. Simultaneously, IRI(1) samples bits SA00-02 of the instruction at the LOT (LAW, OPR, IOT) flip-flop, drawing KC12. If the bits contain octal code 7, the LOT flip-flop sets. IRI(1) also samples the same bits in the IR to produce REP.

MEM STROBE occurs during STROBE 0-8, 9-17 to trigger the CM timing chain for the third CM STROBE. REP allows the IR bits to address control memory, drawing KC17, so that the third CM STROBE extracts the IOT execute word, location 76. The strobe also samples LOT and IR03 at the IOT flip-flop, drawing KC12. LOT(1) and IR03(0) indicate that the instruction is in fact an IOT (op code 70) instruction, setting the IOT flip-flop.

LOT(1) and IR03(0) further check the state of the MB14 bit on drawing KC12. MB14(1) is usually programmed for an input data transfer, consequently generating the OR ACI level at NAND gate R111-E10H. The OR ACI level causes the ACI flip-flop on drawing KC19(2) to set on the third CM STROBE. ACI(1) opens the AC to the O bus. Since the O bus contains nothing at this time, ACI(1) effectively clears the AC by filling it with 0s. The AC is now ready to accept data from the selected device during the IOT execute period.

MB14(0) is programmed for an output data transfer, generating IOT OR ARO at NAND gate R111-F10U, drawing KC12. IOT OR ARO sets the ARO and the I/O BUS ON flip-flops on the third CM STROBE, drawing KC19(3). ARO(1) gates the data from the AR onto the A bus. (The data was gated into the AR from the AC by process word 12.) The contents go directly from the A bus to the ADR, then I/O BUS ON puts them on the I/O bus. Thus, the data is ready for output transfer to the selected device during the IOT execute period.

The third CM STROBE that extracts the IOT execute word goes back to the CM timing chain to generate CM STROBE DLYD after 80 ns. CM STROBE DLYD resets LOT.

The IOT execute word 76 contains CMA20. In the absence of CONT(1) and SM(1) bits, the execute word suspends core memory cycles and control memory processes for the duration of three CLK periods. During this time, the IOT(1) level from the IOT flip-flop and the IOT instruction word bits MB06-17 are available directly to the I/O control logic, drawing KD3, via the CP-I/O interface. Here IOT(1) generates IOT(B) on KD3(1) which gates the device select bits MB06-11 and subdevice 12 and 13 into bus drivers B213, from which they emerge as device select levels DS00-05 and SD00-01. These device and subdevice select levels are sampled within the I/O control logic to enable operation of the standard I/O devices and facilities; they also go directly to the I/O bus for sampling by the optional I/O devices. In both instances, the sampling takes place on the occurrence of IOP pulses derived from bits MB15-17 and on a gray code CLK pulse count (IO0 and IO1) in the I/O control.

IOP pulses are timed with the CLK and CLK POS pulses sent from the clock and run logic, drawing KC10, to the I/O control via the CP-I/O interface. In Figure 5-11, the first CLK pulse in the fetch cycle starts the core memory read/write cycle and the control memory timing chain. In the ensuing 1.5 µs interval the core memory cycle takes place and the control memory sequentially extracts the three fetch cycle process words.

On drawing KD3(3), the first IO CLK POS pulse also resets the pulse counter IO0, IO1 to 00. The second IO CLK POS (IO CLK (B)) pulse steps the counter to 10 and this count generates IOP1P if MB17 = 1. The third CLK pulse steps the counter to 11, generating IOP2P if MB16 = 1. The fourth CLK pulse steps the counter to 01, generating IOP4P if MB15 = 1. The IOP (P) pulses in turn set their respective IOP flip-flops which command the selected I/O devices for the duration of their execute periods.

Further, each IO CLK POS pulse is applied to a 400 ns delay on KD3(3) for a DLY pulse out. This DLY pulse samples the state of the pulse counter; thus an IO RESTART pulse is gated on 400 ns after the count reaches 01. IO RESTART is a 100-ns positive pulse which goes to the CM timing chain, drawing KC16, via the CP-I/O interface. This pulse restarts the CM timing.

At the same time the end of DLY resets the IOP4 flip-flop.

The ensuing CM STROBE extracts the process word in location 20 and resets the IOT flip-flop on drawing KC12 (LOT(1) is now at ground). Process word 20 contains DONE, CONT, and CMA10 (BGN). DONE(1) goes to the clock and run logic, drawing KC10, for manual key control, and the CONT(1) bit permits the CM STROBE to restart the CM timing. The next word thus extracted is the BGN word (10) which starts the next fetch cycle on CLK and gates the new address from the PC to the MB.

For more details on the IOT instructions and the devices which they control, refer to I/O control, Chapter 8 and any supplied I/O option manuals.

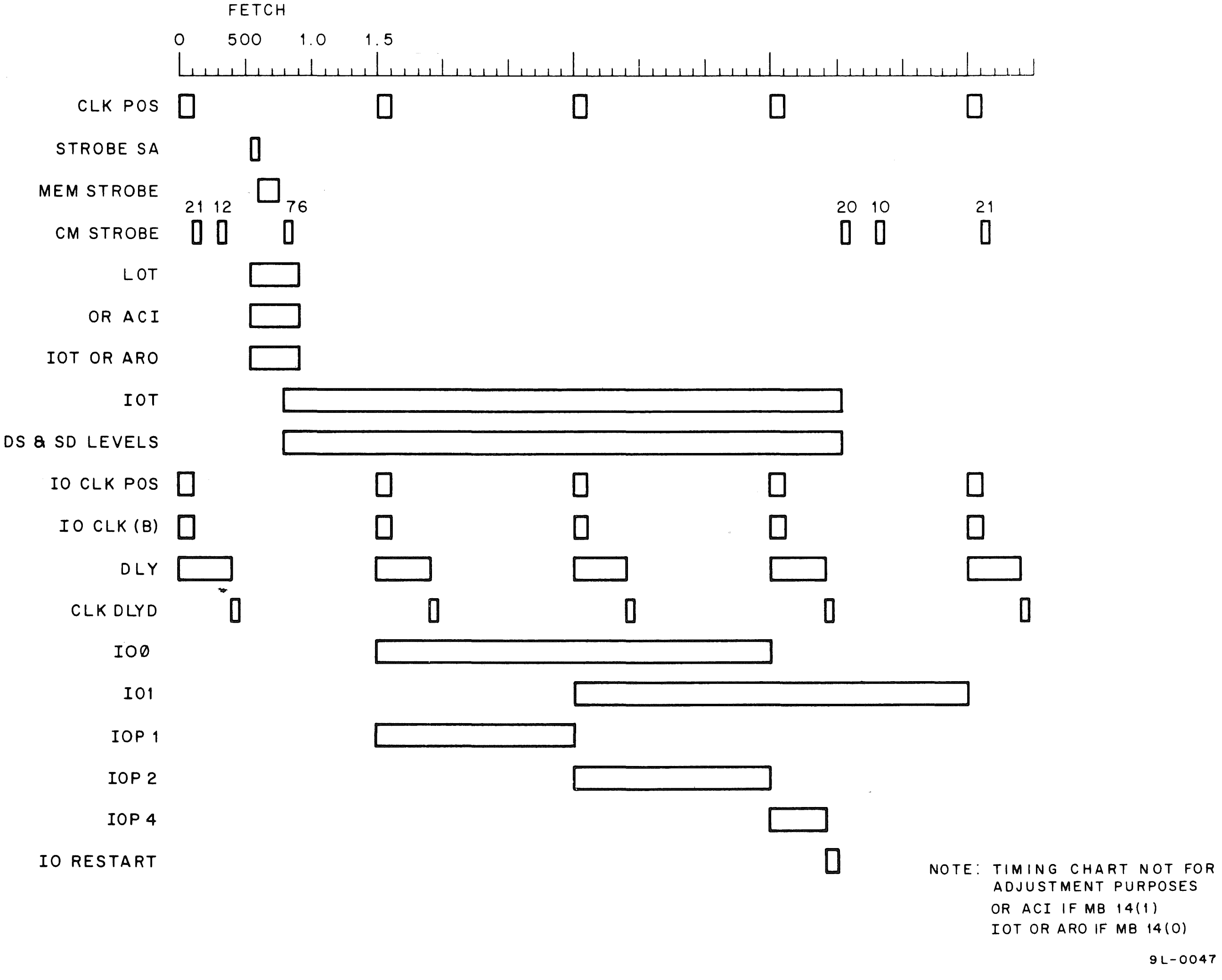


Figure 5-11 IOT Timing

CHAPTER 6
CORE MEMORY SYSTEM

Any core memory, such as the one used in the PDP-9/L, consists of planes of magnetic cores which are arranged in patterns of rows and columns. A set of planes comprises a field (see Figure 6-1).

In order to store the binary states of the bits in any word, the bit positions of that word are assigned to individual planes (plane 0 for bit 0, plane 1 for bit 1, etc.); therefore, any core memory must have as many planes as there are bits in each word, and the number of cores in any one plane then determines the number of words that can be stored in the field, referred to as "word position" in Figure 6-1.

By using binary decoding of both coordinates, an optimum number of word bits can be used for addressing. By converting the addressing bits in each word to coordinate decoding, the word may be stored at a collective address which has been derived by the selected bits in that word.

6.1 ORGANIZATION

The basic PDP-9/L is equipped with a DEC type MC71A Memory System having a 3D organization. The basic system can store 4,096 18-bit words, using a 13-bit address register. Memory capacity can be extended in 4,096-word increments up to a maximum of 32,768 words. Expansion beyond 8,192 words (two 4K fields) requires implementation of the optional Memory Extension Control (DEC Type KG09C). The following discussion covers the operation of an 8K system but is equally applicable to either the basic 4K or extended systems. For information on extended memory, refer to the option manual.

The basic core memory contains 18 planes which, for the purposes of this discussion, are considered to be stacked as illustrated in Figure 6-2. Each of these planes contains 4,096 cores which are arranged in a square pattern with 64 cores on each side. As such this 4K memory can store, in binary electromagnetic form, 4,096 18-bit words by selectively changing the

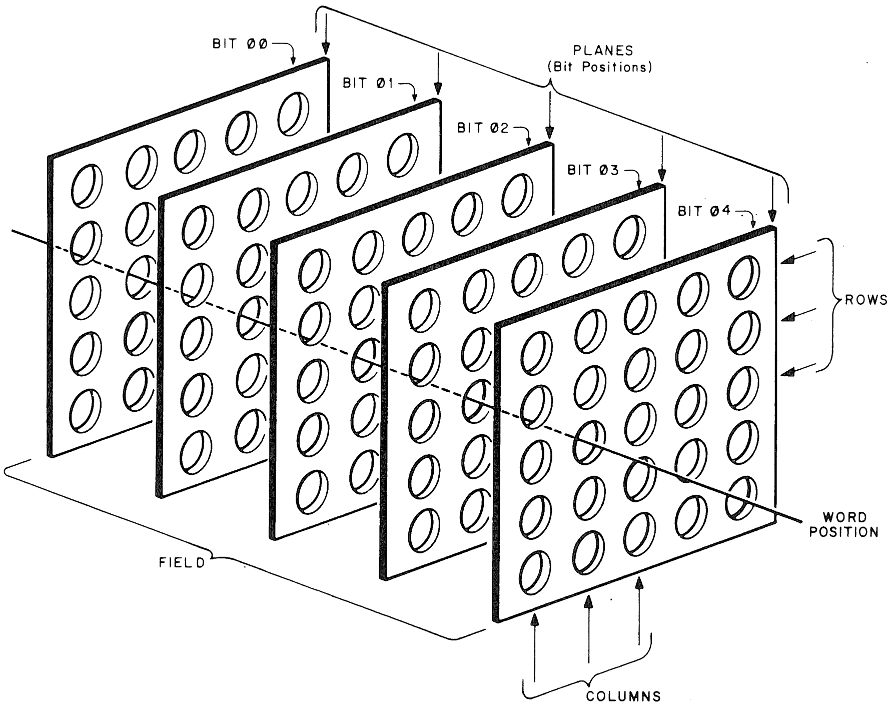


Figure 6-1 Typical Core Memory

magnetic states of 73,728 cores. In the PDP-9/L, this basic 4K memory is referred to as Field 0. A second stack of planes, when optioned, is similarly arranged and designated Field 1. Two 4K stacks comprise a memory bank.

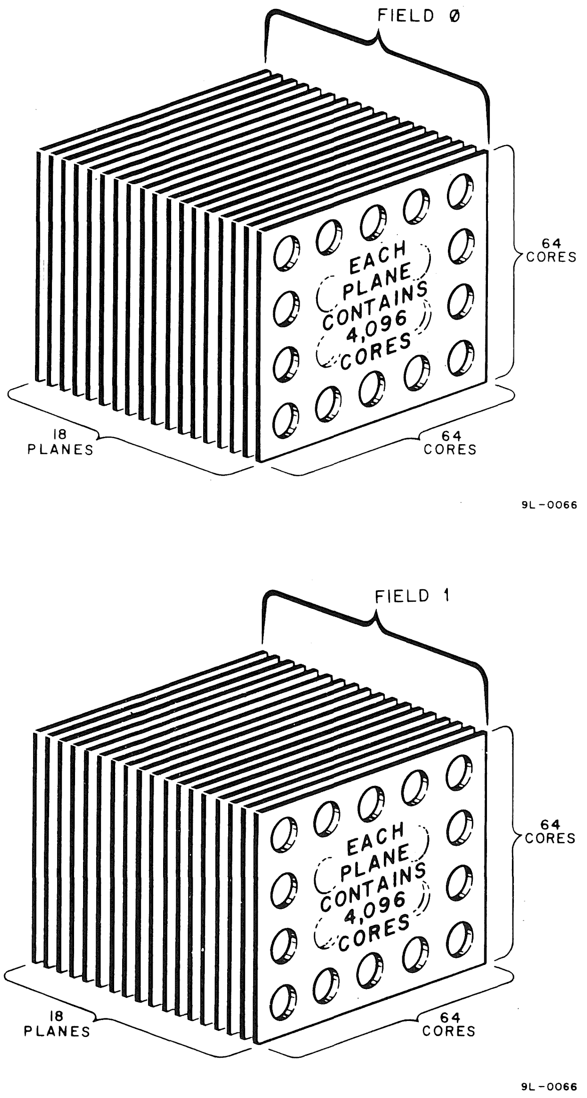


Figure 6-2 Core Memory Stacking Arrangement (Theoretical)

Figure 6-3 shows the elements of the basic MC71A Core Memory System when augmented by an additional 4K of memory. The memory control provides the timing and initiation of the read, inhibit, write, sense, and jam functions. Memory address selection combinations are generated in the MA flip-flops as are the field select signals. Actual address selection is accomplished by applying these MAs to the address decoding and selection matrices. The voltage regulator and X-Y drive circuit provides the inhibiting voltages and source currents used in reading and writing information into and out of the memory. The sense amplifiers and drivers provide the signal interfacing with the central processor, allowing signals, destroyed by reading, to be rewritten into memory.

In the PDP-9/L, address selection is accomplished by the interaction of core-location decoding and bit-sensitive inhibiting. As shown in Figure 6-4, the selection matrices are arranged to accommodate the selective energizing of a single row of cores in the X-axis, and of a single column of cores in the Y-axis. Since the selected X and Y lines each apply half current to a particular row and column of cores; one, and only one core in that plane is energized by full current. Each of the 64 lines for the X coordinate and each of the 64 lines for the Y coordinate continues, in like manner, through all 18 bit planes in sequence before returning to its matrix (see Figure 6-5); therefore when power is applied, the same coincident core in all 18 planes is affected. Thus, the core selection matrices determine the specific set of 18 cores (one for each plane) in which information is to be stored.

To effect this selection, the memory bank utilizes the last 12 bits in any 18-bit word, since combinations of the binary states of 12 bits is all that is required to give two-dimensional selection of a 64 by 64 plane.

Figure 6-6 is a representation of a portion of a core plane. The X- and Y-selection lines can be seen intersecting at each core. When X- and Y-selection current flows in the read direction, the selected core is magnetized in that direction; and when current is reversed (write), the core is magnetized in the opposite direction. Each time a core changes state, a magnetic field is generated by that change.

In the PDP-9/L, the cores are read to a binary 0 state and written to a binary 1 state. If the selected core is found to be in a 0 state during reading, no change of state occurs and no magnetic field is generated. If the core is found to be in a 1 state, the resultant change in state induces a signal in the sense windings (shown in Figure 6-6 as interlacing every core) which is sent to the memory buffer so that the 1 can be regenerated during the write cycle.

Writing into memory consists of duplicating the information contained in the memory buffer. This infor-

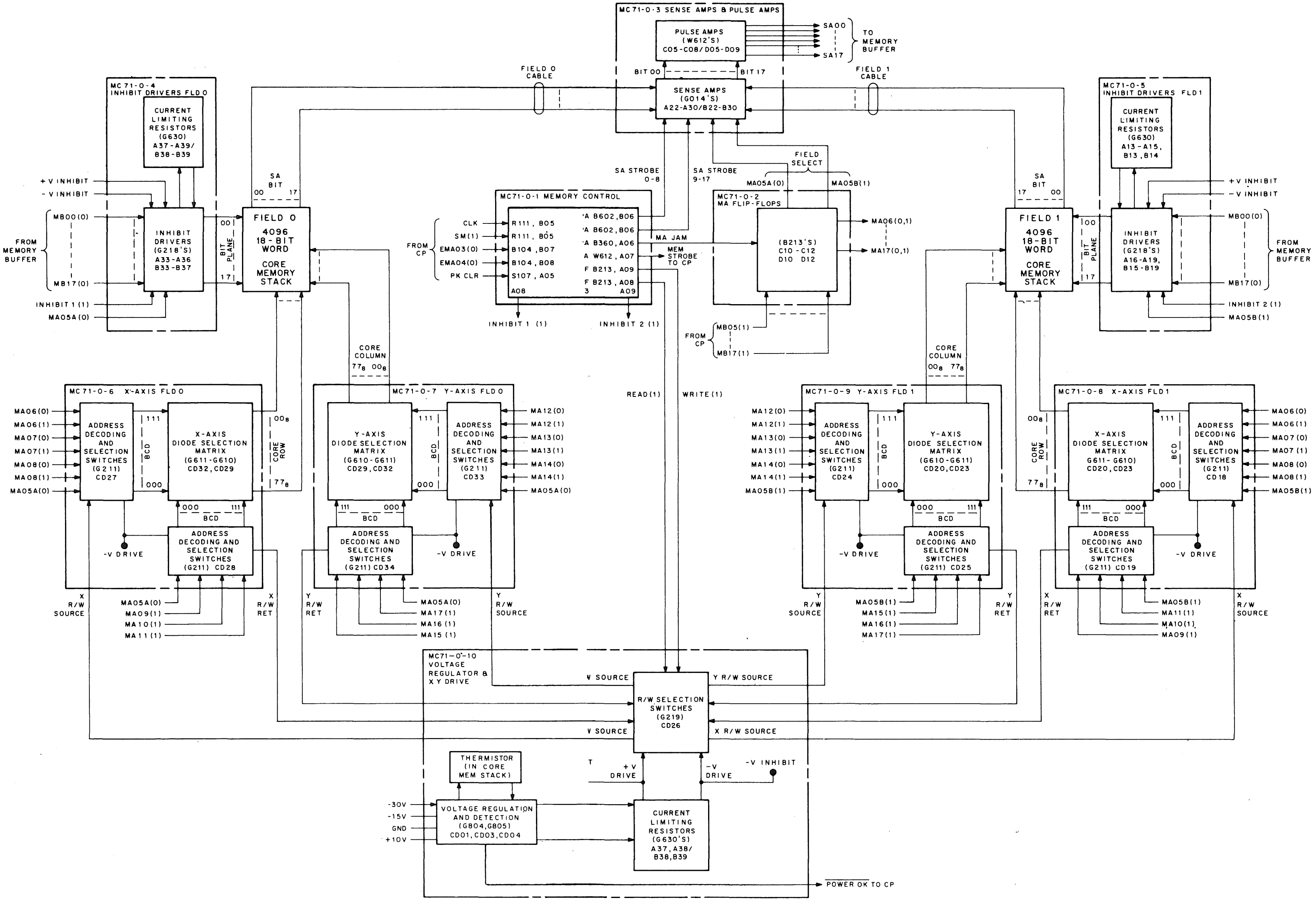


Figure 6-3 MC71A Core Memory System Block Diagram

mation can be that which was placed there as 1s during read, or information placed there from outside the memory. As shown in Figure 6-6, the inhibit windings are systematically strung through each core in the plane. Polarity of current in this winding is never reversed and is oriented toward binary 0. The value of this current is sufficient to offset the current in the Y line. In this way, the writing of a 1 into core is inhibited. The inhibit drivers operate during the write-half-cycle and are controlled by the states of bits in the memory buffer. A 0 MB bit will permit the drivers to inhibit the writing of a 1. A 1 MB bit will disable the drivers allowing the normal write selection current to restore a 1 to that core.

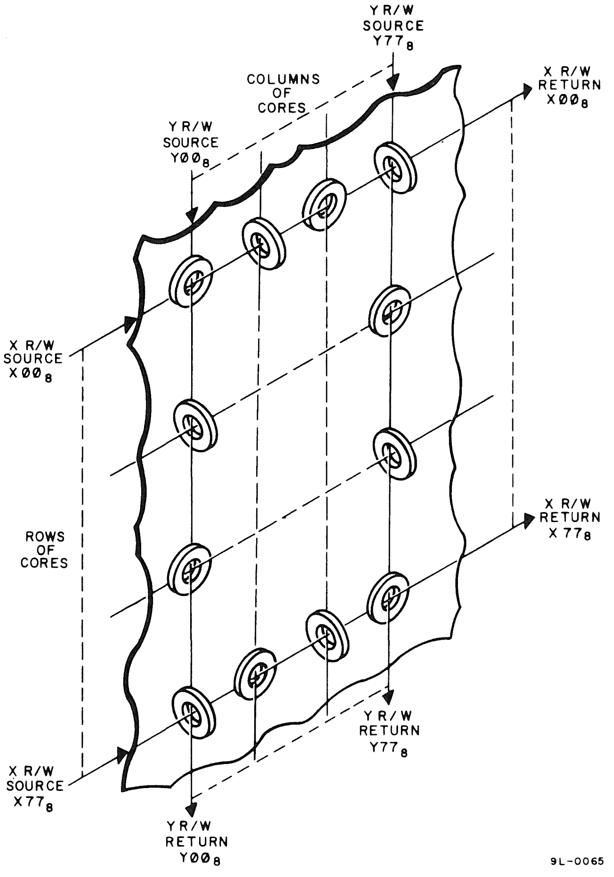


Figure 6-4 Core Row and Core Column Selection

Although this discussion has been based upon the premise that all core planes are stacked one above another, in reality they are placed side-by-side upon several boards. A set of boards then constitutes a "stack." This arrangement of planes, for the convenience of manufacture, does not, however, affect the operation of the memory as described above.

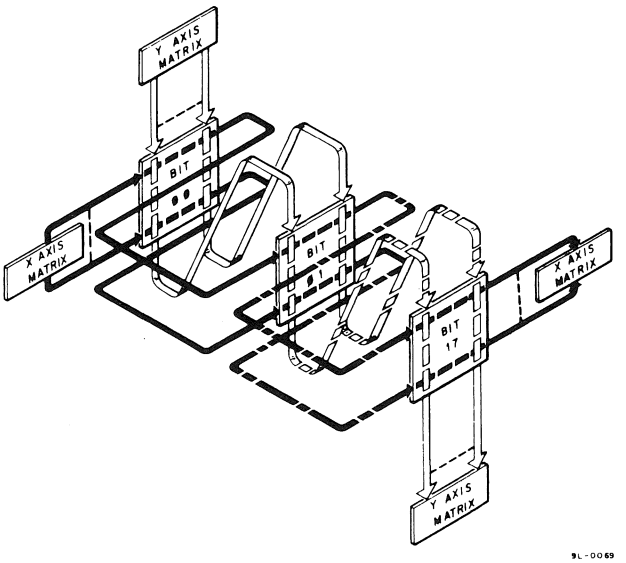


Figure 6-5 X and Y-Axis Selection Scheme for 18 Planes

6.2 DETAILED CIRCUIT ANALYSIS

6.2.1 Memory Control and Timing

The memory control circuits are shown in drawing MC71-0-1. The interrelation of control waveforms are given in drawing MC71-0-13. In this discussion, all intermediate delays are approximations, since they are adjustable to achieve the overall input/output timing precision.

All memory operations are initialized by the generation of a MEM START signal within the memory control (see Figure 6-7). This signal is initiated upon receipt from the CP of a CLK pulse at the K input of B05, provided the SM(1) level is present at pin E. The SM(1) level is also generated in the CP. As shown in drawing MC71-0-13, this begins at time 0 and ends at +110 ns.

MEM START, a 120 ns pulse, applied to delay line W301 at AB01, results in MA JAM (A06N), a 50 ns pulse occurring at +150 ns, which is applied simultaneously to all jam inputs of the MA flip-flops (draw-

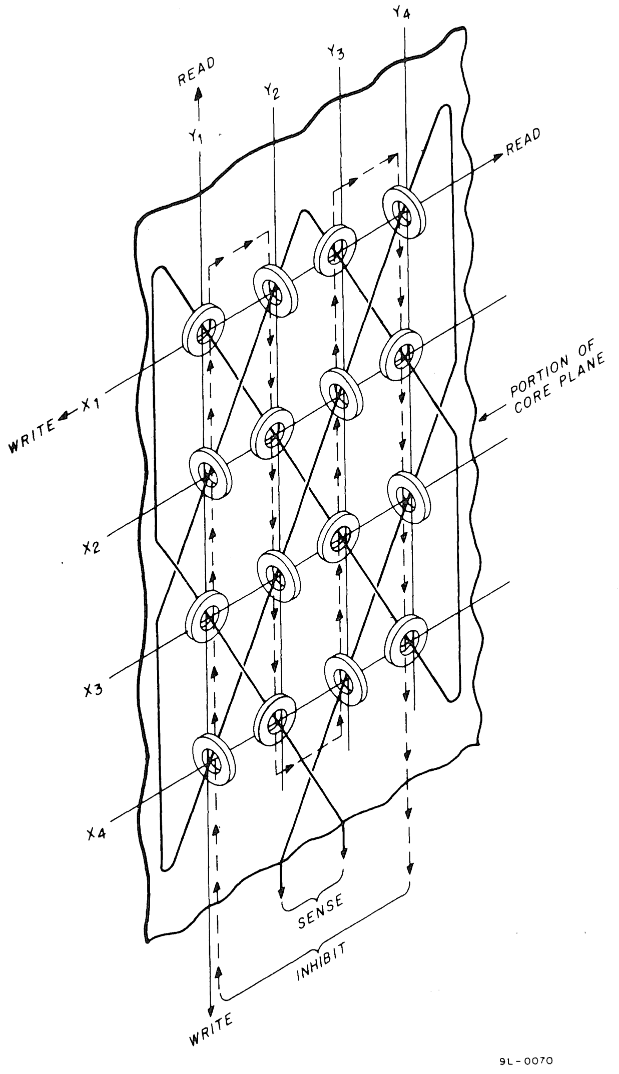


Figure 6-6 Core Memory Winding Scheme

ing MC71-0-2). The output at A06N is also applied to pin AD of delay B310 at AB04. At a nominal delay of 37.5 ns (adjustable by jumpers) the signal is inverted at A05D to produce READ ON used for jamming at A08. If the SEL level is also present, A08 sets to produce READ(1) at between +211.5 and +220 ns (assuming a nominal delay of 12 ns per transfer).

The SEL level is a function of memory bank selection (Field 0 + Field 1). Inverters B104 at B07 and B08 comprise bank selection decoding by sensing coincident combinations of bank selector switches SW3/SW4 and extended memory addressing bits EMA03 and EMA04. SEL will be present only if the states of EMA03 and EMA04 correspond to the states of SW3 and SW4. Table 6-1 shows the switch positions for extended memory capability.

Table 6-1
W712 / B09 Combinations

SW3	SW4	MEM BANK
0	0	0 -8K
0	1	8K-16K
1	0	16K-24K
1	1	24K-32K

When using 8K memory, these switches (located at the rear of each memory bank) are both down (00). Extended memory bits EMA03 and EMA04 are also 00 applying the -3V level (SEL) to the jam input A08K. Other banks are inhibited by these same bits since they do not match the setting of SW3 and SW4 on those banks.

READ(1), a 440-ns wide pulse seen at A08E, is used to initialize the memory read cycle by gating the read transistor switches, on drawing MC71-0-10, thereby completing both the source and return paths for the X- and Y-read current.

READ(1), seen at A08D, is used to condition the second delay on MEM START at A05J to +400 ns. From here it is further delayed and amplified to produce a basis for the SA STROBE signals (B06D)/(B06N) which occur at +560 ns; and for the MEM STROBE (A07V), which occurs at +600 ns. The SA STROBEs (0 - 8 and 9 - 17) are both fed to the sense amplifier (drawing MC71-0-3) where they gate the bit amplifier chains. The MEM STROBE is sent back to the CP where it is used to grant access to the memory by the CP.

The third delay on MEM START at +635 ns indicates the end of the read cycle as READ OFF at B03H which resets the read flip-flop A08 at pin H, and brings up READ(0).

READ(0) from A08E, which begins at +660 ns, is fed to pulse amplifier A07F for application to delay line

AB02. This delay chain establishes the inhibit/write timing.

The first tap at +825 ns is inverted to produce INH ON (A05L) which jams A08 to produce INHIBIT 1 (1) from pin N and A09 to produce INHIBIT 2 (1) from pin N simultaneously at +890 ns. INHIBIT 1 (1) (A08N), a 465 ns-wide pulse, is sent to pins U of the inhibit drivers for Field 0 (drawing MC71-0-4) as one of the conditioning levels in its operation. INHIBIT 2 (1) (A09N) is used by Field 1 (drawing MC71-0-5) for the same purpose.

The second delay tap, at approximately +900 ns, produces WRITE ON (A05N) which jams A08 yielding WRITE(1) at +920 ns. WRITE(1) (A09D), a 410 ns-wide pulse, is used to initialize the memory write cycle by gating the write transistor switches on drawing MC71-0-10, thereby completing both the source and return paths for the X- and Y-write current.

