

PDP-9
MAINTENANCE MANUAL
VOLUME 1

1st printing March 1967
2nd printing June 1967
3rd printing Nov. 1967
4th printing Feb. 1968
5th printing Aug. 1969
6th printing Nov. 1971
7th printing April 1972

The material in this manual is for informational purposes and is subject to change without notice.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC	PDP
FLIP CHIP	FOCAL
DIGITAL	COMPUTER LAB

CONTENTS

CHAPTER 1 INTRODUCTION

1.1	Scope	1-1
1.2	Purpose	1-1
1.3	Physical Description	1-1
1.4	Functional Description	1-3
1.5	Options	1-5
1.6	Reference Documents and Programs	1-5
1.7	Reference Conventions	1-5
1.8	Terminology	1-5
1.9	Engineering Drawings and Circuit Schematics	1-7
1.10	System Specifications	1-7
1.10.1	Dimensions	1-7
1.10.2	Operating Characteristics	1-7
1.10.3	Functional Characteristics	1-8

CHAPTER 2 OPERATION

2.1	Scope	2-1
2.2	Standard Controls and Indicators	2-1
2.2.1	Operator Console	2-1
2.2.2	Marginal Check Panel and Maintenance Panel	2-4
2.2.3	Paper Tape Reader/Punch	2-6
2.2.4	Teletype Unit	2-7
2.2.5	Power Control 841A	2-8
2.2.6	Core Memory Banks	2-9
2.3	Operating Procedures	2-9
2.3.1	Manual Data Storage and Modification	2-9
2.3.2	Storing Binary Data Using READ IN Key	2-10
2.3.3	Storing Data Under Program Control	2-11
2.3.4	Assembling Programs	2-11
2.3.5	Teletype Code	2-12
2.3.6	Local Teletype Operation	2-12
2.3.7	Maintenance Programs	2-12

CONTENTS (cont)

CHAPTER 3 SYSTEM DESCRIPTION

3.1	Computer Organization	3-1
3.1.1	Central Processor	3-1
3.1.1.1	Control Memory (CM)	3-1
3.1.1.2	Adder (ADR)	3-1
3.1.1.3	Accumulator (AC)	3-1
3.1.1.4	AC Link (LINK)	3-5
3.1.1.5	Arithmetic Register (AR)	3-5
3.1.1.6	Optional Multiplier-Quotient Register (MQ)	3-5
3.1.1.7	Program Counter (PC)	3-5
3.1.1.8	Instruction Register (IR)	3-5
3.1.1.9	Memory Buffer Register (MB)	3-5
3.1.2	Core Memory System	3-6
3.1.2.1	Direct Addressing	3-7
3.1.2.2	Indirect Addressing	3-7
3.1.2.3	Autoindexing	3-7
3.1.2.4	Extend Mode Addressing	3-8
3.1.3	I/O Control	3-8
3.2	Instruction Word Formats	3-8
3.2.1	Memory Reference Instructions	3-9
3.2.2	Augmented Instructions	3-9
3.2.2.1	Operate	3-9
3.2.2.2	Input/Output Transfer	3-9
3.2.2.3	Optional EAE	3-10
3.3	Data Formats	3-11
3.3.1	Sign and Magnitude Notation	3-11
3.3.2	Complement Notations	3-11
3.3.3	Data Words	3-13
3.3.3.1	Magnitude of Data Words	3-14
3.3.3.2	Floating-Point Formats	3-14
3.3.4	Scaling	3-15
3.4	Control Memory System	3-16

CONTENTS (cont)

3.4.1	Organization	3-16
3.4.2	Timing and Control	3-18
3.4.3	Control Memory MC09A	3-19
3.4.4	Address Selectors G210	3-19
3.4.5	Current Sources	3-20
3.5	Central Processor Logic	3-20
3.5.1	Fetch Cycle	3-20
3.5.2	Defer Cycle	3-26
3.5.3	Autoindexing	3-28
3.5.4	IA0 Cycle	3-29
3.5.5	Execute Cycle	3-29
3.5.6	Memory Reference Instructions	3-31
3.5.6.1	1s Complement Add (ADD)	3-31
3.5.6.2	Logical AND (AND)	3-33
3.5.6.3	Call Subroutine (CAL)	3-34
3.5.6.4	Deposit Accumulator (DAC)	3-36
3.5.6.5	Deposit Zero in Memory (DZM)	3-37
3.5.6.6	Increment and Skip if Zero (ISZ)	3-38
3.5.6.7	Jump (JMP)	3-39
3.5.6.8	Jump to Subroutine (JMS)	3-40
3.5.6.9	Load the Accumulator (LAC)	3-41
3.5.6.10	Skip if AC Differs (SAD)	3-42
3.5.6.11	2s Complement Add (TAD)	3-43
3.5.6.12	Execute (XCT)	3-44
3.5.6.13	Exclusive OR (XOR)	3-45
3.5.7	Operate (OPR) Instructions	3-46
3.5.7.1	No Operation (NOP)	3-47
3.5.7.2	Complement the Accumulator (CMA)	3-47
3.5.7.3	Complement the LINK (CML)	3-48
3.5.7.4	Inclusive OR the AC/DATA Switches (OAS)	3-48
3.5.7.5	Rotate One Position Left (RAL)	3-49
3.5.7.6	Rotate One Position Right (RAR)	3-49
3.5.7.7	Halt Program (HLT)	3-50

CONTENTS (cont)

3.5.7.8	Skip on Minus Accumulator (SMA)	3-50
3.5.7.9	Skip on Zero Accumulator (SZA)	3-50
3.5.7.10	Skip on Non-Zero LINK (SNL)	3-51
3.5.7.11	Unconditional Skip (SKP)	3-52
3.5.7.12	Skip on Positive Accumulator (SPA)	3-52
3.5.7.13	Skip on Non-Zero Accumulator (SNA)	3-53
3.5.7.14	Skip on Zero LINK (SZL)	3-53
3.5.7.15	Rotate Two Positions Left (RTL)	3-54
3.5.7.16	Rotate Two Positions Right (RTR)	3-54
3.5.7.17	Clear the LINK (CLL)	3-55
3.5.7.18	Set the LINK (STL)	3-55
3.5.7.19	Clear the LINK and Rotate One Position Left (RCL)	3-55
3.5.7.20	Clear the LINK and Rotate One Position Right (RCR)	3-56
3.5.7.21	Clear the Accumulator (CLA)	3-56
3.5.7.22	Clear and Complement the Accumulator (CLC)	3-56
3.5.7.23	Load AC from DATA Switches (LAS)	3-56
3.5.7.24	Get the LINK Content (GLK)	3-56
3.5.7.25	Load the Accumulator with 76XXXX (LAW)	3-57
3.5.8	Input/Output Transfer (IOT) Instructions	3-57
3.6	Core Memory System	3-60
3.6.1	Organization	3-60
3.6.2	Core Array and Field Selection	3-60
3.6.3	Control and Timing	3-63
3.6.4	Address Selectors G219	3-68
3.6.4.1	Digit Drive Selection	3-69
3.6.4.2	Word Drive Selection	3-69
3.6.5	Core Sensing Circuits	3-72
3.6.6	Current Sources	3-73
3.6.6.1	Control Module G804	3-74
3.6.6.2	Negative Regulator G805	3-74
3.6.6.3	Resistor Boards G622	3-74
3.7	Manual Control	3-75
3.7.1	Primary Power Distribution	3-75

CONTENTS (cont)

3.7.2	Power Supplies	3-75
3.7.3	Main Clock	3-76
3.7.4	Manual Controls	3-77
3.7.4.1	Power Turn-On	3-77
3.7.4.2	START Key	3-78
3.7.4.3	PROGRAM STOP Key	3-80
3.7.4.4	CONTINUE Key	3-80
3.7.4.5	DEPOSIT	3-81
3.7.4.6	DEPOSIT NEXT	3-83
3.7.4.7	EXAMINE	3-84
3.7.4.8	EXAMINE NEXT	3-84
3.7.4.9	READ IN Key	3-85
3.7.4.10	I/O RESET Key	3-89
3.7.4.11	REPT and REPEAT SPEED Switches	3-89
3.7.4.12	SING INST and SING STEP Switches	3-90
3.7.4.13	CLK Switch	3-90
3.7.4.14	Other Controls	3-90
3.7.5	Display Indicators	3-91
3.7.5.1	REGISTER Indicator	3-91
3.7.5.2	LINK Indicator	3-92
3.7.5.3	MEMORY BUFFER Indicator	3-92
3.7.5.4	INSTRUCTION Indicator	3-92
3.7.5.5	PIE Indicator	3-93
3.7.5.6	CLK Indicator	3-93
3.7.5.7	SING STEP Indicator	3-93
3.7.5.8	SING INST Indicator	3-93
3.7.5.9	REPT Indicator	3-93
3.7.5.10	PRGM STOP Indicator	3-93
3.7.5.11	DATA Indicator	3-94
3.7.5.12	Other Indicators	3-94
3.7.6	Marginal Check Panel	3-94
3.7.7	Maintenance Panel	3-95
3.7.7.1	NORMAL	3-95

CONTENTS (cont)

3.7.7.2	LOCK	3-96
3.7.7.3	MAINT	3-96
3.7.7.4	EXAMINE	3-98
3.7.7.5	DEPOSIT	3-98
3.8	I/O CONTROL	3-98
3.8.1	Program-Controlled Data Transfers	3-99
3.8.1.1	I/O Bus Connections	3-100
3.8.1.2	Block Diagram Discussion	3-100
3.8.1.3	Device Selector W103	3-103
3.8.1.4	Input Transfers	3-103
3.8.1.5	Output Transfers	3-105
3.8.1.6	I/O Skip Facility	3-105
3.8.1.7	Program Interrupt Facility	3-106
3.8.1.8	I/O Status Check Facility	3-110
3.8.2	Data Channel Transfers	3-111
3.8.2.1	I/O Bus Connections	3-114
3.8.2.2	Multiplexer W104	3-114
3.8.2.3	Break Synchronization	3-114
3.8.2.4	WC Cycle	3-117
3.8.2.5	CA Cycle	3-118
3.8.2.6	Data Input Cycle	3-119
3.8.2.7	Data Output Cycles	3-120
3.8.2.8	Add-to-Memory Facility	3-121
3.8.3	DMA Channel Transfers	3-122
3.8.3.1	Interface Connections	3-122
3.8.3.2	Transfer Operations	3-123
3.8.4	API Channel Transfers	3-125
3.8.5	Transfer Priorities	3-126
3.9	Basic I/O Device Control	3-126
3.9.1	Teletype	3-126
3.9.1.1	Keyboard Control	3-127
3.9.1.2	Transfer to AC	3-129
3.9.1.3	Skip on Keyboard Flag	3-130

CONTENTS (cont)

3.9.1.4	Clear Teleprinter Flag and Load the Buffer	3-130
3.9.1.5	Skip on Teleprinter Flag	3-132
3.9.1.6	LINE/OFF/LOCAL Switch	3-132
3.9.2	Tape Reader	3-132
3.9.2.1	Select Reader Binary	3-133
3.9.2.2	Select Reader Alpha	3-136
3.9.2.3	Transfer to AC	3-137
3.9.2.4	Skip on Reader Flag	3-137
3.9.2.5	Reader No Tape Logic	3-138
3.9.2.6	FEED Button	3-138
3.9.3	Tape Punch	3-138
3.9.3.1	Select Punch Binary	3-139
3.9.3.2	Select Punch Alpha	3-141
3.9.3.3	Skip on Punch Flag	3-141
3.9.3.4	Clear Punch Flag	3-141
3.9.3.5	Clear AC, Clear Flag, and Select Punch	3-141
3.9.3.6	POWER Button and TAPE Switch	3-141
3.9.3.7	FEED Button	3-142
3.9.4	Real-Time Clock	3-142

CHAPTER 4 MAINTENANCE

4.1	Maintenance Philosophy	4-1
4.2	Tools and Test Equipment	4-1
4.3	Module Handling	4-2
4.4	Marginal Check Switches	4-2
4.5	Preventive Maintenance	4-3
4.5.1	Mechanical Checks	4-3
4.5.2	Power Supply Checks	4-4
4.5.3	Core Memory Current Check	4-5
4.5.4	Sense Amplifier Check	4-5
4.6	Corrective Maintenance	4-5
4.6.1	Marginal Checking with Diagnostic Programs	4-6
4.6.2	System Troubleshooting	4-11

CONTENTS (cont)

4.6.2.1	Memory Troubleshooting	4-11
4.6.2.2	Logic Troubleshooting	4-12
4.6.2.3	Signal Tracing	4-12
4.6.2.4	Aggravation Tests	4-13
4.6.3	Circuit Troubleshooting	4-13
4.6.3.1	In-Line Dynamic Tests	4-13
4.6.3.2	In-Line Marginal Checks	4-13
4.6.3.3	Static Bench Tests	4-14
4.6.3.4	Dynamic Bench Tests	4-15
4.6.4	Tape Reader Adjustments	4-16
4.6.4.1	Solenoid Drivers	4-16
4.6.4.2	Photo Amplifier	4-16
4.6.4.3	Clock and Accelerator	4-17
4.6.4.4	Mechanical Adjustments	4-17
4.6.5	Module Repair	4-17
4.7	Replacement Modules	4-17
Appendix		
A	Option List	A-1
B	Basic Instruction Repertoire	B-1

ILLUSTRATIONS

1-1	Programmed Data Processor PDP-9	1-2
1-2	Basic PDP-9 Layout, Front and Rear	1-3
1-3	PDP-9 System Configuration, Block Diagram	1-4
2-1	Operator Console	2-1
2-2	Marginal Check Panel and Maintenance Panel	2-5
2-3	Paper Tape Reader/Punch PC09	2-6
2-4	KSR 33 Teletype Unit	2-7
2-5	Power Control 841A	2-8
3-1	System Block Diagram, Basic PDP-9	3-3
3-2	Memory Reference Instruction Word Format	3-8
3-3	OPR Instruction Word Format	3-9

ILLUSTRATIONS (cont)

3-4	IOT Instruction Word Format	3-10
3-5	IOT Instruction Cycles	3-11
3-6	Data Word Format	3-13
3-7	Floating Point Data Format	3-15
3-8	Control Memory System, Block Diagram	3-17
3-9	CM Line Selection, Line 21	3-19
3-10	Fetch Cycle Timing	3-21
3-11	Fetch Cycle Flow, Sheet 1	3-22
	Sheet 2	3-23
3-12	Defer Cycle Timing	3-27
3-13	Defer Cycle Flow	3-28
3-14	Execute Cycle Timing	3-30
3-15	AND Logic	3-34
3-16	SAD Logic	3-43
3-17	XOR Logic	3-45
3-18	IOT Timing	3-58
3-19	Core Memory System Block Diagram	3-61
3-20	Planar Array Wiring	3-62
3-21	Core Wiring for One Bit Position	3-64
3-22	Core Memory Read/Write Timing	3-65
3-23	Digit Drive Selection, Bit 0, Line Y_0	3-70
3-24	Word Drive Selection, Line X_0	3-71
3-25	Program START Timing	3-78
3-26	Initial Set-up DEPOSIT or EXAMINE Timing	3-82
3-27	READ IN Mode Timing	3-86
3-28	REGISTER DISPLAY Signal Paths	3-92
3-29	Program-Controlled I/O Interface	3-101
3-30	I/O Device Control Logic	3-102
3-31	Device Selector W103, Schematic Diagram	3-104
3-32	Program Interrupt Timing	3-108
3-33	Multiplexer W104, Logic Diagram	3-115
3-34	DCH Break Synchronization	3-116

ILLUSTRATIONS (cont)

3-35	Adapter/Multiplexer DM09A, Block Diagram	3-124
3-36	Paper Tape and Reader Buffer Formats	3-134
4-1	Marginal Check Switches	4-3

TABLES

1-1	Related Documents	1-8
1-2	System Program Tapes	1-9
1-3	Maintenance Program Tapes	1-9
2-1	Operator Console, Controls and Indicators	2-2
2-2	Marginal Check and Maintenance Panels, Controls and Indicators	2-5
2-3	Tape Reader/Punch, Controls and Indicators	2-7
2-4	Teletype Controls	2-7
2-5	Power Control 841A, Controls and Indicators	2-9
2-6	Read-In Mode (RIM) Loader Program	2-10
3-1	Reserved Core Memory Locations	3-6
3-2	Complementary Numbers	3-13
3-3	I/O Status Bit Assignments	3-110
3-4	Teletype Code (ASCII)	3-128
4-1	Tools and Test Equipment	4-1
4-2	Power Supply Output Checks	4-4

CHAPTER 1 INTRODUCTION

1.1 SCOPE

This manual provides operation and maintenance information for the Programmed Data Processor PDP-9, manufactured by Digital Equipment Corporation, Maynard, Massachusetts. The manual consists of two volumes, Volume I containing Chapters 1 through 4, and Volume II containing Chapter 5. Volume I describes the basic computer system (Chapter 1) and discusses the logic circuits in terms of the computer's instruction repertoire (Chapter 3). Manual operations (Chapter 2) and maintenance considerations (Chapter 4) are also included. Volume II contains a complete set of engineering drawings for the basic computer system; customer-ordered options are treated in separate option instruction manuals.

1.2 PURPOSE

This manual is one of several documents related to the PDP-9. It provides the user with a basic understanding of the system capabilities. The levels of discussion assume 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.

1.3 PHYSICAL DESCRIPTION

With the exception of the KSR 33 Teletype Unit, the basic PDP-9 (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 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 8,192-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 core memory wing contains memory parity provisions for use with an optional 19-bit core stack. The central processor wing is prewired for the extended arithmetic element option. The I/O control wing is prewired for the DEC Type 34H 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 housed in two mounting panels above the paper tape reader/punch. Two remaining panels above these house the DMA multiplexer option.