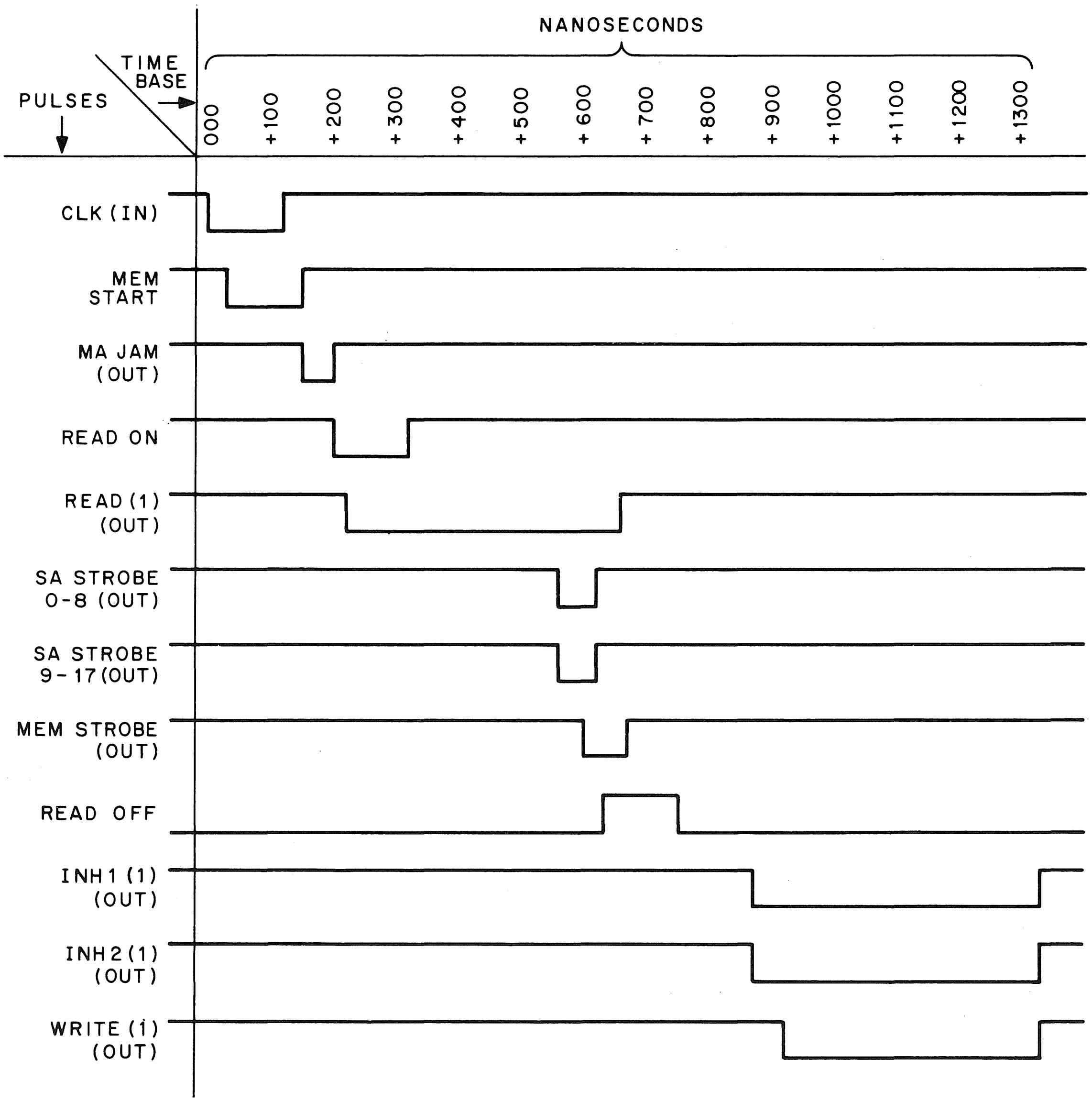


Figure 6-7 MC71-A Timing Diagram

The third delay tap, at approximately +1330 ns, is the WRITE OFF signal which at B03N resets both the inhibit and write flip-flops.

For the time that this sequence is begun by the generation of MEM START to the conclusion of the WRITE(1) pulse, I/O equipment should not send information to memory. This condition is prevented by a power clear-key initialize clear (PK CLR) from the CP. If during this cycle, a printer key is erroneously pressed, or if the computer is turned on or off; a pulse is sent to inverter A05S where it will immediately reset the read, inhibit and write flip-flops (if set) protecting against loss of stored memory information. The cycle will reinitiate upon the next CLK pulse (+1500 ns), if SM(1) is present.

6.2.2 Memory Addressing

The memory address flip-flops are shown in drawing MC71-0-2. This register functions to convert the last 13 bits in the MB word (address field) into equivalent binary form for use in memory address selection. Its inputs are taken from the memory buffer (MB05-17). Its outputs are sent to the X- and Y-address decoding and selection switches (G211) on drawing MC71-0-6 and MC71-0-7, respectively. The B213 configuration used in this application functions as a jam transfer flip-flop in which MA JAM (a -3 Vdc, 50 ns pulse) is applied simultaneously to pins T and J of all flip-flops against either a ground (1) or -3 Vdc (0) level at pins U and K, representing the state of each bit. When these conditions are coincident, the flip-flops are either set or reset and -3 Vdc 1s and/or 0s are seen at pins P and E. The two flip-flops MA05A and MA05B at C12 are configured as buffered drivers in which MB05(0) applied simultaneously to pins K and U will yield a -3 Vdc MA05A(0) at pin E, and MB05(1) will produce a -3 Vdc MA05B(1) at pin P. These pulses are used in NANDing operations on the X- and Y-selection switches for each field. MA05A (0) selects Field 0 and MA05B(1) selects Field 1. The MA JAM bus operates against 150 Ωtermination at pin V of the MA17 flip-flop at D12.

6.2.3 Address Selection

As described previously, the address at which a set of bits is stored is determined by the interaction of the X- and Y-axis address decoding and selection matrices. The X-axis (Field 0) diagram is shown in drawing MC71-0-6, and the Y-axis (Field 0) in drawing MC71-0-7. Inputs are MAs from the MA flip-flops and outputs are electromagnetic effects on a string of cores.

Each axis selector contains two G211 decoders and a diode selection matrix. Selection current (R/W SOURCE), originating in the X-Y drive circuits (drawing MC71-0-10); flows from one decoder, through its selected transistor switch to the matrix. Here it finds its diode path to a selection line (depicted as a coil in the drawings), on which all cores, in any one row or column, for all planes in sequence are strung; and exits through its diode path and selected switch to the R/W RETURN line. The effect is to apply half magnetizing current to a string of 1152 cores in the X-axes and half magnetizing current to a string of 1152 cores in the Y-axes, resulting in one intersected core per plane being magnetized in one of two states.

NOTE

> The fact that matrix lines intersect on any single drawing has no coordinate significance. Layout of components was dictated by drafting convenience.

Each G211 decoder contains eight 4-input logic NAND gate inverters which control the base bias on eight transistor switches. Each switch in turn completes a diode path between selection current source or return and one end of a specific select line (see Figure 6-8). Current in a read direction will flow from R/W SOURCE through D1, Q1, and D2; and from there through the select line, and return through D3, Q2, and D4. Current in a write direction will flow from R/W RETURN through D5, Q2, D6, the select line, D7, Q1, and D8. As can be seen from the schematic, current cannot flow unless both Q1 and Q2 are turned on; a condition enabled by coincidence of two sets of MAs at the inputs of both gates. Negative clamp for operation of all gates is taken from -V DRIVE originating at drive resistors G630(B38) for the X-axis, and at G630(B39) for the Y-axis (drawing MC71-0-10). Distribution of MA binary states across the gates is such to produce selection of any one of 64 lines in both axes. An additional condition is made coincident with all gates in the same field. This is the two MA forms of MB05, used to switch to Field 1 in extended memory operation. Field 0 utilizes the negative level form of MA05A(0), while Field 1 is enabled by the negative level form of MA05B(1). This latter condition is the only difference in operation between Field 0, just discussed, and Field 1 as shown in drawing MC71-0-8 for X-axis selection, and MC71-0-9 for Y-axis selection.

For purposes of address decoding, the X-axis gates use both binary conditions of MA06 through MA11, while the Y-axis gates use both conditions of MA12 through MA17. Because all gates are NAND inverters, the -3 Vdc form of both 1s and 0s is used for coincidence.

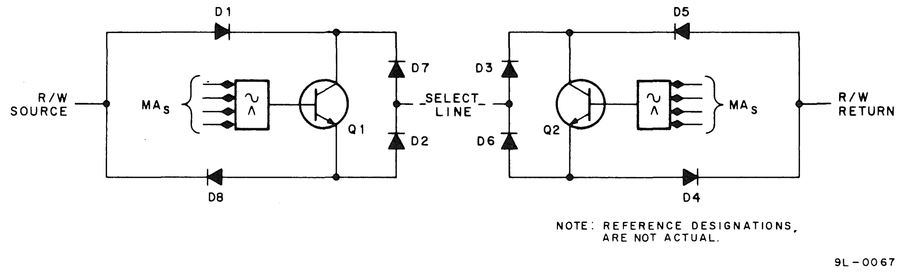


Figure 6-8 Address Selection, Simplified Schematic

6.2.4 Bit Sensing During Read

The sense and pulse amplifiers are shown in drawing MC71-0-3. These circuits function during the read-half-cycle to repeat into the memory buffer all 1s sensed during the core read transition. Inputs are by cables W990 (A31 and B31) from Field 0, and W990 (A21 and B21) from Field 1. Sensed pulses enter their respective dual-input sense amplifiers G014 at pins J and H from Field 0, and F and E from Field 1. Input switching is supplied by positive levels applied to either pins M (for JH input) or L (for FE input), one of which is always present from sense amplifier selectors G010(A12) and (B12).

When sensing the basic 4K stack (field 0), positive level MA05A(0) is fed to G010(A12) at pins F and R. The G010s function as bus drivers to apply a positive select level from pins D and N to pins M of all bit sense amplifiers allowing only Field 0 inputs to be available for amplification. When in extended memory mode MA05B(1) is fed to the G010 at B12, selecting Field 1 inputs at F and E. The SA STROBEs 0-8 and 9-17, occurring simultaneously at +560 ns, turn on all amplifiers for 60 ns allowing any bit 1s to be fed from pins V as positive pulses to their respective W612 pulse amplifers at pin F. Each sense amplifier chain operates into a termination of 150 Ω at pin T. Pins K, N and S receive closely adjusted reference voltages from the master slice control G008 at A11 for use by the sense amp clamping and comparator stages.

The slice voltage (-4.1 Vdc reference +10 Vdc) from A11H, the second-stage clamp voltage (+6 Vdc reference -15 Vdc) at A11N, and the fixed first-stage clamp voltage at A11M are used for pulse forming, level referencing, and noise suppression in the sense amplifiers.

The voltage from A11M is used as a preset reference level against which the amplified signal, sensed by core transition, is compared. This level is set to reject all amplified noise generated in those bit planes in which transitions did not occur, and transmit to the PAs only valid 1s.

The voltage from A11N is used as a reference level for the second differential amplifier stage within each G014, the output of which is compared with the master slice voltage from A11H at an internal rectifying slicer. If the signal voltage fed to the slicer exceeds the master slice voltage, the output gate is enabled producing a "slice window." This window then is sampled at the output gate by the SA STROBE at a time when the ratio of read 1 to read 0 is maximum.

The output positive pulse from G014(V) is applied to pin F of its associated W612 pulse amplifier. Here it is inverted, standardized (312 ns), and fed out via pin L to the memory buffer through the CP/Memory Interface (drawing MC71-0-11).

6.2.5 Voltage Regulation and Selection Drive

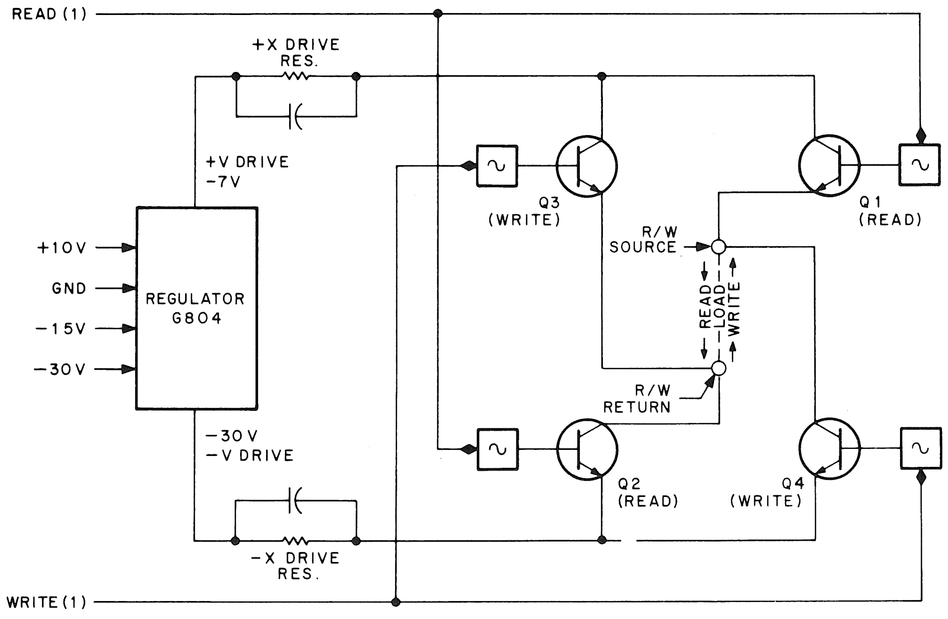
The voltage regulator and X-Y drive circuit is shown in drawing MC71-0-10. A simplified schematic is shown in Figure 6-9. All power required for operation of the core memory system comes from the +10, -15 Vdc and the -30 Vdc supply (Section 7.3.2). The +10 and -15 Vdc outputs are used for transistor logic power and the -30 Vdc supply is used for stack drive current.

In the core memory system, a G804 control module regulates the -30 Vdc supply and prevents operation of the CP if any supply exceeds the limits defined in the overall system specification. The G805 modules deliver positive drive (-7 Vdc) and negative drive (-30 Vdc) to the selector switches G211. The selected switch pair connects the voltage across the appropriate core drive line, and the voltage differential creates the current through the line.

Figure 6-9 is a simplified schematic diagram of the voltage regulator and X-Y drive, drawing MC71-0-10. Although only the X-axis components are show, the Y-axis selection process is identical. During READ(1), Q3 and Q4 are cut off while transistor switches Q1 and Q2 are turned on, allowing drive to flow from + V DRIVE through the +X DRIVE RES, and Q1 to the R/W SOURCE bus. Arrows indicate the direction of read current through the X-selection line (load) to the X R/W RETURN. Drive then flows through enabled Q2, the -X DRIVE RES to the - V DRIVE terminal of regulator G804. During WRITE(1), Q1 and Q2 are cut off and drive flows through Q3, the load (in reverse direction) and Q4. The +X and -Y drive resistors limit drive current to 340 mA during both read and write.

The Control Module G804 at DC04 accepts the common + V (-7 Vdc) and - V (-30 Vdc) outputs from negative regulator modules G805-CD01 and CD03. Zener diode D21 establishes the reference level for differential amplifiers Q6 - Q8 and the voltage adjustment rheostat R12. The differential amplifier output at the collector of Q8 controls the current through transistors Q7 and Q9, whose emitter circuit delivers the control current to the series regulator in G805 via terminals CK,CL.

The thermistor input at the differential amplifiers provides output voltage compensation as a function of core stack temperature. The thermistor is located within the core stack. As the core stack temperature rises, the thermistor resistance increases, and the dif-



NOTE: REFERENCE DESIGNATIONS ARE NOT ACTUAL.

9L-0076

Figure 6-9 X-Axis R/W Drive Selection Simplified Schematic

ferential amplifiers cause the output voltage of the negative regulator modules to decrease.

Other sensing circuits in the control module supply a POWER OK signal to the CP whenever the transistor logic voltage and the negative regulator voltages are correct. The POWER OK remains negative as long as the voltages remain within 3V of their designated values. If not, the level goes to ground preventing the occurrence of spurious memory cycles and consequent disruption of stored information, particularly during initial power turn-on.

On the module schematic, transistor Q10 goes into conduction if +V becomes less negative than -3 Vdc, as governed by the 3V reference levels set up in diode packages E3/E4. Q10 turns on Q11, which places CV at ground. Likewise, transistor Q1 conducts if the -15 and +10 Vdc supply voltages are outside the specified limits. This causes the DR level to go to ground. CV and DR are tied together as $\overline{\text{POWER OK}}$ which goes to the clock and run logic via the CP/ memory interface.

NOTE

> The tab terminal is ground return for the memory voltage relay K2 on drawing 712-0-1.

In the negative regulator G805, series regulator Q1 accepts the control current from the differential amplifiers in the control module G804. Conduction in Q1 places a proportionate voltage drop across resistor R2. The nominal drop across R2 is 23 Vdc, so that the lower end is at -7 Vdc and the higher end sees the full -30 Vdc supply voltage. The five 20 μF capacitors filter the output ripple to less than 50 mV. The discharge time of filter capacitors in the computer supplies allows 10 ms of memory operation following power interruption. Maximum output current of the G805 is 4A. The outputs go to the drive resistor boards G630 and the address selectors.

Four resistor boards G630 are connected between the negative regulator outputs and the source selection switches G219. Each resistor has a parallel peaking capacitor which offsets the inductance of the line and thus presents a better current rise time. The common side of the + Y and - Y drive resistors are decoupled to ground, while the + X and - X drive resistors are returned to the opposite regulator terminal.

6.2.6 Inhibiting During Write

The negative regulator module G805 also supplies -7 and -30 Vdc voltages to the inhibit drivers shown in drawing MC71-0-4 for Field 0 and MC71-0-5 for Field 1. These circuits are identical except for the source of certain control signals. For the purposes of this discussion only Field 0 will be considered. Specific differences will be described at the end of the discussion.

The function of the inhibit drivers is to prevent the writing of a 1 in a selected core by generating a field in opposition to the field in any Y-select line. In this way, the coincidence of current is prevented and the selected core is left in the 0 state. To accomplish this, the MB is conditioned to disable any bit-plane driver in which a 1 has been sensed during read, thereby allowing the X-Y drive current coincidence to rewrite the 1 in core. Conversely, if a 0 had been sensed, the MB would not disable the inhibit driver, coincidence would be inhibited, and the core would remain in the 0 state.

Inputs are taken from the memory buffer (MB00-17), the memory address register (MA05A or MA05B), and from the X-Y driver circuit (+ and - V INH, INH 1 and INH 2). Outputs are by cable to the core stack. The circuit comprises nine G218 modules at locations A33 - A36 and B33 - B36. Current limiting resistors for each half module are contained on G630 resistor boards connected between either pins E and J or M and R of each module. These 56 Ω resistors limit the inhibit current to 310 mA, sufficient to offset the Y-axis drive current during write.

When the particular MB bit is a 0 coincidentally with the correct MA05 bit, an inhibit current will flow from pin F or R through the bit-plane inhibit winding to pin H or P for the duration of the INH 1 (1) signal. Should any MB bit be a 1, no output will be seen.

The operation of the inhibit drivers for Field 1 (drawing MC71-0-4) is identical except the memory address bit required is MA05B(1) from B213/C12(N), and the timing pulse is INH 2(1) from B213/A09(N).

CHAPTER 7
CONSOLE POWER CONTROLS

7.1 MANUAL CONTROLS

The logic sequence to implement functions selected on the PDP-9/L console is described in this chapter. Timing diagrams for the console key functions, the clock/run diagram (KC10), and key flow diagram (KC6) supplement the text and should be referred to while using this chapter. The marginal check (MC) panel and maintenance check panel controls are described in the maintenance section of the manual, Chapter 10.

7.1.1 Power Turn-On

When the console POWER switch is turned ON, PWR OK from the power supplies and/or the ground from Low-Voltage Detector W505 reset the RUN and REPT flip-flops, and enables the application of PWR CLR POS pulses from the main clock to the computer system. RUN(0) enables the REPT CLK. In addition to initializing certain flip-flops throughout the computer system, the PWR CLR POS pulses (KC10) generate IND CLK and PK CLR pulses at pulse amplifiers S602-J25K and W612-H32N. These in turn generate CM STROBE A, C, D on drawing KC16. CM STROBE clears all CM sense flip-flops, drawing KC19, because CM CURRENT is absent.

When the 712 Power Supply is stablized, the PWRCLR POS, IND CLK, and PK CLR pulses are removed. The next REPT CLK pulse sets flip-flop C in conjunction with RUN(0), drawing KC10. Thereafter, flip-flop C alternately resets and sets on the leading edge of each REPT CLK pulse. The REPT CLK pulses remain until the RUN flip-flop sets. These are 100-ns pulses which occur at manually selected intervals of 2 µs to 1s. The five-position REPT switch on the console selects this REPT CLK frequency.

Upon each reset of flip-flop C, flip-flop B sets and resets in a divide-by-two counter mode. When B sets for the first time, the B(1) Λ A(0) condition generates IND EN at R111-J26PN. IND EN goes to the wiper arm of the console REGISTER DISPLAY switch, drawing CS-5408018 to enable the selection of computer registers for display at the REGISTER indicator. IND EN also sets the IO BUS ON flip-flop, drawing KC19(3). C sets on the next REPT CLK pulse. C(1) strobes a DCD input gate to the IND CLK pulse amplifier, conditioned by B(1). IND CLK produces CM STROBE B as before, but now the CM STROBE B will set the PCO, ACO, ARO, or MQO flip-flop if any of these has been selected by the REGISTER DISPLAY switch. The IND EN level is still present by virtue of A(0) Λ B(1); IND EN applies PCO, ACO, ARO, or MQO to the appropriate jam-input gate if the respective register is selected at the REGISTER DISPLAY switch, drawing CS-5408018.

The selected switch position thus sets the appropriate sense flip-flop on CM STROBE B. The flip-flop in the set state gates the contents of the selected register onto the A bus. The contents on the A bus go directly to the ADR. IO BUS ON(1) gates the contents from the ADR to the I/O bus, drawing KC21 to I/O bus (B) via the input mixer, drawing KD7, then via the CP/IO and IO/console interfaces, drawings KC25, KD4, and KD6 to the REGISTER indicators, drawing CS-5408020.

Flip-flops C and B recycle on the REPT CLK pulses, and the computer remains in this no operation state until a console key is operated to allow flip-flop A to set. Once flip-flop A becomes set with REPT(1), IND EN is removed, and the REGISTER DISPLAY circuits are thus disabled. REPT(1) is generated by KEY DLY from DELAY flip-flop. This is true for the operation of any console key; therefore, the register display is meaningful only during the interval between computer no operation state and the setting of A (and RUN) on a key operation.

7.1.2 START Key

The START key and the ADDRESS switches (3-17) on the console operate together to start execution of a program that has been stored in core memory. Drawing KC11 is the START timing. The operator first loads the program's starting address in binary format into the ADDRESS switches (switches up for binary 1s). When he depresses the START key, the key supplies a ground KST level to the NOR gate at R111-J33UV, drawing KC10(1), via the CP/console interface. This level becomes the negative KEY BUS and positive KEY BUS(B) levels. KEY BUS(B) removes $\overline{\text{KEY BUS(B)}}$ from inverter S107-H34F, thereby removing the collector ground from the assertion output of the REPT flip-flop.

KEY BUS(B) also triggers the 50-ms delay at R320-J32V. The negative output recovers after 50 ms to trigger another delay at R302-J32M, which produces a 50-μs negative KEY DLY.

The 50-ms delay period allows sufficient time for settling of switch contact bounce and the 50-μs KEY DLY allows for execution of the longest instruction if the START KEY (or any other key) was operated during a running program. (In a running program the RUN flip-flop is in the set state; an instruction DONE(1) level issued in a control memory process word at some point during the 50-μs KEY DLY resets RUN via NAND gate R111-J28N. Resetting RUN disables the application of CLK POS and CM CLK pulses to the computer system, stopping all operations.)

The KEY DLY recovery sets the REPT flip-flop conditioned by $\overline{\text{KRI}}$ (READ IN key inactive). REPT(1) conditions the DCD set gate of flip-flop A, which sets on the next reset of B. The next REPT CLK pulse sets C, which then strobes the IND CLK gate conditioned by A(1). IND CLK now strobes the KEY INIT POS gate which is conditioned by A(1) ∧ $\overline{\text{KCT}}$. The $\overline{\text{KCT}}$ level is derived from the active CONTINUE key. KEY INIT POS starts the CM timing chain, generates CM CURRENT and CM STROBE, drawing KC16, and generates PK CLR, drawing KC10.

CM CURRENT enables the CM address selectors, drawing KC17, to decode the address in the CMA register. KEY INIT POS and PK CLR act together to clear the MCA flip-flops, drawing KC19(1), for an address of 00. However, the KST level derived from the START key produces levels KIOA3, KIOA4 on drawing KC10. On drawing KC17, the cleared CMA levels CMA0(0), CMA1(0), and CMA2(0) gate KIOA3, KIOA4 into the address selectors, changing the address from 00 to 06.

At CM STROBE time, process word 06 is extracted from control memory. This word contains ADSO, MBI, PCI, SM, and CMA21. ADSO(1) goes to the input mixer on drawing KD7(1) where it is NANDed with $\overline{\text{AUTO RESTART}}$. This level comes from the optional Power Failure Detection KP09A to denote that no automatic restart is in progress. If the option is not installed, the input to the NAND gate is disconnected, and is therefore of no consequence. ADSO(1) produces ADSO(G), which gates the ADDRESS switch levels ADDR SW03-17 into the R141 mixer modules. The outputs are buffered at I/O bus (B).

At the same time ADSO(1) generates LIO in bus driver B213-D12, drawing KC13. LIO (load I/O) gates the address onto the O bus, drawing KC20. PCI(1) gates the address from the O bus into the PC. Likewise, MBI(1) gates the same address from the O bus into the MB.

The next REPT CLK pulse resets C. As C resets, it sets the RUN flip-flop in conjunction with the A(1) condition at the input DCD gate. RUN(1) disables the REPT CLK, resets and holds A and B in the 0 state. The next CLK, ANDed with RUN(1) sets SEN. With SEN(1), the following CLK RUN generates CLK POS. CLK POS pulses are further inverted at pulse amplifiers B602-H33N and W612-E02 for negative CM CLK and CLK CLKD pulses, respectively. The CLK pulse starts the core memory cycle and the CM CLK pulse starts the control memory timing in conjunction with SM(1) of the process word 06.

The processor is now in the state in which it would be at the end of an instruction execute cycle during a running program. The address in the CMA register is 21 (fetch entry), RUN is set, and the MB contains the address of the next instruction to be fetched from core memory. MA JAM in core memory places the address in the MA 120 ns from CLK. The CM timing starts to extract process word 21, and the machine thus starts execution of the program.

CLK · RUN also set the SEN flip-flop on drawing KC10. SEN(1) monitors the states of PCO and ARO at the PCOS and AROS flip-flops throughout the running program. These flip-flops thus continuously reflect the PCO and ARO states.

7.1.3 PROGRAM STOP Key

The spring-loaded down position of the PROGRAM STOP key halts computer operations upon completion of the current instruction. It performs the same functions as the SING INST switch.

KSP (key stop) from the PROGRAM STOP key or SW SGL INST from the SING INST switch applies one enabling input to gate R111-J28U, drawing KC10. The other three inputs to the gate determine when the computer can be stopped. In order to stop, the console must be unlocked, the computer must not be operating in a program break segment (BK0, BK1 states from F35N), and the instruction currently being executed must finish (DONE(1) from the last CM process word in the computer execute cycle). When all these conditions have been met, the RUN flip-flop is collector-pulled to the 0 state by R111-J28U.

RUN(0) enables the REPT CLK, and the REPT CLK pulses start stepping flip-flops C and B. The first REPT

CLK pulse sets C; the next pulse resets SEN in conjunction with C(1), drawing KC10. Reset SEN removes the jam input level to the PCOS and AROS flip-flops.

PCOS and AROS now reflect and retain the states of PCO and ARO at the time the computer stops. PCO and ARO are both cleared by CM STROBE B and CM CURRENT produced from IND CLK on the next REPT CLK pulse.

7.1.4 CONTINUE Key

The spring-loaded down position of the CONTINUE key is used to resume the execution of a program after a programmed HLT or after a manual program stop condition. The KCT level produces KEY BUS, KEY BUS(B), and KEY DLY in the same manner as the START key. KEY DLY upon recovery sets the REPT flip-flop.

While the computer is stopped, REPT CLK pulses are stepping flip-flops C and B. REPT(1) now allows A to set, conditioning the set DCD gate of the RUN flip-flop. On drawing KC10, if PCOS is set, A(1) ∧ B(0) ∧ KCT(B) at R111-H30HJ produces PCO RESTORE which sets the PCO sense flip-flop, drawing KC19(3). A similar logical flow also applies to AROS and ARO RESTORE.

This logic is necessary in order to restore the PCO and ARO flip-flops, since they are cleared by IND CLK pulses when the computer is stopped. (Clearing them enables the use of the REGISTER DISPLAY switch.) PCOS will be set whenever the machine has been stopped during execution of a program, and AROS will be set whenever the machine is stopped at the end of a hardware read-in operation.

The next REPT CLK pulse resets C and the transition sets RUN via the DCD gate conditioned by A(1). RUN(1) clears A and B and inhibits the REPT CLK. The next CLK, ANDed with RUN(1) sets SEN. With SEN(1), the following CLK RUN generates CLK POS. The first CLK POS pulse resets REPT if the REPT switch on the console is off (down) and also starts the main memory and control memory, since SM(1) is present in the last CM process word. The CMA register at this time contains address 21, causing control memory to extract this fetch entry process word for a normal computer fetch cycle. Note that if the operator had activated IO RESET key, SM and address 21 would not be present and improper action of CONT would result.

7.1.5 DEPOSIT THIS

The upper, spring-loaded DEPOSIT THIS position of the DEPOSIT THIS/DEPOSIT NEXT key is used to store a word in core memory. The operator first loads the address of the intended core memory location into the ADDRESS switches and the word itself into the DATA switches. When he raises the DEPOSIT THIS/DEPOSIT NEXT key, the address is gated into the MB and the AR by successive control memory process words. A core memory cycle takes place as the address is jammed into the MA. However, at STROBE 0-8, 9-17 time in the read-half-cycle, the contents of the addressed memory location are kept out of the MB by the absence of the SAO bit. Instead, the word in the DATA switches is gated into the MB so that the write-half-cycle replaces the original contents of the addressed location with this DATA switch word. The original contents are lost.

The DEPOSIT THIS function normally starts from a computer stop condition. Figure 7-1 shows the timing for setting up a DEPOSIT THIS, DEPOSIT NEXT, EXAMINE THIS, or EXAMINE NEXT operation; all require the same set-up conditions. KDP on drawing KC10 sets the REPT flip-flop after the 50-μs KEY DLY as for program START. The REPT CLK pulses step the A, B, and C flip-flops, and KEY INIT POS occurs on the A(1) ∧ IND CLK condition. KEY INIT POS generates PK CLR, drawing KC10, and starts the CM timing, drawing KC16. PK CLR and KEY INIT POS act together to clear the CMA flip-flops, drawing KC19(1), for a CM address of 00. However, the KDP level derived from the DEPOSIT THIS key produces level KIOA5 on drawing KC10. CMA0(0), CMA1(0), and CMA2(0) on drawing KC17 allow the KIOA5 level to change the address to 01. At CM STROBE time, then, the process word 01 is extracted from control memory.

Process 01 contains ADSO, MBI, SM, and CMA25. ADSO(1) produces ADSO(G) to gate the ADDRESS switch levels ADDR SW03-17 into the input mixer, drawing KD7, via the CP/console interface. The gates place the address on the buffered I/O bus. ADSO(1) also generates the LIO level, drawing KC13, in bus drivers B213-D12. LIO gates the address onto the O bus, drawing KC20. MBI(1) jams the address into the MB.

The computer now waits for flip-flops A, B, and C to step through 0, setting the RUN flip-flop. RUN(1) disables the REPT CLK; the CLK POS (CLK and CM CLK) pulse after RUN(1) starts the core memory cycle and the CM timing chain in conjunction with SM(1), resets RUN in conjunction with KIOA5, and resets the REPT flip-flop. The address in the MB gets jammed into the MA at the start of the core memory cycle. The process word read out of control memory location 25 contains MBO, ARI, KEY, and CMA26. MBO(1) gates the contents of the MB (address) onto the B bus, the contents go directly through the ADR, and NOSH places them on the O bus. ARI(1) then gates the contents into the AR. For DEPOSIT THIS, the address does not get past the AR (see DEPOSIT NEXT).

STROBE 0-8, 9-17 occurs in core memory to read out the word at the address specified by the MA. However, this word is inhibited by the absence of SAO(1) and is therefore lost. MEM STROBE starts another CM cycle, drawing KC16.

KEY(1) is NANDed with a KDP+KDN level on drawing KC13 to produce DASO. KEY(1) also generates IN CLR in the CM timing, drawing KC16, as the timing progresses toward CM STROBE. IN CLR produces 1 → MBI, setting the MBI flip-flop, drawing KC19(2). Note that KEY ∧ (KDP+KDN) on drawing KC13 goes to the jam input of SAO, drawing KC19(3). This level keeps SAO in the reset state on the CLR pulse.

DASO gates the DATA switch levels DATA SW00-17 into the input mixer gates, drawing KD7, via the CP/console interface. The input mixer gates place the data word on I/O bus (B).

The KEY ∧ (KDP+KDN) level on drawing KC19(3), also generates KDP+KDN+RI. The KDP+KDN+RI level produces LIO on drawing KC13 in conjunction with MBI(1). LIO places the data word on the O bus from I/O bus (B). MBI(1) jams the word into the MB.

The core memory write-half-cycle stores the data word at the preselected address. The CM STROBE produced

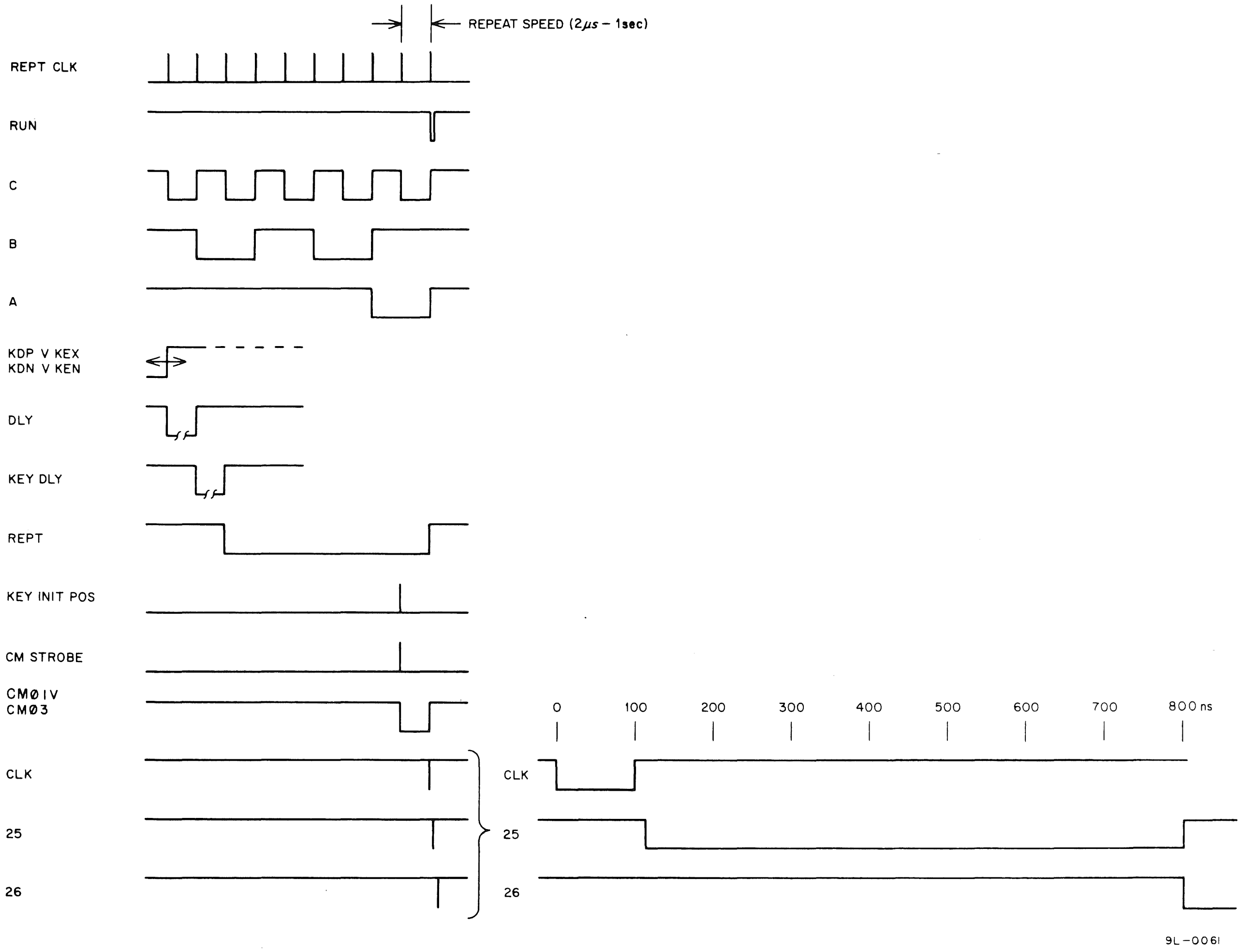


Figure 7-1 Initial Set-Up DEPOSIT or EXAMINE Timing

by MEM STROBE extracts process word 26 from control memory. Process word 26 contains PCO, SM, and CMA21. Since RUN was reset by KIOA5 and the CLK POS pulse, the computer stops and the control memory retains process word 26 in its sense flip-flops until the computer is made to proceed from CONTINUE,START, DEPOSIT NEXT, or EXAMINE THIS. From CONTINUE THIS, process word 26 accomplishes the operations performed by the BGN word at the end of an instruction. That is, it obtains the next address from the PC for the start of the next fetch cycle. For START, DEPOSIT NEXT, and EXAMINE THIS, the address from the PC does not reach the MB and the CM address (21, fetch entry) changes appropriately.

7.1.6 DEPOSIT NEXT

Having deposited a single word in core memory with the DEPOSIT THIS position of the DEPOSIT THIS/ DEPOSIT NEXT key, the operator may use the DEPOSIT NEXT position to store a series of words in consecutive locations. For each depression to DEPOSIT NEXT, the current address in the AR is incremented by 1, then placed in the MB for the start of the next core memory cycle. ADDRESS switch entries after the initial DEPOSIT THIS function are unnecessary and are in fact inhibited. The DATA switches must be used to enter the new data word before each depression to DEPOSIT NEXT.

On drawing KC10 and in Figure 7-1, KDN produces the same set-up conditions as KDP, but now the levels KIOA4 and KIOA5 are gated on. Thus the first process word is taken from location 03. Process word 03 contains ARO, +1, MBI, SM, and CMA25. ARO(1) gates the current address in the AR onto the A bus, and the A bus contents enter the ADR. Process +1(1) produces CI17, drawing KC14, to initiate a carry through the ADR. NOSH takes the incremented address of the ADR to the O bus, and MBI(1) places it in the MB.

The new address is thus ready in the MB for the core memory cycle that starts with the CLK POS pulse. Process word 25 stores the address in the AR as for DEPOSIT THIS. The entire process repeats for each depression to DEPOSIT NEXT.

7.1.7 EXAMINE THIS

The upper spring-loaded EXAMINE THIS position of the EXAMINE THIS/EXAMINE NEXT key transfers a single word from core memory to the MB for display at the MEMORY BUFFER indicator on the console. The address of the word to be examined is first loaded into the ADDRESS switches. When the key is raised, the address is gated into both the MB and the AR. The core memory cycle fetches the word and places it in the MB in conjunction with the control memory processes. The address remains in the AR, so that it can be displayed in the REGISTER indicator by turning the REGISTER DISPLAY switch to the AR position.

The EXAMINE THIS function is normally started from a computer stop condition, e.g., following a DEPOSIT THIS/DEPOSIT NEXT operation. On drawing KC10 and in Figure 7-1, KEX produces the same set-up conditions as KDP to take the initial process word from location 01. Process word 01 places the address in the MB, and the succeeding process word (25) places it in the AR for an EXAMINE NEXT operation.

IN CLR and CLR from the CM timing, drawing KC16, set the MBI and SAO flip-flops during the core memory read-half-cycle to gate the word from the sense amplifiers to the MB. IN CLR and CLR are derived from the KEY(1) bit of process word 25 as for DEPOSIT THIS. The computer stops until made to proceed from CONTINUE THIS, START, or EXAMINE NEXT. It is in this stop condition that the MEMORY BUFFER indicator and the AR selection of the REGISTER indicator can be observed.

7.1.8 EXAMINE NEXT

Having examined a single word in core memory with the EXAMINE THIS position of the EXAMINE THIS/ EXAMINE NEXT key, the operator may use the EXAMINE NEXT position to examine a series of consecutively stored words. For each depression to EXAMINE NEXT, the current address in the AR is incremented by 1, then placed in the MB for the start of the core memory cycle. ADDRESS switch entries after the initial EXAMINE THIS function are unnecessary and are in fact inhibited.

On drawing KC10 and in Figure 7-1, KEN performs the same set-up functions as KEX, but now the levels KIOA4 and KIOA5 are gated on. Thus the first process word is taken from location 03. Process word 03 contains ARO, +1, MBI, SM, and CMA25. ARO(1) places the current address in the AR on the A bus, and the A bus contents enter the ADR. Process +1(1) produces CI17, drawing KC14, to initiate a carry through the ADR. NOSH takes the incremented address of the ADR to the O bus, and MBI(1) places it in the MB.

The new address is thus ready in the MB for the core memory cycle that starts with the CLK POS pulse.

Process word 25 stores the address in the AR as for EXAMINE THIS. IN CLR and CLR derived from the KEY(1) bit of process word 25 set MBI and SAO during the core memory read-half-cycle to gate the word from core memory's sense amplifiers to the MB. The entire process repeats for each depression to EXAMINE NEXT.

7.1.9 READ IN Key

The READ IN key stores 18-bit binary words from punched paper tape in consecutive core memory locations. The operator first loads the address of the first word to be stored into the ADDRESS switches. When he depresses the READ IN key, the CP selects the paper tape reader for binary mode operation, then waits for the reader to read three lines of tape. The reader control logic in the I/O control section of the computer assembles these lines in a reader buffer (RB), drawing KD9(2). In the binary mode tape format for READ IN operations, tape channels 1 through 6 of each line contain one 6-bit character of an 18-bit word; channel 7 is punched only in the last line of the last word to be read, and channel 8 is punched in every line to control the gating of the 6-bit characters into the proper RB bit positions. When the RB is full, its contents are transferred to the MB and then are deposited in core memory. The current address of the core memory location is retained in the AR where it is incremented by 1 to store successive words in consecutive memory locations. The process continues until the reader encounters a line of tape in which channel 7 is punched indicating that this is the last line of the last word. The reader then stops and the CP executes the instruction encoded in the last word. This last word is usually a JMP instruction to the starting address of the program just loaded, or a HLT instruction to afford manual control of the start of the program.

The READ IN function normally starts from a computer stop condition, where flip-flops C and B, on drawing KC10(1), are cycling on the REPT CLK pulses (Figure 7-2). When the operator depresses the READ IN key, the KRI ground obtained from the key initiates KEY DLY as for all other key functions. For other keys, the trailing edge of KEY DLY sets the REPT flip-flop. REPT(1) would then permit the initial setting of flip-flop A, ultimately resulting in KEY INIT POS and in the setting of the RUN flip-flop. For READ IN operations, however, $\overline{\text{KRI}}$ at ground prevents KEY DLY from setting REPT so that the reader can have time to read three lines of tape into the RB. For this reason, the reader control logic determines when three lines of tape have been read and when flip-flop A shall set, as follows.

KRI and KEY DLY go to the read-in mode control, drawing KD8, via the CP-I/O interface. Both signals are applied to the DCD set gate of the READ IN 1 flip-flop. KRI conditions the gate and the positive-going trailing edge of KEY DLY strobes the gate to set the flip-flop.

READ IN 1(1) generates RSB (reader select binary) at pulse amplifier S602-F04K, and is also inverted at S107-H05D for a negative RI1(1)B level. RSB goes to the reader control, drawing KD9(1), where it resets the RDR ALPHA flip-flop and generates an IOT0104 command at pulse amplifier S603-D10T. The IOT0104 command clears the RB, RDR 1, RDR 2, and RDR FLG flip-flops, and sets RDR RUN, thereby starting the tape reader mechanism. RDR COUNT pulses, timed with the appearance of the tape channel 8 holes, step the RDR 1 and RDR 2 flip-flops; the first RDR COUNT sets flip-flop RDR 1. RDR 1(1) strobes the first line of tape into the reader buffer, RB00-05. The second RDR COUNT sets RDR 2. RDR 2(1) strobes the second line of tape into RB06-11. The third RDR COUNT sets the RDR FLG, conditioned by RDR 2(1). RDR FLG(1) strobes the third line into RB12-17, and resets RDR RUN, to stop the reader.

RDR FLG(1) is buffered at S107-E06N, drawing KD9, for a negative RDR FLG(1)B level. RDR FLG(1)B and RI1(1)B generate RD START RQ on drawing KC10(1). RD START RQ generates KIOA5 and conditions a DCD set gate at flip-flop A. On the next reset of flip-flop B flip-flop A sets.

A(1) conditions the KEY INIT POS gate and the RUN set gate. The next REPT CLK pulse sets C, and C(1) turns on IND CLK to produce the KEY INIT POS pulse. C resets on the next REPT CLK pulse, resetting A and setting RUN.

KEY INIT POS generates PK CLR, both pulses reset the CMA 00,01,02 flip-flops, and KIOA5 changes the resulting CM address from 00 to 01, drawing KC17 KEY INIT POS starts the CM timing chain, drawing KC16, to extract process word 01. Processes ADSO(1) and MBI(1) transfer the ADDRESS switch contents to the MB and SM(1) starts the core memory cycle on the CLK pulse after RUN(1) as for the DEPOSIT THIS operation. SM(1) and CM CLK also restart the CM timing to extract process word 25. CLK POS resets the RUN flip-flop in conjunction with KIOA5. KEY(1) of process word 25 sets the READ IN 2 flip-flop on drawing KD8. READ IN 1(1) ∧ READ IN 2(1) set the IOT0102 flip-flop, drawing KD9(1).

IOT0102(1) goes to the input mixer, drawing KD7(1), where it generates the RDR ON BUS level. RDR ON

BUS gates the RB00-17 contents into the input mixer gates, whose outputs are NORed onto I/O bus (B).

Other processes evolved from process word 25 gate the current address held in the MB into the AR, gate the I/O bus (B) contents onto the O bus, and the O bus into the MB, as for DEPOSIT THIS. The core memory write-half-cycle stores the MB contents at the current address previously transferred from the MB to the MA.

Process word 25 contains CMA26. But now R12(1)B derived from READ IN 2(1) boosts the address to 27 in conjunction with KEY(1), drawing KC17. The CM STROBE derived from core memory's MEM STROBE at the CM timing chain therefore extracts process word 27.

Process word 27 is a do-nothing process which merely contains CMA00. At this time, another RSB pulse occurs by virtue of READ IN 1(1) and KEY(1) of process word 25, delayed 1.2 μs in R302-F05M, drawing KD8. RSB produces IOT0104 to turn on the tape reader as before. The reader control assembles the next word in the RB.

RI1(1)B $\wedge$ RDR FLG(1) generates RD START RQ and KIOA5 as before, and additionally RI2(1)B generates

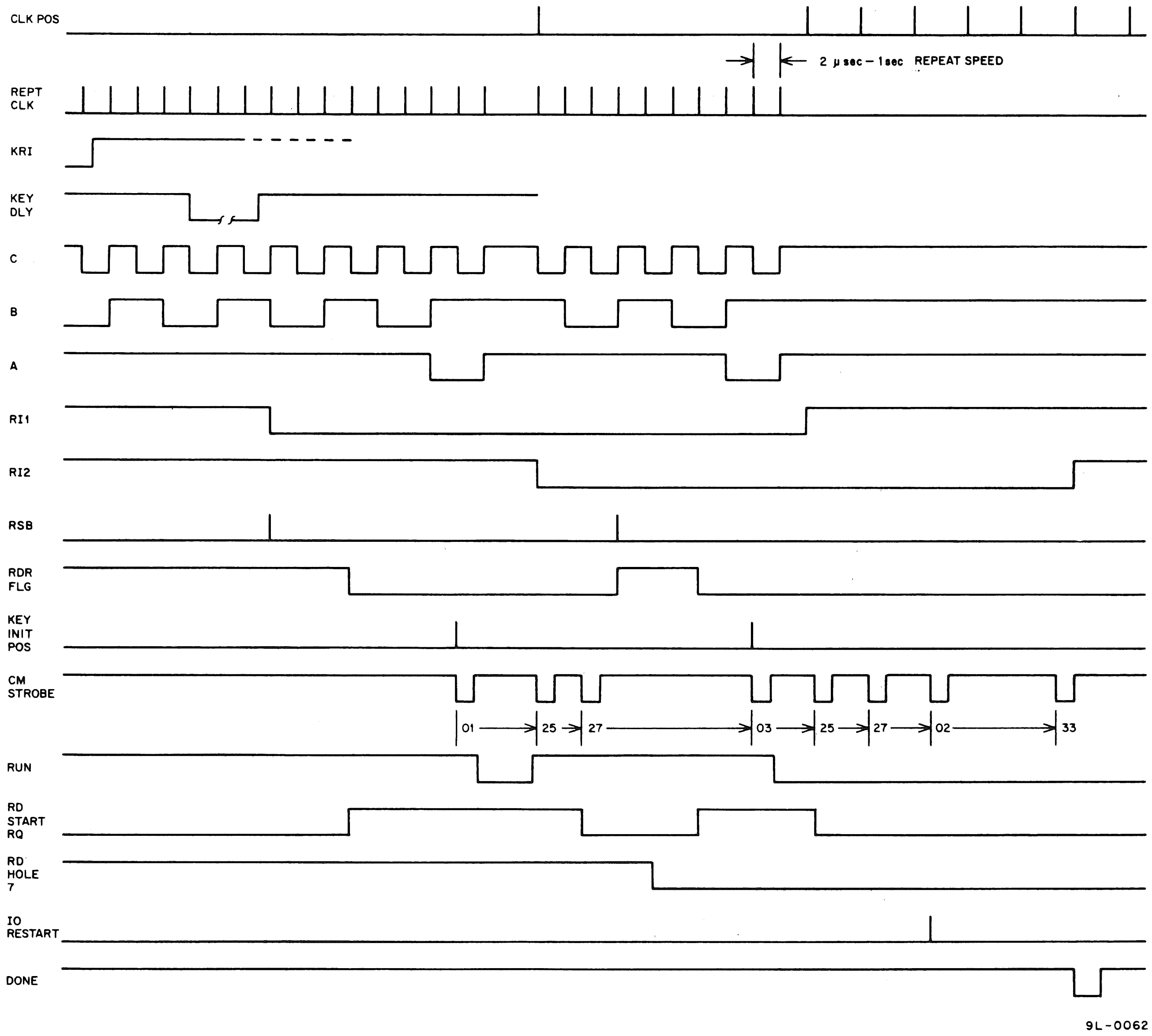


Figure 7-2 READ IN Mode Timing

KIOA4. Therefore, the CM STROBE initiated by the next KEY INIT POS pulse extracts process word 03. Process word 03 increments the current address in the AR by 1 and gates the incremented address into the MB. The computer proceeds to process word 25, storing the new word in the next consecutive core memory location. Process word 27 is extracted as for the first word. These processes (03,25,27) repeat to store all successive RB words in consecutive locations, until the last word is detected by the presence of a RD HOLE 7 level from the last line in the reader control logic.

Figure 7-2 is a practical illustration of the timing using a two-word tape record. Since the second word is the last, the RD HOLE 7 level appears with the third line of information at W023-A17L, drawing KD9(2). RD HOLE 7 becomes RD HOLE 7(B) and RD HOLE 7(C) at inverters S107-C06T, S107-E06T. On drawing KD8, RD HOLE 7(B) conditions the DCD reset gate at the READ IN 1 flip-flop and the DCD gate at the INPUT IO RESTART pulse amplifier S602-F04U.

KEY INIT POS occurs to extract the first of the three process words 03, 25, 27, and RUN sets to issue the CLK POS pulse. CLK POS cannot reset RUN because of RD HOLE 7(C) at the reset input gate.

KEY(1) of process word 25 resets the READ IN 1 flip-flop, and triggers the 1.2 μs delay in R302-F05M, drawing KD8. Reset READ IN 1 removes RI1(1)B and consequently RD START RQ from flip-flop A, drawing KC10(1). The removal of RI1(1)B also removes KIOA5 from the CM address gates. RI2(1)B remains to place KIOA4 at the address gates.

Delayed KEY(1) recovers after 1.2 μs to produce INPUT IO RESTART. Since READ IN 1 is reset, the delay upon recovery cannot produce another RSB pulse. With the RUN flip-flop remaining set, the REPT CLK is disabled, flip-flops A, B, and C cannot recycle, and KEY INIT POS cannot occur. The next process word must be extracted, therefore, by INPUT IO RESTART. This pulse goes to the I/O control logic, drawing KD3(3), where it triggers pulse amplifier S602-H20K for IO RESTART. The IO RESTART pulse triggers the CM timing chain, drawing KC16.

The next CM STROBE thus obtained will extract process word 02 from control memory. (The CM address in process word 27 is 00; this is boosted to 02 by KIOA4.)

Process word 02 contains ARO, SM, and CMA33. ARO(1) gates the current address in the AR onto the A bus, the address on the A bus goes through the ADR, and NOSH places it on the O bus. In addition, ARO(1) produces RQ MBI in conjunction with SM(1), RUN(1), and CLK on drawing KC19(2). RQ MBI generates 1 → MBI, setting the MBI flip-flop. MBI(1) gates the current address from the O bus into the MB. Note that this is the address of the last word stored in core memory by the previous 03, 25, and 27 processes.

SM(1) waits for the next CLK pulse, at which time the core memory cycle starts and the CM timing chain cycles to extract the next process word from location 33. Process word 33 is the XCT entry word which causes the computer to execute the instruction encoded in the last stored word. DONE(1) in the succeeding instruction execute process word resets the READ IN 2 flip-flop.

7.1.10 I/O RESET Key

The I/O RESET key clears all flags in I/O devices, control flip-flops in the I/O control logic, and all CP registers except the PC.

The I/O RESET function normally starts from a computer stop condition. When the operator depresses the I/O RESET key, the ground KIO level obtained from the key conditions input DCD gates at pulse amplifiers S602-H11K and S603-J10F, drawing KD3(1). These gates are strobed constantly by positive CLK pulses from the main clock, drawing KC10. Unlike the gated negative CLK and CLK POS pulses, the positive CLK pulses are present independent of the RUN flip-flop status. These ungated CLK pulses, therefore, generate IO PWR CLR POS and IOT PWR CLR pulses at the amplifiers as long as the KIO conditioning level is present.

The IO PWR CLR POS pulses reset the BK0, BK1 flip-flops, drawing KD3. They also go to the flags of optional I/O devices via the I/O bus, and to certain flip-flops within the I/O control section controlling the standard I/O equipment. The IOT PWR CLR pulses clear the DCH SYNC, PIE, and IO0, IO1 flip-flops, drawing KD3. IO PWR CLR POS generates IO CLR on KD3(2). IO CLR resets PROG SY, PROG SYNC, and BK flip-flops.

The ground KIO level also produces KEY DLY on drawing KC10 as for other manual key functions, and generates KIOA3, KIOA5. KEY DLY upon recovery sets the REPT flip-flop, allowing flip-flops C and B to set A, and generating KEY INIT POS and setting RUN. The ensuing CLK POS pulse resets RUN shortly thereafter, in conjunction with KIOA5 and the negation states of RDR HOLE 7(C), RI1(1)B.

The CM STROBE produced in control memory by KEY INIT POS extracts process word 05 because of the KIOA3, KIOA5 address levels at the CM address gates, drawing KC17. Process word 05 contains ACI, ARI, MBI, MQI, LI, KEY, CONT, and CMA27. ACI(1), ARI(1), MBI(1), and MQI(1) open their respective registers to the contents of the O bus. Since the O bus contains nothing at this time, the AC, AR, MB, MQ registers are filled with 0s. LI(1) strobes the jam input gate of the LAR, drawing KC15. Since the "normal" input gates are disabled by a ground KEY(0), LI(1) resets the LAR.

The CM STROBE produced by KEY INIT POS restarts the CM timing in conjunction with CONT(1), to extract process word 27. Process word 27 is the do-nothing process which contains merely the CM address 00. LI(0) resets the Link.

Reset RUN allows REPT CLK pulses to recycle flip-flops C and B, and the computer remains in the 00 KEY NOP state until another key is operated.

7.1.11 REPT and REPEAT SPEED Switches

The latched up position of the REPT switch disables the DCD reset gate of the REPT flip-flop, drawing KC10, so that the flip-flop cannot become reset by the CLK POS pulse after RUN(1). With the REPT flip-flop always set, the REPT CLK pulses will recycle flip-flops A, B, and C to set RUN each time RUN resets. The recurrence rate of REPT CLK pulses determines the time interval from RUN reset to RUN set. The REPEAT SPEED switch determines the REPT CLK recurrence rate. The REPT and REPEAT SPEED switches are normally used in this manner with START, CONTINUE THIS, READ IN, and with built-in maintenance provisions. REPEAT SPEED is a five-position rotary switch, drawing CS-5408018, which selects various capacitors to tune the REPT CLK frequency for speeds ranging from 2 μs to 1s. The capacitor selected by position 1 is mounted externally, while all others are located within the REPT CLK module R401. Note that READ IN operations require that REPEAT SPEED be placed in position 5 (2 μs).

7.1.12 SING INST and SING STEP Switches

The latched up positions of the SING INST and SING STEP switches are normally used in conjunction with CONTINUE THIS to advance a program one instruction or one cycle at a time. If both switches are up, the SING STEP function overrides SING INST.

If the SING STEP switch is operated during a running program, the computer will halt at the end of the current cycle, drawing KC10. The CONTINUE THIS key can then be used to step the instructions one cycle at a time, with each CONTINUE THIS depression.

The CONTINUE THIS key can also be used with the REPT switch for continuous SING STEP or SING INST operations. Ordinarily, turning on the SING INST switch alone will cause the RUN flip-flop to reset upon completion of the current execute cycle. The CONTINUE THIS key would then be depressed for the execution of the next instruction. If the REPT switch is turned on (up), the REPT flip-flop remains set as RUN resets, and the REPT CLK pulses step flip-flops A, B, C to eventually set RUN. Thus, the program continues one instruction at a time, at intervals determined by the REPEAT SPEED switch. This eliminates the necessity for depressing the CONTINUE THIS key for each advance.

7.2 DISPLAY INDICATORS

7.2.1 REGISTER Indicator

The 11-position REGISTER DISPLAY switch on the console selects the contents of the following registers for display in the 18-bit REGISTER indicator.

AC contents
AR contents
MQ contents
PC, Link, memory extend mode status, memory protect mode status, EPC
RDR - reader buffer contents
TTI - keyboard buffer contents
API - on/off, interrupt request, and priority level status of eight priority levels
I/O A - 15-bit address word in DCH or API operations
I/O B - 18-bit data word on I/O bus from/to any device
DPY - 9-bits each of X, Y buffers of optional Type 34HL Display

The REGISTER DISPLAY switch and the REGISTER indicator are enabled by IND EN when the computer is in the stop condition. The IND EN level from drawing KC10 goes to the REGISTER DISPLAY switch wiper, drawing CS-5408018 via the CP/console interface, drawing KC23.

If any of the four CP registers (AC, AR, PC, MQ) were preselected, the appropriate enabling level (ACD, etc.) goes from the switch contact to the respective register gate sense flip-flop, setting the flip-flop on a CM STROBE B generated by IND CLK. IND EN also sets IO BUS ON at CM STROBE B time. The IND CLK pulses are derived from the stepping of flip-flops B and C during the RUN reset condition (computer NOP). The sense flip-flop opens the A bus to the appropriate register contents. As shown in Figure 7-3, a direct signal path to the input mixer is afforded by the A bus, ADR, and I/O bus. At the input mixer, drawing KD7, the contents on the I/O bus are inverted (buffered) at NOR gates R123, then fed from the IO/CP interface (drawings KC25, KD6) to indicator driver transistors in the console, drawing CS-5408020. The transistors supply drive current of 30 mA at -2V to the appropriate REGISTER indicator lamps (REG 00-17) for all binary 1 levels received from IO BUS00(B) - IO BUS17(B). The indicator lamps remain illuminated as long as the operator holds the computer in the stop condition.