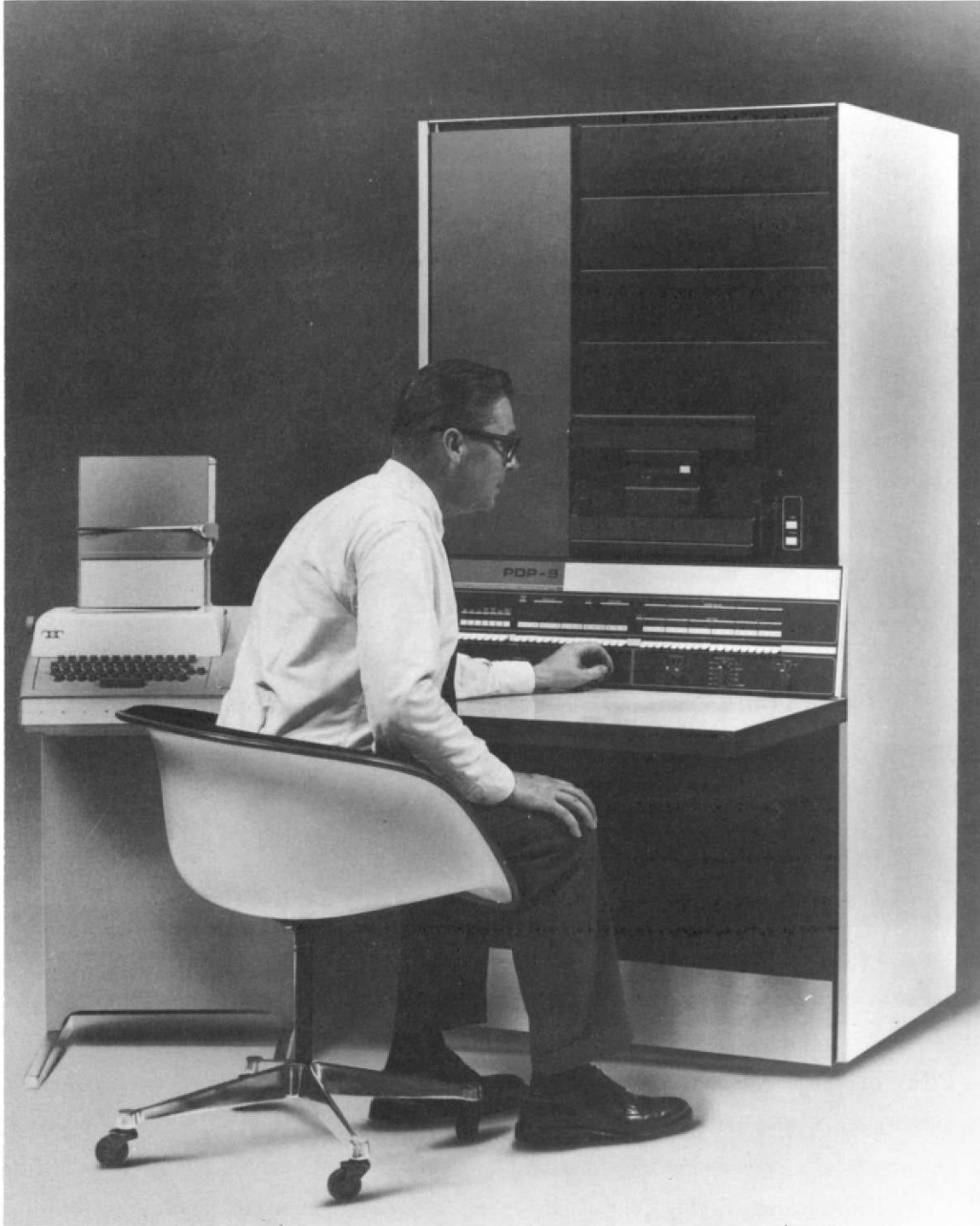


Figure 1-1 Programmed Data Processor PDP-9

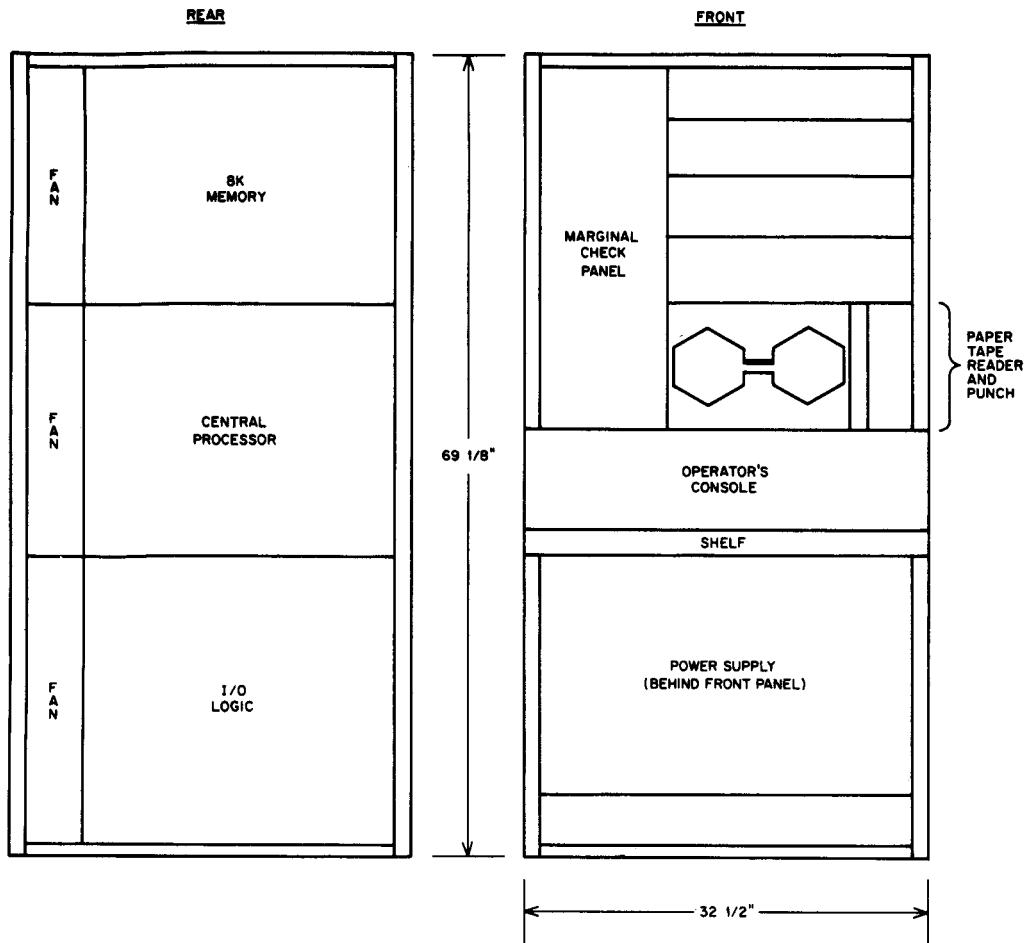


Figure 1-2 Basic PDP-9 Layout, Front and Rear

1.4 FUNCTIONAL DESCRIPTION

The PDP-9 (Figure 1-3) 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. The system is a single-address, 18-bit computer using 1s complement arithmetic which is program-convertible to 2s complement notation to facilitate multiple-precision operations. Indirect addressing to one level and autoindexing features afford programming flexibility. The 8,192-word core memory provides random-access to any word within 1 μ s.

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, the paper tape reader, the paper tape punch, and the real-time clock as basic furnished equipment. Program-controlled transfer operations include program

interrupt (PI), I/O skip, and I/O status checking facilities. The real-time clock (RTC) is located in the I/O control section of the computer. When turned on, the RTC permits programming of long delays by incrementing a counter in core memory 60 times per second (50 times for 50-cycle powered machines). A program interrupt occurs when the preset count is reached.

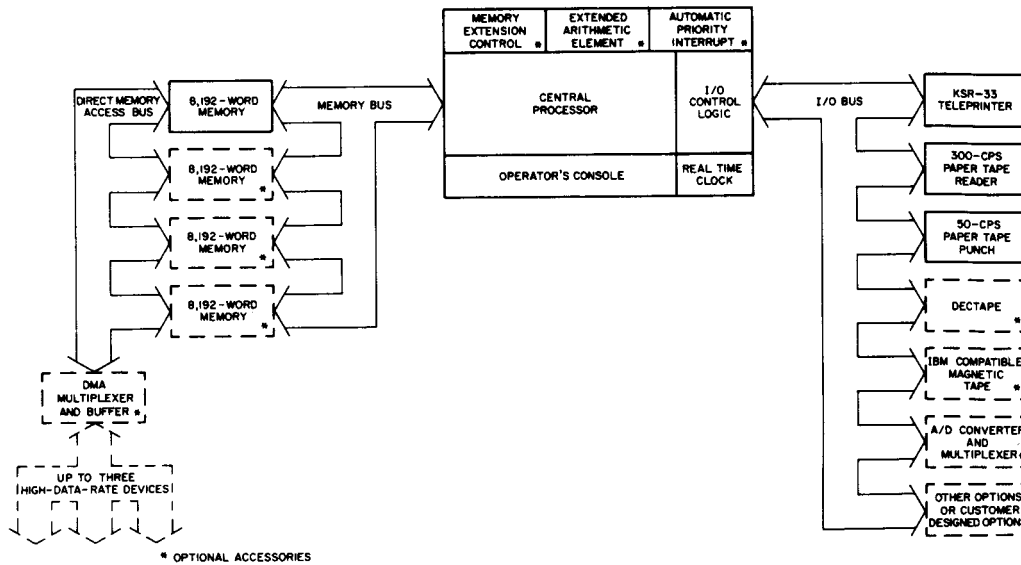


Figure 1-3 PDP-9 System Configuration, Block Diagram

The memory bus provides for the addition of up to three memory expansion modules for a total of 32,768 words of memory. The direct memory access (DMA) bus permits direct access to core memory on a cycle-stealing basis for up to three extremely fast (up to 1,000,000 wps) peripheral devices via the DMA channel.

Two DEC Type 709 Power Supplies produce all required dc operating voltages from a single source of 120V or 240V, 60-cycle, single-phase power. The supply input transformers can be tapped for operation with 50-cycle power. The supplies provide +10V and -15V for use throughout the basic PDP-9 system; one supply also provides -30V to the core memory system, and the other provides +10V and -30V to the paper tape punch. A variable 0 to 20V output checks the operation of the computer under marginal power supply limits. By substituting this output for the normal +10V and -15V outputs, the technician can rapidly troubleshoot an existing fault or forecast a future system failure.

Input ac power to the supplies is controlled from a DEC Type 841A Power Control Unit attached to one of the supplies. The 841A contains the ac input power receptacle and associated circuit breaker. The console POWER switch must be ON for application of ac power to the supplies.

1.5 OPTIONS

Appendix A lists all central processor, core memory, and I/O options available for use with the basic PDP-9. Separate manuals on specific options are supplied per individual customer purchases.

1.6 REFERENCE DOCUMENTS AND PROGRAMS

Tables 1-1 through 1-3 at the end of this chapter list the standard documents and program tapes supplied with the basic PDP-9. Others may be furnished as appropriate to customer requirements.

1.7 REFERENCE CONVENTIONS

In the interest of effective text presentation and readability this manual observes certain reference conventions noted below.

a. Numerical Notation - Unless otherwise indicated, all binary 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; viz., 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 Chapter 5.

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

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. See also Section 1.9.

1.8 TERMINOLOGY

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

a. 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 micro-coded computer commands. These commands are executed during either normal (OPR) or extended (IOT) computer fetch cycles, as opposed to Memory Reference Instruction.

b. 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. Up to four process words may be extracted within any 1 μ s computer cycle.

c. 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.

d. Core Memory Cycle - The 1 μ s 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).

e. Computer Fetch Cycle - The 1 μ s 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.

f. Computer Defer Cycle - The 1 μ s 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.

g. Computer Execute Cycle - The 1 μ s 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, and the result is stored in core memory. The result may be the original operand, or may be an operand modified by the manipulating processes. In the latter case, the original operand is lost.

h. Computer IAO Cycle - The 1 μ s 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.

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

j. 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.

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

l. 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.

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

n. Program Break - An exit from the main program into a device service routine requested by one of several sources below and granted by the central processor upon completion of the current execute cycle, in the following descending order of priority.

- DMA channel breaks
- DCH breaks
- RTC breaks
- API breaks
- PI breaks

1.9 ENGINEERING DRAWINGS AND CIRCUIT SCHEMATICS

A complete set of engineering drawings and module circuit schematics is delivered with the PDP-9. Chapter 5 contains a set of reduced engineering drawings applicable to the basic system only indexed by their drawing number codes. Logic symbols used on the drawings are defined in the Logic Handbook, Document C-105.

In isolated cases of custom-design or late design changes, the logic drawings appearing in Chapter 5 may conflict with those furnished with the system. The furnished drawings always reflect the system as delivered, and therefore take precedence over the drawings in Chapter 5.

1.10 SYSTEM SPECIFICATIONS

1.10.1 Dimensions

Cabinet Height	69-1/8 in.
Cabinet Width	32-1/2 in.
Cabinet Depth	27-3/4 in.
Shelf Width	32-1/2 in.
Shelf Depth	22 in.
Door Clearance (Rear)	31 in.
Cabinet Weight	750 lb.
Teletype Height	8-3/8 in.
Teletype Width	18-5/8 in.
Teletype Depth	18-1/2 in.
Teletype Weight	40 lb.

1.10.2 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	2 KW
Power Receptacle	Hubbell Twistlock, flush, 250V, 30A
Power Supply outputs	+10, -15, -30, ±20 Vdc
Logic Levels	0V = logic 0, -3V = logic 1
Tested Temperature Range	55-122°F
Relative Humidity Range	10-95%
Heat Dissipation	6830 Btu/hr

1.10.3 Functional Characteristics

Word Length	18 bits plus optional memory parity bit
Cycle Time	1 μ s (1.2 μ s with parity)
Core Memory Operation	Read/restore or read/modify/write cycle
Core Memory Capacity	8,192 words expandable to 32,768 words
Core Memory Access	Single direct-addressing in any 8,192-word memory bank; single indirect-addressing from one bank to another
Computation Rate	500,000 additions/s
KSR 33 Teletype	10 char/s
DEC PC02 Paper Tape Reader	300 char/s
DEC PC03 Paper Tape Punch	50 char/s
Real-Time Clock	60 pps (50 pps for 50-cycle powered system)
Program-Controlled I/O Capacity	Up to 64 devices, 4 mode selections each device
API Channel Capacity	Up to 28 devices
Data Channel Capacity	Up to 8 devices
Direct Memory Access Channel Capacity	Up to 3 devices

Table 1-1
Related Documents

Document Number	Title	Publisher
C-105	Logic Handbook	DEC
F-95	PDP-9 User Handbook	DEC
DEC-00-IP2A-D	Paper Tape Reader PC02, Instruction Manual	DEC
DEC-9A-BSAA-D	PDP-9 Software Summary	DEC
DEC-9A-GSAA-D	PDP-9 Advanced Software System	DEC
DEC-9A-SFA0-D	PDP-9 FAST System	DEC
DEC-9A-USA0-D	PDP-9 Multianalyzer Programs	DEC
DEC-9B-GSAA-D	PDP-9 Basic Software System	DEC
Bulletin 273B	KSR 33 Teletype, Technical Manual	Teletype Corp.
Bulletin 1184B	KSR 33 Teletype, Parts	Teletype Corp.
	Paper Tape Punch Model 500, Maintenance Manual	Royal McBee Corp.
	Instruction Manuals, Optional Equipment	DEC

Table 1-2
System Program Tapes

Number*	Program	Number*	Program
ABAA-PU	Symbolic Assembler	FSDA-PA	Divide S.P. Signed
AFT1-PU	FORTRAN II Compiler	FSDB-PA	Divide S.P. Unsigned
AFT2-PU	FORTRAN II Assembler	FSMA-PA	Multiply S.P. Signed
AFT3-PU	FORTRAN II OTS	FSMB-PA	Multiply S.P. Unsigned
AFT5-PU	FORTRAN II IO Library	LFFA-PA	F.F. Loader
AFT6-PU	FORTRAN II 6DD Library	LR1A-PH	RIM Loader
AFT7-PU	FORTRAN II 9DD Library	POT1-PU	Teletype Octal Dump (SA:22)
CDDA-PU	DDT	POT2-PU	Teletype Octal Dump (SA:17300)
CT1A-PU	Extended Trace	PT1A-PA	TIC TOC
ESSA-PU	Symbolic Editor	PTMA-PU	Master Tape Duplicator
FC2A-PA	CAL Handler Type II	PYP1-PU	RIM Puncher (SA: 0100)
FC3A-PA	CAL Handler Type III	PYP2-PU	RIM Puncher (SA: 17000)
FCRA-PA	CAL Handler Revised	QDPA-PA	Double Precision Integer Test
FCXA-PA	Execute Subroutine	QF1A-PA	FORTRAN II Operating Test
FODA-PA	Decimal Integer Print	QFPA-PA	Floating Point Test
FOOA-PA	Octal Print Subroutine	SDPA-PA	Double Precision Integer Package
FOTA-PA	Teletype Output Package	SFPA-PA	Floating Point Package

*PA - Paper Tape ASCII
PH - Paper Tape Hardware Read-In
PU - Paper Tape Funny Format

Table 1-3
Maintenance Program Documents

Number	Program
9A-D01A-PH	Instruction Test Part I
9A-D02A-PH	Instruction Test Part II
9A-D0BA-PH	ISZ Test
9A-D0CA-PH	Memory Address Test
9A-D0DB-PH	JMP Self Test
9A-D0EA-PH	JMP-Y Interrupt Test
9A-D0FA-PH	JMS-Y Interrupt Test

Table 1-3 (cont)
Maintenance Program Documents

Number	Program
9A-D1AA-PH	Basic Memory Checkerboard
9A-D1BA-PH	Extended Memory Checkerboard
9A-D2BA-PH	TTY Test
9A-D2CB-PH	High Speed Reader Test
9A-D2DB-PH	Punch Test
9A-D7AB-PH	Basic Exerciser

CHAPTER 2 OPERATION

2.1 SCOPE

This chapter contains operating information, tabular descriptions of system controls and indicators, and programming considerations for the PDP-9 operator.