The select levels for all other display selections go from the REGISTER DISPLAY switch to the input mixer via the CP/console (drawing KC23) and CP/IO interface (CP/H40 to IO/H01, drawing KC25). At the input mixer the select level (RDRD, etc.) passes a NOR gate in R111-D17 to produce a RDR ON BUS etc. signal which gates the selected information into the input mixer modules. The information is then buffered at IO BUS00(B) - IO BUS17(B) as for the CP registers.

Note that the I/O BUS position of the switch is actually the off position. In this position, the indicator displays whatever happens to be on the data lines of the I/O bus.

7.2.2 Link Indicator

The set side of the Link is wired through the CP/IO and IO/console interface to its indicator drive transistor in the console, then to the Link indicator lamp. When the Link sets, the indicator illuminates. Although displayed continually, the Link indication is meaningful only when the AC contents are selected in the computer stop condition. The LINK flip-flop is shown on drawing KC15.

7.2.3 MEMORY BUFFER Indicator

The MB register bits MB00-17 are wired through the CP/IO and IO/console interface to their indicator driver transistors in the console, then to the MEMORY BUFFER indicator lamps. Although displayed continually, the indication is meaningful only in the computer stop condition.

7.2.4 INSTRUCTION Indicator

The five INSTRUCTION indicator lamps continually display the contents of the IR. The first four lamps

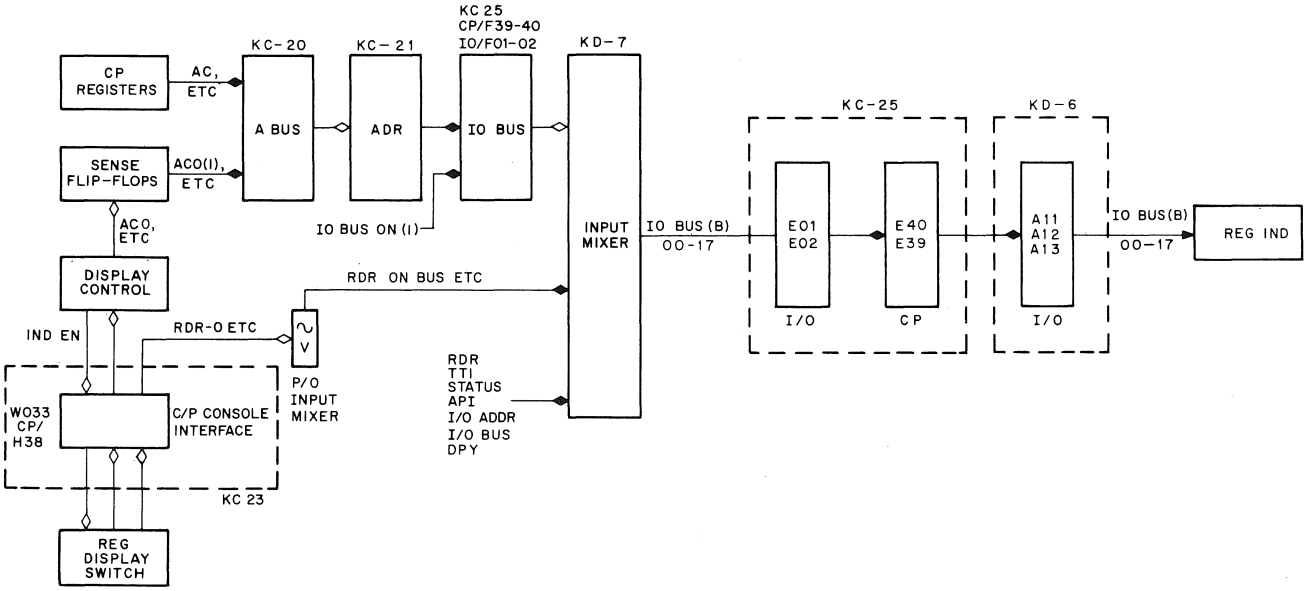


Figure 7-3 REGISTER DISPLAY Signal Paths

indicate the op code IR00-03 of the instruction being executed. The fifth lamp illuminates when the memory reference instruction contains an indirect address (IR04 = 1). The IR bits are wired to their indicator driver transistors in the console via the CP/console interface, then to the indicator lamps.

7.2.5 PIE Indicator

The PIE indicator illuminates when the program interrupt facility is enabled by an IOT0042 instruction (ION). In the I/O control logic, drawing KD3(2), the decoded instruction sets the PIE flip-flop. The set side of the flip-flop is wired to its indicator driver transistor in the console via the IO/console interface, then to the indicator lamp. The lamp remains illuminated until a program interrupt occurs or until an IOT0002 instruction (IOF) is issued. Both events reset the PIE flip-flop, disabling the facility and extinquishing the lamp.

7.2.6 CLK Indicator

The CLK indicator illuminates when an IOT0044 instruction (CLON) enables the optional real-time clock. The CLK switch must be down before the IOT instruction can enable the clock.

7.2.7 SING STEP Indicator

The SING STEP indicator illuminates when the SING STEP switch is turned on. The SW SGL STP level from the switch goes via the CP/console interface to its indicator driver transistor in the console, then to the indicator lamp.

7.2.8 SING INST Indicator

The SING INST indicator illuminates when the SING INST switch is turned on. The SW SGL INST level from the switch goes via the CP/console interface to its indicator driver transistor in the console, then to the indicator lamp.

7.2.9 REPT Indicator

The REPT indicator illuminates when the REPT flip-flop is set. From drawing KC10 the REPT(1) level goes through the CP/console interface to its indicator driver transistor in the console, then to the indicator. The REPT flip-flop is normally controlled by the program, but may be kept in the set state by the REPT switch for maintenance purposes.

7.2.10 PRGM STOP Indicator

The PRGM STOP indicator illuminates when the RUN flip-flop resets. The reset side of the RUN flip-flop is wired through the CP/console interface to its indicator driver transistor in the console, then to the indicator lamp.

7.2.11 DATA Indicator

The DATA indicator illuminates when a DCH break is in progress. From drawing KD3(1) a DCH BK DLY level goes through the I/O interface to the DATA indicator driver transistor in the console, then to the indicator.

7.3 POWER CONTROL

7.3.1 Primary Power Distribution

The PDP-9/L will accept several primary power inputs. This flexibility permits it to be operated in many power environments. Adjustment to different conditions is done by changing various input transformer taps in the power distribution system. Input power requirements are 3 KVA. For 60 Hz power sources, adjustment can be made for either 120 or 240V. For 50 Hz sources, adjustments may be made for 100V, 150V, 200V, 215V and 230V ac. Primary power distribution is shown on drawing 9-L-3. Primary power is applied to TB1 of the main Type 712 Power Supply located at the bottom right of the front panel as viewed from the rear (drawing 712-0-1).

Power is applied to the system by 40A circuit breaker CB1, located on the 712 Power Supply between the two distribution boards and accessible from the rear (panel doors are not interlocked). Closing this breaker applies power to the console and the maintenance panel and energizes the main power contactor K1, from which power is taken for the utility outlets and power supply transformers T1 and T2. Adjusted ac voltage then is distributed by auto tap to the High-Speed Punch PC09, the Memory MC71, the fans, the Central Processor KC09-C, and the I/O Control KD09-C. The elapsed time meter is tied across the high-speed punch auto-tap and indicates to the nearest tenth of an hour, the cumulative hours of system power on.

All jumper modifications are usually made by the manufacturer to suit a particular customer's needs. A quick check, however, is usually in order before applying power. There are four categories of adjustments:

a. Power contactor current limit adjustments,

b. Auto-taps,

c. Power transformer primaries, and

d. Power transformer secondaries.

These various conditions are given in Table 7-1. Check first the power contact jumpers; regardless of line frequency, when input voltage is 110V, jumpers must be connected between J-12 and J-13 and between J-14 and J-15. If the input voltage is 220V, these jumpers should be removed, allowing current limiting resistors R3 for I1 and R9 for K1 to function.

The next check is the auto-taps, Number 1 for the high-speed punch and Number 2 for the fans. If operating conditions are 50 Hz, 115V, the line should be connected between terminals 10 and 13 on both T1 and T2. With 60 Hz, 120V conditions, connect the line from pins 11-13.

The power transformer primary connections and jumpers should then be checked against Table 7-1 for the particular line voltage and frequency conditions at the operating site.

Table 7-1
Input Power Adjustments

Category	Line Frequency	Line Voltage	Jumpers	Line Connections
Power Contactor	50/60	110	J12-J13 J14-J15	Not Applicable
		220	None	
Auto Taps	50	115	Not Applicable	10-11
	60	120	Not Applicable	11-13
Transformer Primary	50	100	11-14 12-13	11-12
		115	10-12 11-15	11-12
		200	13-14	11-12
		215	10-14	11-12
		230	10-15	11-12
	60	120	11-14 12-13	11-12
		240	13-14	11-12
Transformer Secondary	50	Not Applicable	Not Applicable	50 Hz Blk.
	60	Not Applicable	Not Applicable	60 Hz Blk.

Finally, the transformer secondary connectors should be plugged into the proper block for the power frequency used. These connections are changed by removing and replacing two modular quick-disconnect fittings. This information is repeated for quick reference in Table 7-1.

7.3.2 Power Supply

The 712 Power Supply, drawing 712-0-1, is a dual 50/60 Hz unit designed for ease in conversion to any of the input and frequency combinations listed in the drawing. The user simply matches the line frequency and then chooses the input voltage value which most nearly matches his nominal line voltage. He then arranges the transformer taps to the chosen values. Output voltages and current drains are as shown.

Figure 7-4 is a functional block diagram of the 712 Power Supply. Its purpose is to provide the -30 Vdc required by the high-speed punch and the memory; and also to supply the logic dc voltages +15 and -10. The 712 Power Supply is made up of two groups of heavy-duty power supply modules. One group is fed by T1 the other by T2. The first group comprise a fixed -15 Vdc, a fixed +10 Vdc, and an ungrounded supply made variable between 0 and approximately 20 Vdc. The second group contains a +10, a -15, and a -30 Vdc supply. The -30 Vdc supply feeds the punch directly, and the memory through operation of memory voltage relay K2. Operation of K2 is from a delayed ground from the memory control module to -15V in this supply.

The marginal check supplies operate in conjunction with the MC switch S1 to vary the voltages to the logic, thereby determining the life expectancy of selected logic modules. This becomes part of the preventive maintenance schedule for the computer. In use, certain sets of logic modules are selected by marginal check switches located within the rack. The marginal check switch on the maintenance panel is then set to either + MC or - MC position and a selected diagnostic routine is run. During the routine, the variable supply is adjusted until the computer fails. At this point the voltage, as indicated by the marginal check voltmeter M2, is recorded for future reference. The same procedure is followed for the opposite polarity. These periodically recorded failure points are then used to forecast impending problems in modules.

The function of the marginal check switch S1 is illustrated in Figure 7-5. In OFF position (Figure 7-5a), the meter M1 is disconnected and fixed positive and negative voltages are sent to the logic modules. Note also that a ground is placed on the output of the variable supply. In the + MC position (Figure 7-5b) the positive voltage to the logic is varied while the negative voltage is fixed. In - MC position (Figure 7-5c) the negative voltage is varied while the positive voltage is fixed. In either MC position the buckoff meter

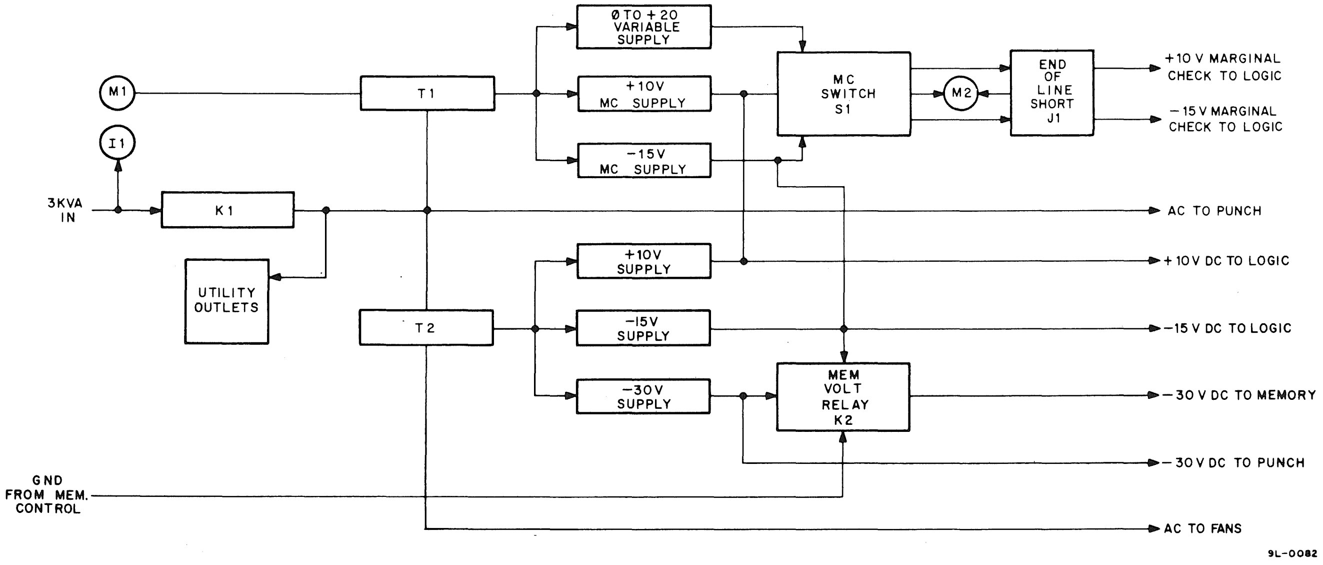
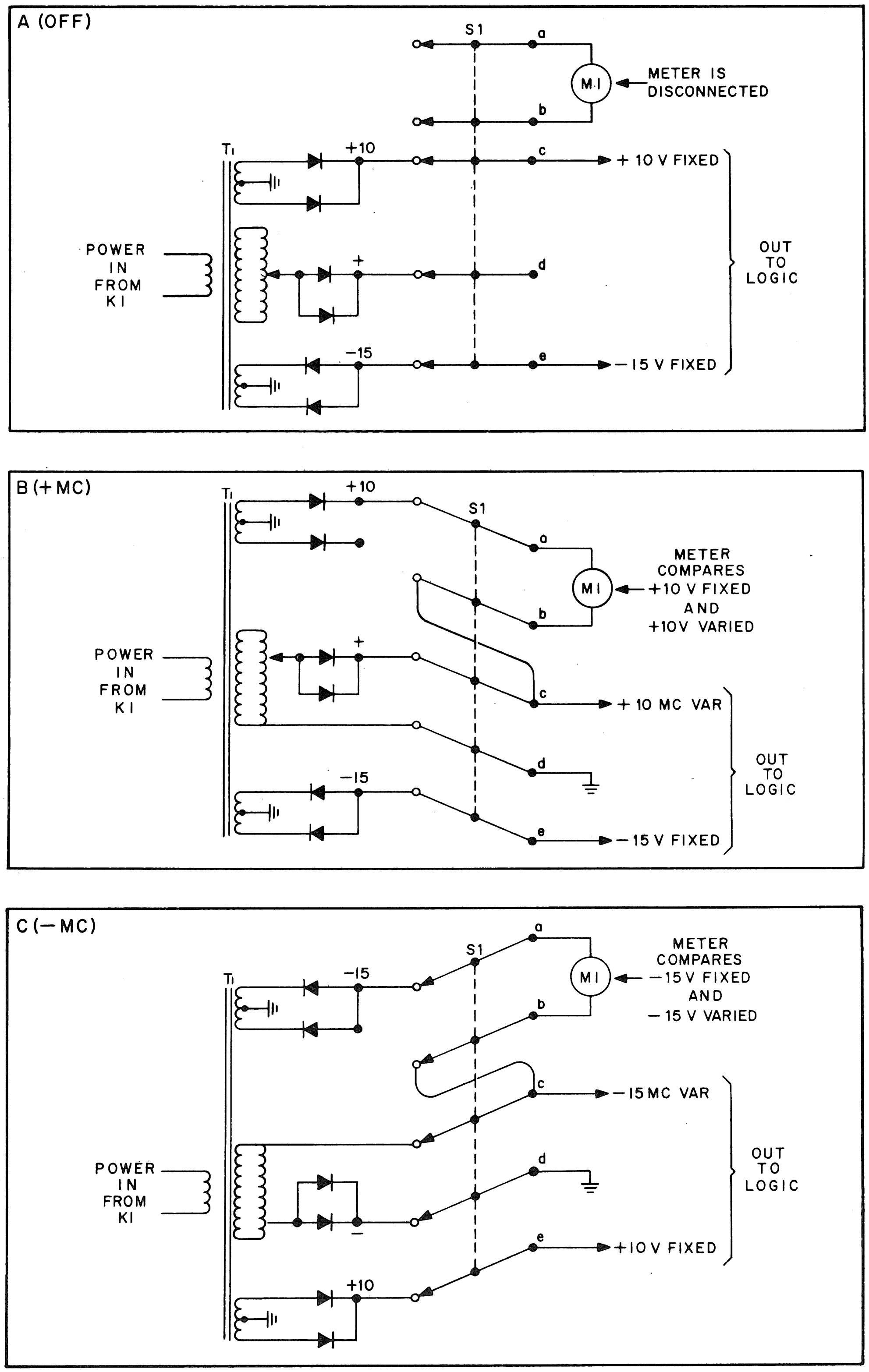


Figure 7-4 Power Supply 712, Block Diagram



9L-0083

Figure 7-5 Marginal Check Switch Positions, Simplified Schematic

compares the variable voltage against the fixed supply of the same polarity and is an indication of voltage above or below the fixed value.

For example, with the MC switch in +10 MC position: if the meter reads 8 divisions to the right of center scale, the actual MC voltage being sent to logic is +18 Vdc, whereas if the meter reads 3 divisions to the left of center scale, the actual voltage is +7 Vdc. The same would be true if the MC switch were in - MC position except that a reading 5 positions to the right would indicate a voltage of -20 Vdc.

CAUTION

> The -15V margin should not be varied beyond ±5V. Doing so may damage logic elements.

The ac output at J10 and J11, marked CONSOLE SW, supplies the main ac input to the console assembly (see drawing 9-L-3). This output also is sent to the maintenance panel where it completes through the MAINT switch in LOCK position.

The maintenance switch S2 is shown in greater detail in drawing 712-0-1(2). Those console functions which are necessary for maintenance are completed through this switch when it is in NORMAL position. Placing this switch in LOCK position disables the console controls and prevents any accidental interference with the diagnostic program. The MAINT, EXAMINE and DEPOSIT positions interact with the repeat speed control and REPT switch on the console which in turn determine the rapidity at which each function is cycled. The MAINT position initializes a built-in test program which circulates a self-incrementing count through all active CPU registers. The EXAMINE position simulates a repetition of the EXAMINE THIS function, and the DEPOSIT position similarly simulates the DEPOSIT THIS function.

CHAPTER 8
INPUT/OUTPUT CONTROL

The PDP-9/L input/output control has two modes of operations - program controlled transfer and data channel transfer.

Program controlled transfers occur as the result of IOT (input/output transfer) instruction execution. These instructions, contained in the body of a main program or in appropriate subroutines, are microcoded to effect response of a specific device interfaced to the I/O bus system. The microcoding includes issuing of a unique device selection code and appropriate processor-generated pulses to initiate device operations, such as taking data from the bus or placing data on the bus or clearing device flags. All program controlled transfers are executed through the accumulator in parallel bytes of up to 18 bits in length.

The data channel facility provides for relatively high speed transfers of data in blocks between peripherals (DECtape, magnetic tape, etc.) and system core memory. The transfers are controlled by word counter registers and current address registers contained in memory. Eight pairs of these registers are provided to control non-overlapping data channel transfers to or from as many as eight devices. A data channel transfer request "breaks" program control at completion of the current instruction and suspends execution of the program in progress until the current word transfer is completed (three machine cycles for input to memory, four cycles for output to device). Successive breaks are granted to either the active device or another data channel device if the request for service is made prior to the completion of the current channel transfer action. The maximum transfer rate for the facility is between 160,000 and 220,000 words per second, depending on the mix of input and output transfers.

All I/O data transfers function within the precedence of the following priority structure.

a. Data channel (DCH) requests

b. Real-time clock counting (optional).

c. Automatic priority interrupts (API), 8 levels (optional)

d. Program interrupts (PI)

e. Main program in progress (lowest priority)

A higher priority request for service interrupts any in-process service of a lower priority at the end of the current instruction. Program interrupts and priority interrupts require that the main program transfer control to specific service subroutines. These routines must be programmed to restore control to the interrupted program at completion of the service interval. Computer-granted breaks satisfy data channel requests; i.e., program execution is delayed but not disturbed while the data channel transfers information between memory and the requesting device via the MB.

The I/O control section offers flexibility for increasing the capability and efficiency of the PDP-9/L system. The I/O bus structure is prewired to accommodate certain common options without extensive modifications in the CP. The control logic for the options is distributed within the I/O control section and within the options themselves. Thus, no extensive wiring changes or logic modifications are necessary when installing optional devices. Such additions merely involve connections between the options and the I/O bus, and installation of control logic modules at the prewired frame locations. Physical location requirements of the options themselves depend on their nature and on the available space. Refer to the PDP-9/L User Handbook, Document No. DEC-9L-GRVA-D for descriptions of the options available with the PDP-9/L.

The interface logic installed in each device is unique to the device's data transfer rate and to its relative importance in the program. Slow-speed devices (up to 33,000 wps) are normally connected to the I/O bus for program-controlled transfer operations. High-speed devices (up to 218,000 wps) are normally connected to the I/O bus for data channel (DCH) transfer operations. High-speed and critical-use devices may also be connected to the I/O bus in the optional automatic priority interrupt (API) system. The devices connected into the DCH and optional API channels are multiplexed for servicing from the same bus connections.

8.1 PROGRAM-CONTROLLED DATA TRANSFERS

I/O devices interfaced for program-controlled data transfers are under the control of the computer's input/

output transfer (IOT) instructions. The IOT instructions contain microcoded bits for device selection, data transfer and transfer direction commands, program interrupt selection, I/O skipping, and device status checking. Data transfer takes place in bytes of up to 18 bits between the selected device and the AC or AR register of the central processor.

All program-controlled devices are chain-connected in parallel at the I/O bus. The number of devices is limited only by the 6-bit 64 device-select code 06-11 in the IOT instruction, and by signal strength versus cable length considerations. Two additional IOT instruction bits 12-13 are used to select special operating modes of a device. I/O skip operations and transfer commands are microcoded in the remaining bits, 14-17.

Six of the 64 device-selected codes are allocated to the basic PDP-9/L paper tape reader, paper tape punch, Teletype, program interrupt and I/O status check. Data transfers to and from the standard devices take place via the input mixer rather than the I/O bus.

8.1.1 I/O Bus Connections

The I/O bus for program-controlled data transfers terminates at module connector receptacles H850 at AB25 and AB26, as shown on drawing KD2(1). Connections are made from AB25/26 to the first device by one set of 36-pair, Type W031 Ribbon Cables, and parallel chain connections are made between successive devices. The cabling terminates in Type W850 Male Connectors at each end, mating with H850 receptacles at the I/O bus and at the devices. In some cases, one end of the cable may be free to attach to device terminal strips or other special connections. One wire in each pair is grounded and common-connected to the ground mesh of the devices. The connections at AB25 and AB26 are also wired to H850 receptacles at AB29 and AB30 for use as spares, should the chain of peripherals exceed loading and driving considerations. All signals appearing on the I/O bus are driven by B213 Bus Drivers. These are adequate to drive the signals without appreciable decay in chain-connected cabling up to 50 ft in length. In certain applications, the spare connectors may be used for a secondary chain of peripherals where the primary chain would otherwise exceed the 50-ft limitation. For example, devices using the program interrupt facility could be grouped in the secondary chain without affecting other devices in the primary.

8.1.2 Block Diagram Discussion

Figure 8-1 shows the interface for program-controlled data transfers and Figure 8-2 shows the typical logic contained in a device.

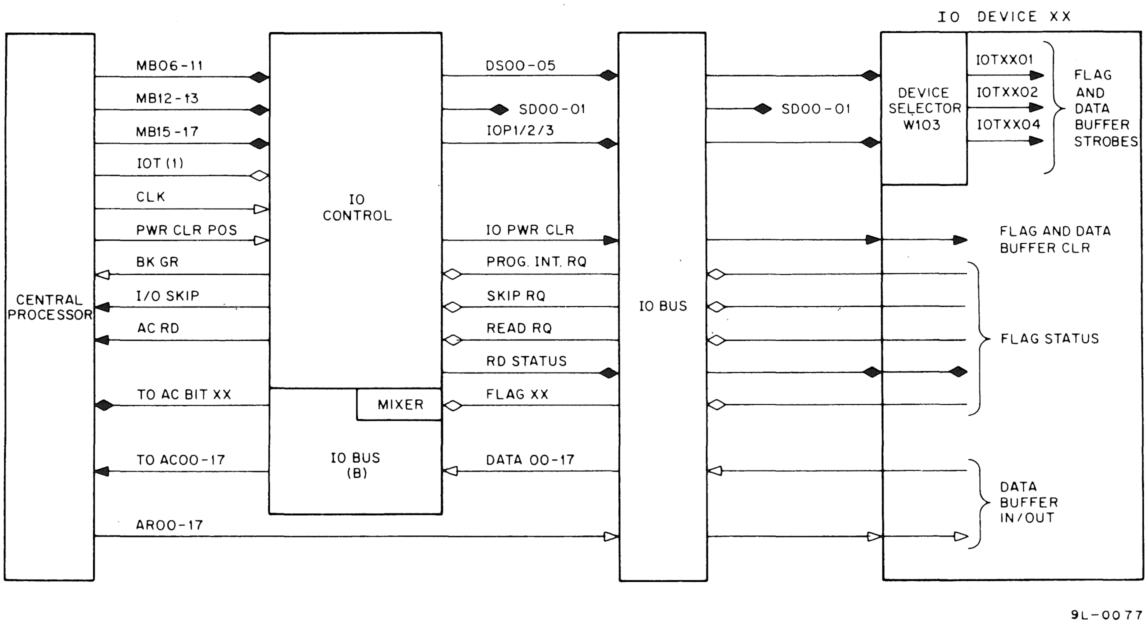


Figure 8-1 Program-Controlled I/O Interface

To effect an output transfer, the program first issues an IOT skip instruction which senses the selected device's data flag. The program repeatedly returns to this IOT instruction by means of a JMP until the device is ready to accept data. When ready, the device flag allows a skip to occur when the flag is sensed by the CP. The skip request is processed to set the SKIP flip-flop in the CP, whereupon the program skips the JMP instruction and goes to a subroutine for the output transfer. The subroutine contains a LAC instruction which loads the data word from core memory into the AC and the AR; an IOT output transfer

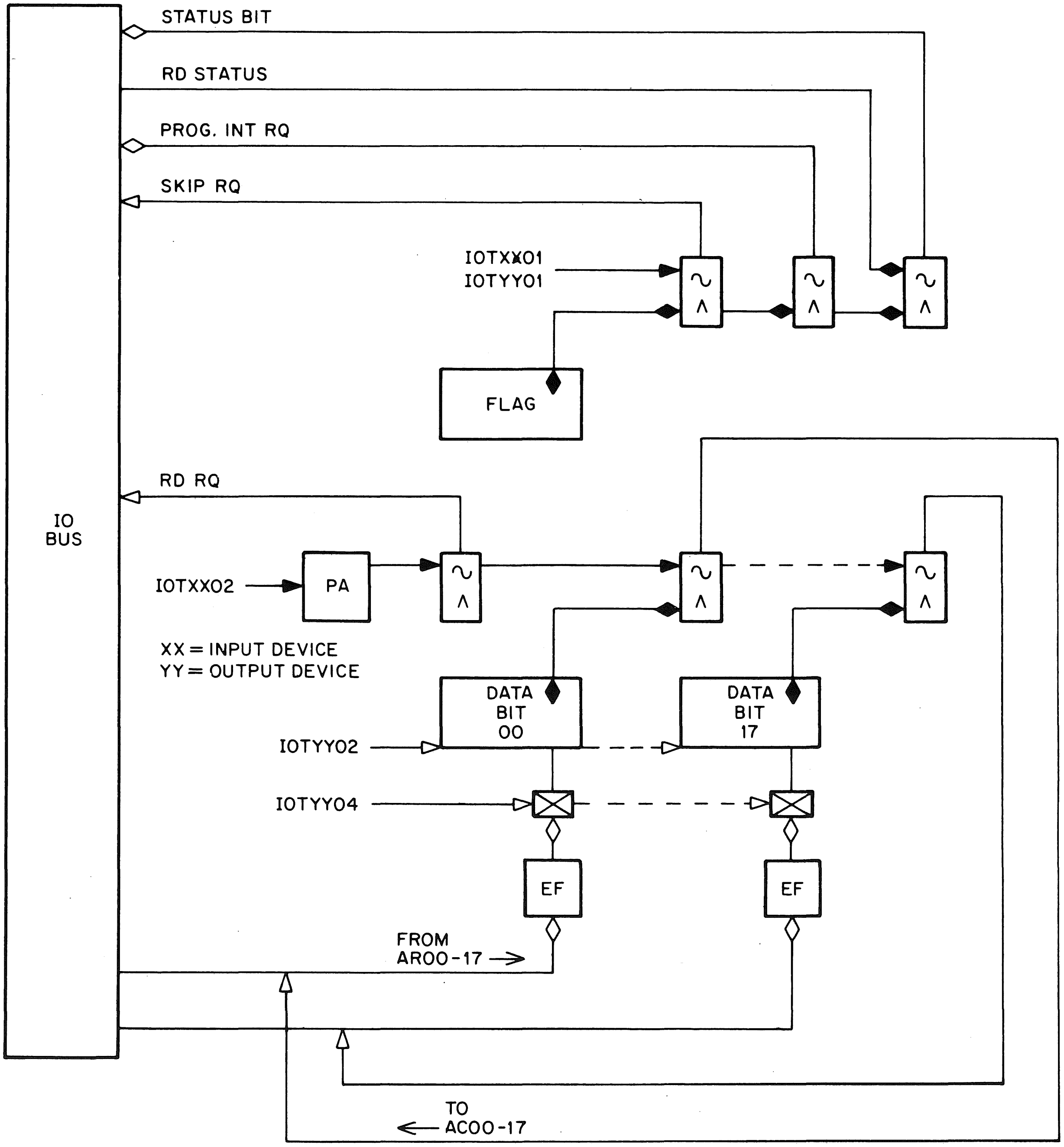


Figure 8-2 I/O Device Control Logic

instruction gates the data from the AR onto the I/O bus via the A bus and ADR, and strobes it into the device's data buffer.

To effect an input data transfer the process is reversed. When the IOT skip instruction senses that the device is ready, the program skips to an IOT read instruction. The IOT read instruction strobes the device's data buffer contents onto the I/O bus and generates a read request pulse in the device which goes to the CP as the AC RD pulse. AC RD gates the contents of the I/O bus into the AC via the buffered I/O bus (B) and the O bus.

Each device contains a device selector such as the W103 which responds only to its assigned code in the IOT instruction, bits 06-11. In addition to the select code, the device selector receives sequential IOP levels from an IOP generator in the I/O control section. These levels are derived in the generator from IOT instruction bits 15-17, and are gated along with the device select code to produce sequential IOT levels out of the device selector. The IOP/IOT levels occur at 1.5 μs intervals following the 1.5 μs computer fetch cycle which fetched the IOT instruction. The instruction may call for any combination of up to three IOP/IOT sequential levels, depending on the situation. Thus the total IOT instruction time consists of a 1.5 μs fetch cycle and an extended execute period of 4.5 μs. Normally, IOP1/IOTXX01, derived from MB17(1), is generated by an IOT skip instruction to test a device flag. IOP2/IOTXX02, from MB16(1), is generated by an IOT read instruction.

IOP4/IOTXX04, from MB15, is generated by an IOT load, write, etc., instruction. Certain combinations may be encoded in a single IOT instruction, such as using IOTXX02 to clear a data buffer and IOTXX04 to load new data into the buffer during the next 1.5-μs execute period.

Since several devices may be connected to the I/O bus as program-controlled transfer devices, periodic IOT skip and jump loops to test the ready status of each device would involve considerable delay of the main program. To alleviate this need for prolonged flag sensing, one of two facilities may be used: the program interrupt (PI) facility or the status checking (IORS) facility.

To use the PI facility, the program issues an IOT ION (interrupt on) instruction. This instruction enables program interrupt operation of all peripheral devices connected for program-controlled transfers, allowing the computer to continue with the main program.

When any device is ready for data servicing, it raises its flag, which sends a program interrupt request to the CP. Upon receipt of this request, the program completes its current instruction, stores the contents of the PC and certain program status information (Link, memory extend mode, memory protect mode, EPC) in core memory location 00000, then enters a flag search routine from location 00001 to find and service the device that requested the interrupt. Upon completion of the servicing routine, the computer returns to the interrupted program stored at 00000 by means of JMP *. If two or more devices request a program interrupt simultaneously, the computer honors the first flag sensed in the I/O search routine.

The status of 18 device flags and control flip-flops of certain commonly-used optional peripherals can be read simultaneously onto the 18 data lines of I/O bus (B) by an IOT IORS (input/output read status) instruction. This instruction is useful when it is desired to test the status of any of these peripherals on an individual basis. The IORS instruction gates the status bits onto I/O bus (B) via the input mixer, then generates AC RD to gate them from I/O bus (B) into the AC via the O bus. Rotate instructions (OPR RAL, RTL) may then be used to shift the applicable bit into AC00 or into the Link. Other OPR instructions (SMA, SZL) will sense the shifted status bit for conditional branching into service routines

8.1.3 Device Selector W103

The double-height W103 Device Selector selects an I/O device by decoding bits 06-11 in the IOT instruction held in the MB. All W103 modules are fabricated uniformly with 14-input diode gating, and output gating for the production of IOTXXXX pulses. Figure 8-3 shows the module logic. The 14 input diodes permit selection of any arbitrary code at installation time. The R107 inverters are used to arrange the six device-select levels DS0-5 derived from MB06-11 so as to present six negative enabling levels to the input diode gates. The remaining input diodes are disconnected internally or externally at the time of installation in the I/O device. The input IOP1, IOP2, IOP4 levels complete the gating to the output pulse amplifiers which then issue the sequential IOTXX01, IOTXX02, IOTXX04 pulses.

8.1.4 Input Transfers

Input transfer of a device data buffer is normally accomplished by an IOTXX12 instruction. During the instruction fetch cycle the op code detection circuits in the CP generate an IOT(1) ground level. IOT(1) goes to the I/O control logic, drawing KD3(1) where it becomes IOT(B) at S107-F22D. IOT(B) gates the select code bits MB06-11 into bus drivers B213 to produce the select levels DS0-5. These enable the input gates at device selector XX and the gates condition the DCD inputs to the output pulse amplifiers, which now await the commanded IOP levels (in this case IOP2 only) for triggering.

The IOP generator circuits are synchronized with the CLK POS pulses from the clock and run logic, drawing KC10. On drawing KD3(2), the CLK POS pulses generate IO CLK POS pulses at S603-J10M when an IOT instruction is detected. IO CLK POS pulses strobe the Gray code pulse counter IO0, IO1 on drawing KD3(3). When an IOT instruction is detected, IOT(B) and the next IO CLK(B) pulse set IO0 for a count of 10. IO CLK(B) pulses are derived on drawing KD3(3) from IO CLK POS pulses at S107-J21D.

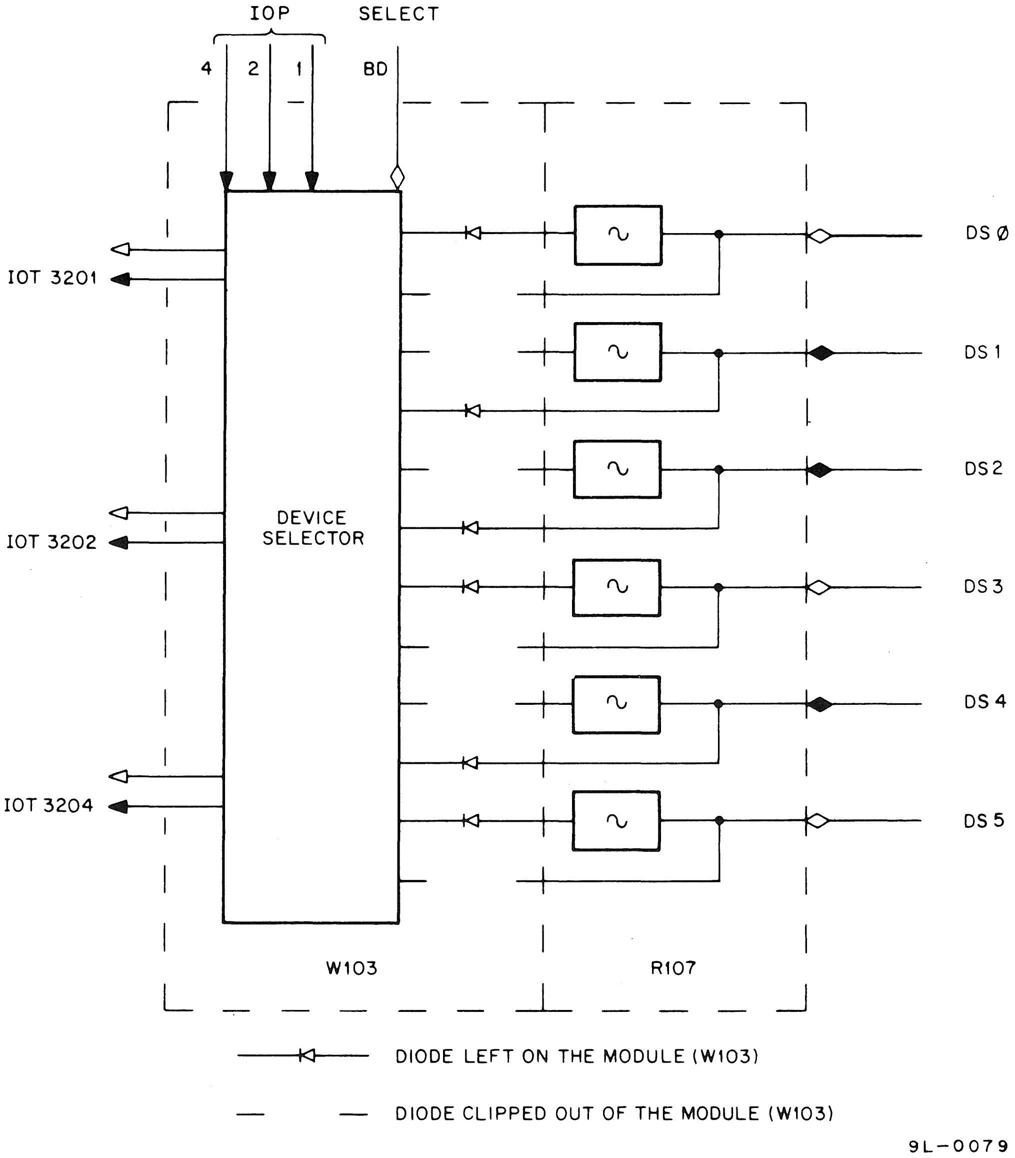


Figure 8-3 Device Selector W103, Schematic Diagram

This is the first CLK pulse after IOT(1); i.e., the beginning of the IOT execute period. IO0(1) strobes the IOP1P gate on drawing KD3(3), but the gate is disabled for input transfers because MB17(1) is absent.

The next IO CLK POS pulse sets IO1 because of IO0 (1). IO0 remains set, so that the count is now 11; IO1(1) generates IOP2P in conjunction with MB16(1). IOP2P sets the IOP2 flip-flop. This set flip-flop gates on the IOTXX02 pulse in the device selector.

IOTXX02 generates the RD RQ in the device and gates the device's data buffer contents onto the I/O bus. The contents on the I/O bus are ORed onto the I/O bus (B) at the input mixer logic, drawing KD7. Here the IOT(B) level holds RD IO BUS at ground. RD IO BUS in turn keeps the I/O bus contents on I/O bus B independently of the IOP2P pulse.

The RD RQ pulse enters the I/O control logic, drawing KD3(3), from the I/O bus. Here RD RQ is NORed at R111-F19V, then NANDed with IO1(1) and CLK DYL'D at R002-F20N for AC RD, 400 ns after IO CLK POS triggers delay B301-H22D.

AC RD goes to the CM sense flip-flops, drawing KC 19(2) where it becomes AC RD(B) and produces the 1 → ACI pulse. AC RD(B) goes to drawing KC13 to produce LIO. LIO gates the contents of I/O bus (B) onto the O bus, drawing KC20. 1 → ACI sets the ACI flip-flop. ACI(1) gates the contents on the O bus into the AC. AC RD returning to ground resets ACI.

The AC now contains the data word read in from the selected device. The next IO CLK POS pulse resets IO0 as CLK starts the third IOT execute period. The IO CLK POS pulses continuously trigger the 400-ns delay in B301-H22. With the count at 01, IO0(0) and IO1(1) condition an IO RESTART input DCD gate and the DLY upon recovery strobes the gate. This same DLY resets IOP4. IO RESTART triggers the CM timing to extract the next process word (20) in the execute period. The BGN process word (10) follows to end the IOT execute period and to prepare the computer for the next fetch cycle. The IO CLK POS pulse concurring with the fetch cycle start resets IO1 for a return to the 00 count.

8.1.5 Output Transfers

Readout to a device data buffer is normally accomplished by an IOTYY06 instruction. In this case, octal code 6 in the instruction orders the generation of two IOP pulses in sequence, IOP2 and IOP4, producing IOTYY02 and IOTYY04 in the device selector YY.

IOTYY02 normally clears the device data buffer and IOTYY04 subsequently gates in the new data from the AR. Note that in Figure 8-2 emitter followers are used at the input DCD gates to isolate the load of the gates from the I/O bus.

As for input transfers, the IOT instruction is detected to start the IOP generation and to select the appropriate device. But MB14 is 0. MB14(0) sets the ARO and IO BUS ON flip-flops by generating an IOT OR ARO signal drawing KC12. ARO (1) gates the contents of the AR onto the A bus, the A bus contents go through the ADR, and IO BUS ON puts them on the I/O bus. Thus the data is ready for loading into the device buffer. The IO0, IO1 pulse counter steps as for input transfers to generate the IOP2, IOP4 pulses. These pulses set the IOP2, IOP4 flip-flops in turn to produce IOTYY02, IOTYY04, clearing the buffer and loading the data.

8.1.6 I/O Skip Facility

Before transferring data into or out of a device, the device's data flag is usually tested for its ready status. IOTXX01 instructions generate an IOTXX01 pulse in the device selector, which is then NAND gated with the output of the device flag flip-flop. If the device is ready for a transfer, the flag is in the set state and the output gate enables to send a SKIP RQ to the I/O control logic via the I/O bus. The SKIP RQ pulse is NANDed with IO0(1) and CLK DLY'D, drawing KD3 (3), to produce the IO SKIP pulse 400 ns after IO CLK POS. IO SKIP sets the SKIP flip-flop in the CP, drawing KC14. Upon completion of the IOT skip instruction, SKIP(1) and PCO(1) of the BGN process word produce the CI17 level on drawing KC14. CI17 increments the ADR as the current address in the PC is brought through the ADR to the MB for the next instruction fetch cycle. Thus the current address is incremented by 1 so that the computer skips one instruction. The instruction then reached usually starts an input/output service routine which includes an instruction to reset the data flag.

8.1.7 Program Interrupt Facility

When computer time is at a premium, it is advantageous to have the computer perform other tasks rather than remain in an I/O skip and jump loop waiting for a device flag. The program interrupt (PI) facility allows the program to ignore devices until one of the devices signals that it has completed its previous operation and is ready for another. When such a request is present, the computer completes the current instruction, then stores the conditions of the main program, disables the PI facility to future interrupt requests, and enters a PI break subroutine which contains an I/O skip and jump instruction for each device connected to the PI facility. The I/O skip instructions sense the flags of each device in turn until the subroutine finds the device that requested the interrupt. The subroutine then skips to a service routine to service that device. The service routine for any device usually resets the device flag and consequently removes its interrupt request signal. Moreover, since a program interrupt disables the PI facility, the service routine must conclude with an IOT ION instruction, which again enables the facility for use by other devices requesting interrupts, and an IOT DBR instruction, which restores the conditions of the main program. The program can enable and disable the facility in accordance with its needs, by means of the two IOT instructions:

ION	700042	Interrupt ON
IOF	700002	Interrupt OFF

The PI facility has the lowest priority among all program break segments (DCH, RTC, API).

The interrupt enable logic is contained in the I/O control, drawing KD3. IOT00042 is detected during fetch to start the IOP generation. In this case, device select bits MB06-11 are 00, and the subdevice select bit MB12 is 1, as indicated by octal code 4 in the IOT instruction. The IOT(B) level derived from IOT(1) is NANDed with MB12(1) on drawing KD3(1) to produce the positive SDOP level. SDOP conditions the DCD set gate to the PIE flip-flop, drawing KD3 (2).

Since MB06-11 are 00, DS0-5 are at ground on drawing KD3(1). These ground levels and IOT(1) produce the negative 0XEN and 00EN(B) levels. The 00EN(B) level waits for the IO0, IO1 pulse counter and MB16(1) to set the IOP2 flip-flop. IOP2(1) and 00EN(B) produce a positive IOT0002 pulse. The leading edge of IOT0002 strobes the conditioned PIE set gate, setting the flip-flop.

PIE(1) removes the ground PIE(0) level from the NOR gate at R111-J12U, drawing KD3(2). At this gate any one of four conditions can hold the PROG SY flip-flop S203-H07 in the reset state via collector pulling: PIE(0), CLK SYNC(1) DCH SYNC SAVE(1), and PI DISABLE. CLK SYNC(1) is present during a program break caused by the optional real-time clock; DCH SYNC SAVE(1) is present during a DCH program break, PI DISABLE during an API program break, and PIE(0) when the PI facility is disabled. PIE(1) removes the PROG SY reset hold only if the other three conditions are absent. A PI break cannot occur, therefore, if another program break has already been initiated by one of the above conditions.

After setting PIE on IOT0002, the IOT ION instruction idles through its remaining execute period and the program continues. A PROG INT RQ thereafter from any device conditions the DCD set gate of the PROG SY flip-flop. Provided that the asserted PIE and negated conditions above remove the collector ground, PROG SY will set on the next permissible IO SYNC POS pulse. IO SYNC POS pulses can only occur on IO CLK(B) pulses under the above stated circumstances, plus one other necessary condition. At the IO CLK(B) gating at R111-J12H, IO SYNC POS can develop only on an IO CLK(B) which occurs at some time other than during an IOT execute period (IOT(0)). If a PROG INT RQ comes in at the middle of the extended IOT execute period, it must wait until the entire instruction is completed for an IO SYNC POS pulse.

The PRE API SYNC(0) level that conditions the IO SYNC POS input DCD gate is available from the API option when no API break is pending or after a DCH break request has been accepted. This input is necessary only to delay the start of an API break following a DCH break. The input at this point is jumpered to ground if the API option is not installed.

In Figure 8-4, IO SYNC POS and a PROG INT RQ set PROG SY at the start of an instruction execute cycle other than an IOT execute. For IOT instructions, PROG SY sets at the start of the next instruction fetch. PROG SY(1) is ANDed at R111-F10J on drawing KD3(2), generating a negative BK SYNC level. This level on drawing KC17 waits for the DONE(1) bit in the execute process word of the current instruction, at which time the ODD ADDR level turns on.

The current execute process word containing DONE(1) also contains CMA10 (BGN). ODD ADDR boosts this address to 11, so that the last process word in the current instruction is the BK entry word rather than BGN.

Whereas the BGN word deposits the next sequential address from the PC into the MB for the subsequent fetch cycle, the BK entry word deposits address 00000 in the MB and commands a subsequent IA0 cycle. BK entry contains the processes EXT, IRI, SM, and CMA30. EXT(1) sets the BK flip-flop on KD3(2), produces LIO on drawing KC13, and IO ADDR ON BUS, drawing KD7(1). IRI(1) strobes the IR register input gates, drawing KC12. Since the IR gates contain nothing at this time, the IRI(1) bit effectively clears the IR register. IO ADDR ON BUS gates the IO ADDR bits from the I/O bus into the input mixer, where they appear at I/O bus (B). LIO gates the contents of I/O bus (B) onto the O bus. Since I/O bus (B) also contains nothing at this time, LIO puts 0s on the O bus. EXT(1) on drawing KC19(2) produces 1 → MBI in conjunction with SM(1) of the BK entry word, and CLK. 1 → MBI sets the MBI flip-flop, and MBI(1) gates the 0s from the O bus into the MB. BK(1) sets a second control flip-flop, PROG SYNC S202-H08, on drawing KD3(2).

At the CM address gates on drawing KC17, PROG SY(1)B and EXT(1) boost the next CM address from 30 to 32 (IA0 entry). SM(1) of the BK entry word and the next CM CLK pulse will start the IA0 cycle, extracting process word 32 from control memory as CLK initiates the core memory cycle.

The IA0 entry word, and in fact the entire IA0 cycle, executes a pseudo-CAL instruction, going from process word 32 to 23 to 60 to 10, since the IR register contains op code 00. The only difference is that for PI breaks the exit conditions of the main program are stored in location 00000 rather than 00020, and the program takes its next instruction from 00001 rather than 00021. This results because address 20 is not produced.

Returning to the I/O control logic, the CLK pulse that initiated the IA0 cycle develops another IO

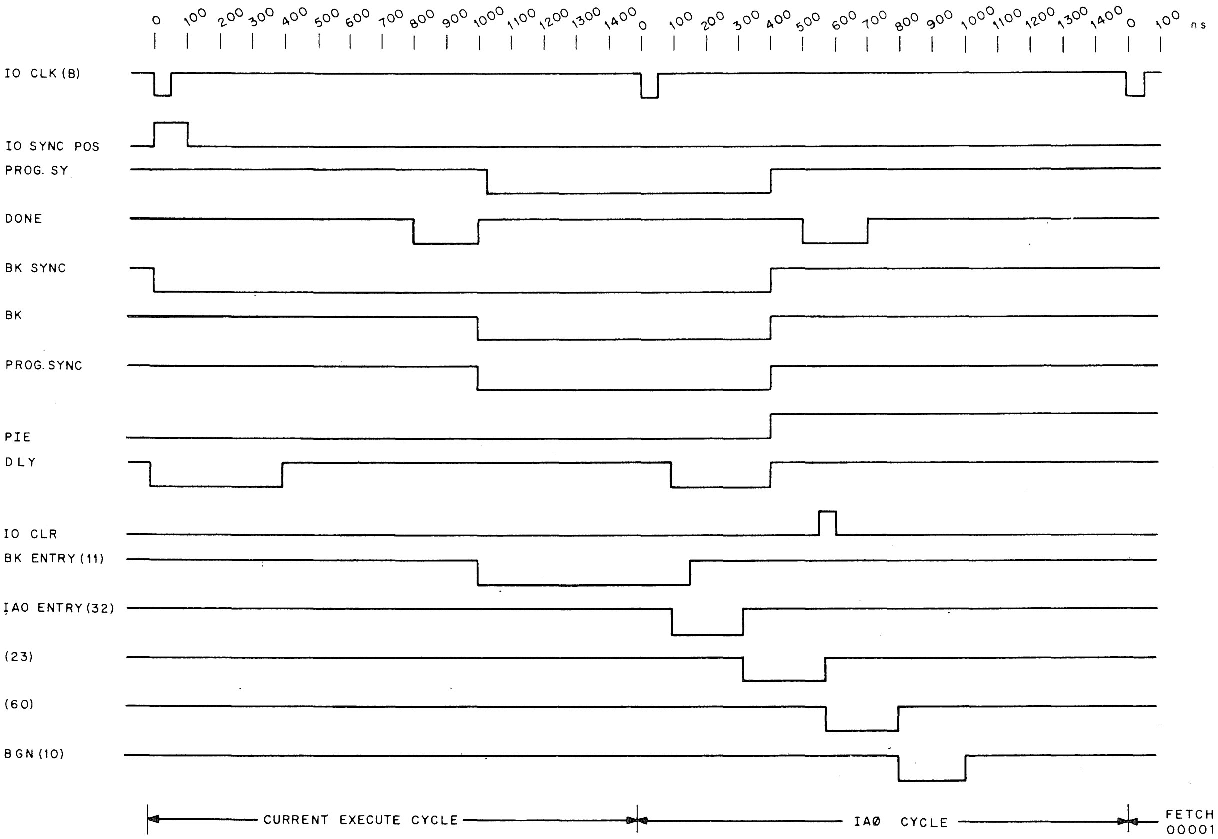


Figure 8-4 Program Interrupt Timing

CLK(B) pulse which appears at the IO SYNC POS gate and which triggers the B301 Delay Multivibrator on drawing KD3(3). At the IO SYNC POS gate IO CLK(B) cannot generate another IO SYNC POS pulse because of PROG SY(0) at ground (PROG SY is still set).

The B301 Delay Multivibrator recovers 400 ns after IO CLK(B) so that the DLY output goes to ground ($\overline{\text{DLY}}$). $\overline{\text{DLY}}$ and BK(1) trigger pulse amplifier S602-H11U for IO CLR, drawing KD3(2). IO CLR resets PROG SY, PROG SYNC, and BK. The PROG SYNC flip-flop, upon resetting, triggers pulse amplifier S602-J09K to generate a ground going pulse. This pulse pulls the collector of the PIE flip-flop to ground, resetting the flip-flop. PIE(0) from the reset flip-flop holds the PROG SY flip-flop in the reset state, so that future PROG INT RQs cannot cause a program interrupt unless an ION instruction has been issued.

The next computer CLK pulse starts a fetch cycle to fetch the instruction word located at 00001. This is usually a JMP to the flag search subroutine which finds and services the device that requested the interrupt.

The service routine includes an IOT instruction which resets the device flag as it performs the data transfer(s). This removes the PROG INT RQ from the common I/O bus line, to allow other devices to use the line. The service routine must also include an IOT ION instruction to re-enable the PI facility for the other devices, since PIE was reset. The ION instruction is normally programmed just before the service routine exit.

To exit from the service routine and return to the main program, provisions must also be made to restore the status of the Link, memory EXD mode, memory PRTCT mode, and EPC. An IOT DBR instruction (703344) issued before the JMP * 00000 that returns to the main program will restore the status.

The typical exit sequence, therefore, is:

```
ION         /RE-ENABLE PI
DBR         /PRIME SYSTEM TO RESTORE
            /PC,L,EXD,PRTCT,EPC STATUS
JMP * 00000     /RESTORE INTERRUPTED STATUS
```

On drawing KD3(1) the DBR instruction (703344) is decoded to produce an IOP4 and, consequently, an IOT3344 pulse at R111-F10UV. IOT3344 sets the DB RESTORE flip-flop S202-H10. DB RESTORE(1) then conditions the DCD set gate to the second flip-flop in S202-H10.

The next computer cycle fetches the JMP * 00000 instruction from memory and places it in the MB. The op code detection circuits recognize the indirect address, and the computer therefore goes into a defer cycle to fetch the contents (effective address) of location 00000. The interrupted PC count, the Link status, memory extend mode status, memory protect mode status, and extended program count comprise the contents of location 00000, now read into the MB by the defer entry process word (31).

Process word 31 also detects the JMP op code to produce REP, in which case the JMP execute process 74 follows. DEI going to 0 at the start of process 74 sets the second flip-flop in S202-H10. This flip-flop then resets DB RESTORE and produces a DBR pulse at W612-E16D, which goes through the CP-I/O interface to drawing KC15. Here DBR is NANDed with MB00 at R111-D05H. If MB00 is 1, indicating that the stored Link status was 1, a ground level A BUS LINK results. A BUS LINK causes the ADRL to go negative.

Process word 74 takes the contents of the MB and gates the address portion into the PC via the B bus, ADR, and O bus. As this occurs, the ADRL is strobed into LAR by LI(1) of the process. Process word 10 (BGN) follows, during which the restored PC contents are gated back into the MB via the A bus, ADR, and O bus, and the LAR state is strobed into the Link by LI(0). The interrupted Link status is thus restored to the Link, the interrupted program address is restored to the PC and MB, and the computer will now resume the program on the coming fetch cycle.

For EXD mode, PRTCT mode, and EPC status restoration, refer to the respective option manuals.

The IOT IOF instruction may be programmed to disable the PI facility as necessary. In this case, IOT0002 produced as for IOT ION resets the PIE flip-flop via its DCD reset gate because of the detected SD0 ground level.

8.1.8 I/O Status Check Facility

The IOT IORS instruction (700314) loads the AC with a word comprising the status of various device flags and control flip-flops. IOT0314 is detected during fetch to select the facility and to start the IOP generation. In this case device 03 is the ASR33 Teletype keyboard, MB14(1) puts 0s in the AC, and MB15(1) generates IOP4 to read the status bits from the I/O bus into the AC. This differs from the normal scheme where IOP2 is used for input transfers. Of the 18 possible status bits thus deposited in the AC for further examination, 11 have been preassigned as listed in Table 8-1.

On drawing KD3(1), DS0-2 are sampled by IOT(1) from the CP detection circuits to produce 0XEN. The 0XEN level is further gated with DS03P, DS04, DS05 on drawing KD11(1), turning on the KBD SEL level.

The pulse counter flip-flops IO0 and IO1 on KD3(3) produce IOP4P on the fourth IO CLK POS pulse, and IOP4P sets the RD STATUS flip-flop on KD11(1). RD STATUS(1) generates STATUS ON BUS at the input mixer, drawing KD7(1). STATUS ON BUS becomes RD STATUS at bus driver B213-F07, drawing KD11(1). The RD STATUS level goes through the I/O bus to the preassigned device flag and control flip-flop output gates, placing their status on the bus. STATUS ON BUS gates the status bits into the input mixer diode gates, whose outputs are NORed onto I/O bus (B).

RD STATUS(1) also appears at diode mixer R141-F24D, drawing KD3(3), to produce the INT RD RQ BUS signal. INT RD RQ BUS is NORed at R111-F19UV for RD RQ(B). 400 ns after the IO CLK POS pulse produces IOP4 with IO1(1), the RD RQ(B), IO1(1), and CLK DLY'D signals generate AC RD at W612-F18N. AC RD(B) turns on LIO, drawing KC13, and 1 →ACI, drawing 19(2) to gate the contents from I/O bus (B) to the O bus and into the AC.

Table 8-1
I/O Status Bit Assignments

Bit	Status
00	Program Interrupt On
01	Tape Reader Flag
02	Tape Punch Flag
03	Teletype Keyboard Flag
04	Teletype Printer Flag
05	Oscilloscope Display Option Flag
06	Real Time Clock Option Overflow Flag
07	Clock Enable
08	Tape Reader No Tape
09	Tape Punch No Tape
10	DECtape Option Flag(s)
11-14	Unassigned
15-17	Reserved for special customer devices

8.2 DATA CHANNEL TRANSFERS

Four data channels (DCH) provide a high-speed data transfer path between core memory and four optional devices for the transfer of blocks of data. Address and control lines go through a secondary I/O bus, and the data lines go through the primary I/O bus which is used for program-controlled transfers.

Each of the four devices are assigned a pair of sequential channel registers in core memory:

Channel Registers		Device
00030	00031	0
00032	00033	1
00034	00035	2
00036	00037	3

These registers are initialized by the program before the device begins transferring data. The first register of a pair is a word count (WC) register initialized to the 2's complement of the number of data words to be transferred. The second is a current address (CA) register which is initialized to the address, -1, of the location from/to which the first data word is transferred. Four additional devices can be attached to the existing configuration; however appropriate channel register addresses must be assigned.

Once the registers have been initialized, the computer instructs the device, by means of IOT instructions, to prepare for transfer operations. The device itself contains essentially the same control logic as for program-controlled transfers, plus other DCH logic. Most of the additional logic is contained in a W104 Multiplexer.

When the device is ready for the input/output transfer, it issues a DCH RQ to the computer on an IO SYNC pulse. The CP honors the request at the completion of the current instruction by issuing a BK SYNC to the control memory logic and a DCH GR to all four devices. The DCH GR allows the requesting device to place the address of its WC register on the secondary I/O bus; the BK SYNC causes the computer to exit the program and as for program interrupt to go into the BK entry process.