2.2 STANDARD CONTROLS AND INDICATORS

The following sections list and define standard manual controls and indicators, arranged as to the system elements: operator console, marginal-check and maintenance panels, paper tape reader/punch, teletype, power control, and core memory banks.

2.2.1 Operator Console

Table 2-1 lists the functions of the standard controls and indicators shown in Figure 2-1. Other controls and indicators are optional and are not listed here. Indicators on the console show the existing states of specific register bits and control flip-flops by illuminating for binary 1s and extinguishing for binary 0s. The console can be electrically locked by a control on the maintenance panel (Figure 2-2) to prevent accidental alteration of the program in progress. While the console is locked, operation of any key or switch (with the exception of the DATA switches) will not be detected by the system.

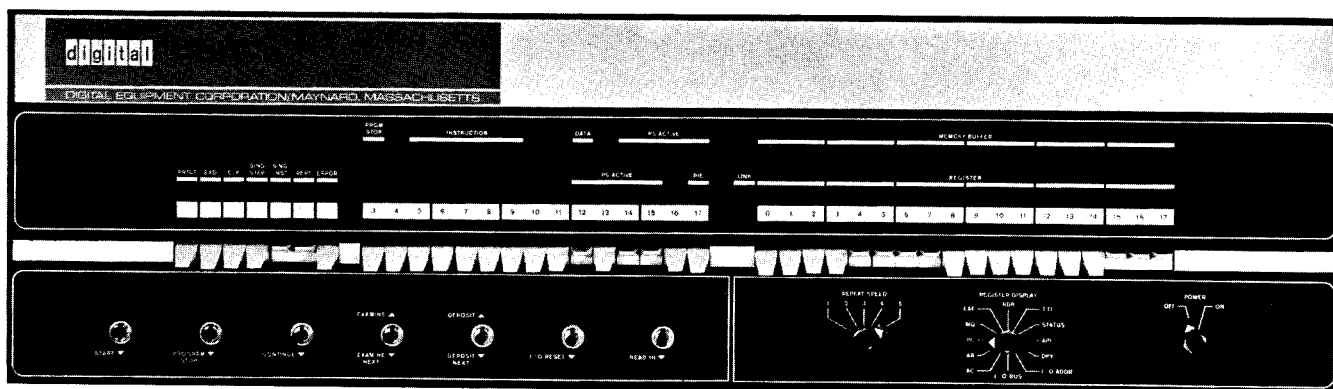


Figure 2-1 Operator Console

Table 2-1
Operator Console, Controls and Indicators

Name	Function
START key	Three positions: off (center), operate (down, spring-loaded return), and locking (up). Depressing START starts program execution at the location specified by the ADDRESS switches. Up position for MAINT functions, Table 2-2.
I/O RESET key	Two positions: off (center) and operate (down, spring-loaded return). Depressing key clears all I/O device flags, all central processor registers except program counter (PC), and turns off the real-time clock and program interrupt facility.
PROGRAM STOP key	Two positions: off (center) and operate (down, spring-loaded return). Depressing key halts program operation at completion of the current instruction.
CONTINUE key	Three positions: off (center), operate (down, spring-loaded return), and locking (up). Depressing key resumes program execution from the point at which it stopped. Up position for use with REPT, SING STEP, SING INST switches.
EXAMINE/EXAMINE NEXT key	Three positions: off (center), EXAMINE (up, spring-loaded return), and EXAMINE NEXT (down, spring-loaded return). Raising the key to EXAMINE transfers the contents of the memory location specified by the ADDRESS switches to the memory buffer register (MB). After the transfer, the computer stops, and the MB contents can be viewed in the MEMORY BUFFER indicator. The contents of the ADDRESS switches transfer to the arithmetic register (AR) for viewing in the REGISTER indicator. Depressing the key to EXAMINE NEXT increments the contents of the AR by 1 and transfers the contents of the newly addressed memory location to the MB. Using EXAMINE NEXT after EXAMINE facilitates monitoring of sequential memory locations as the ADDRESS switches need only be set the first time to the lowest memory location.
DEPOSIT/DEPOSIT NEXT key	Three positions: off (center), DEPOSIT (up, spring-loaded return), and DEPOSIT NEXT (down, spring-loaded return). Raising the key deposits the contents of the DATA switches into 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 into which the data was entered. Depressing the key increments the AR contents by 1, and deposits the contents of the DATA switches into the memory location specified by the new address. Using DEPOSIT NEXT after DEPOSIT facilitates entering data in sequential memory locations as the ADDRESS switches need only be set the first time to the lowest order address.

Table 2-1 (cont)
Operator Console, Controls and Indicators

Name	Function																
READ IN key	Two positions: off (center) and operate (down, spring-loaded return). Depress the key to initiate read-in of paper tape punched in binary code. (Three 6-bit lines form one 18-bit word.) Storage of words begins at the memory location specified by the ADDRESS switches. At the completion of tape read-in, the computer reads and executes the last word stored. Read-in occurs at the selected repeat speed (see REPEAT SPEED).																
REPEAT SPEED switch	With REPT switch up, REPEAT SPEED establishes one of five selected speeds at which certain manual operations repeat without operator intervention. The repeat speeds range from approximately 8 μ s (position 5) to 60 ms. Switch must be placed in position 5 during READ IN operations.																
REPT switch and indicator	The up position causes operations initiated by START, CONTINUE, SING STEP, SING INST, or READ IN to repeat. REPEAT SPEED establishes the rate of repetition. Also used for MAINT function, Table 2-2. The indicator lights when the switch is up.																
REGISTER DISPLAY switch and REGISTER indicator	<p>Twelve-position switch. Each position selects a specific register to display its contents in the REGISTER indicator. Display is enabled only when the machine stops. The standard register selections are as follows.</p> <table border="0" data-bbox="662 1150 1404 1915"> <tr> <td data-bbox="662 1150 841 1213">RDR</td> <td data-bbox="857 1150 1404 1213">Displays contents of paper-tape reader buffer.</td> </tr> <tr> <td data-bbox="662 1234 841 1297">TTI</td> <td data-bbox="857 1234 1404 1297">Displays contents of teletype keyboard buffer in the eight least-significant indicator lamps.</td> </tr> <tr> <td data-bbox="662 1339 841 1402">STATUS</td> <td data-bbox="857 1339 1404 1402">Displays status of I/O device flags connected to status checking facility (Table 3-3).</td> </tr> <tr> <td data-bbox="662 1444 841 1507">I/O ADDR</td> <td data-bbox="857 1444 1404 1507">Displays address word from address lines of I/O bus for DCH and optional API operations in the 15 least significant indicator lamps.</td> </tr> <tr> <td data-bbox="662 1549 841 1612">I/O BUS</td> <td data-bbox="857 1549 1404 1612">Displays data word on I/O bus to/from any device using the bus.</td> </tr> <tr> <td data-bbox="662 1654 841 1717">AC</td> <td data-bbox="857 1654 1404 1717">Displays contents of the AC.</td> </tr> <tr> <td data-bbox="662 1738 841 1801">AR</td> <td data-bbox="857 1738 1404 1801">Displays contents of the AR.</td> </tr> <tr> <td data-bbox="662 1822 841 1915">PC</td> <td data-bbox="857 1822 1404 1915">Displays contents of the PC in the 13 least significant indicator lamps, status of LINK, memory extend mode, memory protect mode, EPC in the five most significant lamps.</td> </tr> </table>	RDR	Displays contents of paper-tape reader buffer.	TTI	Displays contents of teletype keyboard buffer in the eight least-significant indicator lamps.	STATUS	Displays status of I/O device flags connected to status checking facility (Table 3-3).	I/O ADDR	Displays address word from address lines of I/O bus for DCH and optional API operations in the 15 least significant indicator lamps.	I/O BUS	Displays data word on I/O bus to/from any device using the bus.	AC	Displays contents of the AC.	AR	Displays contents of the AR.	PC	Displays contents of the PC in the 13 least significant indicator lamps, status of LINK, memory extend mode, memory protect mode, EPC in the five most significant lamps.
RDR	Displays contents of paper-tape reader buffer.																
TTI	Displays contents of teletype keyboard buffer in the eight least-significant indicator lamps.																
STATUS	Displays status of I/O device flags connected to status checking facility (Table 3-3).																
I/O ADDR	Displays address word from address lines of I/O bus for DCH and optional API operations in the 15 least significant indicator lamps.																
I/O BUS	Displays data word on I/O bus to/from any device using the bus.																
AC	Displays contents of the AC.																
AR	Displays contents of the AR.																
PC	Displays contents of the PC in the 13 least significant indicator lamps, status of LINK, memory extend mode, memory protect mode, EPC in the five most significant lamps.																

Table 2-1 (cont)
Operator Console, Controls and Indicators

Name	Function
POWER switch	Controls application of primary power to power supplies, fans, teletype motor, tape punch motor, and interfaced optional external devices.
CLK switch and indicator	The down position allows the program to enable the real-time clock. The indicator lights when the clock is enabled.
SING STEP switch and indicator	The up position halts the program by preventing the initiation of the next memory cycle (except during DCH operations, which cannot be single-stepped). The indicator lights when the switch is up.
SING INST switch	The up position halts the program after completion of one instruction. The indicator lights when the switch is up.
ADDRESS switches (3-17)	Establish a 15-bit core memory address to be entered in the PC by operation of the START key, or in the AR by operation of the EXAMINE or DEPOSIT key. Up position for a 1 bit, down position for a 0 bit.
DATA switches (0-17)	Establish an 18-bit data or instruction word to be entered into memory by DEPOSIT or DEPOSIT NEXT, or to be entered in the AC by a programmed OAS or LAS instruction. Up position for a 1 bit, down position for a 0 bit.
PRGM STOP indicator	Lights when the program halts.
INSTRUCTION indicator	The five indicator bits reveal the current op code in the IR (instruction register). A lighted indicator lamp denotes a 1 bit, extinguished lamp a 0 bit.
DATA indicator	Lights to indicate that data is being transferred between core memory and a DCH device.
PIE indicator	Lights when the program interrupt facility has been enabled by program control.
LINK indicator	Shows the content of the LINK register.
MEMORY BUFFER indicator	Shows the contents of the MB (memory buffer) register.

2.2.2 Marginal Check Panel and Maintenance Panel

These panels (Figure 2-2) are concealed behind the hinged red cover on the left of the cabinet. Table 2-2 details the functions of the panel-mounted controls and indicators.

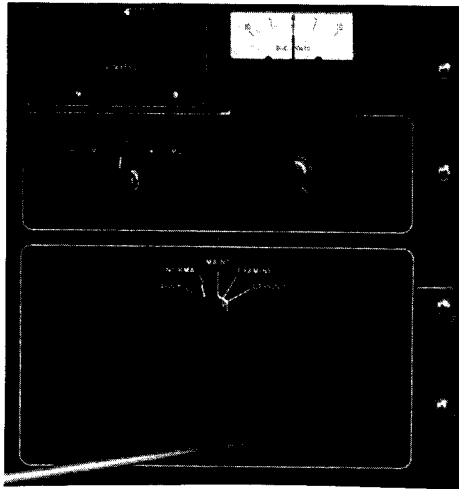


Figure 2-2 Marginal Check Panel and Maintenance Panel

Table 2-2
Marginal Check and Maintenance Panels, Controls and Indicators

Name	Function
Voltmeter*	<p>Marginal Check Panel</p> <p>Indicates the marginal-check voltage output of the 709 Power Supply. The center of the scale (0V) equates with the reference voltage selected, +10 or -15 Vdc. The right side of the scale indicates an increase in magnitude beyond the selected voltage. The left side indicates a decrease.</p>
Selector switch*	<p>Three positions:</p> <p>OFF Removes the marginal check voltages from the computer system.</p> <p>+10 MC Selects the +10V marginal check output.</p> <p>-15 MC Selects the -15V marginal check output.</p>
Variac control knob*	<p>Adjusts the marginal check voltage above or below the selected reference voltage.</p>
TOTAL HOURS meter	<p>Indicates to the nearest tenth of an hour the cumulative "power on" time of the system. Meter counts from 00000.0 to 99999.9 hr.</p>
	<p>Maintenance Panel</p>
Selector switch	<p>Five positions:</p> <p>LOCK Electrically locks the console controls (except DATA switches).</p> <p>NORMAL Console is unlocked; all controls effective.</p> <p>MAINT With the switch in this position and the START key and REPT switch up, the built-in</p>

*Used with internally-mounted marginal check switches; see Chapter 4.

Table 2-2 (cont)
Marginal Check and Maintenance Panels, Controls and Indicators

Name	Function
Selector switch (cont)	<p style="text-align: center;">Maintenance Panel (cont)</p> <p style="text-align: center;">maintenance test program circulates a self-incrementing count through all active CP registers to verify register operation and internal transfer-path gating. The test proceeds at the rate selected by the console REPEAT SPEED switch.</p> <p style="text-align: center;">EXAMINE With console REPT switch off (down), simulates the console EXAMINE function. With REPT switch up, simulates console EXAMINE-REPT function.</p> <p style="text-align: center;">DEPOSIT With console REPT switch off (down), simulates the console DEPOSIT function. With REPT switch up, simulates console DEPOSIT-REPT function.</p>

2.2.3 Paper Tape Reader/Punch

The PC09 Tape Reader/Punch appears in Figure 2-3. The PC09 consists of the PC02 Tape Reader (left) and the PC03 Tape Punch (right). Table 2-3 lists the tape reader/punch controls and their functions.

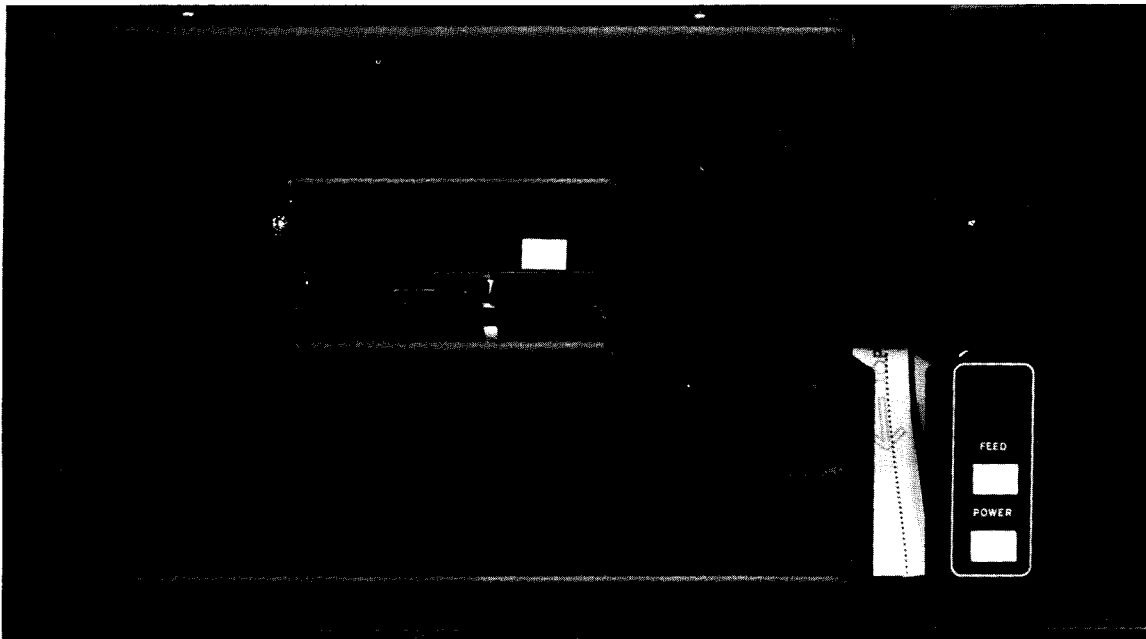


Figure 2-3 Paper Tape Reader/Punch PC09

Table 2-3
Tape Reader/Punch, Controls and Indicators

Name	Location	Function
FEED button	Reader	Depressing button advances tape without reading. Spring-loaded release position allows program control or manual control or read-in operations.
POWER button and indicator	Punch	Off position simulates PUN NO TAPE condition. On position removes the condition. Indicator lights when tape is secured and button is pressed to on position.
FEED button	Punch	Depressing button advances tape and punches feed holes. Off, spring-loaded release position, allows program control of punching operations. POWER indicator extinguishes when button is pressed to on position.

2.2.4 Teletype Unit

The KSR 33 Teletype Unit appears in Figure 2-4. Table 2-4 lists the teletype controls and their functions.

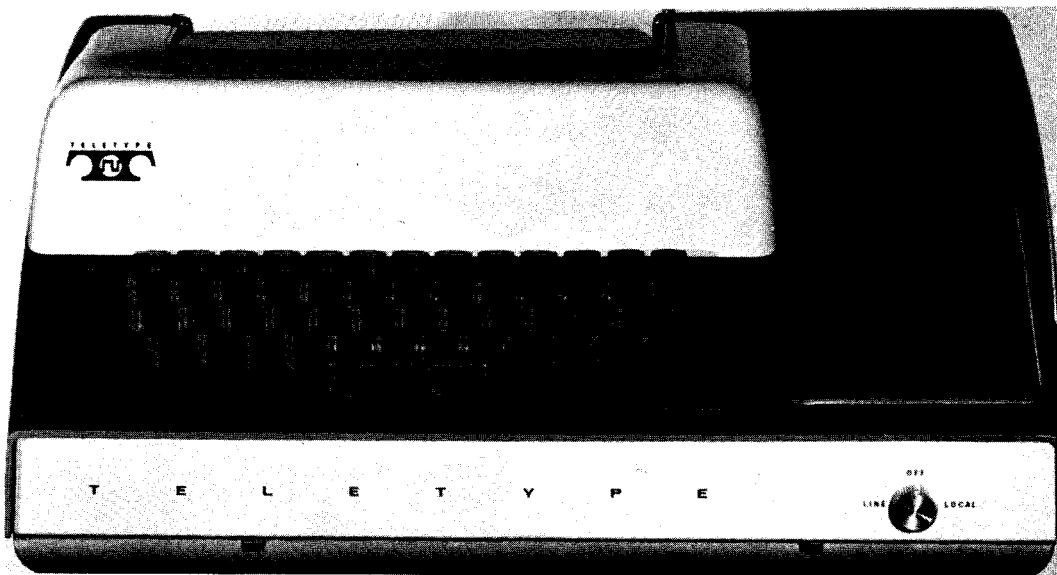


Figure 2-4 KSR 33 Teletype Unit

Table 2-4
Teletype Controls

Name	Function
Teletype keyboard	Supplies input data to the computer and/or the page printer, depending on the setting of the LINE/OFF/LOCAL switch.
LINE/OFF/LOCAL switch	Controls the application of primary power to the Teletype and controls data connection between the Teletype and the central processor. In the LINE position, the Teletype is energized and connected as an I/O device to the computer. In the OFF position the teletype motor is de-energized but the logic circuits remain energized. In the LOCAL position the Teletype is energized for off-line operations, breaking the signal connections to the computer. Use of the Teletype requires that the computer be energized through the console POWER switch.

2.2.5 Power Control 841A

Figure 2-5 illustrates the Power Control 841A. Table 2-5 lists the controls and functions.

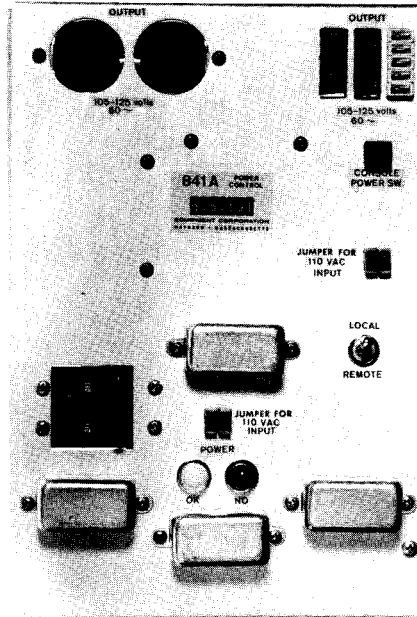


Figure 2-5 Power Control 841A

Table 2-5
Power Control 841A, Controls and Indicators

Name	Function
Circuit Breaker CB1	Protects primary power source from overloads due to failure of 709 Power Supplies.
ON Indicators I1, I2	Indicate the presence of primary power at the input plug (P1).
LOCAL/REMOTE switch S1	LOCAL position applies primary power in conjunction with CB1, the console POWER switch, and the maintenance panel switch to the basic computer system power supplies, fans, etc. REMOTE position for use with 841A units located in optional free-standing equipment cabinets.

2.2.6 Core Memory Banks

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

2.3 OPERATING PROCEDURES

Several methods are available for loading or unloading PDP-9 information, as described in the PDP-9 User Handbook, F-95. 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 configuration. Although used infrequently, they are valuable in preparing the initial programs and learning the functions of machine input/output transfers.

2.3.1 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 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 User Handbook and in Digital Program Library descriptions. The RIM Loader program is listed in Table 2-6 for convenience as an exercise in manual data storage. To store the RIM Loader manually in PDP-9 core memory:

- 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 17762.)
- 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. Momentarily lift the DEPOSIT/DEPOSIT NEXT key 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 momentarily pressing the DEPOSIT/DEPOSIT NEXT key to the DEPOSIT NEXT position 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 momentarily set the EXAMINE/EXAMINE NEXT key to the EXAMINE position. After the first core memory cell has been checked at the MEMORY BUFFER indicator, the remaining cells may be examined in sequence by repeatedly setting the key to the EXAMINE NEXT position 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.

Table 2-6
Read-In Mode (RIM) Loader Program

Address	Content	Tag	Mnemonic	Comments
17762	0	R,		/READ ONE BINARY WORD
17763	700101		RSF	/WAIT FOR WORD TO COME IN
17764	617763		JMP .-1	/
17765	700112		RRB	/READ BUFFER CONTENTS INTO AC
17766	700144		RSB	/READ ANOTHER WORD INTO BUFFER
17767	637762		JMP I R	/EXIT SUBROUTINE
17770	700144	GO,	RSB	/ENTER HERE, START READER
17771	117762	G,	JMS R	/GET NEXT BINARY WORD
17772	057775		DAC OUT	/
17773	417775		XCT OUT	/EXECUTE CONTROL WORD
17774	117762		JMS R	/GET DATA WORD
17775	0	OUT,	0	/STORE DATA WORD
17776	617771		JMP G	/CONTINUE

2.3.2 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 into the tape storage bin to the right of the tape reader; the tape must be positioned with its feed holes toward the inside and the title end exiting from the bin first. Raise the snap-action tape retainer. Place the title end of the tape across the tape platform with the feed holes engaged by the feed-hole sprocket. Lower the tape retainer to secure the tape in the reading position.

c. Press the FEED button above the tape feed platform momentarily to insure proper tape alignment and observe that the tape is fan-folding properly into the left-hand storage bin.

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

e. Press the READ IN key. The tape will be read automatically. When tape motion stops, depress the FEED button to completely advance the tape into the left-hand storage bin.

f. 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.

2.3.3 Storing Data Under Program Control

Information can be stored or modified in the computer automatically only 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, following the instructions in Section 2.3.2b.

c. Using the ADDRESS switches, set the starting address of the RIM Loader program (17770).

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

2.3.4 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. Since the assembler tape is in RIM format, it can be loaded by method described in Section 2.3.3. 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 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 assembler 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.

2.3.5 Teletype Code

The 8-bit code used by the KSR 33 Teletype is the American Standard Code for Information Interchange (ASCII), modified. A complete description of the code and the manner in which it is read appears in Section 3.9.1.

2.3.6 Local Teletype Operation

The Teletype can be used as an ordinary typewriter as follows:

- a. Turn the maintenance panel switch to NORMAL and the console POWER switch ON.
- b. Set the teletype LINE/OFF/LOCAL switch to LOCAL.
- c. Type the desired information on the page printer using the teletype keyboard.

2.3.7 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 4, as they apply to preventive or corrective maintenance of the PDP-9 system. To exercise these routines the user should be familiar with the machine programming described in the User Handbook.

CHAPTER 3 SYSTEM DESCRIPTION

3.1 COMPUTER ORGANIZATION

Figure 3-1 illustrates the basic organization of the PDP-9 computer system. Employing a transfer bus system, data is jam-transferred (no logic delays) between registers at dc levels to minimize timing problems.

3.1.1 Central Processor

The central processor section of the computer includes the active registers and control elements described below.

3.1.1.1 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 very fast, read-only, prewired magnetic core storage unit. A control register (CR) in the CM addresses the magnetic core matrix drivers to make the CM deliver specific processing gate levels to the transfer busses and to the active registers. The CR supplies new address information to the CM based on the instruction to be executed and on the conditions sensed by the previous process levels.

3.1.1.2 Adder (ADR) - The 18-bit ADR functions as a fast, nonstoring adder-subtractor for arithmetic operations, and as a common transfer bus 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 mutually 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.

3.1.1.3 Accumulator (AC) - The 18-bit AC retains the result of arithmetic/logical operations during the interim between instructions. The AC contents 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 and the result left in the AC. The contents of the AC and MB registers can be combined in the ADR by logical AND or exclusive OR instructions, the result left in the AC. An inclusive OR can be formed in the ADR between the AC contents 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.

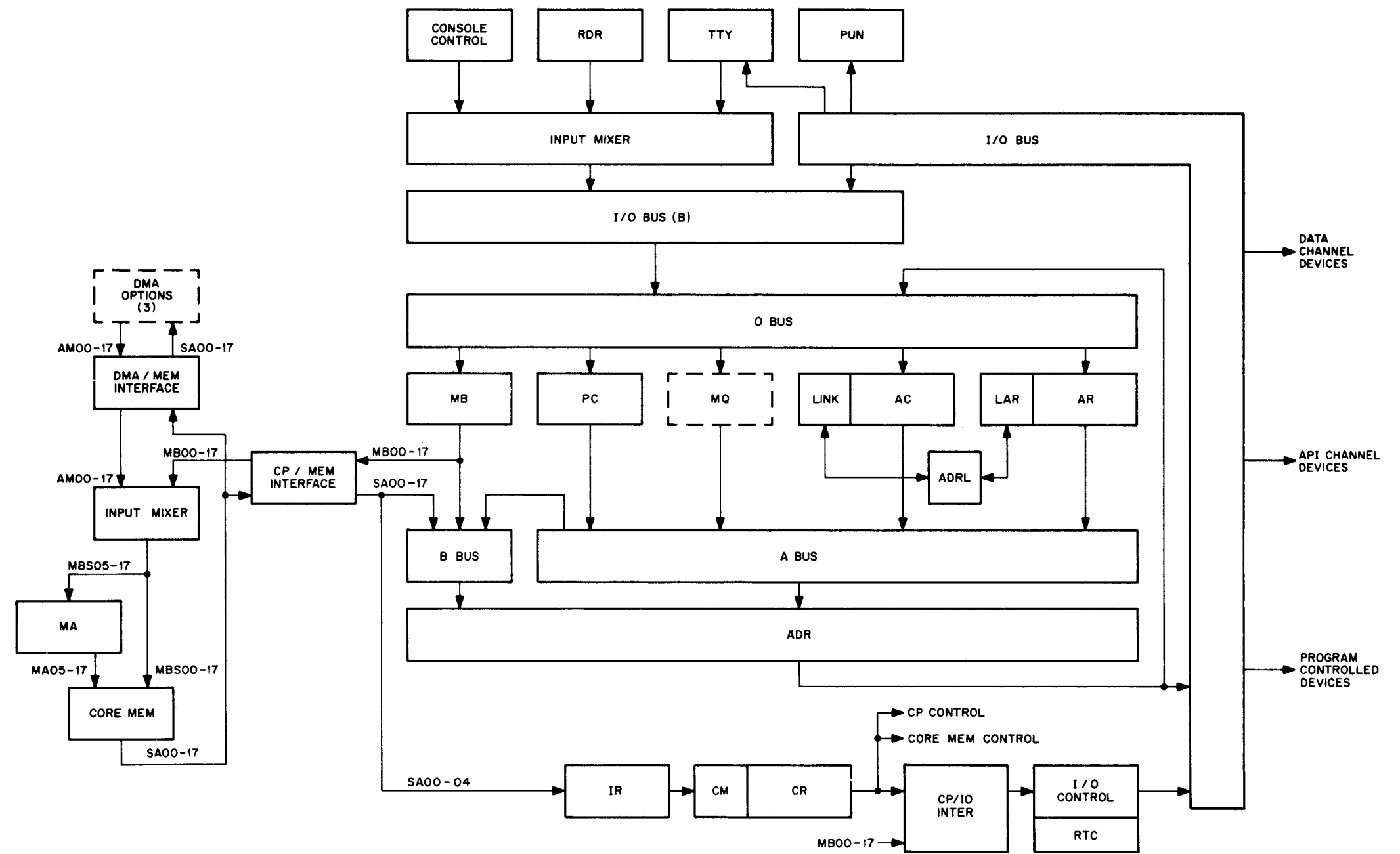


Figure 3-1 System Block Diagram, Basic PDP-9

3.1.1.4 AC Link (LINK) - This 1-bit register is used to extend the arithmetic capability of the AC. In 1s complement arithmetic, it is an overflow indicator; in 2s complement arithmetic, it extends the AC to 19 bits and functions as a carry register. The program can check 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 with the AC in rotate operations.

3.1.1.5 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. The 19th-bit AR link (LAR) stores the content of the AC link (LINK). For program-controlled output data transfers, information is transferred from the AR to an external device via the I/O bus.

3.1.1.6 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.

3.1.1.7 Program Counter (PC) - The PC determines the program sequence; i.e., the order 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 extended memory options adds two extended program counter (EPC) bits to the system addressing scheme.

3.1.1.8 Instruction Register (IR) - The IR accepts the five most-significant bits of each instruction fetched from core memory. The four most-significant bits constitute the instruction operation code (op code). The fifth bit indicates whether the instruction contains a direct (effective) address of an operand to be manipulated, 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.

3.1.1.9 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 cell into the MB and are rewritten into the cell 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 cell and placed in the MB. The operand is then rewritten into the memory cell undisturbed or modified, 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 (see Section 3.1.2).

3.1.2 Core Memory System

The PDP-9 core memory uses a new design concept (2-1/2 D) for speed, compactness, and reliability; it operates with a complete cycle time of 1 μ s. Each 8,192-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 8,192 words up to 32,768 words in 8,192-word increments. Such expansion requires the implementation of the memory extension option to extend the PDP-9 addressing capability.

Each PDP-9 memory package has a two-port capability for data entry and retrieval. One port, connected to the CP/memory interface, services the central processor. The other allows a peripheral device direct and immediate access to memory for very fast data transfers (up to 1,000,000 wps) via the direct memory access (DMA) channel. A device request for DMA service has priority over a processor or program request. The latter is deferred until DMA operation terminates. An optional DMA Multiplexer MP09A permits servicing of up to three DMA peripherals and assigns each a fixed priority.

The PDP-9 offers four modes for addressing core memory: direct addressing, indirect addressing, autoindexing, and extended mode addressing for optional extended memory modules. Addresses of locations (memory word registers) in the basic 8,192-word core memory stack are 00000 through 17777. For systems having additional core memory modules, the basic module is referred to as bank 0 and subsequent modules are referred to as bank 1, bank 2, and bank 3. Their addresses are 20000 through 37777 for bank 1, 40000 through 57777 for bank 2, and 60000 through 77777 for bank 3. Programs prepared for the PDP-9 should reserve locations 00000 through 00077 as specified in Table 3-1.

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 real-time clock count

Table 3-1 (cont)
Reserved Core Memory Locations

Address	Purpose
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	Stores the 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 unique entry instructions for each of 32 optional automatic priority interrupt channels.

3.1.2.1 Direct Addressing - Directly addressed memory reference instructions (bit 04 = 0, Figure 3-2) take the 13-bit address (05 to 17) specified in their instruction words as the effective address of the memory register which contains the required operand. The 13-bit address allows direct addressing of up to 8,192 locations in a currently addressed memory bank. Locations in memory banks other than that currently addressed must be accessed by indirect addressing.

3.1.2.2 Indirect Addressing - Indirectly addressed memory reference instructions (bit 04 = 1, Figure 3-2) take the 13-bit address (05 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 I 100

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.

3.1.2.3 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.

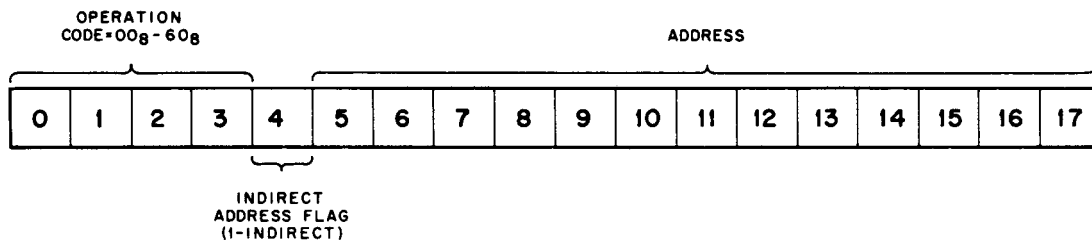


Figure 3-2 Memory Reference Instruction Word Format

3.1.2.4 Extend Mode Addressing - Installation of additional memory packages requires the memory extension control option for addressing a memory location outside the currently addressed memory bank.

3.1.3 I/O Control

The I/O control section includes logic for program-controlled transfers between the central processor and as many as 64 devices. Program-controlled transfers make use of program interrupt, I/O skip, and I/O status checking facilities. A real-time clock installed in the I/O section as part of the basic equipment operates at 60 pps (50 pps for 50-cycle powered systems), and is tied to the program interrupt facility and the API option. 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, magnetic tape and drums normally use DCH access to core memory, in order that they operate at their maximum transfer rates. Slower asynchronous devices such as line printers, teletype keyboards and punched card equipment may operate at their 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.

3.2 INSTRUCTION WORD FORMATS

The PDP-9 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.

3.2.1 Memory Reference Instructions

Memory reference instructions (Figure 3-2) consist of an 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 13-bit address, 05 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.

3.2.2 Augmented Instructions

3.2.2.1 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 jointly or separately. 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.

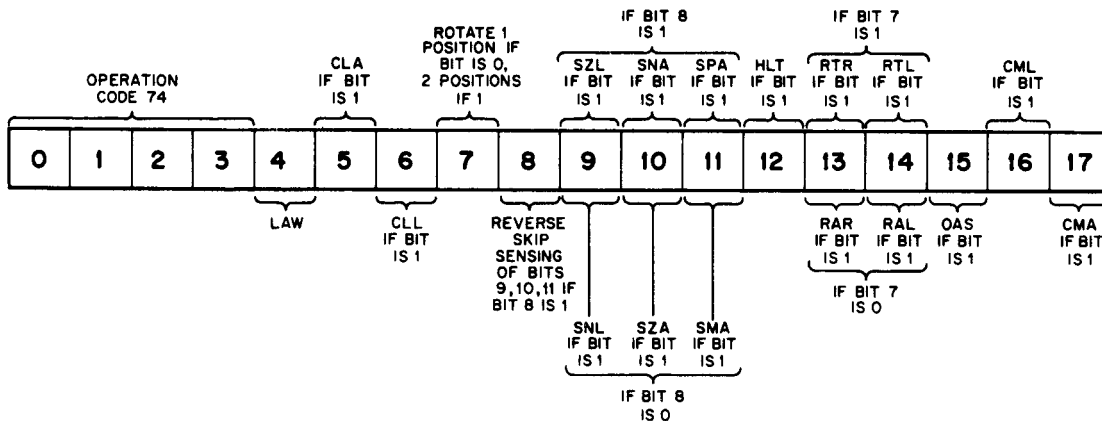


Figure 3-3 OPR Instruction Word Format

3.2.2.2 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 device'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 I/O devices via the I/O bus.