The BK entry process replaces the BGN process at the end of the normal computer execute cycle. The process transfers the WC address from the secondary I/O bus to the MB. The WC address also goes from the MB into the MA for the coming computer cycle. The computer enters a WC cycle, incrementing the WC ADDR by 1 and storing it in the AR. The core memory read-half-cycle reads out the contents of the addressed WC register, +1, into the MB. If the contents (word count) have incremented to 0, an IO OFLO signal goes to the device, telling it not to request further transfers since the last data word is about to be transferred. If the device is also connected to the PI facility and the facility is enabled, it may also use the IO OFLO to issue a PROG INT RQ to the computer. For example, the PI can be honored to branch to a subroutine which re-initializes the WC and CA registers.

In any case, the core memory writes the incremented word count back into the WC register, and other processes place the incremented WC ADDR from the AR into the MB and MA for the next computer cycle. The incremented address is the address of the device's CA register.

The next computer cycle is a CA cycle, in which the address contained in the CA register is read from core memory into the MB. On its way through the ADR to the MB, the address is incremented by 1 (unless searching). The incremented address is the address of the core memory location from/to which the data word is to be transferred. For an input transfer, the device has issued a RD RQ by this time, which generates an IOP2 in the I/O control logic. IOP2 goes to the device's W103 Device Selector to generate an IOT XX02 pulse. The device uses this pulse to gate the data word out of its data register onto the primary I/O bus. Other processes gate the data word into the AR as the core memory writes the incremented address into the CA.

For an output transfer, the processes write 0s into the AR because no RD RQ has been issued and, consequently, no data is placed on the primary I/O bus.

The next computer cycle(s) performs the actual data transfer. For input transfer core memory reads out the contents of the addressed location, but they are flocked from the MB and are therefore lost. Other processes gate the data word from the AR to the MB, and core memory writes the word into the addressed location. This completes the input transfer, and a BGN word (or another BK entry word) follows.

For an output transfer the core memory read-half-cycle does place the contents of the addressed location in the MB, since the contents represent the data word to be transferred to the device. Other processes gate the data word from the MB onto the primary I/O bus, and the core memory restores the same data word to the addressed location.

This completes one computer output cycle. An additional idle cycle ensues which gives the data word time to settle on the primary I/O bus and the device time to strobe it into its data register. During the idle cycle, the device's WR RQ generates an IOP4 in the I/O control logic. IOP4 generates an IOT XX04 pulse in the device's device selector, which is used to strobe the data in. A BGN process (or BK entry) word follows as for the single input transfer cycle.

Successive DCH requests and DCH breaks can occur from the same device or another device provided the device flag has been raised at the start of the current CA cycle. Otherwise, the program resumes and executes at least one instruction before the DCH can cause another break entry. If the instruction is an IOT, XCT, or optional EAE instruction, a considerable delay between DCH breaks is possible, since these are multi-cycle instructions and their DONE(1) levels appear in the last cycle only. For maximum operating efficiency, the program must be planned with the device transfer rates and the break entry point in mind.

Priority among I/O devices making simultaneous DCH requests is determined by their physical placement. Devices closer to the I/O bus have priority over those farther away. An enabling level from the computer's I/O control section is chain-connected through the W104 multiplexer in each device. The DCH grant to all four devices causes the removal of the enabling level from the lower priority devices.

DCH operations also include an add-to-memory feature (Section 8.2.8).

8.2.1 I/O Bus Connections

The DCH uses the data, IO SYNC, RD RQ, and IO PWR CLR lines of the primary I/O bus, AB25, 26 drawing KD2(1), and certain address and control lines of the secondary I/O bus, drawing KD2(2). The same cabling considerations for the primary bus apply to the secondary. Of the 15 IO ADDR lines shown on KD2(2) the DCH uses the six least significant for the four assigned pairs of channel addresses.

The basic PDP-9/L allots four pairs of WC and CA registers in core memory for use with four optional devices in the data channels. Because of the time delay encountered in propagating signals through the W104 modules, the number of additional devices is limited to four (total eight) provided the total I/O bus cable length does not exceed 50 ft. The additional pairs of core memory registers must be assigned and protected, and the devices must contain the W104 multiplexer or equivalent logic interfaced to the I/O bus.

8.2.2 Multiplexer W104

Figure 8-5 is the logic diagram for the W104 module. The device flag and IO SYNC pulse set the REQ flip-flop. The set REQ flip-flop sends a DCH RQ to the computer and places the EN OUT level to succeeding W104s at ground (EN IN), holding their REQ flip-flops in the reset state until the currently requesting device relinquishes control by resetting its flag.

8.2.3 Break Synchronization

The device flag raises asynchronously when the device is ready for a data transfer. Thereafter, the DCH break synchronizes on IO SYNC and IO SYNC POS pulses. IO SYNC pulses occur on computer CLK POS pulses where no IOT instruction is currently in progress, drawing KD3(1). Under these conditions, IO SYNC occurs to set the REQ flip-flop in the W104 in conjunction with the device flag, Figure 8-5 and drawing KC31. REQ(1) sends a ground DCH RQ to the DCH SYNC flip-flop, drawing KD3(2), and grounds the EN OUT signal to succeeding DCH devices. The EN IN level is supplied by the I/O control, drawing KD3(1) at W005-H19H, labeled DCH EN. DCH EN goes to the first W104 from the secondary I/O bus. Assuming that the first device has raised its flag, IO SYNC has set its REQ flip-flop. The 0 side of REQ in going to ground blocks the EN IN level at the R111 input gate. Thus, EN OUT goes to ground, resetting and holding the lower priority REQ flip-flops.

The main CLK pulse, of course, starts a normal computer cycle. The DCH RQ sent to the I/O control waits for the next CLK pulse. IO CLK(B) derived from the next CLK pulse (IO CLK POS) generates IO SYNC POS on drawing KD3(2) if the conditions IOT (0), CLK SYNC(0), etc. are present. This means that IO SYNC POS occurs only on an IO CLK(B) during which no IOT, optional API, RTC or PI, operation is in progress. The DCH, therefore, cannot interrupt any of these current operations.

IO SYNC POS strobes a DCD gate conditioned by DCH RQ to set the DCH SYNC flip-flop. DCH SYNC(1) holds the optional CLK SYNC and PRE API SYNC flip-flops in the reset state, and sets and holds the DCH SYNC SAVE flip-flop. DCH SYNC SAVE (1) holds the PROG SY flip-flop in the reset state. Therefore, once the DCH SYNC flip-flop has set, these operations cannot begin; DCH has the higher priority.

DCH SYNC(1) also triggers the 100-μs DCH BK DLY, generates DCH GRANT on drawing KD3(1), and INC + DCH on KD3(2). DCH GRANT sets the ENA flip-flop in the W104 in conjunction with REQ(1). ENA (1) puts the device's IO ADDR on the secondary I/O bus. The IO ADDR bits go to drawing KD5, where they are buffered to IO ADDR(B). The DCH BK DLY illuminates the console DATA indicator.

DCH GRANT also produces a CLR FLAG pulse in the W104. CLR FLAG resets the REQ flip-flop and the device's data flag. EN OUT goes negative, allowing lower priority devices to make future DCH requests.

DCH SYNC(1) generates BK SYNC on drawing KD3(2). BK SYNC waits for DONE(1) in the instruction being executed. Since the device data flag was raised asynchronously, the DCH SYNC flip-flop can set on an IO CLK(B) pulse which starts any one of the four normal computer cycles (fetch, defer, IA0, execute). Assuming that the flag is raised before a normal two-cycle computer instruction begins, the CLK pulse that initiates fetch results in DCH RQ and the CLK pulse that initiates execute results in DCH SYNC(1), as shown in drawing KC31.

BK SYNC goes to the CP via the CP-I/O interface to generate ODD ADDR in conjunction with DONE(1), drawing KC17. The execute process word containing DONE(1) also contains CMA10. Process word 10 is the BGN word which normally places the next address from the PC in the MB for the next computer fetch cycle. Now ODD ADDR on drawing KC17 boosts

CMA10 to 11. Therefore, the last process word in the execute cycle is taken from CM location 11.

Process word 11 is the BK entry word which starts the DCH break operation. The word contains EXT, IRI, SM, and CMA30. EXT(1) produces IO ADDR ON BUS, drawing KD7(1), and LIO, drawing KC13. IO ADDR ON BUS gates the device's IO ADDR 12(B)-17(B) through the input mixer modules onto I/O bus (B). LIO gates the address from I/O bus (B) onto the O bus.

EXT(1) on drawing KC19(2) produces 1 → MBI in conjunction with SM(1) and CLK. 1 → MBI sets the MBI flip-flop, and MBI(1) then gates the address on the O bus into the MB.

IRI(1) puts 0s in the IR, drawing KC12. On drawing KC17, EXT(1) and INC + DCH boost the CM address from 30 to 34. Process word 34 will be extracted from control memory on the CM STROBE that results from the next CM CLK pulse to start the WC cycle. Drawing KC5 shows the break flow from the BK entry process through completion.

8.2.4 WC Cycle

Process word 34 is extracted from control memory on the CM CLK pulse in conjunction with SM(1) of the BK entry word. CLK and SM(1) also start the core memory read/write cycle.

IO SYNC, derived from IO CLK POS, sets ENB in the W104, Figure 8-5. ENB(1) sends a SELECT level to the W103 Device Selector in the DCH device. This level is applied to W103-BD, Figure 8-3, bypassing the device select code input gates to force selection of the device.

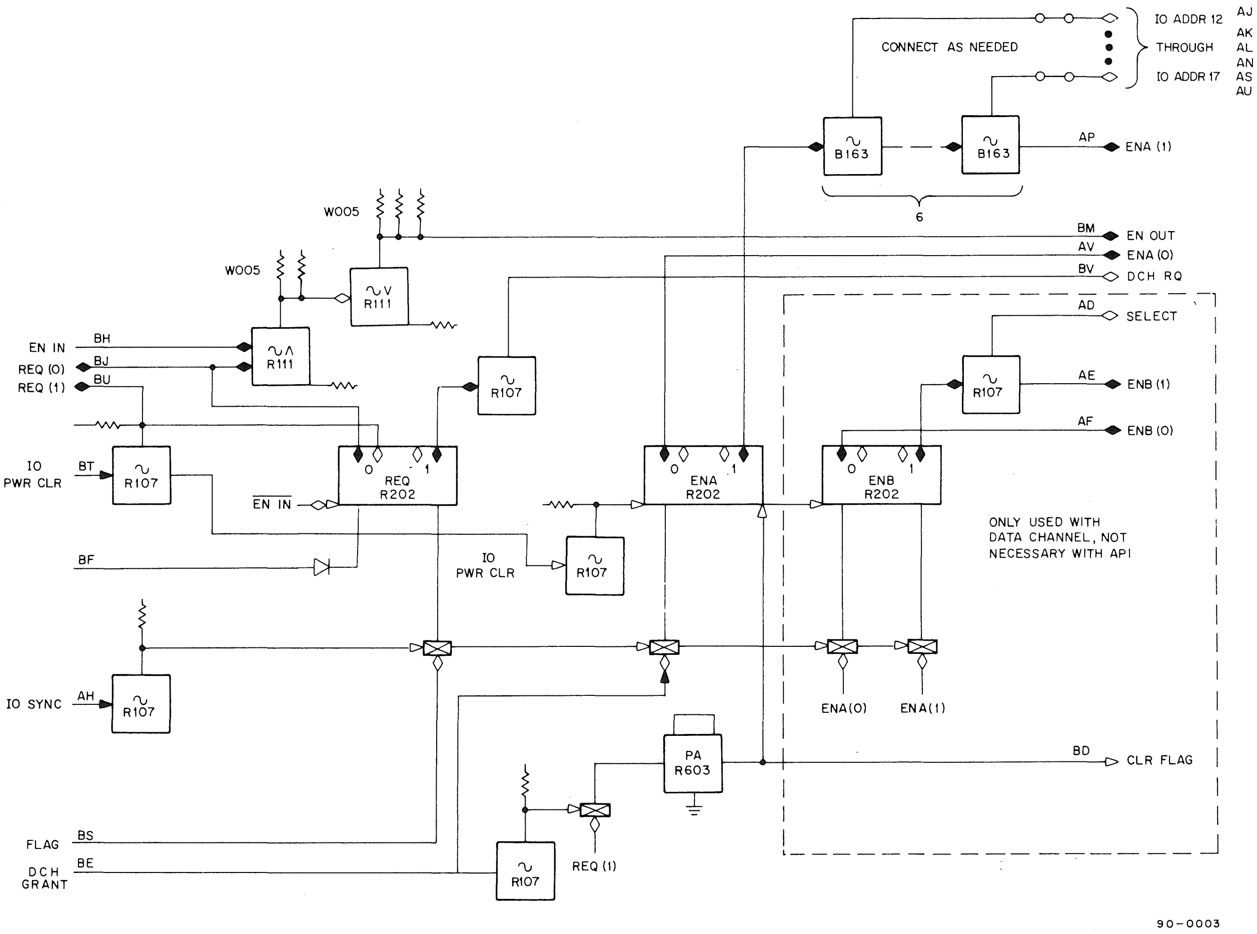


Figure 8-5 Multiplexer W104, Logic Diagram

Process word 34 contains MBO, + 1, ARI, DCH, and CMA10. MBO(1) gates the WC ADDR from the MB to the B bus. The address on the B bus goes through the ADR, and NOSH places it on the O bus. As it goes through the ADR, process +1(1) generates CI17 on drawing KC14, and CI17 increments the ADR contents by 1. ARI(1) gates the incremented address from the O bus into the AR.

Process word 34 remains active for two normal process-word periods because the CM STROBE cannot retrigger the CM timing chain, on drawing KC16, in the absence of a CONT(1) bit. CM STROBE does get as far as the CLR gating, however, at which time DCH(1) generates IN CLR and CLR. IN CLR sets MBI via 1 → MBI and resets MBO, ARI. CLR sets SAO and resets +1.

DCH(1) sets BK0 on drawing KD3(3) for a break count of 10. BK0(1) · BK1(0) generates DCH INX on drawing KD3(3). DCH INX produces CI17 on drawing KC14, in conjunction with SAO(1).

The core memory read-half-cycle reads out the contents of the addressed WC register. SAO(1) places them on the B bus, the B bus contents go through the ADR, and NOSH placed them on the O bus. As the contents pass through the ADR they are incremented by 1. MB(1) gates the contents into the MB. IA0(0) and the MEM STROBE that caused core memory readout retrigger the CM timing chain. If the word count has incremented to 0, ADRA=0 and ADRB=0 result at the ADR output, drawing KC21. ADR=0, SAO(1), BK0(1), DK1(0), and the CM STROBE triggered by MEM STROBE produce OFLO on drawing KC14. OFLO goes through the CP-I/O interface to the IO OFLO flip-flop on drawing KD3(2), setting the flip-flop. IO OFLO(1) goes through the I/O bus to a control flip-flop in the device. The control flip-flop acts to shut down the device, since the current data transfer is the last transfer of the block of data. The control flip-flop may also be used to send a program interrupt request to the computer. The core memory write-half-cycle writes the incremented word count into the WC register.

Process word 34 contains CMA10. On drawing KC17, DCH(1), $\overline{\text{INC MB}}$, and BK1(0) boost this address to 14. The negative $\overline{\text{INC MB}}$ level is present because the add-to-memory capability (Section 8.2.8), and optional real-time clock are inactive during normal DCH operations.

The CM STROBE triggered by MEM STROBE extracts process word 14. Process word 14 contains ARO, SM, and CMA37. ARO(1) gates the contents of the AR onto the A bus. The contents go through the ADR and NOSH places them on the O bus. ARO(1) on drawing KC19(2) produces 1 → MBI in conjunction with SM(1) and CLK. 1 → MBI sets the MBI flip-flop, and MBI(1) gates the contents of the O bus into the MB. The contents represent the address +1 of the WC register, incremented and stored in the AR earlier. This incremented address is the address of the CA register; it is jammed into the MA, at MA JAM time, for the coming CA cycle. The CA cycle begins on the next CLK pulse with SM(1) of process word 14.

8.2.5 CA Cycle

SM(1) and CLK start the control memory and the core memory for the CA cycle. CM STROBE produced by SM(1) · CM CLK extracts process word 37, which contains DCH and CMA13. DCH(1) steps the BK counter to 11 by setting BK1, with DK0 remaining set.

If the device intended to make an input data transfer it has placed a RD RQ and $\overline{\text{WR RQ}}$ on the I/O bus at ENB(1) time. On drawing KD3(3), the RD RQ conditions a DCD gate so that an IOP2P pulse is generated by S602-H21K when BK1 sets. IOP2P sets the IOP2 flip-flop. IOP2(1) then gates on an IOTXX02 pulse in the device's W103 Device Selector in conjunction with the force SELECT level from the W104. The IOT XX02 pulse is used by the device to strobe the data word from its data register onto the I/O bus.

Also, as BK1 sets, it resets the DCH SYNC flip-flop in conjunction with $\overline{\text{WR RQ(B)}}$ on drawing KD3(2). Reset DCH SYNC removes the BK SYNC, INC + DCH and DCH GRANT levels, releases the reset hold on the CLK SYNC, and optional PREAPI SYNC flip-flops, and the set hold on the DCH SYNC SAVE flip-flop.

BK0(1), BK1(1), and $\overline{\text{+1 → CA INH}}$ produce $\overline{\text{DCH INX}}$ on drawing KD3(3). The +1 → CA INH level is the negation of a special signal generated in devices which automatically search, e.g., tape systems. The assertion level would inhibit DCH INX, preventing the CA increment described here.

The CM STROBE that extracted process word 37 cannot produce another CM STROBE; nevertheless, it triggers the chain to generate IN CLR and CLR in conjunction with DCH(1) on drawing KC16. IN CLR sets MBI and CLR sets SAO as for the WC cycle. The core memory read-half-cycle reads out the address in the CA register and MEM STROBE retriggers the CM chain. SAO(1) and MBI(1) gate the address in the CA register into the MB via the B bus, ADR, and O bus. As the address goes through the ADR, CI17, resulting from DCH INX and SAO(1) on drawing KC14, increments the address by 1 as for the WC cycle.

The incremented address in the MB represents the address of the core memory location from which the data word is transferred to the device (output transfer), or to which the data word is transferred from the device (input transfer). This address is written into the CA register during the core memory write-half-cycle.

The CM STROBE triggered by MEM STROBE extracts process word 13. This word contains ARI, CONT, and CMA16. ARI(1), BK0(1), and BK1(1) produce LIO on drawing KC13. LIO places the data word from I/O bus(B) onto the O bus, and ARI(1) gates it into the AR (input transfer). CM STROBE and CONT(1) retrigger the CM chain to extract process word 16. This word contains SM and CMA36. SM(1) waits for the next CLK pulse to start the data cycle(s).

8.2.6 Data Input Cycle

SM(1) and CLK extract process word 36 and start the core memory cycle. Process word 36 contains DCH and CMA17. DCH(1) steps the BK counter to 01 by resetting BK0, with BK1 remaining set. IO SYNC resets ENA.

In the absence of CONT(1), the CM STROBE cannot produce another CM STROBE; nevertheless, it retriggers the CM chain to generate IN CLR and CLR in conjunction with DCH(1). IN CLR sets MBI as usual, but now BK0(0), BK1(1), and $\overline{\text{WR RQ}}$ on drawing KC19(3) prevent CLR from setting SAO. Thus, the STROBE 0-8, 9-17 in core memory reads out the contents of the addressed memory location, but they are lost because of reset SAO. The BK count and RD RQ(B) set the ARO flip-flop. ARO(1) gates the data word from the AR onto the A bus. The data word on the A bus goes through the ADR, and NOSH places it on the O bus. MBI(1) gates it into the MB.

MEM STROBE and IA0(0) allow the generation of another CM STROBE. This CM STROBE extracts process word 17, which contains MBO, DONE, and next CMA10 (BGN). MBO(1) is used for an output transfer only. In the input transfer case, it gates the data word onto the B bus and through the ADR, but the data word stops there. During this period the core memory write-half-cycle writes the data word into the addressed memory location.

The CM STROBE also sets CONT on drawing KC19(1) in conjunction with the BK count, DCH(1), and $\overline{\text{WR RQ}}$ to generate another CM STROBE. This next CM STROBE then extracts process word 10 (or 11). If the data word just transferred into memory was the last, process word 10 transfers the current program address from the PC to the MB, resuming the interrupted program. If the DCH device flag was again set at the CLK pulse of the CA cycle, another DCH RQ went to the I/O control on the IO SYNC pulse derived from CLK. Consequently, DCH SYNC(1) occurs at CLK of the current data cycle and the BK entry word 11 replaces BGN. The DCH break synchronization repeats for another word transfer.

Ultimately, IO SYNC following BGN on the last DCH transfer resets ENB, removing the force SELECT level from the device's W103. The IO CLK POS pulse developed from CLK POS resets BK1 for a count of 00, and the program resumes on this CLK pulse. DLY resets DCH SYNC SAVE 400 μs later in conjunction with BK1(0). DCH SYNC SAVE(1) has held off any PI requests at the PROG SY flip-flop in order to give the optional API requests the priority over the next break.

8.2.7 Data Output Cycles

For an output transfer, the RD RQ and IOP2 levels are absent during the CA cycle. Because of this, ARI(1) of process word 13 merely strobes 0s into the AR, since the I/O bus(B) is disabled. During the data cycle, the CLR pulse generated by DCH(1) of process word 36 and CM STROBE sets SAO in addition to IN CLR setting MBI. Thus, the contents of the addressed memory location (data word) do get into the MB. The ARO flip-flop remains reset in the absence of RD RQ, so that there is no interference from the 0s in the AR. Also, process word 36 steps the BK count to 01 as for the input transfer, and BK0, upon resetting, resets DCH SYNC with the WR RQ condition, drawing KD 3(2). Reset DCH SYNC removes the BK SYNC, INC + DCH, and DCH GRANT levels, and releases the reset hold on the CLK SYNC and optional PRE API SYNC flip-flops.

MBO(1) of process word 17 gates the data word onto the B bus, through the ADR, and onto the I/O bus (with IO BUS ON). Process word 17 remains active throughout the next CLK period, because WR RQ now prevents CONT from setting. On the IO CLK POS pulse, IO SYNC resets ENA and IO CLK POS resets BK1 as for an input transfer, but the absence of an SM(1) bit prevents CLK from starting the core memory and control memory cycles. Thus an idle cycle follows, during which the data word has time to settle on the I/O bus and the device has time to strobe the word into its data register.

As BK1 resets, it strobes a DCD gate conditioned by WR RQ on drawing KD3(3) to generate IOP4P. IOP4P sets the IOP4 flip-flop. IOP4(1) and the force SELECT level from the W104 (ENB is still set) produce an IOT XX04 pulse in the device's W103. The device uses this pulse to strobe the data word from the I/O bus into its data register.

IOP4(1) also conditions a DCD gate to the IO RESTART logic on drawing KD3(3). The IO CLK POS pulse triggers the 400-ns delay multivibrator B301-H22. The DLY recovers $\overline{(DLY)}$ to trigger IO RESTART 400 ns later. IO RESTART goes to the CM timing chain to produce a CM STROBE. This CM STROBE extracts the BGN word (or BK entry word if another DCH RQ was present) as for an input transfer. Note that DLY also resets IOP4, and DCH SYNC SAVE as for the input transfer. IO SYNC derived from the next CLK pulse resets ENB.

8.2.8 Add-to-Memory Facility

This facility permits incrementing the contents of a specified memory register (WC register) using one DCH cycle, or permits the contents of the specified memory register to be added to the contents of a device data buffer using all four DCH cycles. A DCH device can request these actions through the appropriate I/O bus lines.

A device using the INC MB facility is connected to the DCH in the usual manner. A DCH RQ is initiated by the device flag, the flag is cleared when the request is granted, and the DCH RQ is removed from the I/O bus as usual. The device's WC register address is gated onto the IO ADDR lines by ENA(1) generated in the W104. The ENA(1) level is used by the device to issue a ground INC MB level to the I/O bus. The WC cycle starts on the next CLK pulse after break entry by setting ENB. DCH INX of process word 34 in the WC cycle increments the contents of the addressed memory register as for normal DCH operations. IO OFLO goes to the device as usual, if the WC register increments to 0. The ground INC MB level from the device causes the control memory to extract process word 10 (BGN) on the next CM STROBE. INC MB and BK0(1) of the break counter set DONE on this CM STROBE. The next CLK pulse resets BK0 with DONE(1) as the computer reverts to the main program. BK0(0) and INC MB reset DCH SYNC. Reset DCH SYNC removes DCH GRANT so that the two succeeding IO SYNC pulses reset ENA and ENB in the W104. If the device requests another INC MB break by again raising its flag, one instruction of the main program is executed before the next break is honored, as the break entry word synchronizes on the DONE bit of the executed instruction.

A device using the add-to-memory facility is connected to the DCH in the usual manner. A DCH RQ is initiated by the device flag, the flag is cleared when the request is granted, and the DCH RQ is removed from the I/O bus as usual. The device's WC register address is placed on the IO ADDR lines by ENA(1) generated in the W104. The WC cycle starts on the next CLK pulse after break entry by setting ENB. ENB(1) causes the device to issue both a RD RQ and a WR RQ to the I/O control logic.

During the CA cycle, IOP2 gates the device data onto the I/O bus as usual, and process word 13 gates it into the AR. During the first data cycle, process word 36 reads the contents of the addressed memory register into the MB via the B bus, ADR, and O bus. At the same time, the device data in the AR is gated into the MB via the A bus, ADR, and O bus. An ADD operation therefore takes place in the ADR and the sum is deposited in the MB. The sum in the MB is later placed on the I/O bus by process word 17 and may be gated into the device data buffer with the IOP4 pulse that occurs during the second data cycle, if desired. Successive add-to-memory DCH breaks can occur if the device flag is up at CLK time of the current CA cycle.

During ADD, a DATA OFLO occurs when

$$\text{MB} + \text{I/O Bus} > 2^{17}-1$$

or

$$\text{MB} + \text{I/O Bus} < -2^{17}$$

DATA OFLO(1) is a 200 μs pulse, gated onto the I/O bus in the middle of the third add-to-memory DCH cycle, which signals that an incorrect sum has occurred. This happens when the MB and I/O bus are of like signs and their sum has the opposite sign. If the MB and I/O bus have opposite signs DATA OFLO(1) cannot occur.

8.3 API CHANNEL TRANSFERS

The 32-channel Automatic Priority Interrupt option KF09A permits device-initiated data transfers at four high priority levels and program-initiated data transfers at four lower priority levels. The eight priority levels take precedence over program interrupt breaks and the main program. API transfers take place via the I/O bus as for DCH transfers. The API system interface contains essentially the same logic as the DCH, including W103 Device Selectors and W104 Multiplexers for each device, plus synchronization and priority determination logic within the option.

Up to eight I/O devices can be multiplexed by as many W104 Multiplexers for operation at the same level of priority. Among devices on the same level of priority, the device closest to the I/O bus has precedence, as for DCH transfers.

Each device is assigned an address in core memory as for the DCH. Assignments are made independent of priority levels; a device may be assigned more than one priority. The four software priority levels command subroutines entered at core memory addresses 00040 through 00043. The remaining locations 00044 through 00077 are assigned to the devices themselves.

A device ready flag causes its W104 to issue an interrupt request as for DCH transfers. The interrupt request goes to the API option logic for determination of priority. If the issuing device has a higher API priority, the option logic interrupts a lower API interrupt in process, issues an API grant to the higher priority W104 Multiplexer, and sets a SYNC flip-flop in the option. The API grant defers requests from all lower priority devices by disabling their W104 multiplexers, and enables the priority device to place its core memory address on the I/O bus. The API SYNC flip-flop in the option will cause a BK SYNC in the I/O control logic as for DCH transfers. The BK SYNC causes the control memory to enter the BK process (upon the completion of the current instruction) as for DCH transfers. The control memory recognizes the BK as an API BK, and goes into the XCT instruction process on the next CLK pulse. The XCT process causes the computer to execute the instruction contained in the core location addressed by the device. This instruction is usually a JMS to the device service routine. Service routine exit and return to the main program is accomplished by a DBR/JMP * instruction.

Since the I/O devices are assigned address locations independent of priority, the API logic affords three different methods to change active device priorities according to the needs of the program. Priority reallocation and determination are discussed in detail in the KF09A option manual.

8.4 TRANSFER PRIORITIES

The following descending order of interrupt priorities is established where simultaneous interrupt requests occur.

DCH
RTC (Optional)
API (Optional)
PI
Main Program

CHAPTER 9
ASR-33 TELETYPEWRITER

The Model 33 Automatic Send-Receiver (ASR) Teletypewriter Set, the standard input/output equipment supplied with the PDP-9/L, has two basic modes of operation-keyboard/reader and teleprinter/punch. During the keyboard/reader mode, data is serially transferred from the teletype keyboard or paper-tape reader to the PDP-9/L central processor. In the teleprinter/punch mode, the data stored in the PDP-9/L is serially transferred to the ASR33 for printout on the teleprinter or punching on paper tape. The data transfer rate is ten characters per second.

The ASR33 has a full-duplex and half-duplex interface capability. Full-duplex operation permits the transfer of data to proceed independently, i.e., data can be transferred from the keyboard/reader to the PDP-9/L while different data is being transferred and recorded on the teleprinter/punch. Half-duplex interface does not permit independent input/output operations.

The information transferred between the PDP-9/L and ASR33 is assembled for parallel transfer to the accumulator by circuitry within the PDP-9/L I/O. The control circuitry also provides program flags to enable program interrupts, program skips, and data transfer. When data is transferred from the ASR33 keyboard/reader to the PDP-9/L accumulator, the Teletype-In (TTI) data path is used in the I/O circuitry. Conversely, the Teletype-Out (TTO) data path is used to transfer data from the PDP-9/L to the ASR33.

The paper tape and reader formats are shown in Figure 9-1.

9.1 KEYBOARD/READER

9.1.1 Functional Description (Refer to Figure 9-2)

Data from the keyboard and paper reader is transferred to the PDP-9/L via the keyboard/reader control circuitry. Data transfer is initiated by pressing a key on the keyboard or with programmed instructions for the paper tape reader.

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

When the teletype tape reader data is to be entered into the TTI, the control de-energizes a relay in the teletype unit to release the tape feed latch. When released, the latch mechanism does not stop tape motion until the complete character has been sensed.

A keyboard flag is set and causes a program interrupt request when the 8-bit teletype character has been assembled in the TTI. The program senses the condition of this flag with a KSF instruction and issues a KRB. This instruction clears the AC and keyboard flag and transfers the content of the TTI into the AC.