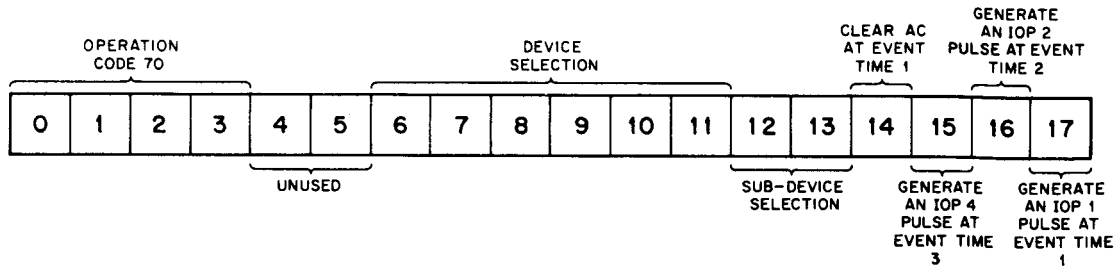


Figure 3-4 IOT Instruction Word Format

Execution of an IOT instruction requires four computer cycles consisting of an instruction fetch cycle and three execute cycles of 1 μ s duration each, designated event times 1, 2, and 3 (Figure 3-5). Only the fetch cycle contains a core memory read/write cycle. Thereafter, the core memory is idle until completion of the IOT execute cycles. Bit 17 generates an IOP1 pulse during event time 1. 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 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 at 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.

3.2.2.3 Optional EAE - The EAE option adds the hardware necessary to implement the EAE instructions. This class of instructions (op code 64) performs high-speed data manipulation and multiply-divide operations as specified by microprogramming individual instructions.

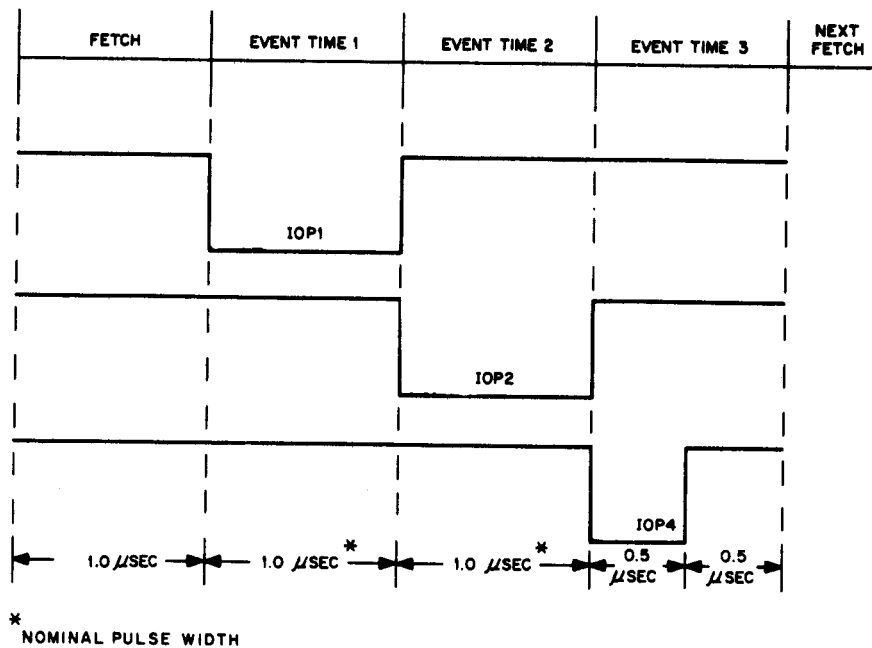


Figure 3-5 IOT Instruction Cycles

3.3 DATA FORMATS

The PDP-9 uses three notations to represent signed data. They are sign and magnitude, 1s complement, and 2s complement. In each, bit 0, or the most-significant bit of a single- or multi-precision data word, serves as the sign indicator: 0 for a positive quantity, 1 for a negative quantity.

3.3.1 Sign and Magnitude Notation

In sign and magnitude notation, quantities of equal magnitude but opposite signs differ only in the content of the sign-indicator bit; i.e., the positive number contains a 0 in bit 0, and the negative number contains a 1 in bit 0. For example:

$$\begin{array}{rcl}
 +131 & 071_{10} & \begin{array}{l} s \\ 011 \ 111 \ 111 \ 111 \ 111 \ 111_2 \end{array} \\
 -131 & 071_{10} & \begin{array}{l} s \\ 111 \ 111 \ 111 \ 111 \ 111 \ 111_2 \end{array}
 \end{array}$$

Observe that conversion of a positive number to a negative number, and vice versa, requires only that the sign bit be complemented; the magnitude bits are not affected.

3.3.2 Complement Notations

In both complement notations (1s and 2s), the sign indicator (bit 0) is 0 for positive quantities and 1 for negative quantities. The 1s complement of a quantity is equivalent to the logical complement of its magnitude and sign; i.e., all binary 1s are replaced by 0s and all binary 0s are replaced by 1s. The 2s complement of a quantity is equivalent to its 1s complement, plus 1 added to the lowest order,

or least-significant bit. Positive quantities in either notation have identical representations. For example, $+15_{10}$ is represented in a PDP-9 data word as

$$\begin{array}{c} s \\ 000\ 000\ 000\ 000\ 001\ 111 \end{array}$$

in either 1s or 2s complement notation. The 1s complement of -15_{10} is represented by

$$\begin{array}{c} s \\ 111\ 111\ 111\ 111\ 110\ 000 \end{array}$$

The 2s complement of -15_{10} appears as

$$\begin{array}{c} s \\ 111\ 111\ 111\ 111\ 110\ 001 \end{array}$$

Zero has two representations in 1s complement notation:

$$\begin{array}{c} +0_{10} \\ s \\ 000\ 000\ 000\ 000\ 000\ 000_2 \end{array}$$

and

$$\begin{array}{c} -0_{10} \\ s \\ 111\ 111\ 111\ 111\ 111\ 111_2 \end{array}$$

2s complement notation has one representation for zero:

$$\begin{array}{c} +0_{10} \\ s \\ 000\ 000\ 000\ 000\ 000\ 000_2 \end{array}$$

In 2s complement notation minus zero likewise appears in binary form as

$$\begin{array}{c} -0_{10} \\ s \\ 000\ 000\ 000\ 000\ 000\ 000_2 \end{array}$$

since complementing each bit and then adding 1 to the low order bit results in the propagation of an arithmetic carry through the entire word, as follows:

$$\begin{array}{c} +0_{10} \\ 000\ 000\ 000\ 000\ 000\ 000_2 \end{array}$$

is complemented to be

$$\begin{array}{c} s \\ 111\ 111\ 111\ 111\ 111\ 111_2 \\ + 1_2 \end{array}$$

adding 1

$$\begin{array}{c} \text{results in} \\ -0_{10} \\ s \\ \hline 000\ 000\ 000\ 000\ 000\ 000_2 \end{array}$$

The binary 1 carried out of the sign bit "overflows" the 18-bit capacity of the PDP-9 word. Since two identical representations of 0 are ambiguous to the computer, convention has adopted the +0 2s complement notation representation of 0.

Table 3-2 indicates the representations in 1s and 2s complement notation of a range of numbers from +5 to -5; a five-bit word is used for simplicity.

Table 3-2
Complementary Numbers

<u>Number</u>	<u>1s Complement</u>	<u>2s Complement</u>
+5	00101	00101
+4	00100	00100
+3	00011	00011
+2	00010	00010
+1	00001	00001
+0	00000	00000
-0	11111	Not considered
-1	11110	11111
-2	11101	11110
-3	11100	11101
-4	11011	11100
-5	11010	11011

3.3.3 Data Words

PDP-9 hardware and software capabilities include add, subtract, multiply, divide, and rotate of data in single- and multi-precision formats. For signed data words, bit 0 serves as the sign indicator, with a 0 for a positive quantity, and a 1 for a negative quantity. A signed single-precision data word includes a sign bit and 17 magnitude bits (Figure 3-6a). A signed double-precision data word consists of two computer words or a total of 36 bits (Figure 3-6b). The first word contains the sign bit and the 17 most-significant bits; the second word contains the 18 least-significant bits. The words are normally stored in consecutively addressed core memory locations for ease of programming.

3.3.3.1 Magnitudes of Data Words - For 1s complement signed and sign-and-magnitude notations, the permissible magnitude of any quantity, X is in the range of

$$-(2^{n-1}-1) \leq X \leq 2^{n-1}-1$$

where n is the number of bits allocated to the storage of data in a data word. For a single-precision data word (sign bit and 17 data bits), this relationship becomes

$$-(2^{17}-1) \leq X \leq 2^{17}-1$$

or

$$-131\ 071_{10} \leq X \leq +131\ 071_{10}$$

For 2s complement signed notation, the permissible magnitude of any quantity, X, is in the range of

$$-2^{n-1} \leq X \leq 2^{n-1}-1$$

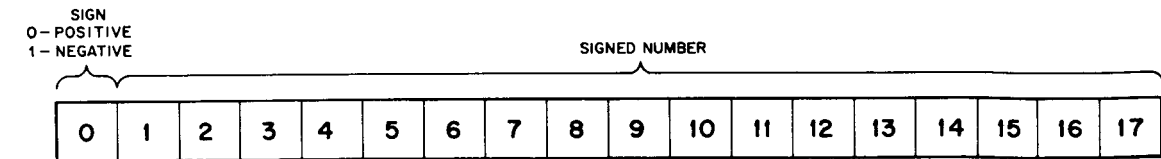
where n is again the number of bits allocated to the storage of data. A single-precision data word has the range

$$-2^{17} \leq X \leq 2^{17}-1$$

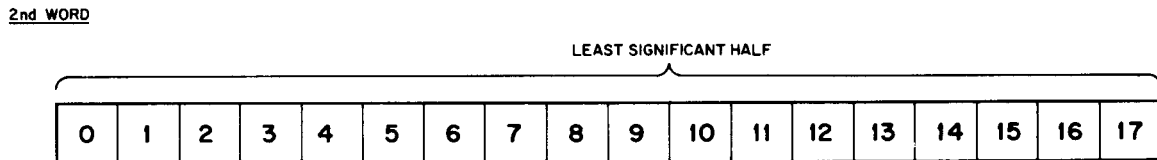
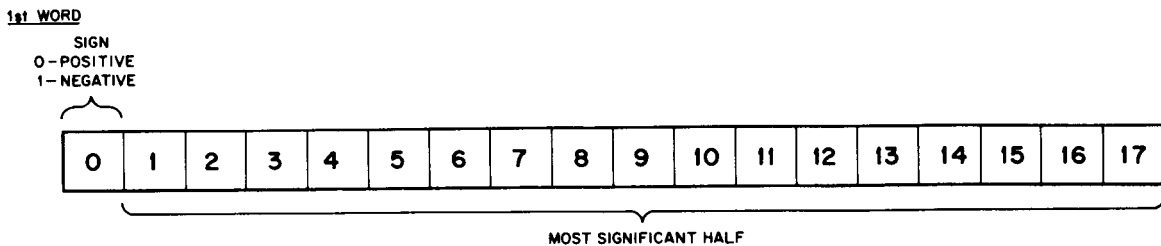
or

$$-131\,072_{10} \leq X \leq +131\,071_{10}$$

The position of the decimal point is implied in the above ranges.



a. Single-Precision Data



b. Double-Precision Data

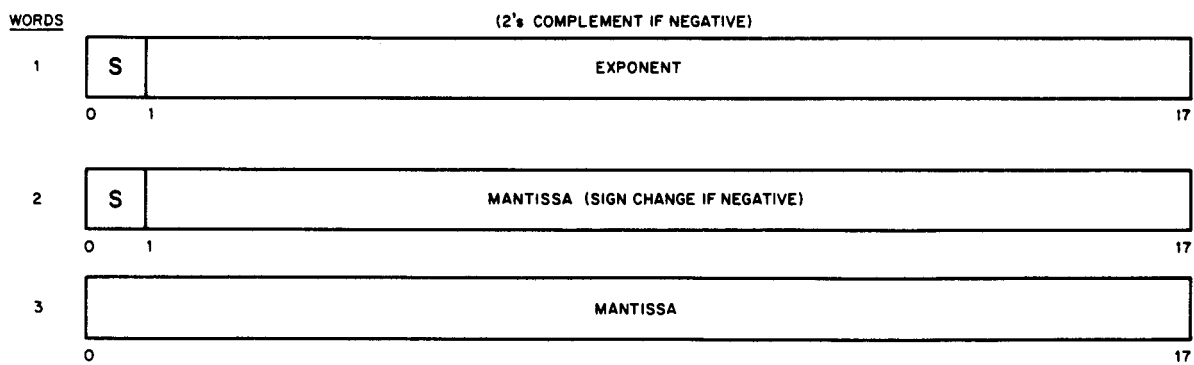
Figure 3-6 Data Word Format

3.3.3.2 Floating-Point Formats - Floating-point representation of a binary number consists of two parts: the exponent and the mantissa. The mantissa is a fraction with the binary point assumed to be positioned between the sign bit and the most significant data bit. The mantissa is always stored in a normalized state; i.e., leading 0s are eliminated from the binary representation so that the high order bit is always a 1. The exponent as stored represents the power of 2 by which the mantissa is multiplied to obtain the number's value for use in computation.

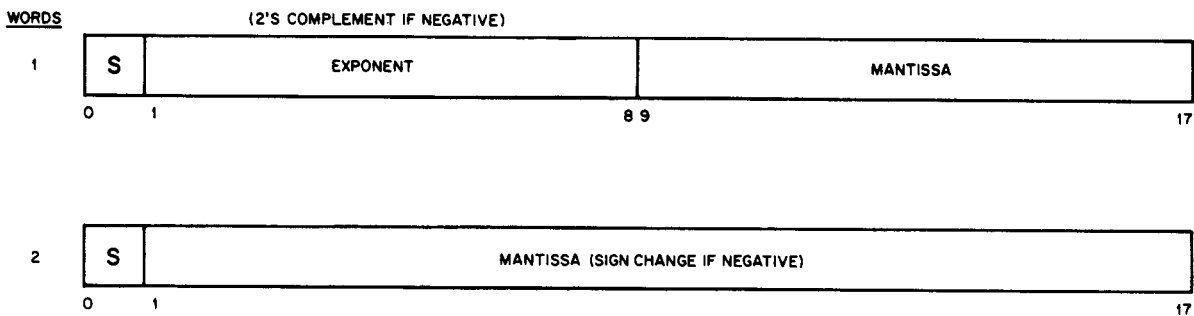
The PDP-9 floating-point software system offers two modes for storage of floating-point numbers: three-word mode and two-word mode. Three-word mode requires three memory locations for storage

of a floating-point binary number (Figure 3-7a). The exponent, a signed 17-bit integer in 2's complement notation, occupies the first word, or memory location. The mantissa, a 35-bit quantity in sign and magnitude notation, is stored in the second and third words. The sign of the mantissa is stored in the high-order bit of the second word.

Two-word mode requires two memory locations for storage of a floating-point binary number, (Figure 3-7b). The exponent, an 8-bit integer in 2's complement notation, and its sign occupy the 9 high-order bits of the first word. The mantissa, a 26-bit quantity in sign and magnitude notation, is stored in the 9 low-order bits of the first word and in the 17 low-order bits of the second word. The sign of the mantissa is stored in the high-order bit of the second word.



a. Three-Word Mode



b. Two-Word Mode

Figure 3-7 Floating Point Data Format

3.3.4 Scaling

For the maintenance technician interested in arithmetic data handling, the PDP-9 User Handbook F-95 contains further information on scaling considerations for fixed-point arithmetic, and descriptions

of the addition and subtraction processes. Multiply and divide processes are presented in the "Mathematical Subroutines" section of the Basic Software System Reference Manual, DEC-9B-GSAA-D. This document also includes descriptions of single- and multiple-precision arithmetic, plus the use of the floating-point software system.

3.4 CONTROL MEMORY SYSTEM

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 execution of instructions and also specify the timing of control memory readout. The CMA may be modified by conditions sensed during the processes, or 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.

3.4.1 Organization

Figure 3-8 is a functional block diagram of the control memory system. The system consists of CM timing logic, drawing KC16; two G210 Address Selectors, on drawing KC17, driving the MC09A Control Memory core array, drawing KC18(2); and 36 core-sensing control register Flip-Flops B213, drawing KC19.

The importance of keeping the control memory processes synchronized with the main core memory cycles may readily be appreciated. Both the main memory cycle and the control memory timing are started by the logical AND of the main clock (CLK) which produces CM CLK pulses every 1 μ s while the processor is running, and a start memory (SM) level from a CM sense flip-flop that is always set in the last CM process word during any computer cycle (fetch, defer, execute, IA0). The last CM process word in an execute cycle always contains the SM level and address 21 in the CM sense flip-flops. The next computer cycle must, of course, be a fetch cycle, and the next CM process word will be extracted from address 21. Both of these are initiated by SM (1) and CM CLK. 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 presented to the Address Selectors G210, and will always be address 12. The CM will now extract the process word at address 12. During this process the main core memory reads the instruction word into the MB and the op code portion into the IR.

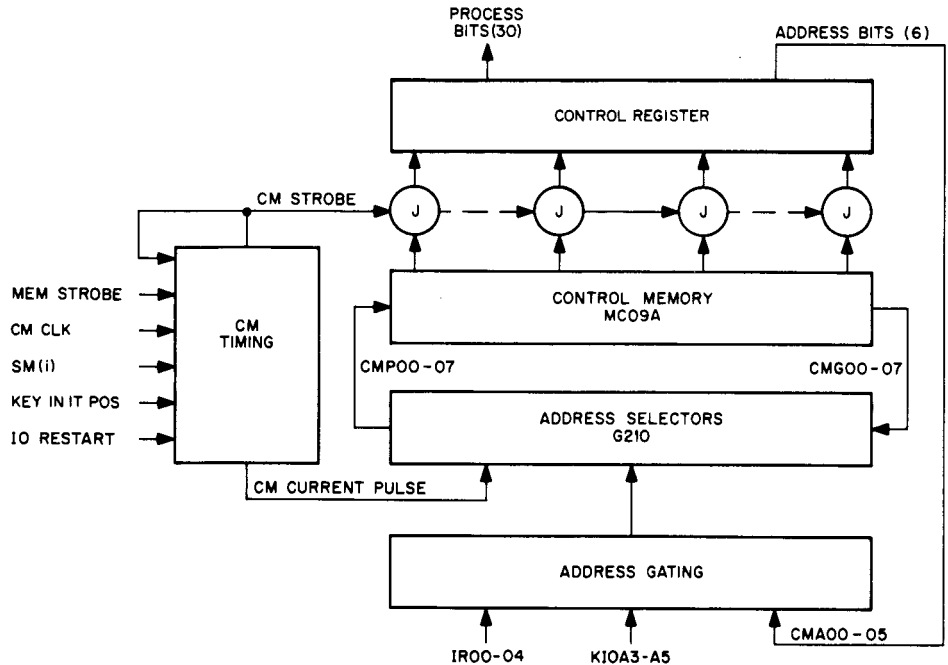


Figure 3-8 Control Memory System, Block Diagram

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 retriggering.

The address selectors contain positive drive and negative sink transistor switches operating 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 direction of the winding determines the state a core will assume when the line is driven. Core windings in one direction induce positive voltages in their respective sense lines, setting the sense flip-flops in the control register. Core windings in the other direction 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 3-8, the address in the control register may be preempted by manual operations from the console, where the levels K10A3-A5 perform the CM addressing function. See Manual Control, Section 3.7. Section 3.5 explains in detail the functions of all process words in executing the computer instructions. Drawing KCI8(1) is the Control Memory Program Chart in which one process word (LOC column) addresses 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. Below each block are the results of the processes. 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, denoted by dotted lines.

3.4.2 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. $CM\ CLK \wedge SM(1) \wedge AM\ SYNC\ BUS(0)$
- c. $CM\ STROBE \wedge CONT(1)$
- d. $MEM\ STROBE \wedge CONT(0)$
- e. IO RESTART

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 (Section 3.7). This pulse starts the chain to extract a series of manual entry process words from locations 00-07 using the KIO levels for CM addressing. The switches are used to set the starting address of a program into the MB, for example. In this case, the last manual entry process word contains SM (1) and address 21 for the subsequent fetch-entry process word.

$CM\ CLK \wedge SM(1)$ starts the timing chain to extract the entry word of any computer cycle concurrently with a core memory cycle (Section 3.5).

$CM\ STROBE \wedge CONT(1)$ starts the chain to extract the second word, and $MEM\ STROBE \wedge CONT(0)$ extracts the third, in any computer cycle. For execute cycles, $CM\ STROBE \wedge CONT(1)$ 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 3 μ s execute period (Section 3.5.8). This word prepares the computer for the next fetch cycle.

For manual READ IN operations, IO RESTART extracts a process word when the tape reader reads a hole in channel 7 (Section 3.7.4.9). This word also prepares the computer for the next fetch cycle.

The optional EAE logic also retriggers the chain with IO RESTART. Refer to the EAE Instruction Manual.

3.4.3 Control Memory MC09A

Control Memory MC09A 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. Complete wiring patterns of the MC09A are detailed in drawings MC1 through MC66, delivered with the system. 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.

3.4.4 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 drive-select and four sink-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 sink to the other. The module is similar to the G219 Address Selector Modules in core memory.

Figure 3-9 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.

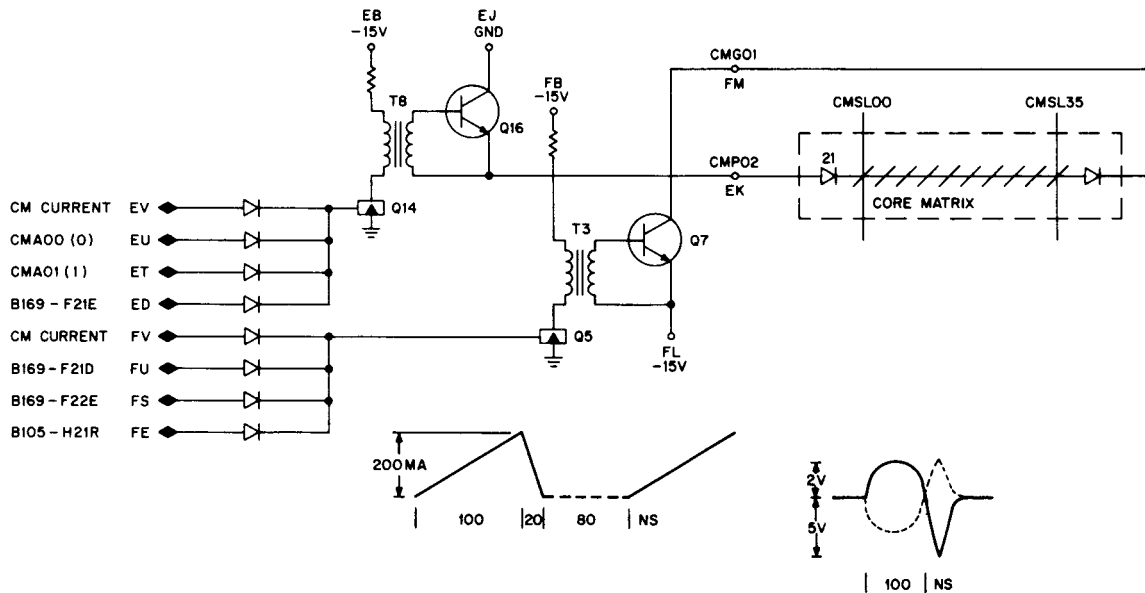


Figure 3-9 CM Line Selection, Line 21

For the fetch entry word extraction, the address in control register bits CMA00-05 is 21. In Figure 3-9 CMA00 (0), CMA01 (1), and the level from B169-F21E are all negative. On drawing KCl7, the output at B169-F21E comes from the paralleled NAND gates controlling address bit 02. 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 CMP02 connection to one end of core drive line 21; the Q7 collector output at FM is the CMG01 connection to the other end. Current flows through the line from CMP02 to CMG01.

3.4.5 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 purely by the inductance of the drive lines to 200 mA, inducing positive or negative signals of 2V (with 5V flyback) in the sense lines. The idealized waveforms are shown in Figure 3-9.

3.5 CENTRAL PROCESSOR LOGIC

The discussions that follow correlate the CP logic with the execution of computer instructions. The functions common to most instructions; i.e., fetch cycle, defer cycle, autoindexing, are treated as an introduction to the individual execute cycle descriptions. Flow charts and timing diagrams supplement the text.

3.5.1 Fetch Cycle

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 (Section 3.5.6). 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 Δ MB level, drawing KCl9(2). Δ MB is NANDed with SM(1) of the BGN word, MEM DONE from core memory, and RUN (1) from the clock and run logic, drawing KCl0(1). The NAND gate output at R111-E22N produces 1 \rightarrow MBI at Pulse Amplifier B602-E23D when MEM DONE of the execute cycle occurs. The 1 \rightarrow MBI pulse sets the MBI flip-flop in control memory. 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. MA JAM WORD and MA JAM DIGIT in core memory transfer this address from the MB to the MA (Section 3.6.3). 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.

Figure 3-10 is the timing diagram for the fetch cycle and Figure 3-11 is the flow diagram. At CLK time in the BGN process, CM CLK and SM(1) produce CM CURRENT and CM STROBE 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- μ s fetch cycle period or interval between CLK pulses. CLK and SM(1) also start the core memory cycle.

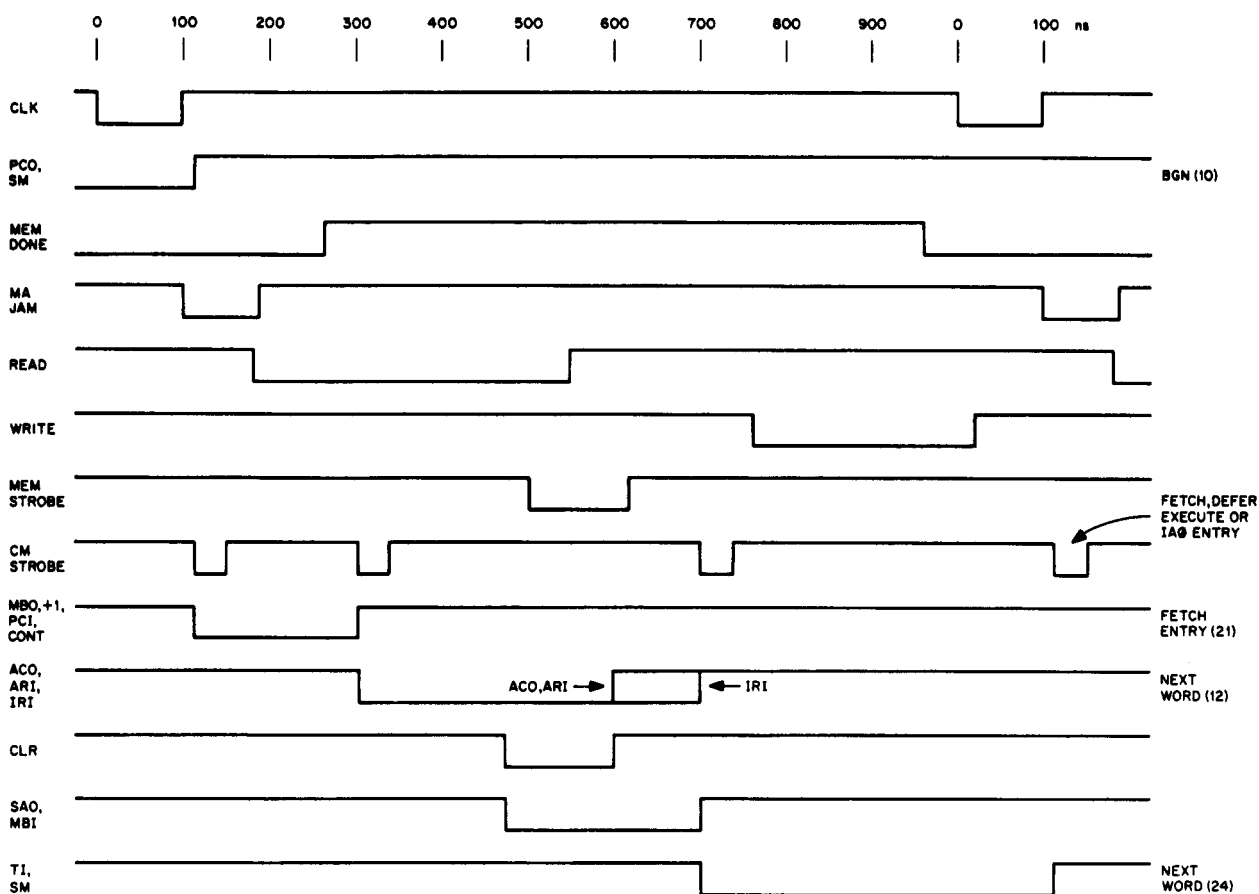


Figure 3-10 Fetch Cycle Timing

The fetch entry word 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 other than during shifting (SHL, SHR) 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 on drawing KC16. This pulse triggers a 50 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. TESTER is always negative, turning this inverter on. The collector passes NOR gate R111-F26U for CM CURRENT and pulses B602-F30D for CM STROBE after a 50-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.

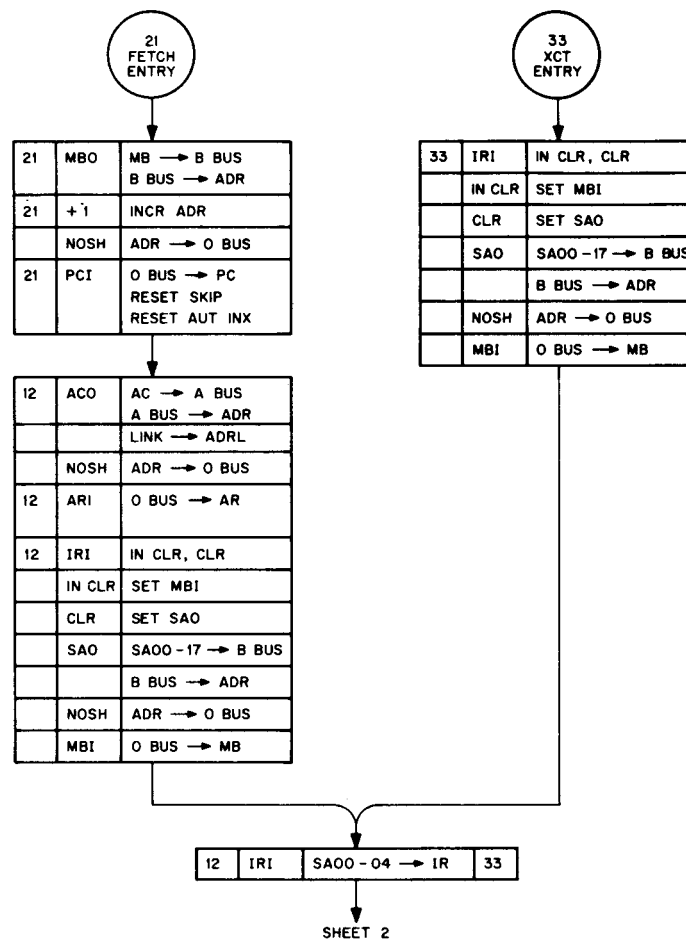


Figure 3-11 Fetch Cycle Flow, Sheet 1

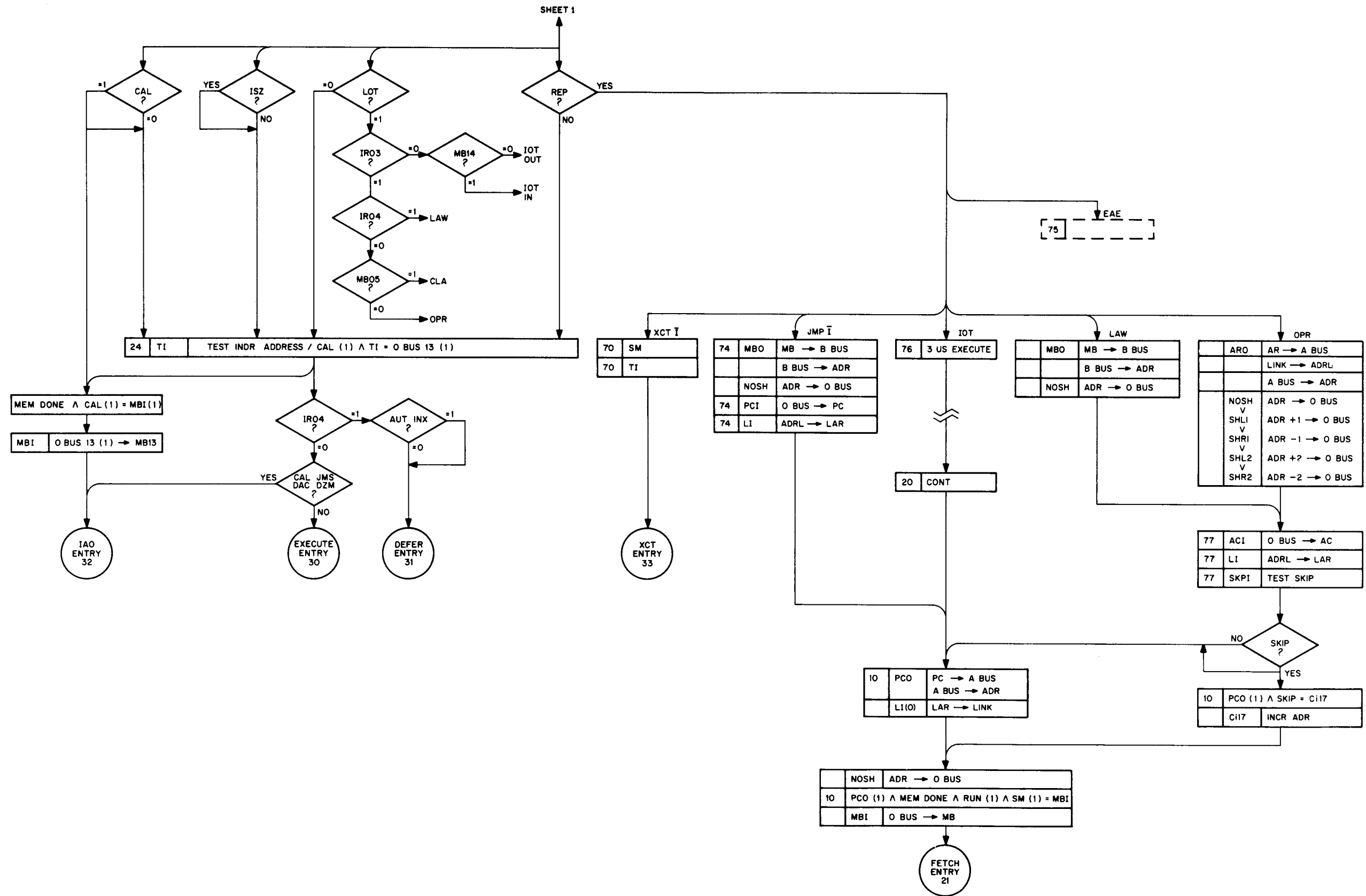


Figure 3-11 Fetch Cycle Flow, Sheet 2

The second word thus 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, and the content of the LINK is gated into the ADRL. 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 in conjunction with MBI(0) and CM STROBE DLYD. The inverter output goes to ground, triggering 50 ns delay B310-EF29FU. Upon recovery the delay produces an IN CLR pulse and a CLR pulse. IN CLR produces $1 \rightarrow$ MBI to set the MBI gate, drawing KC19(2), and generates CLR I, same drawing. CLR I resets ACI, ARI, PCI, MQI, and MBO. CLR resets +1 and ACO, and sets SAO. MEM STROBE, STROBE SAL, and STROBE SAR occur in core memory, drawing MC2. SAL and SAR strobe the sense amplifier contents SA00-17 out to the CP/memory interface. The sense amplifiers contain the instruction word read out of 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.