9.1.2 Keyboard Control

The keyboard control logic is shown on drawing KD11 (1). A simplified functional diagram is shown in Figure 9-2. When a key is pressed on the console, the start space derived from the keyboard generator results in a negative TT KBD IN level. (The switch shown at terminals 3-4 opens.) TT KBD IN and KBD FLG(0) ·TT HD place the solenoid driver output W040-B33S at -15V, releasing the printer solenoid, and enabling the printing circuits for simultaneous printout.

TT KBD IN becomes a ground level TT KBD IN(B) at the output of inverter S107-C33R. This ground level conditions the DCD reset gate at the TTI shift register's most significant bit TTI0, and conditions the DCD set gate at the TT IN ACT flip-flop.

The central processor's 1.5-µs CLK pulses continuously strobe the DCD gate of TT IN ACT. Once conditioned by the start space, the flip-flop sets on the next CLK pulse.

TT IN ACT(1) generates a TTI INITIALIZE pulse at pulse amplifier S603-C39M, starts a TTI CLK, and disables a TTO CLK on drawing KD11(2). TTI INITIALIZE resets IN LAST UNIT, and sets all TTI register flip-flops (as set TTI represents a "space"). IN LAST UNIT(0) conditions a DCD gate at the output of

TTI CLK, enabling a TTI LOAD on the next positive transition of the TTI CLK. This pulse resets TT RDR RUN to assure that the paper tape does not advance. The TTI CLK, enabled by TT IN ACT(1), is adjusted internally to generate pulses at 110 pps after an initial delay of 4.54 ms. The initial delay places the pulses at the center of the 9.09-ms code units for read-in accuracy. The first TTI CLK pulse strobes the DCD gate conditioned by IN LAST UNIT(0) to produce TTI LOAD. TTI LOAD then strobes the input gates of the TTI register. $\overline{\text{TT KBD IN(B)}}$ produced by the start space resets TTI0. All other bits remain at 1.

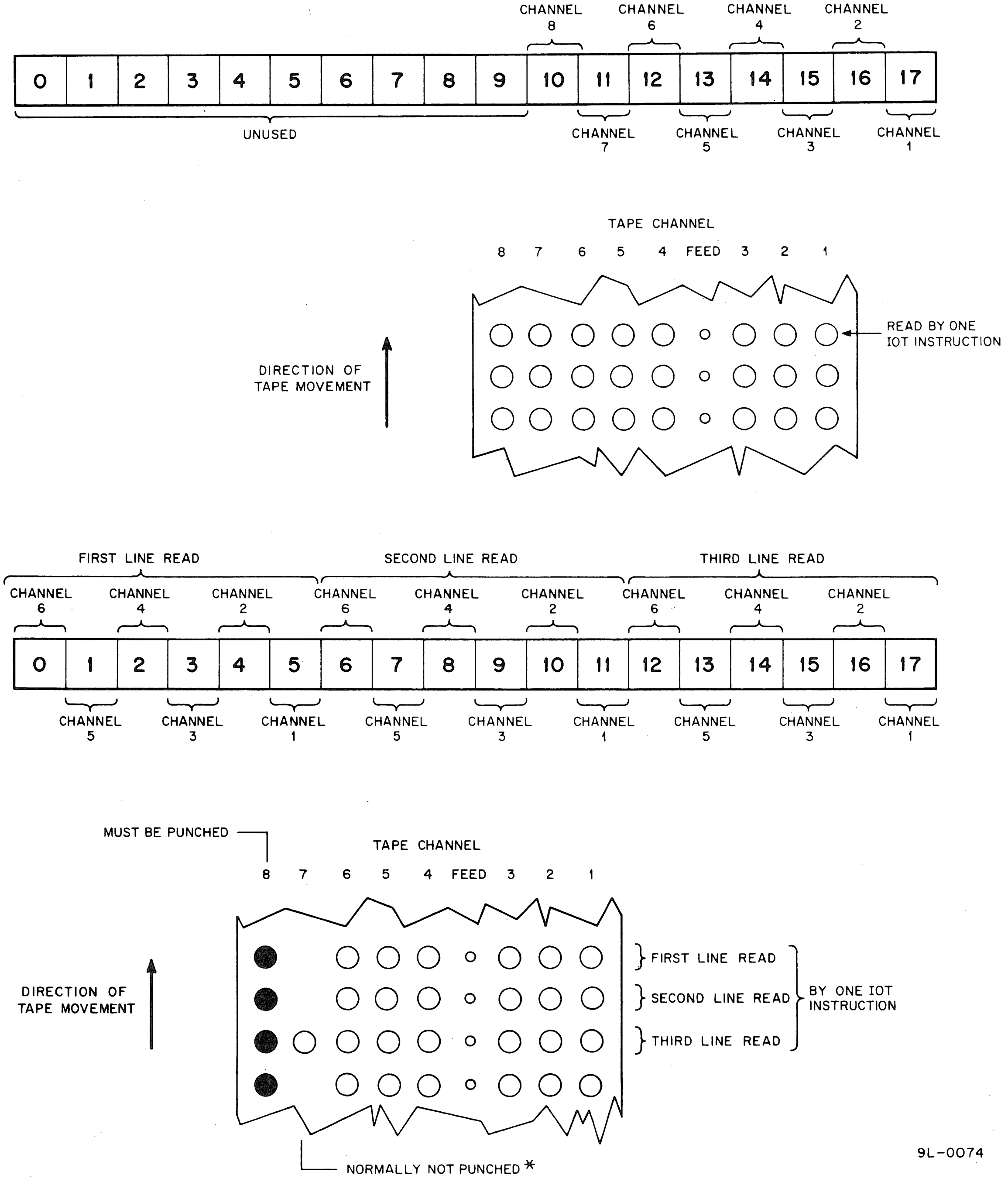


Figure 9-1 Perforated Tape Format

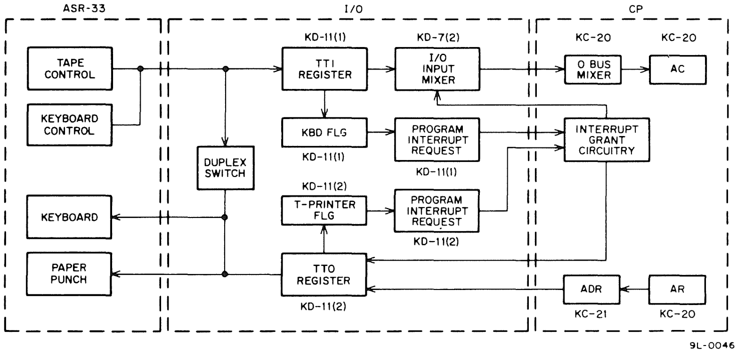


Figure 9-2 Basic I/O Data Transfer, Functional Diagram

Successive TTI LOAD pulses appear synchronously with successive keyboard character units to serially shift each mark or space through the TTI register. On the eighth TTI LOAD, the start space becomes the least significant bit, TTI7, and the seventh code unit goes into TTI0. TTI7 (0) conditions the DCD set gates of the KDB FLG and IN LAST UNIT flip-flops. The ninth TTI LOAD puts the last code unit into TTI0, shifts the first into TTI7, and sets KBD FLG and IN LAST UNIT.

IN LAST UNIT(1) conditions the DCD reset gate of TT IN ACT and disables the DCD gate at TTI LOAD. The next TTI CLK pulse cannot issue another TTI LOAD, but resets TT IN ACT to stop the TTI CLK and all shifting operations.

Initially, the TTI register was set to 1s, producing TTI FULL. The negative TTI FULL level operates with TT KBD IN(B) to reset TT IN ACT if a false start space is created by a noisy keyboard generator. If the noise is sufficient, a false TT KBD IN(B) level could set the TT IN ACT flip-flop, as for a true space, on the next computer CLK pulse. By the time the first TTI CLK pulse occurs (4.54 ms) the noise level has disappeared to remove TT KBD IN(B). $\overline{\text{TT KBD IN(B)}}$ · TTI FULL causes TT IN ACT to reset on the first TTI CLK pulse.

KBD FLG(1) causes a ground PROG INT RQ, drawing KD11(1), at inverter R111-D39H. During keyboard/reader operation KBI DIS TT RDR is negative because of reset RDR RUN and RDR FLAG flip-flops. The PROG INT RQ goes to the I/O control logic, drawing KD3(2), to cause a program interrupt if the PIE flip-flop is set.

9.1.3 Reader Control

The paper tape reader data is transferred to the PDP-9/L in the same manner that keyboard data is transferred. The difference between the two transfers is the method of initiating the transfer sequence. For keyboard operation, depressing a key enables the transfer of teletype character codes to the TTI shift register, which in turn, sets the KBD FLG. However, to initiate paper tape reader operation, the reader advance relay is activated to pass one line of paper tape through the reader. The data on the tape is then coupled to the TTI register as TT KBD IN pulses. As with keyboard operation, the KBD FLG is set when the data transfer is complete.

To initiate the paper tape data transfer, the PDP-9/L issues a KRS instruction (700322) to the reader control, drawing KD11(1). This KRS instruction is decoded by the R111-R002 gate to set the TT RDR RUN flip-flop. A set TT RDR RUN grounds the solenoid driver and activates the ASR33 reader-advance relay. As the tape passes through the reader, TT KBD IN pulses are generated, and the sequence described for keyboard operation is repeated. When the transfer of one line of data is completed, the KBD FLG is set, and a PROG INT RQ causes a PDP-9/L program interrupt.

9.1.4 Data Transfer Instructions

During the PDP-9/L program interrupt, a flag search subroutine is performed to determine what device caused the interrupt and, upon detection of the device code, jump to a service routine. The search and service instructions pertaining to the reader/keyboard are as follows:

IOT0301	KSF	Skip on keyboard flag
IOT0312	KRB	Clear keyboard flag and AC. Read TTI register into AC.
IOT0322	KRS	Advance paper tape one line.

An example of basic programming of the keyboard and reader instructions follows.

Read Tape	Read Keyboard
KRS	KSF
KSF	JMP. -1
JMP. -1	KRB
KRB	

9.1.4.1 Skip on Keyboard Flag - The KSF instruction (700301) senses the status of the KBD FLG. If the flag is set, the keyboard control logic issues a skip request to the CP and the program skips the next instruction.

MB06-11 are detected to produce KBD SEL on drawing KD11(1). When IO0 sets, signaling the start of the first IOT execute period, it produces IOP1P in conjunction with MB17(1). IOP1P sets IOP1, IOP1(1), a KBD SEL(B), and KRI DIS TT RDR sample the state of the KBD FLG. If the KBD FLG is set, the INT SKP RQ BUS goes to ground, triggering pulse amplifier W012-F18D on drawing KD3(3). This PA pulse is the IO SKIP signal which sets the SKIP flip-flop in the CP, drawing KC14. The KSF instruction idles through its remaining execute periods, until PCO(1) of the BGN process word generates CI17 in conjunction with SKIP(1). CI17 initiates a carry in the ADR as PCO(1) gates the contents of the PC through the ADR to the MB for the next instruction fetch cycle. Thus, the address in the MB contains the PC address + 1.

9.1.4.2 Transfer to AC - The KRB instruction (700312) clears the KBD FLG and the AC, then gates the contents of the TTI register into the AC via the input mixer and I/O bus (B).

During the IOT fetch, the KRB instruction is decoded in the CP's op-code detection circuits for IOT(1). On drawing KD3(1), the IOT(B) level derived from IOT(1) samples the MB06-11 bits in the instruction to produce the appropriate DS0-5 levels, and IOT(1) further decodes DS0-2 to produce OXEN. OXEN samples DS3-5 at the keyboard control, drawing KD11(1), resulting in KBD SEL and KBD SEL(B).

MB14(1) is also decoded in the op-code detection circuits to set ACI. ACI(1) places 0s in the AC from the inactive O bus, thus clearing the AC.

When the pulse counter IO1 in the I/O control logic sets signaling the start of the second IOT execute period, it produces IOP2P on drawing KD3(3) in conjunction with MB16(1). IOP2P, in conjunction with KBD SEL, sets IOT 0302, drawing KD11(1), and also resets the KBD FLG.

IOT0302(1) generates an INT RD RQ BUS level on drawing KD3(3). This level is NORed at R111-F19 UV for RD RQ(B), then NANDed for AC RD at CLK DLYD time. AC RD sets ACI and becomes AC RD(B) on drawing KC19(2), to generate LIO on drawing KC13.

At the input mixer, drawing KD7(1), IOT0302(1) generates TTI ON BUS. The TTI ON BUS level gates TTI0-7 into the input mixer gates, positions 10-17 on drawing KD7(2), whose outputs are NORed onto I/O bus (B). LIO gates the contents of I/O bus (B) onto the O bus, and ACI(1) gates them from the O bus into the AC.

9.2 TELEPRINTER/PUNCH

9.2.1 Functional Description

The transfer of data from the PDP-9/L accumulator to the ASR33 teleprinter/printer is initiated by a teletype load and select (TLS) IOT instruction. This instruction is program controlled and is formed by combining the clear teleprinter flag (TCF) IOT0404 and teletype load and print (TPC) IOT0402 instructions.

When the PDP-9/L initiates a TLS command, an 8-bit character code is set in parallel from the arithmetic register (AR) to a TTO shift register. During the fetch cycle, the control of the AC is transferred to the AR. The teleprinter/punch control generates the start space, shifts the eight character bits serially into the teletype printer selector magnets, and then generates the stop marks. This transfer of information from the TTO into the teleprinter/punch unit is accomplished at the normal teletype rate and requires 100 ms for completion.

The flag in the teleprinter control is set when the last bit of the character code had been sent to the teleprinter/punch, indicating that the TTO is ready to receive a new character from the AR. The flag is connected to both the program interrupt synchronization element and the instruction skip element. The program then checks the flag by means of a TSF instruction and, if set, issues the TLS instruction to clear the flag and send a new character from the AC to the TTO.

9.2.2 Teletype Load Sequence (TLS)

During the IOT fetch, the TLS instruction is decoded in the central processor op-code detection circuits for IOT(1). On drawing KD3(1), the IOT(B) level samples the MB06-11 bits in the instruction to produce the appropriate DS0-5 levels, and IOT(1) further decodes DS0-2 to produce 0XEN. 0XEN goes to the teleprinter control, drawing KD11(2), to sample DS3-5, resulting in T-PRNTR SEL and T-PRNTR SEL(B).

When the pulse counter IO1 in the I/O control logic sets, signaling the start of the second IOT execute period, it produces IOP2P in conjunction with MB16 (1). IOP2P resets the T-PRNTR FLG in conjunction with T-PRNTR SEL. When IO1 resets, signaling the start of the third IOT execute period, it produces IOP4P in conjunction with MB15(1). IOP4P generates IOT0404 on drawing KD11(2) in conjunction with T-PRNTR SEL.

IOT0404 sets TTO EN and TTO STOP, and gates the 8-bit character into TTO0-7 from the I/O bus. (The character is placed on the I/O bus from the AR during fetch.)

TTO EN(1) sets TTO START on the next TTO CLK pulse. The TTO CLK, operating at the same baud rate as TTI CLK, is running during full-duplex operation or when the keyboard is disabled. TTO START (1) sets TTO OUT ACT(1) on the second TTO CLK, and TTO OUT ACT(1) conditions the DCD gate of the TTO LOAD pulse amplifier. The third TTO CLK after TTO EN(1) thus generates the first TTO LOAD pulse.

TTO OUT ACT(1) sets the TT LINE flip-flop. TT LINE(1) causes the output of solenoid driver W040-B33K to become -15V, thus releasing the printer select solenoid and causing the generation of a start space in the printer, drawing KD11(1).

The first TTO LOAD pulse shifts TTO EN(1) into TTO STOP, TTO STOP into TTO0, and all character bits into the next low-order TTO bit positions. TTO LOAD also resets TTO EN.

Successive TTO LOAD pulses shift the bits down through the register and into TT LINE to produce appropriate marks and spaces. The ninth TTO LOAD shifts TTO STOP(1) into TTO7 and TTO EN(0) into TTO6. Since TTO EN(0) was successively shifted through all positions, they are now all 0s, producing TTO EQ 0 in conjunction with TTO OUT ACT(1).

On the tenth TTO LOAD pulse, TTO EQ 0 resets TTO START and TTO OUT ACT while setting the T-PRNTR FLG. The tenth TTO LOAD also resets TT LINE with TTO7(1), which was the initial TTO STOP state. Thus TT LINE(1) creates a stop mark in the printer.

T-PRNTR FLG(1) causes a PROG INT RQ at R111-D39N. The PROG INT RQ goes to the I/O control logic, drawing KD3(2), to cause a program interrupt request. When the CP honors the interrupt, the TSF instruction in the flag search subroutine determines that the teleprinter has caused the interrupt, and jumps to a service routine which contains another TLS to clear the flag and load another character into the TTO buffer.

9.2.3 Skip on Teleprinter Flag

The TSF instruction (700401) senses the status of the T-PRNTR FLG. If the flag is set, the teleprinter control logic issues a skip request to the CP and the program skips the next instruction.

On drawing KD3(1), MB06-11 are detected to produce T-PRNTR SEL on drawing KD11(2). When IO0 sets, signaling the start of the first IOT execute period, it produces IOP1P in conjunction with MB17(1). IOP1P sets IOP1. IOP1(1) and T-PRNTR SEL(B) sample the state of T-PRNTR FLG. If the T-PRNTR FLG is set, the INT SKP RQ BUS goes to ground, triggering pulse amplifier W012-F18D on drawing KD3(3). This PA pulse is the IO SKIP signal which sets the SKIP flip-flop in the CP, drawing KC14. The TSF instruction idles through its remaining execute period until PC0(1) of the BGN process word generates CI17 in conjunction with SKIP(1). CI17 initiates a carry in the ADR as PC0(1) gates the contents of the PC through the ADR to the MB for the next instruction fetch cycle. Thus the address in the MB contains the PC address + 1.

9.3 HARDWARE READ-IN (HRI) OPERATION

9.3.1 Functional Description

Data on hardware read-in (HRI) tapes is loaded into the PDP-9/L core memory with the key read-in (KRI) mode of operation. To transfer the HRI data, the operator places the tape into the ASR33 tape reader, selects the starting address for the tape on the console, and depresses the READ IN key. The PDP-9/L then selects the paper tape reader for binary mode operation (RSB) and commands the reader to advance three lines of tape. The 9/L reader control logic assembles the data punched in the tape in a reader buffer (RB) and transfers it to the memory buffer.

In the binary mode tape format for read in operations, channels 1 through 6 of each line contain one 6-bit character of an 18-bit word (refer to Figure 9-1). Channel 7 is punched only in the last line of the last word to be read and channel 8 is punched in every line to control the gating of the 6-bit characters into the proper RB bit positions. The current address of the core memory location is retained in the AR where it is incremented by 1 to store successive words in consecutive memory locations. The process continues until the reader encounters a line of tape which has the channel 7 hole punched. The reader then stops and the CP executes the instruction encoded in the last word.

9.3.2 Reader Control

When the READ IN key is depressed, a READ IN (1) pulse generates a RSB pulse (reader select binary) on drawing KD8. RSD is applied to pulse amplifier S603-D10 on drawing KD9(1) and generates an IOT0104. This pulse conditions the reader control circuitry for the start of a RSB sequence. The RDR RUN flip-flop is set and the RDR ALPHA, RDR FLG, RDR2, RDR1, and RB flip-flops are reset. IOT0104 also generates KBD FLG(0) and TT RDR ST pulses via pulse amplifier S603-A01. TT RDR ST collector resets KBD FLG flip-flop, assuring that a premature program interrupt is not generated, and sets the TT RDR RUN flip-flop. A set TT RDR RUN grounds the solenoid driver, permitting one line of tape to advance in the reader.

As the tape passes through the reader, negative and positive TT KBD IN pulses are applied to the TTI shift register. The serial pulses are then shifted through the register in the same sequence described for keyboard operation. When TTI7 resets, representing the teletype start space, the KBD FLG is set. KBD FLG (1) becomes KRI KBD FLG(1) after being jumpered through the W990-A15 module connector, KD9(1). KRI KBD FLG(1) sets the STOP DLY flip-flop to generate STOP DLY pulses. The STOP DLY pulses, in conjunction with RD HOLE 8P + ALPHA pulses, generate a RDR COUNT pulse for each line of tape passing through the reader.

The first STOP DLY pulse and RDR RUN (1) sets the RDR1 flip-flop. This combination enables six bits of data to be transferred from the TTI register to reader buffers 00 through 05, drawing KD9(2). While the RDR COUNT is setting RDR 1, a RDR INDEX triggers the 1-μs RDR GO delay which, in turn, generates another TT RDR ST pulse. This pulse resets the KBD FLG and advances another line of tape into the reader.

The sequence described for the first line of tape data transfer is repeated again. The KBD FLG triggers another STOP DLY, and a second RDR COUNT pulse is generated to set RDR 2. RDR RUN(1) and RDR2 enable the second line of tape data to be transferred from the TTI register to reader buffers 06 through 11. As with the first data transfer, a TT RDR ST pulse resets the KBD FLG and advances another line of tape into the reader.

The data transfer is thus repeated for the third time. However, at the end of this transfer, a program interrupt will occur because a 18-bit tape data word is ready for transfer to the accumulator. With the third STOP DLY pulse, RDR FLG is set to enable the TTI buffer data to be transferred to reader buffers 12 through 17 while the data is being transferred, the KBD FLG and RDR RUN flip-flops are being reset. Resetting the KBD FLG conditions it for the next data transfer and a reset RDR RUN disables the tape reader.

RDR FLG(1) is buffered at S107-E06N, drawing KD9 (1), for a negative RDR FLG(1)B level. This level generates a RD START RQ in the central processor control and starts the sequence for generating IOT0102. (Refer to KEY READ IN description, Section 7.1.9.) IOT0102(1) is applied to the I/O input mixer, drawing KD7(1) and produces a RDR ON BUS level. This level gates the reader buffer data into the input mixer gates and then onto the I/O bus (B). Control memory process word 25 gates the data on the I/O bus into the memory buffer.

9.3.3 High Speed Reader/Punch Circuitry

Some of the circuitry utilized with the high-speed reader/punch option is supplied with the basic PDP-9/L. The circuits in this configuration are three R111 NAND gates and a RDR NO TAPE S202 flip-flop (refer to KD9(1)). During low-speed reader/punch operation (ASR33), these circuits are inhibited with four NO PCO TO GHO levels.

CHAPTER 10
MAINTENANCE

PDP-9/L maintenance theory is directed to the module-replacement level. Downtime caused by malfunctions is thus minimized and the system is more readily kept on-line. The effort is divided into preventive and corrective maintenance.

Preventive maintenance consists of routine periodic checks such as visual inspections, standard maintenance procedures involving cleaning and lubricating, and periodic marginal checking to expose weakening conditions before failure occurs.

Corrective maintenance is instituted when a malfunction occurs, to isolate the problem and to make proper adjustments or replacements. This involves the use of diagnostic routines prepared on paper tape and designed to test the functional units of the system. The procedures and techniques of periodic checking aid in fault isolation. Intermittent error conditions occurring during system operation can be caused to occur continuously by simulating marginal power conditions.

10.1 EQUIPMENT REQUIRED

Maintenance activities for the PDP-9/L system require the standard test equipment and special materials listed in Table 10-1, plus standard hand tools, cleaners, test cables and probes.

10.2 PREVENTIVE MAINTENANCE

Preventive maintenance consists of tasks performed periodically. They prevent failures caused by progressive deterioration or minor damage. A preventive maintenance log book should be established and entries made at each recurring schedule. This data then can be useful in forecasting impending component failure and result in replacement of modules before breakdown.

Preventive maintenance tasks consist of mechanical checks, electrical checks, and marginal voltage checks. All maintenance schedules should be established by conditions at the installation site. Mechanical checks should be performed monthly or as often as required to allow efficient functioning of the air filters and fans. All other tasks should be performed on a regular schedule determined by reliability requirements. A recommended schedule is every 600 operating hours or every four months, whichever comes first.

10.2.1 Mechanical Checks

a. Clean the exterior and interior of equipment cabinets with a vacuum cleaner or clean cloth moistened in nonflammable noncorrosive solvent.

b. Remove and clean the air filters in each added section of the computer. The main rack fans are equipped with a 1/4 in. mesh which should be vacuumed. The floor fans in added racks should be removed and the aluminum mesh filters washed in a mild detergent.

c. Lubricate door hinges and casters with a light machine oil. Wipe off excess oil.

d. Inspect all wiring and cables for cuts, breaks, fraying, deterioration, kinks, strain, and mechanical security. Tape, solder, or replace any defective wiring or cable covering.

e. Inspect the following for mechanical security: keys, switches, control knobs, lamp assemblies, jacks, connectors, transformers, fans, capacitors, elapsed time meter, etc. Tighten or replace as required.

f. Inspect all module mounting panels to assure that each module is securely seated in its connector.

g. Inspect power supply capacitors for leaks, bulges, or discoloration, and replace if necessary.

10.2.2 Electrical Checks

Perform the power-supply output checks listed in Table 10-2. Use a multimeter to make the output voltage measurements under normal load and an oscilloscope to measure the peak-to-peak ripple content on all dc outputs of the supply. The +10 and -15 Vdc supplies are not adjustable; therefore, if output voltage or ripple content is not within specifications, consider the power supply defective and initiate troubleshooting procedures.

Table 10-1
Equipment Required

Equipment	Manufacturer	Designation
Multimeter	Triplett or Simpson	Model 630-NA or 260
Oscilloscope	Tektronix	Type 547
Plug-in-Unit	Tektronix	Type CA
Clip-on Current Probe	Tektronix	Type P6016
X10 Probe	Tektronix	P6008
Recessed tip, 0.065 inch for wire-wrap terminals	Tektronix	206-052
Current Probe Amplifier	Tektronix	Type 131
Hand Unwrapping Tool	Gardner-Denver	500130
Hand-Operated Wire-Wrap Tool with a 26263 bit for 24 AWG Wire and 18840 Sleeve	Gardner-Denver	14H1C
Module Extender	DEC	Type W980
Diagnostic Self-Test Routines	DEC	Memory Address Test Basic Memory Checkerboard Test Instruction Test, Part I Instruction Test, Part II TTY Test, Part I TTY Test, Part II JMP Self Test JMP-Y Self Test JMP-Y Interrupt Test JMS-Y Interrupt Test ISZ Test

Check the operation of the variable output which produces the marginal check voltages. With all marginal check switches in the +10, -15, and MC positions make the following measurements at the supply:

a. Connect a multimeter between the green (-) and orange (+) terminals; set the selector switch on marginal check panel to the -15 MC position, and turn the Variac control knob clockwise to assure that the supply can produce at least -20 Vdc as indicated on the multimeter. Record the indications given on both the marginal check panel voltmeter and on the multimeter. These indications should be equal, ±1V (20V on multimeter = 5V on voltmeter). Connect an oscilloscope to the green terminal, and measure the peak-to-peak ripple content to assure that it is no more than 0.7V. Turn the control knob fully counterclockwise; set the selector switch to the OFF position, and disconnect the multimeter and oscilloscope.

b. Connect the multimeter between the orange (+) and green (-) terminals; set the selector switch to the +10 MC position, and turn the control knob clockwise to assure that the supply can produce at least +20 Vdc. Record as above (20V on multimeter= 10V on voltmeter). Turn the control knob fully counterclockwise, set the selector switch to the OFF position, and disconnect the multimeter.

10.2.3 Marginal Checks

Marginal checking uses MAINDEC Diagnostic programs to test the functional capabilities of the computer while the module operating voltages are biased above and below the nominal levels within specified margins. Biasing the operating voltages aggravates borderline circuit conditions within the module to produce failures detected by the program. Upon error detection, the program usually provides a printout of visual indication which aids in locating the source of the fault, and then halts. Therefore, replacement of modules with marginal components is also possible during scheduled preventive maintenance checks.

The biased operating voltages at which circuits fail should be recorded on the Marginal Check Forms (see Figures 10-1 through 10-8). By plotting the bias voltages obtained during each scheduled preventive maintenance, progressive deterioration can be observed and failures can be predicted, thus providing a means of planned replacement. These checks can also be used as a troubleshooting aid to locate faulty components.

CAUTION

> The -15 Vdc margin should not be increased beyond the ±5V point to prevent possible damage to the logic elements.

Table 10-2
Power Supply Output Checks

Measurement Terminals at Power Supply Output	Nominal Output (Vdc)	Acceptable Output Range (V)	Maximum Output Current (A)	Maximum Peak-to-Peak Output Ripple (V)
Red (+) to Black (-)	+10	+9.5 to 11.5	5.0	0.5
Black (+) to Blue (-)	-15	-13.5 to 16.5	22.0	0.7
Black (+) to Yellow (-)	-30	-28.5 to 32.5	10.0	0.9
Black (+) to Brown (-)	-30	-28.5 to 32.5	5.0	0.9

PDP-9/L MARGINAL CHECK

Revision 4-7-67

Frame Serial

PROGRAM 34H DISPLAY TEST

Frame Type IO (KD09A)

34H Serial No.

MC Switch	Area Tested	Notes	Specification				Measured			
			+10		-15		+10		-15	
			R.T.	E.T.	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.
4	Input Mixer		+5		+2.5					
			-4		-2.5					
9	34H Display Logic		+5		+2.5					
			-4		-2.5					

PROGRAM API TEST

KF09A Serial No.

1	I/O Address Mixer		+5		+2.5					
			-4		-2.5					
8	API Control		+5		+2.5					
			-4		-2.5					

UNLESS OTHERWISE INDICATED ELEVATED TEMPERATURE (E.T.)
SPECIFICATIONS ARE IDENTICAL TO ROOM TEMPERATURE (R.T.)

Signature

Date

Supervisor

White - Field Service Yellow - Q.C. Blue - Customer Pink - Regional Office

Figure 10-1 Display MC Form

PDP-9/L MARGINAL CHECK

Revision 4-7-67

Frame Serial____________

Frame Type IO

PROGRAM TELETYPE TEST PART I

MC Switch	Area Tested	Notes	Specification				Measured			
			+10		-15		+10		-15	
			R.T.	E.T.	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.
2	Reader Control		+5		+2.5					
			-4		-2.5					
3	Reader AMP		+5							
			-4							
4	Input Mixer		+5		+2.5					
			-4		-2.5					
5	IO Control Part 1	1	+5		+2.5					
			-4		-2.5					
6	IO Control Part 2	1	+5		+2.5					
			-4		-2.5					
PROGRAM TELETYPE TEST PART II										
7	Punch Control	2	+4.5		+2.5					
			-4.0		-2.5					
PROGRAM TELETYPE TEST PART II										
4	Input Mixer		+5		+2.5					
			-4		-2.5					
10	Teletype Input		+5		+2.5					
			-4		-2.5					
11	Teletype Output		+5		+2.5					
			-4		-2.5					

NOTES:

1) When taking margins do not exceed this positive-margin since module damage may result.
2) Limited by W520 analog module.