IRI(1) now examines the op code bits for certain instructions, drawing KC12. REP (repeat fetch) results if IRI(1) detects an IOT (R111-E14H), OPR (R111-E14H), $XCT \bar{I}$ (R111-E14U), $JMP \bar{I}$ (R111-E14U), or an optional EAE instruction (R111-E14N). IRI(1) also sets the LOT 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 CONT(0) start the CM timing, drawing KC16, for the third CM STROBE. MEM STROBE \wedge CONT(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 3 μ s execute period during which the main core memory is idle, and the next fetch cycle ensues. Optional EAE instructions can extend up to 17 μ s during which the main core memory is idle.

Assuming that there is no REP here, 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 IR04 at R111-F23H on drawing KC17 for indirect addressing. If IR04=1, then TI(1) \wedge IR04(1) at R111-F23H

boosts the CM address from 30 to 31 (defer entry). At the same time, $TI(1) \wedge IR04(1)$ 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 $IR04=0$, then $TI(1) \wedge IR04(0)$ examines the IR bits at R111–F234 for $CAL \bar{I}$, $JMS \bar{I}$, $DAC \bar{I}$, or $DZM \bar{I}$, drawing KC17. If any of these are detected, the address gates boost the address from 30 to 32 (IA0 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, so that 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 IA0 cycle for $JMS \bar{I}$, $CAL \bar{I}$, $DAC \bar{I}$, or $DZM \bar{I}$, or an execute cycle.

3.5.2 Defer Cycle

During the third process of the fetch cycle the address in the CMA for the next process word is 30 (execute entry). The TI (1) level samples the IR04 bit on drawing KC17. If $IR04=1$, 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, $TI(1) \wedge IR04(1)$ on drawing KC17 enables the CMA05 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 CMA05 gate boosts the existing address (30) to 31, from which the defer entry process word is taken at CM STROBE time.

Figure 3-12 is the defer cycle timing and Figure 3-13 shows the flow. Process word 31 contains the DEI (defer/execute initiate) process bit and CMA24. DEI(1) resets IR04 (and CAL) on drawing KC12. CM STROBE DLYD, MBI(0) and DEI(1) produce an IN CLR and a CLR pulse after a recovered delay, drawing KC16. IN CLR produces $1 \rightarrow MBI$ to set the MBI sense flip-flop, drawing KC19(2), and CLR sets SAO on drawing KC19(3), as for the fetch cycle. CM STROBE is prevented from restarting the CM timing chain for the next CM STROBE because of the absence of the 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 SAL, and STROBE SAR 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 IR04 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 IR04, it can now sample the IR for JMP I and XCT I.

MEM STROBE \wedge CONT(0) produces the CM STROBE which extracts the next process word from location 24 if not changed by REP as for fetch. Since DEI(1) has reset IR04 and CAL, TI(1) can neither enable CMA05 for another defer cycle, nor process a true CAL instruction in a subsequent IA0 cycle. If a CAL I instruction is programmed, it will be treated in the defer cycle as a JMS I 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 3.5.6.3 and 3.5.6.8. Similarly, an IA0 cycle follows defer if TI(1) \wedge IR04(0) detects a JMS I, DAC I, or DZM I 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.

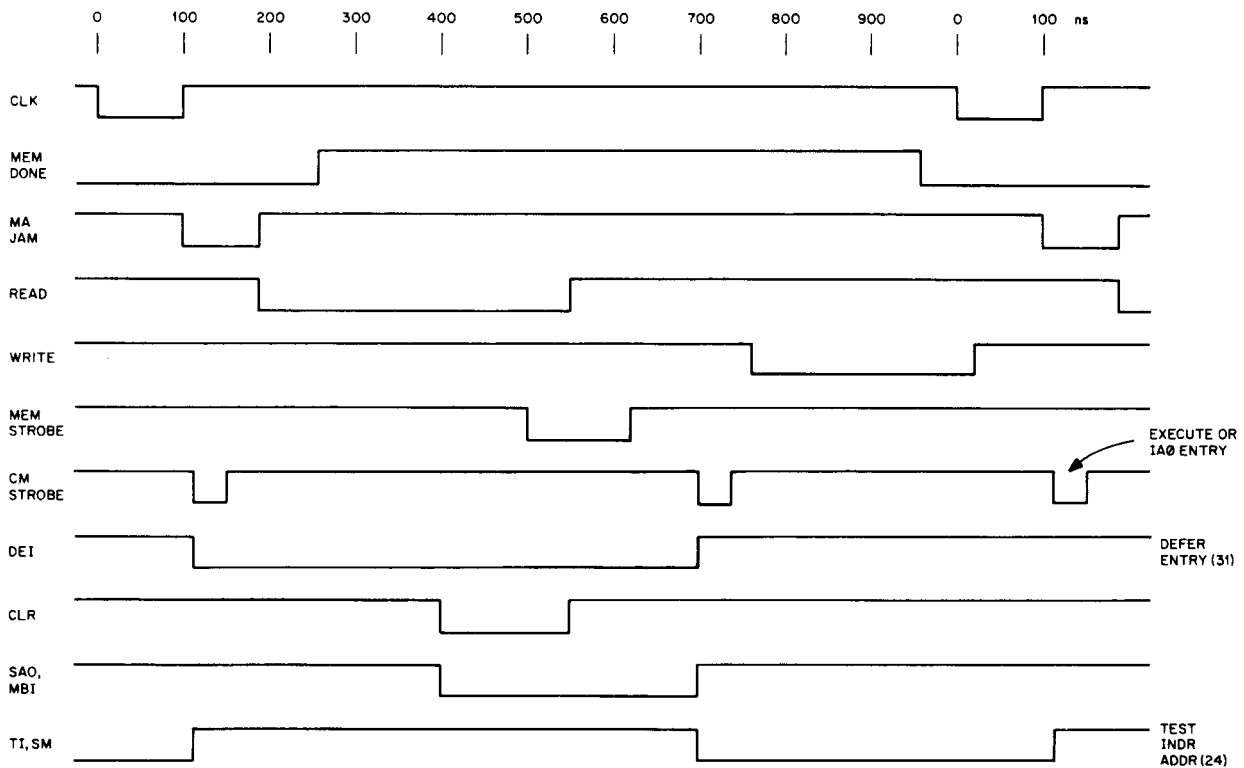


Figure 3-12 Defer Cycle Timing

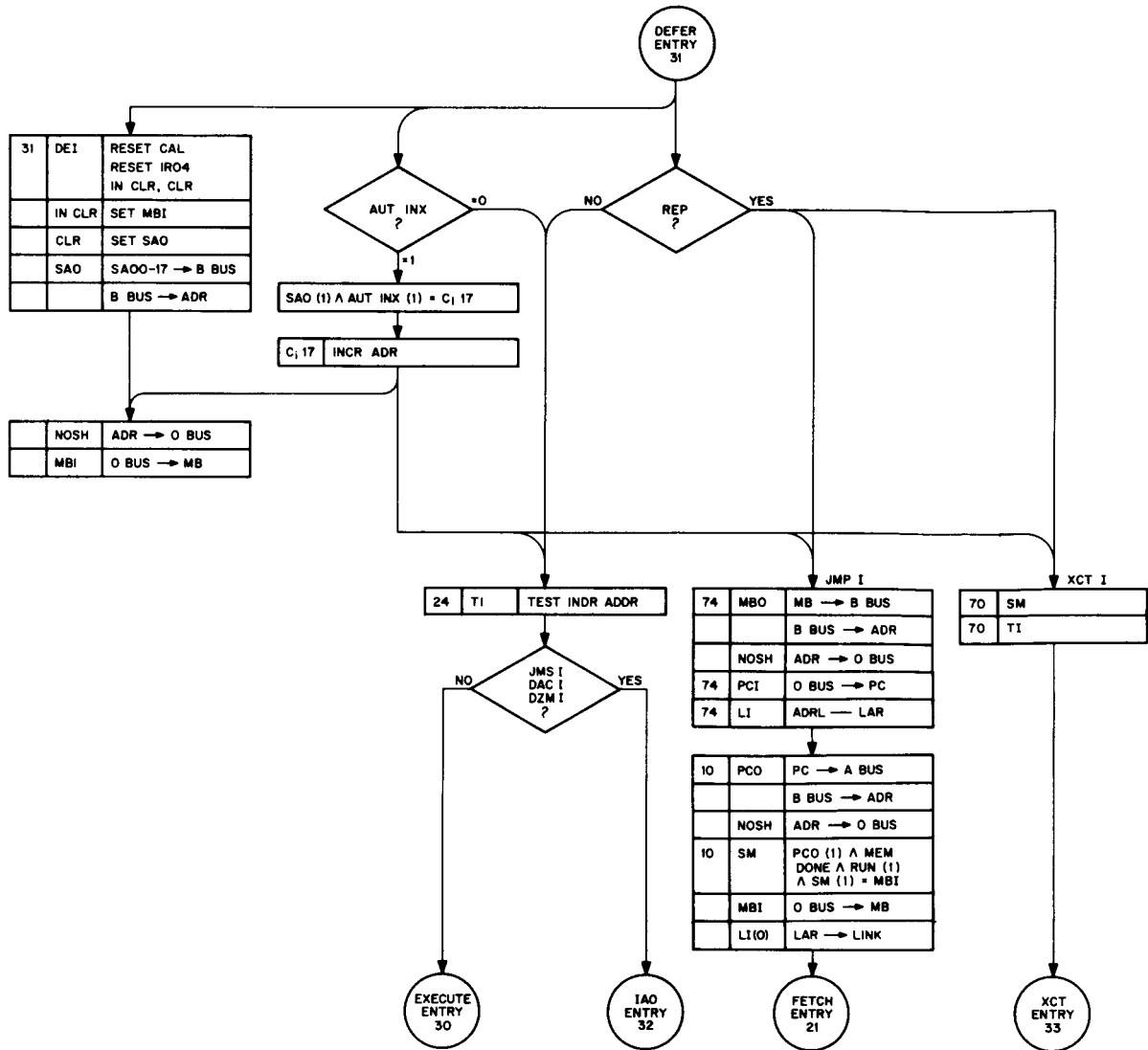


Figure 3-13 Defer Cycle Flow

3.5.3 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 3.5.1) where IR04 is examined by TI(1) for indirect addressing. For indirect addressing IR04=1, so the TI(1) and IR04(1) levels sample address bits MB05-14 of the instruction word at the AUT INX flip-flop, drawing KC14 and Figure 3-11. 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 and Figure 3-13 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 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. $TI(1) \wedge IR04(0)$ resets AUT INX during process word 24 or 70, and PCI(1) resets AUT INX during process word 74.

3.5.4 IA0 Cycle

The IA0 cycle replaces the normal execute cycle (Section 3.5.5) 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. See the appropriate descriptions in Section 3.5.6.

3.5.5 Execute Cycle

Figure 3-14 is the timing diagram for the execute cycle. The computer enters the execute cycle from either the fetch cycle (Section 3.5.1) or the defer cycle (Section 3.5.2). 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.

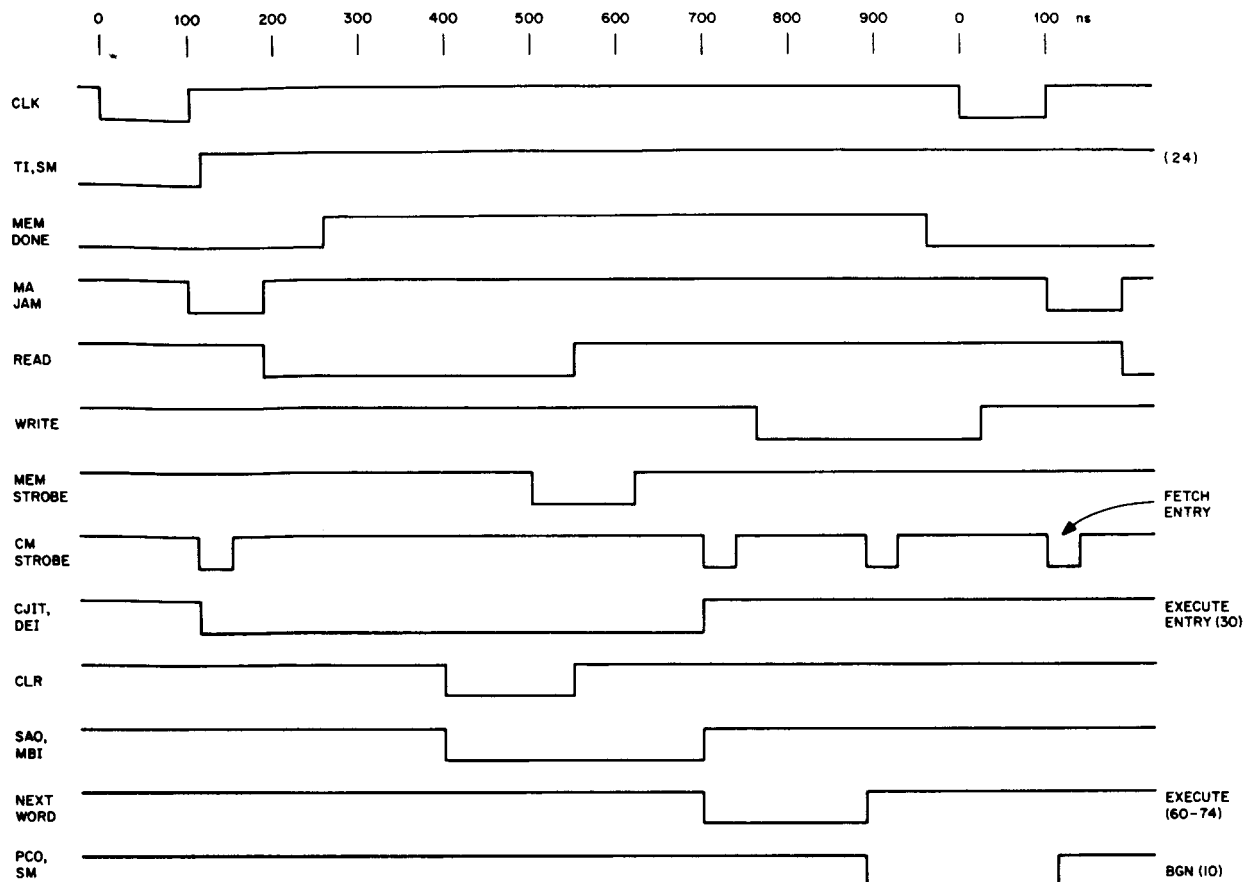


Figure 3-14 Execute Cycle Timing

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) and MBI(0), drawing KC16. IN CLR produces $1 \rightarrow$ 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 \wedge CONT(0) restarts the chain. STROBE SAL and STROBE SAR occur in core memory to place the operand in the MB. While the operand is en route through the ADR to the MB, CJIT(1) will produce CI17 on drawing KC14 if the ISZ instruction op code was detected, on drawing KC12, during fetch. CI17 then increments the operand by 1 in the ADR. If no ISZ, the operand remains unchanged.

MEM STROBE \wedge CONT(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 (CMA00 and CMA01 = 1). This is the starting point for the extraction of the execute word determined by the op code in the IR register. CMA00(1) and CMA01(1) on drawing KC17 allow the IR bits to address the control memory, so that the address 60 is boosted to the address specified by the op code.

All memory reference instructions (except XCT) require a single 200-ns process word for execution. For these instructions, the execute process word contains CMA 10 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 (ADD, AND, JMP, LAC, SAD, TAD, XOR), 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 (Section 3.5.4). In the last two cases, the original operand is lost.

IOT, OPR, and optional EAE 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.

3.5.6 Memory Reference Instructions

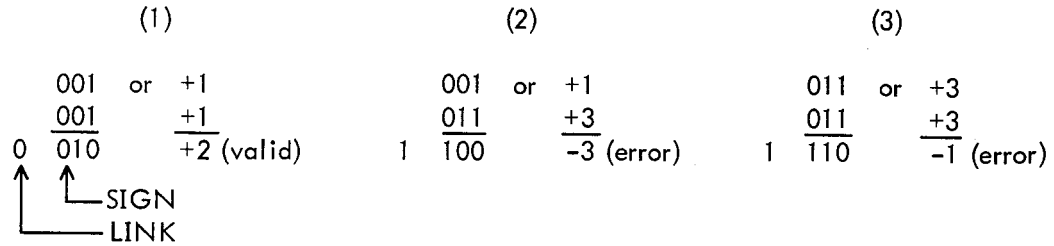
The following paragraphs describe the logic functions for execution of memory reference instructions. Whereas the preceding paragraphs cover the functions that are common to all instructions, the following discussions cover those functions that are unique to the individual instructions. The instructions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

3.5.6.1 1s Complement Add (ADD) - The ADD instruction (30) adds the contents of the addressed memory location (addend) to the contents of the AC (augend) in 1s 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 be reset previously, and remains reset unless an arithmetic overflow occurs.

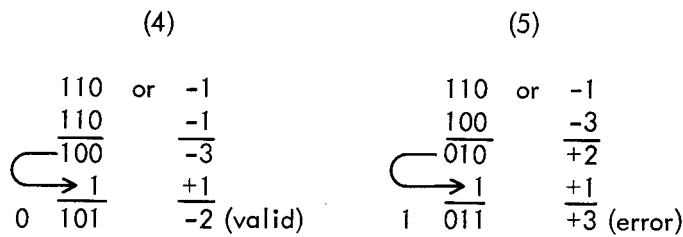
Under the rules of 1s complement arithmetic (Section 3.3), the sign bits 00 in both the addend and the augend are added as an integral part of the magnitude bits 01-17 during add. An end carry out of the sum sign-bit 00 is end-around carried into the sum magnitude-bit 17 to establish the final magnitude. 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 .