UNLESS OTHERWISE INDICATED ELEVATED TEMPERATURE (E.T.) SPECIFICATIONS ARE IDENTICAL TO ROOM TEMPERATURE (R.T.)

Signature____________

Date____________

Supervisor____________

White - Field Service Yellow - Q.C. Blue - Customer Pink - Regional Office

Figure 10-2 Teletype Tests MC Form

PDP-9/L MARGINAL CHECK

Revision 3-22-68

Frame Serial ____

Frame Type CP

PROGRAM ISZ TEST

MC Switch	Area Tested	Notes	Specification				Measured			
			+10		-15		+10		-15	
			R.T.	E.T.	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.
4	CM Sense Flops	1,2	+5		+2.5					
			-4		-2.5					
10	CM Timing	1,2	+5		+2.5					
			-4		-2.5					

NOTES:

1) When taking margins do not exceed the positive-margin since module damage may result.
2) Operate the program with AC SW's 6, 7 and 8 as ones.

UNLESS OTHERWISE INDICATED ELEVATED TEMPERATURE (E.T.) SPECIFICATIONS ARE IDENTICAL TO ROOM TEMPERATURE (R.T.)

Signature ____

Date ____

Supervisor ____

White - Field Service   Yellow - Q.C.   Blue - Customer   Pink - Regional Office

Figure 10-3 ISZ MC Form

PDP-9/L MARGINAL CHECK

SPECIAL OPTIONS

OPTION

SERIAL NO.

PROGRAM

PROGRAM NO.

DEC ORDER NO.

KEY SHEET NO.

			Specification				Measured			
			+10		−15		+10		−15	
MC Switch or Panel	Area Tested	Notes	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.

NOTES:

UNLESS OTHERWISE INDICATED ELEVATED TEMPERATURE (E.T.)
SPECIFICATIONS ARE IDENTICAL TO ROOM TEMPERATURE (R.T.)

Signature

Date

Supervisor

White - Field Service   Yellow - Q.C.   Blue - Customer   Pink - Regional Office

Figure 10-4 Special Options MC Form

PDP-9/L MARGINAL CHECK

Revision 4-7-67

Frame Serial____________

Frame Type IO

PROGRAM HIGH SPEED READER TEST

MC Switch	Area Tested	Notes	Specification				Measured			
			+10		-15		+10		-15	
			R.T.	E.T.	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.
2	Reader Control		+5		+2.5					
			-4		-2.5					
3	Reader AMP		+5							
			-4							
4	Input Mixer		+5		+2.5					
			-4		-2.5					
5	IO Control Part 1	1	+5		+2.5					
			-4		-2.5					
6	IO Control Part 2	1	+5		+2.5					
			-4		-2.5					
PROGRAM HIGH SPEED PUNCH TEST										
7	Punch Control	2	+4.5		+2.5					
			-4.0		-2.5					
PROGRAM TELETYPE TEST PART II										
4	Input Mixer		+5		+2.5					
			-4		-2.5					
10	Teletype Input		+5		+2.5					
			-4		-2.5					
11	Teletype Output		+5		+2.5					
			-4		-2.5					

NOTES:

1) When taking margins do not exceed this positive-margin since module damage may result.
2) Limited by W520 analog module.

UNLESS OTHERWISE INDICATED ELEVATED TEMPERATURE (E.T.) SPECIFICATIONS ARE IDENTICAL TO ROOM TEMPERATURE (R.T.)

Signature____________

Date____________

Supervisor____________

White - Field Service   Yellow - Q.C.   Blue - Customer   Pink - Regional Office

Figure 10-5 High-Speed Reader MC Form

PDP-9/L MARGINAL CHECK

Revision 3-22-68

Frame Serial ____________ PROGRAM EAE PART 1

Frame Type PROCESSOR EAE Serial No. ____________

MC Switch	Area Tested	Notes	Specification				Measured			
			+10		-15		+10		-15	
			R.T.	E.T.	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.
6	Link, AC	1	+5		+2.5					
			-4		-2.5					
11	MQ	1	+5		+2.5					
			-4		-2.5					
12	EAE Control	1	+5		+2.5					
			-4		-2.5					
4	CM Flip Flops	1	+5		+2.5					
			-4		-2.5					

PROGRAM EAE PART 2

6	Link, AC	1	+5		+2.5					
			-4		-2.5					
11	MQ	1	-5		+2.5					
			-4		-2.5					
12	EAE Control	1	+5		+2.5					
			-4		-2.5					
4	CM Flip Flops	1	+5		+2.5					
			-4		-2.5					

NOTES:

1) When taking margins do not exceed positive margins since module damage may result.

UNLESS OTHERWISE INDICATED ELEVATED TEMPERATURE (E.T.) SPECIFICATIONS ARE IDENTICAL TO ROOM TEMPERATURE (R.T.)

Signature ____________

Date ____________

Supervisor ____________

White - Field Service Yellow - Q.C. Blue - Customer Pink - Regional Office

Figure 10-6 EAE MC Form

PDP-9/L MARGINAL CHECK

Revision 4-7-67

Frame Serial________________

Frame Type PROCESSOR

PROGRAM BASIC EXERCISER

MC Switch	Area Tested	Notes	Specification				Measured			
			+10		−15		+10		−15	
			R.T.	E.T.	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.
1,2	Adder, A Bus	2, 3	+2	+1	+2	+1				
	IO Bus		−2	-1	-1	-1				
3	O Bus, SHX2,		+5		+2.5					
	B Bus		-4		-2.5					
4	CM Sense		+5		+2.5					
	Flip Flops		-4		-2.5					
5	PC, Skip,	1	+5		+2.5					
	CI17		-4		-2.5					
6	Link, AC		+5		+2.5					
			-4		-2.5					
7	IR, MB	1	+5		+2.5					
			-4		-2.5					
8	OPR, AR		+5		+2.5					
			-4		-2.5					
9	CM Address	1	+5		+2.5					
			-4		-2.5					
10	CM Timing, CLK,	1	+5		+2.5					
	RUN, DPY		-4		-2.5					

NOTES:

1) When taking margins do not exceed the positive-margin since module damage may result.
2) All switches indicated must be on together while taking this margin.
3) The W505 low voltage detector module in CP location J21 must be removed before testing this margin.

UNLESS OTHERWISE INDICATED ELEVATED TEMPERATURE (E.T.) SPECIFICATIONS ARE IDENTICAL TO ROOM TEMPERATURE (R.T.)

Signature________________

Date________________

Supervisor________________

White - Field Service Yellow - Q.C. Blue - Customer Pink - Regional Office

Figure 10-7 Basic Exerciser Test MC Form

PDP-9/L MARGINAL CHECK

Frame Serial ____________ Revision ____________

Frame Type MC71 MEMORY MARGIN FORM Program BASIC EXERCISER

MC Switch	Area Tested	Notes	Specification				Measured			
			+10		-15		+10		-15	
			R.T.	E.T.	R.T.	E.T.	R.T.	E.T.	R.T.	E.T.
1,7	Control		+6	+6	+4	+4				
			-4	-4	-3	-3				
2,8	MA Flip-Flops		+6	+6	+4	+4				
			-6	-6	-4	-4				
3,9	Sense Amps	1,2								
4,10	Sense Amp PA's Slice & G010's		+6	+6						
			-4	-4						
5,11	Inhibit Drivers		+6	+6	+4	+4				
			-6	-6	-4	-4				
6,12	Matrix Selector & Drivers		+6	+6	+4	+4				
			-6	-6	-3	-3				
4,10	Sense Amp PA's Clamp & G010's				+4	+4				
					-4	-4				

NOTES:
1) 8.0 volt spread not unbalanced by more than 2V.
2) 6.0 volt spread not unbalanced by more than 2V.

UNLESS OTHERWISE INDICATED ELEVATED TEMPERATURE (E.T.)
SPECIFICATIONS ARE IDENTICAL TO ROOM TEMPERATURE (R.T.)

Signature ____________

Date ____________ Supervisor ____________

White - Field Service Yellow - Q.C. Blue - Customer Pink - Regional Office

Figure 10-8 MC71 Memory MC Form

Since the marginal voltages attainable vary for different circuit changes and/or system configurations, determine the expected check voltages for a specific system from the initial factory test records and any subsequent test records in the maintenance log. A record of margins obtained at the factory for a specific system is provided and serves as a base for all preventive and corrective maintenance procedures.

Margins decrease with time and normal circuit operation deterioration, but this decrease does not affect reliable operation of the machine until there is little or no margin at all. The normal slow rate of margin decay can be used to predict the time at which the system should be examined to prevent sudden failure; margins do provide a measure of circuit performance and can be used to certify corrective or defective operation. However, failure of a system to obtain the same margins year-after-year does not constitute a defect in the operation of the system. For example, if a specific margin decreases at the rate of 0.5V per year, no trouble is indicated, but if this margin suddenly decreases by 1.5V in six months, a need for troubleshooting is indicated.

Marginal check voltages are supplied to various sections of the processor through connections made to the module connectors on each mounting panel via the fan housing connectors. Each marginal check voltage may be adjusted throughout the range of 0 to 20V by means of the control knob and voltmeter located on the marginal check panel (see Figure 10-9). The selector switch on this panel selects either the +10 or the -15 marginal check voltage. Power supply leads to the fan housing connectors, and the mounting panels are color-coded as follows:

Color	Voltage
Orange	+10V marginal check supply
Red	+10V fixed power supply
Black	ground
Blue	-15V fixed power supply
Green	-15V marginal check supply

Marginal check and fixed supply voltages are distributed to each module through the marginal check switches, Figure 10-10. These switches, mounted on the fan housings in each computer section, apply either fixed operating voltages or variable marginal checking voltages from the marginal check panel to pins A (+10V) and B (-15V) of the computer modules.

Three sets of twelve switches each are designated 01 through 12. Each set applies the selected voltages to specific modules in its associated section. Drawings KC09-C-9, MC71-0-15, and KD09-C-13 are the MC switch configuration drawings for the CP, core memory, and I/O control sections, respectively. The drawings designate by number the MC switches which apply their voltages to the specific modules.

The diagnostic routines which are used with the marginal check facilities are supplied as perforated paper tapes. A complete description and instructions accompany each tape. The following list summarizes the functions of each basic diagnostic routine. Other test routines are available for optional equipment. The programs are listed in the recommended execution sequence.

a. Memory Address Test - Checks the memory system to ensure that all memory locations not occupied by the program in a given 4K memory stack can be uniquely addressed. It does this by writing the address of a memory location into itself and checking to see that it is there. The complement of the address is also written to ensure that all bits of a word can be accessed. Checks are also made to ensure that only one memory location is affected whenever memory is addressed, and that cores of different memory locations are not shorted inside the memory stack. Errors are indicated to the operator via the teleprinter.

b. Basic Memory Checkerboard Test - Checks for core failure on half-selected lines under worst case conditions in the 4K memory system.

c. Instruction Test Part I - Tests all operate group instructions and the memory reference instructions LAC, ADD, SAD, TAD, and XOR. Individual error halts are used to describe the failing tests.

d. Instruction Test Part II - A continuation of Part I, testing the following instructions and machine functions: DZM, DAC, ISZ, JMP, JMS, XCT, AUTO-INDEX, INDIRECT ADDRESS, TIME CLOCK, DBR and INTERRUPT. Individual error halts are used to describe the failing tests.

e. TTY Test Part I - Verifies the operational status of an ASR33/35 Teletype and associated logic. This part tests the input and output logic and the teletype reader. The programs are selected and controlled by the accumulator switches.

f. TTY Test Part II - Verifies the operational status of an ASR33/35 Teletype and associated control logic. This part tests the teleprinter, punch,

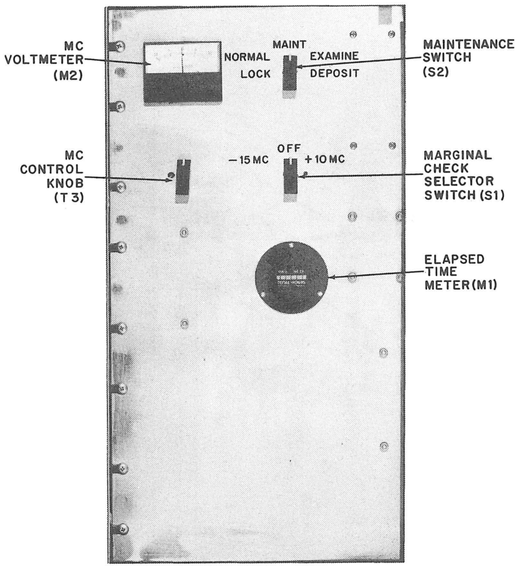


Figure 10-9 Marginal Check Panel

keyboard and the printer, punch, reader combinations. The programs are selected and controlled by the accumulator switches.

g. JMP Self Test - Checks the PDP-9/L to ensure that the JMP␣. instruction can be executed properly. The computer is held in a JMP to the current location instruction for a definite time interval. If, during this interval, the JMP instruction fails, the error will be indicated to the operator. If the JMP instruction does not fail, it is moved elsewhere and the check is repeated. All memory locations not occupied by the program are tested.

h. JMP-Y Interrupt Test - Determines if the PDP-9/L will complete a JMP-Y (where Y is some random value) instruction before it goes into program interrupt. This is done by setting an I/O flag and then transferring control to an ION/JMP Y instruction group (located at a random place in memory). The computer should complete the JMP Y instruction before it goes into program interrupt. If no error occurs, the ION/JMP Y instruction group is moved to other random memory locations and the test is repeated. Errors are indicated to the operator via the Teletype or error halts.

i. JMS-Y Interrupt Test - Determines if the PDP-9/L will complete a JMS Y (where Y is some random value) instruction before it goes into program interrupt. This is done by setting an I/O flag and then transferring control to an ION/JMS Y instruction group (which is located at some random place in memory). The computer should complete the JMS Y instruction before it goes into program interrupt. If no error occurs, the ION/JMS Y instruction group is moved to other random memory locations and the test is repeated. Errors are indicated to the operator via the Teletype or error halts.

j. ISZ Test - Checks the operation of the ISZ instruction. Various checks of the ISZ instruction are made, including ISZ of 777777_8 to 0_8 on all memory locations, and ISZ of random numbers stored in random memory locations from random memory locations. Errors are indicated to the operator via the teleprinter.

To perform the marginal checks, proceed as follows:

a. Turn power off. (Note: Always do so before selecting switches.)

b. See that all marginal check switches are in the FIXED position.

c. Set the selector switch on the maintenance panel to the +10 MC position.

d. Adjust the output of the marginal check power supply so that the voltmeter indicates 0V.

e. Refer to drawings KC09-C-9, MC71-0-15, and KD09-C-13, and set the appropriate FIXED/MC switches for the module(s) to be checked to the MC positions.

f. Start computer operation in a diagnostic program or routine which fully utilizes the circuits in the modules to be tested.

g. Decrease the marginal check power supply output until normal system operation is interrupted, and record the marginal check voltage. At this point, marginal transistors can be located and replaced, if desired. Readjust the marginal check power supply output to the nominal 0V level.

h. Restart computer operation. Increase the marginal check supply output until normal computer operation is interrupted and record the marginal check voltage. Again it is possible to locate and replace transistors. Readjust the marginal check power supply to the nominal 0V level.

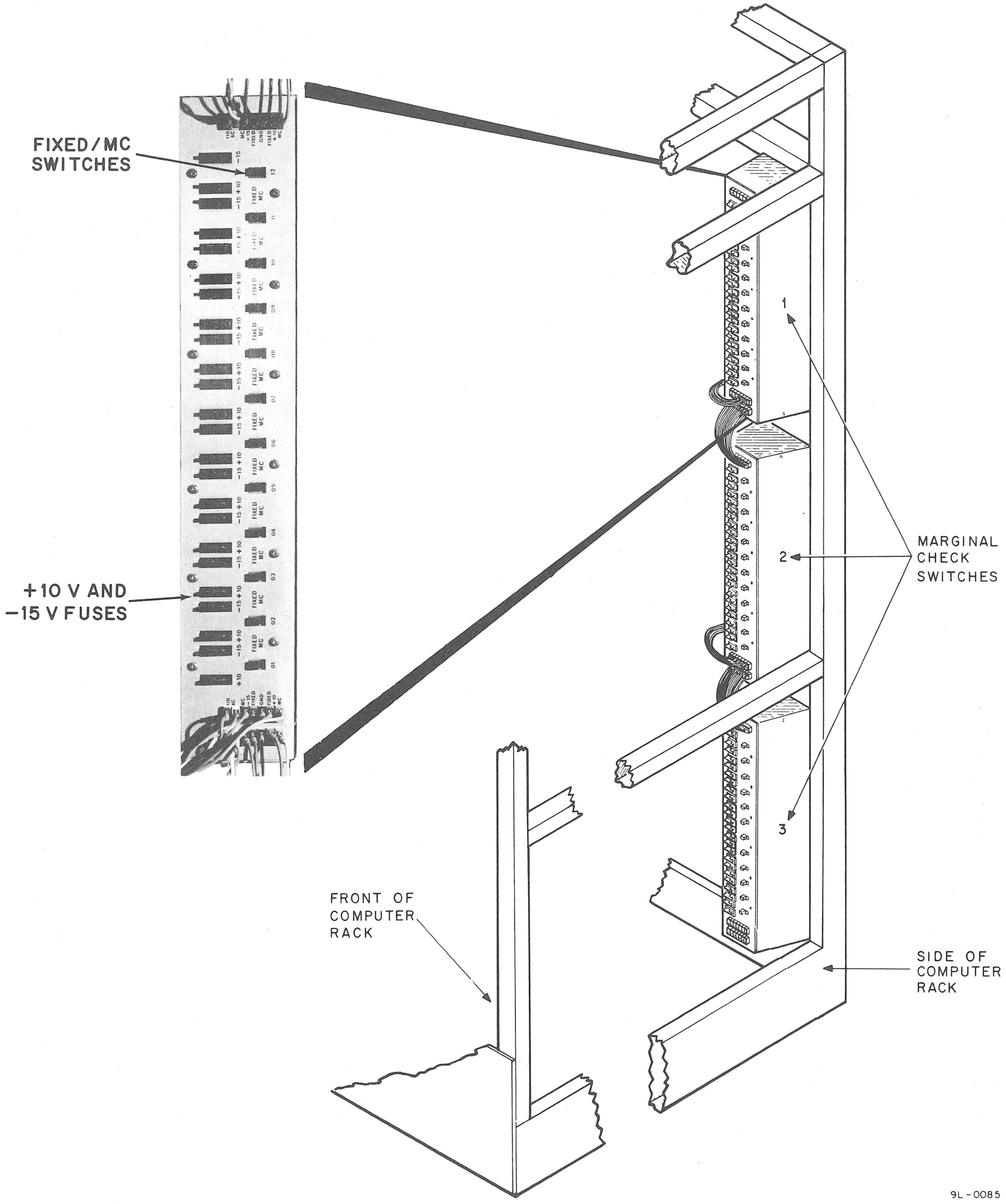


Figure 10-10 Marginal Switch Panel

i. Return the FIXED/MC switches to the FIXED positions.

j. Repeat steps e through h for other modules to be checked.

k. Set the selector switch on the maintenance panel to the -15 MC position and adjust the output until the marginal check voltmeter indicates 0V.

l. Refer to the drawings listed in step e above and set the appropriate FIXED/MC switches for the modules to be checked to the MC positions.

m. Repeat steps f through h. Return the FIXED/MC switches to the FIXED positions.

n. Repeat steps k through m for each test.

o. Set the maintenance panel selector switch to the OFF position.

10.3 CORRECTIVE MAINTENANCE

When a malfunction occurs, the condition should be analyzed and corrected as suggested in the following procedures. No test equipment or special tools are required for corrective maintenance other than a broad-band oscilloscope and a standard multimeter. However, a clip-on current probe such as the Tektronix Type P6016 with a Type 131 Current Probe Amplifier is very helpful in monitoring memory currents. The best corrective maintenance tool is a thorough understanding of the physical and electrical characteristics of the equipment. Persons responsible for maintenance who are thoroughly familiar with the system concept, the logic drawings, operation of specific module circuits, and the location of mechanical and electrical components can readily interpret diagnostic routine printouts for isolating malfunctions.

Diagnosis and remedial action for a fault condition usually proceed in the following steps:

a. Preliminary investigation: gather all information to determine the physical and electrical security of the computer.

b. System troubleshooting: define the errors by locating the fault to within a module, through use of diagnostic routines, control panel troubleshooting or signal tracing.

c. Replace defective module or modules to get the system on-line.

d. Log entry to record pertinent data.

Once the system is again in operation, defective parts within a module can be located and repaired or replaced. Repaired modules should be verified by a validation test after repair.

Before commencing troubleshooting procedures record all unusual functions of the machine prior to the fault and all observable symptoms. In addition, note the program in progress, condition of operator console indicators, etc. This information should be referenced to the maintenance log to determine whether this type of fault has occurred before or if there is any cyclic history of this fault, and to ascertain how the condition was previously corrected.

When the entire machine fails, perform a visual inspection to determine the physical and electrical security of all power sources, cables, connectors, etc. Assure that the power supplies are working properly and that there are no power short circuits by performing the power supply checks as described under preventive maintenance. Check the conditions of the air filters in the fan housings. If the filters become clogged, the temperature within the cabinet might rise sufficiently to cause marginal semiconductors to become defective.

10.3.1 Module Handling

Turn off all power before extracting or inserting modules. Access to controls on the module for use in adjustment, or access to points used in signal tracing can be gained by removing the module (use a straight, even pull to prevent twisting of the printed-wiring board), connecting a Type W980 Module Extender into the vacated module connector in the mounting panel, and then inserting the module into the extender.

10.3.2 Built-in Checks

Prior to performing a more complex troubleshooting routine, some flaws can be located by using the built-in testing routines activated by the maintenance switch on the maintenance panel. With this switch in MAINT position; and with the console REPT, START, and START HOLD switches in the up position, all active CPU registers and internal transfer paths may be verified. This test circulates a self-incrementing count at a rate which is adjustable by the repeat speed control on the console. The DEPOSIT position allows an adjustable DEPOSIT function, while EXAMINE permits examination of contents at an adjustable speed.

10.3.3 System Troubleshooting

Begin troubleshooting by performing the operation in which the malfunction was initially observed, using the same program that indicated the fault. Thoroughly check the program for proper control settings. Assure that the PDP-9/L, and not the peripheral equipment, is actually at fault before continuing with corrective maintenance procedures. Faults in equipment that transmits or receives information, or improper connection of the system frequently give indications very similar to those caused by computer malfunctions. Faulty ground connections between peripheral equipment and the computer are a common source of trouble. By analyzing the portion of the program being performed and the general condition of the indicators, the fault can usually be isolated.

10.3.4 Section Troubleshooting

If the fault has been isolated to the computer but cannot be immediately localized to a specific logic function, it can be further isolated to either the core memory or the central processor sections. When the fault has been narrowed to a logic element, continue troubleshooting to locate the defective module or component by means of signal tracing.

If the entire memory system fails, use a multimeter to check the outputs of the 712 Power Supplies. If the supply is defective, troubleshoot it and correct the cause of the trouble. For a selected core a current spike can be observed on an oscilloscope, and a missing spike then represents a malfunctioning address. Perform the Memory Address Test to locate defective core memory addresses. Record all addresses which fail and inspect the record for common bits. Refer to engineering drawings, and check the selectors that decode common bits of the failing addresses. Also check the associated resistor board and memory matrix module.

To locate the cause of a specific address failure, use an oscilloscope and current probe to trace read and write current while using the DEPOSIT and ADDRESS switches, or the DEPOSIT and EXAMINE positions of the maintenance selector switch on the maintenance panel. If an address is dropping bits, deposit all 1s. If an address is picking up bits, deposit all 0s. The Memory Checkerboard Test can be used to troubleshoot all other memory conditions.

10.3.5 Logic Troubleshooting

If the instructions do not seem to be functioning properly, perform the Instruction Test. This test halts to indicate instructions that fail. When an instruction fails, as indicated by the console indicators when the program stops, consult the descriptive listing to obtain an interpretation that will localize the fault.

If the computer interrupt system or the teleprinter do not seem to be functioning properly, perform the TTY Test Program. If the fault has been located within a functional logic element, program the computer to repeat an operation which uses all functions of that element. Use the oscilloscope to trace signal flow through the suspected logic element. Oscilloscope sweep can be synchronized by control signals or clock pulses, which are available on individual module terminals at the wiring side of the module mounting panels. Circuits transferring signals to/from external equipment are most likely to cause difficulty. Trace output signals from the interface connector back to the origin, and trace input signals from the connector to the final destination. The signal tracing method can be used to certify signal qualities such as pulse amplitude, duration, rise time, and the correct timing sequence.

10.3.6 Module Troubleshooting

Once the fault has been isolated to a specific module turn off computer power and carefully remove the suspected module. Inspect the receptacle for wear or damaged contacts. If found to be good, replace the module with one known to be good and rerun the last Diagnostic Program. If the computer performs properly, return the system to operating status.

APPENDIX A
BASIC INSTRUCTION REPERTOIRE

MEMORY REFERENCE INSTRUCTIONS

Mnemonic Symbol	Octal Code	Machine Cycles	Operation Executed
CAL	00	2	Call subroutine. The address portion of this instruction is ignored. The action is identical to JMS 20.
DAC Y	04	2	Deposit AC. The contents of the AC are deposited in the memory at location Y.
JMS Y	10	2	Jump to subroutine. The contents of the PC and the Link, memory EXD mode, and memory project mode status are deposited in memory Y. The next instruction is taken from Y + 1.
DZM Y	14	2	Deposit zero in memory. Zero is deposited in memory Y.
LAC Y	20	2	Load AC. The contents of Y are loaded into the AC.
XOR Y	24	2	Exclusive OR. The exclusive OR is performed between the contents of Y and the contents of the AC, with the result left in the AC.
ADD Y	30	2	Add (1's complement). The contents of Y are added to the contents of the AC in 1's complement arithmetic and the result is left in the AC.
TAD Y	34	2	2's complement add. The contents of Y are added to the contents of the AC in 2's complement arithmetic and the result is left in the AC.
XCT Y	40	1+	Execute. The instruction in memory Y is executed.
ISZ Y	44	2	Increment and skip if 0. The contents of Y are incremented by 1 in 2's complement arithmetic. If the result is 0, the next instruction is skipped.
AND Y	50	2	AND. The logical operation AND is performed between the contents of Y and the contents of the AC with the result left in the AC.

Memory Reference Instructions (Cont)

Mnemonic Symbol	Octal Code	Machine Cycles	Operation Executed
SAD Y	54	2	Skip if AC is different from Y. The contents of Y are compared with the contents of the AC. If the numbers are different, the next instruction is skipped.
JMP Y	60	1	Jump to Y. The next instruction to be executed is taken from memory Y.

INPUT/OUTPUT TRANSFER INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
		Program Interrupt
IOF	700002	Interrupt off. Disable the PI facility.
ION	700042	Interrupt on. Enable the PI facility.
		I/O Equipment
IORS	700314	Input/output read status. The status of given flags replace the contents of the AC.
CAF	703302	Clear all flags
DBR	703344	Debreak and restore the main program following program interrupt.
		Teletype Keyboard
KSF	700301	Skip if the keyboard flag is set to 1.
KRB	700312	Read the keyboard buffer. The contents of the buffer are placed in AC 10-17 and the keyboard flag is cleared.
KRS	700322	Select keyboard reader if START switch is engaged.
		Teletype Teleprinter
TSF	700401	Skip if the teleprinter flag is set.
TCF	700402	Clear the teleprinter flag.
TLS	700406	Load teleprinter buffer. The contents of AC 10-17 are placed in the buffer and printed. The flag is cleared before transmission takes place and is set when the character has been printed.

OPERATE INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
NOP	740000	No operation. Causes a 1-cycle program delay.
CMA	740001	Complement accumulator. Each bit of the AC is complemented.
CML	740002	Complement Link.
OAS	740004	Inclusive OR DATA switches. The word set into the DATA switches is OR combined with the contents of the AC, the result remains in the AC.
RAL	740010	Rotate accumulator left. The contents of the AC and Link are rotated one position to the left.
RAR	740020	Rotate accumulator right. The contents of the AC and Link are rotated one position to the right.
HLT	740040	Halt. The program is stopped at the conclusion of the execute cycle.
SMA	740100	Skip on minus accumulator. If the contents of the AC are negative (2's complement) the next instruction is skipped.
SZA	740200	Skip on zero accumulator. If the contents of the AC equal 0 (2's complement), the next instruction is skipped.
SNL	740400	Skip on non-zero Link. If the Link contains a 1, the next instruction is skipped.
SKP	741000	Skip. The next instruction is unconditionally skipped.
SPA	741100	Skip on positive accumulator. If the contents of the AC are 0 (2's complement) or a positive number, the next instruction is skipped.
SNA	741200	Skip on non-zero accumulator. If the contents of the AC are not 0 (2's complement), the next instruction is skipped.
SZL	741400	Skip on zero Link. If the Link contains a 0, the next instruction is skipped.
RTL	742010	Rotate two left. The contents of the AC and the Link are rotated two positions to the left.
RTR	742020	Rotate two right. The contents of the AC and the Link are rotated two positions to the right.

Operate Instructions (Cont)

Mnemonic Symbol	Octal Code	Operation Executed
CLL	744000	Clear Link. The Link is cleared.
STL	744002	Set Link. The Link is set to 1.
RCL	744010	Clear Link, then rotate left. The Link is cleared, then the Link and AC are rotated one position left.
RCR	744020	Clear Link, then rotate right. The Link is cleared, then the Link and AC are rotated one position right.
CLA	750000	Clear accumulator. Each bit of the AC is cleared.
CLC	750001	Clear and complement accumulator. Each bit of the AC is set to 1.
LAS	750004	Load accumulator from switches. The word set into the DATA switches is loaded into the AC.
GLK	750010	Get Link. The content of the Link is set into AC17.
LAW N	76XXXX	Load the AC with 76 +n, where n equals 00000 → 07777.

Digital Equipment Corporation
Maynard, Massachusetts

printed in U.S.A.