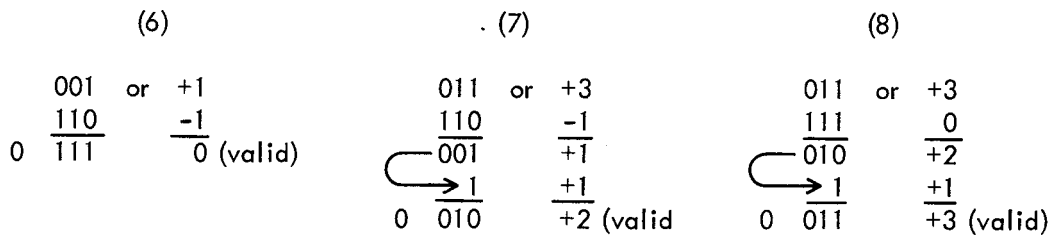
Positive Sign ADD



Negative Sign ADD



Unlike Sign ADD



Close examination reveals two basic rules of 1s complement addition: (1) in like-sign addition, overflow is possible and the sum sign differs if overflow occurs; (2) in unlike-sign addition, no overflow is possible and the sum sign may be plus or minus.

The fetch cycle (Section 3.5.1) 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 content of the LINK goes into the ADRL. The op code is sampled but does not alter the execute entry address (30).

During execute (Section 3.5.5) 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. CMA00(1) and CMA01(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. The contents of the buses are added in the ADR, with carries resulting where corresponding bits are 1s (A and B bus bits at ground). 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 algebraically summing the carries CO00 and CO01 out of ADR00 and ADR01. The following functions apply to overflow detection (see ADD examples).

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

$\text{ADOF} \wedge \text{AXS}(1) \wedge \text{LI}(1)$ 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 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. LI(0) sets the LINK in conjunction with LAR(1) as an indication of overflow.

An "add overflow" gate R111-C03N 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).

3.5.6.2 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 (Section 3.5.1) 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 (Section 3.5.5) 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. CMA00(1) and CMA01(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 contents of the A bus are gated onto the B bus by the AND(1) level ($\overline{A \text{ BUS}}$, drawing KC21). AND(1) on drawing KC13 produces CMPL, which is applied to each bit of the ADR to complement the half-add results. Figure 3-15 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.

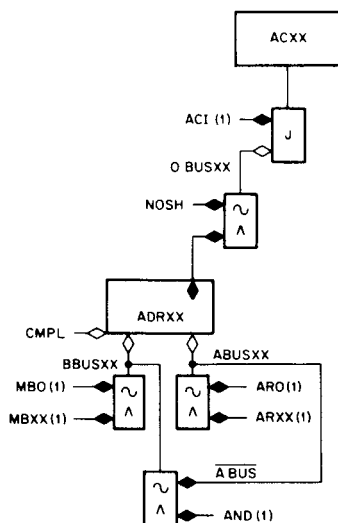


Figure 3-15 AND Logic

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 $\overline{A \text{ BUS}}$ 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 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.

3.5.6.3 Call Subroutine (CAL) - The CAL instruction (00) is equivalent to instruction JMS 20, Section 3.5.6.8. 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 (Section 3.5.1) 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 IR00, IR01, and IR04 at the CM address gating. For a CAL \bar{I} instruction (and DAC \bar{I} , DMZ \bar{I} , and JMS \bar{I}), 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 MEM DONE on drawing KC19(2) at the 1 \rightarrow MBI gate. MEM DONE occurs in core memory just before the next CM CLK pulse starts the IA0 cycle (Section 3.6.3). At the 1 \rightarrow MBI gating, $CAL(1) \wedge SM(1) \wedge MEM\ DONE \wedge RUN(1)$ produce the 1 \rightarrow 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. 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 (00020) to the B bus, and the contents go from the B bus directly to the ADR. Process +1 produces CII7 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 \rightarrow PCI on drawing KC12 in conjunction with the IR00, IR01, and IR03 bits (all 0s). The 1 \rightarrow 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 SAL, and STROBE SAR 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 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 I instruction can be used to return to the sequence stored at 00020. JMP I should be preceded by an IOT DBR instruction in order to restore the status bits to the system (Section 3.8.1.7).

3.5.6.4 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 (Section 3.5.1) 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 IR00, IR01, and IR04 at the CM address gating. For a DAC \bar{I} instruction (and CAL \bar{I} , JMS \bar{I} , DZM \bar{I}) 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, the LINK, the 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 $CJIT(1) \wedge IR03(0)$ produce 1 \rightarrow PCI on drawing KC12 for program count and status storage. For DAC, $IR03=1$, so that the PCI gate does not set.

Note that MEM STROBE, STROBE SAL, and STROBE SAR 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. CMA00(1) and CMA01(1) allow the IR bits to address the control memory, in which case $IR03(1)$ 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, 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 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.

3.5.6.5 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 (Section 3.5.1) 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 $IR00$, $IR01$, and $IR04$ at the CM address gating. For a DZM \bar{I} instruction (and CAL \bar{I} , DAC \bar{I} , JMS \bar{I}) 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, the LINK, the 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 $CJIT \wedge IR03(0)$ on drawing KC12 produce 1 \longrightarrow PCI for program count and status storage. For DZM, $IR03=1$, so that the PCI gate does not set.

Note that MEM STROBE, STROBE SAL, and STROBE SAR 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. CMA00(1) and CMA01(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 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.

3.5.6.6 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. For proper execution of ISZ, the addressed memory location must initially contain a negatively-signed, 2s complement quantity representing a positive integer. ISZ increments the magnitude of the quantity toward 0 (overflow).

The fetch cycle (Section 3.5.1) 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 (Section 3.5.5) 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 CONT(0) restart the CM timing on drawing KC16. Since the CM address in process word 30 is 60, CMA00(1) and CMA01(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. 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 KC15. 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, during 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.

3.5.6.7 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 ($\overline{\text{JMP I}}$, Figure 3-11). If the JMP instruction contains an indirect address, the computer enters a defer cycle to fetch the effective address. REP during defer then transfers the effective address to the PC (Figure 3-13). 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 (Section 3.5.1) IRI(1) detects the $\overline{\text{JMP I}}$ op code in the IR bits to provide the REP ground level, drawing KC12. During defer (Section 3.5.2) DEI(1) resets IR04, 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. 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 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 I is preceded by DBR, Section 3.8.1.7.

3.5.6.8 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 (Section 3.5.1) 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 IR00, IR01, and IR04 at the CM address gating. For a JMS \bar{I} instruction (and DAC \bar{I} , DZM \bar{I} , CAL \bar{I}) 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 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 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, which increments 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 $1 \rightarrow \text{PCI}$ on drawing KC12 in conjunction with the IR00, IR01, and IR03 bits (all 0s). The $1 \rightarrow \text{PCI}$ 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 SAL, and STROBE SAR 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 (CMA00, CMA01=1). CMA00(1) and CMA01(1) allow the IR bits to address the control memory, in which case IR02(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 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 I instruction can be used to return to the stored sequence at Y. JMP I should be preceded by an IOT DBR to restore the stored status bits to the system (Section 3.8.1.7).

3.5.6.9 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 (Section 3.5.1) 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 (Section 3.5.5) 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. CMA00(1) and CMA01(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. 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 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.

3.5.6.10 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 so that 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 (Section 3.5.1) 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 (Section 3.5.5) 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. CMA00(1) and CMA01(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 complementary outputs of the MB to the B bus, and ACO(1) takes the direct outputs of the AC to the A bus (see Figure 3-16). Both A bus and B bus contents go directly to the ADR. If the output bus levels are both at ground or 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 IR03(1) on drawing KC13. All ADR bit outputs are inverted and 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 $\overline{\text{ADRA}}=0$ or $\overline{\text{ADRB}}=0$ 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 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 CII7, 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.

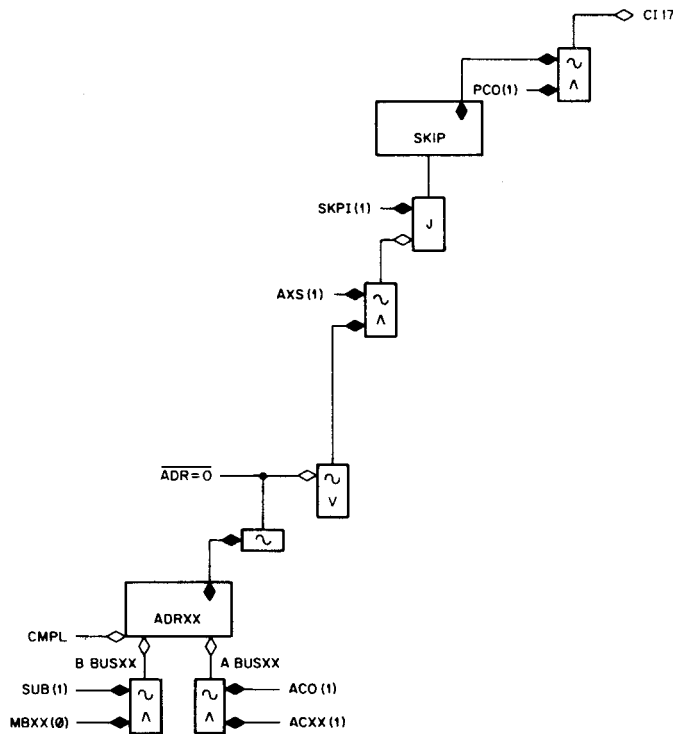


Figure 3-16 SAD Logic

3.5.6.11 2s Complement Add (TAD) - The TAD instruction (34) adds the contents of the addressed memory location to the contents of the AC in 2s 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. An end carry out of the sum sign bit 00 complements the LINK.

The fetch cycle (Section 3.5.1) 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 (Section 3.5.5) 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 (CMA00, CMA01=1). CMA00(1) and CMA01(1) allow the IR bits to address the control 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. The contents of the buses are added in the ADR, with carries resulting where corresponding bits are 1s (A bus and B bus at ground).

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 end carry resulted from ADR00, ADRL is negative. If the LINK was set and an end 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 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.

3.5.6.12 Execute (XCT) - The XCT instruction (40) causes the computer to execute the instruction contained in the addressed memory cell. If the XCT instruction contains a direct address, REP is detected during fetch (XCT \bar{I} , Figure 3-11). The computer waits for the next CLK pulse, then enters a quasi-fetch cycle (XCT entry, Figure 3-11) 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 (Section 3.5.1) IRI(1) in process word 12 detects XCT \bar{I} in the IR bits to provide a ground REP level, drawing KC12 (IR04=0 at R002-E13M). During defer (Section 3.5.2) DEI(1) resets IR04 and thus also detects an XCT op code for REP. In either case REP and CMA01(1) at the CM addressing logic, drawing KC17, gate the IR bits into the address selectors. CMA01(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 CONT(0) causes the next (third) fetch cycle process word 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 I to be used in the optional extend mode addressing scheme.

For XCT entry, Figure 3-11, the CM STROBE initiated by SM(1) and CM CLK extracts the XCT entry word in 33. This word 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) and IRI(1). IN CLR produces 1 \rightarrow MBI to set the MBI flip-flop, drawing KC19(2), and CLR sets SAO on drawing KC19(3). STROBE SAL and SAR 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 CONT(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.

3.5.6.13 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 (Section 3.5.1) 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 (Section 3.5.5) 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 (CMA00, CMA01=1). CMA00(1) and CMA01(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 3-17 illustrates the XOR logic for one bit position. SUB(1) takes the complementary contents 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 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 IR03(1) on drawing KC13. NOSH gates the ADR bits to the O bus, and ACI(1) gates them into the AC.

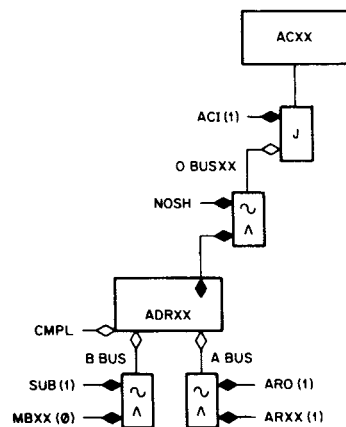


Figure 3-17 XOR Logic

DONE(1) goes to the clock and run logic, drawing KC10(1), for manual key control, and 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.

3.5.7 Operate (OPR) Instructions

OPR instructions (op code 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 logical operations on the current contents of the AC and/or LINK. The operations to be performed are encoded in bits 04-17 in the instruction word as described in Section 3.2.2.1.

During process word 12 of the fetch cycle (Section 3.5.1 and Figure 3-11) the OPR instruction is placed in the MB and the op code portion is placed in the IR. Also, the contents of the AC and the LINK are transferred to the AR and the ADRL, in anticipation of OPR and certain other instructions. 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 IR03-04 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 into the ADR. NOSH places the ADR contents on the O bus. ACI(1) of process word 77 gates the instruction into the AC. Thus, the LAW instruction transfers a constant within the instruction (MB05-17) into the AC. This technique obviates the necessity for storing and reading the constant at a separate core memory location.

If IR03 is 1 and IR04-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.

Also on the third CM STROBE, ARO RESTORE sets the IO BUS ON gate, drawing KC19(3). This gate applies the ADR contents to the I/O bus.

If MB05 is 1 at the ARO RESTORE gate, 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), IR03(1), and IR04(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 from the PC into the 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, and the foregoing discussion is common and preparatory to all instructions. The descriptions that follow are arranged in numerical order.

3.5.7.1 No Operation (NOP) - The NOP instruction (740000) is a "do nothing" instruction which delays the computer program for the duration of one cycle. NOP may be used to synchronize program timing with I/O device timing. During process word 12 the NOP instruction is placed 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 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 (MB06-17 are all 0s at the operate logic, drawing KC13).

3.5.7.2 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. During process

word 12 the CMA instruction is placed 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 process word 77 on the next CM STROBE. CM STROBE extracts the process word and sets the OP flip-flop, drawing KC12.

$OP(1) \wedge MB17(1)$ 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.

3.5.7.3 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.

During process word 12 the state of the LINK enters the ADRL, drawing KC15, and $ACO(1)$ gates the contents of the AC onto the A bus. If the LINK is set, the ADRL (adder link) goes negative (set). $ADRL(1)$ applies a set level to the LAR in conjunction with the negation levels $AXS(0)$, \overline{SHIFT} , etc., at the gate labeled "normal". $ARI(1)$ of process word 12 gates the contents of the O bus into the AR. The O bus contains the contents that were gated out of the AC.

The CM STROBE extracts the OPR process word 77 and sets the OP flip-flop drawing KC12. $LI(1)$ strobes the LAR during the OPR process. CML forces the ADRL to ground. CML is generated by $OP(1) \wedge MB16(1)$ on drawing KC13. Since the ADRL is now at ground the "normal" gate disables, and $LI(1)$ resets the LAR. At BGN the LI level goes to ground. In so doing, it resets the LINK in conjunction with reset LAR.

The converse is true for the case where the LINK is originally in the reset state. In this case, the ADRL goes negative through CML, so that its output sampling gate sets the LAR. $LI(0)$ at BGN time then sets the LINK.

3.5.7.4 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.

During process word 12 the OAS instruction is placed 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 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). $OP(1) \wedge MB15(1)$ 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.

3.5.7.5 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.

During process word 12 the RAL instruction is placed 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 generate ARORESTORE, 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). $OP(1) \wedge MB07(0) \wedge MB14(1)$ 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 on drawing KC15. 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 negative \overline{SHIFT} 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.

3.5.7.6 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.

During process word 12 the RAR instruction is placed in the MB, the op code portion in the IR, and the contents of the AC in the AR and LAR. 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). $OP(1) \wedge MB07(0) \wedge MB13(1)$ 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, 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 the AC17. ACI(1) of the OPR process word transfers the shifted contents of the O bus into the AC. On drawing KC13, the negative \overline{SHIFT} 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.

3.5.7.7 Halt Program (HLT) - The HLT instruction (740040) stops program execution. During process word 12 the HLT instruction is placed 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 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). $OP(1) \wedge MB12(1)$ results in $RUN(0)$ on drawing KC10(1). $RUN(0)$ 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.

3.5.7.8 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 SMA 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 samples the ADR00 bit at the AC SIGN flip-flop, drawing KC15. ADR00 represents the sign bit, AC00. If ADR00=1, $ARI(1)$ sets the AC SIGN flip-flop.

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 OR ARO RESTORE, drawing KC19(3). $AC\ SIGN(1) \wedge MB11(1)$ 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 $ARO(1)$ 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 goes through the ADR to the O bus and MB. $LI(0)$ recirculates the content of the LAR into the LINK.

3.5.7.9 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.

During process word 12 the SZA 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, 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 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). ADR=0 SAVE(1) \wedge 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) merely 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.

3.5.7.10 Skip on Non-Zero LINK (SNL) - The SNL instruction (740400) tests the status of the LINK. If the LINK is set, the computer skips the next instruction. If the LINK is reset, the computer executes the next instruction. The contents of the AC and the LINK remain unchanged.

During process word 12 the SNL instruction is placed 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 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).

If the LINK is set, LINK(1) \wedge 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. 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 content of the LAR into the LINK.

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

During process word 12 the SKP instruction is placed 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 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). 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.

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 content of the LAR into the LINK.

3.5.7.12 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 samples 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 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). The OP(0) level at the paralleled NAND gates in B169-D38 goes to ground. This holds the output at D38E at a negative level since all other gates are disabled. 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) 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 content of the LAR into the LINK.

3.5.7.13 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 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, an output bus on the ADR (ADRA=0, ADRB=0) 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 KC15. ARI(1) of process word 12 resets the flip-flop in conjunction with the ground level(s).

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). The OP(0) and ADR=0 SAVE(1) levels are both at ground, therefore, at the respective NAND gates in B169-D38, drawing KC14. Since the other paralleled gates are disabled, the output at D38E is negative. 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 recirculates 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.

3.5.7.14 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.

During process word 12 the SZL instruction is placed in the MB, the op code portion in the IR, and the contents of the AC in the AR. If the LINK is reset, LINK(1) at the B169-D38 gate is at ground, drawing KC14.

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 ARO flip-flop in conjunction with ARO RESTORE, drawing KC19(3). On drawing KC14 the around OP(0) level at the B169-D38 NAND gate holds the output at B169-B38D at a negative level. 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.

3.5.7.15 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.

During process word 12 the RTL instruction is placed 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 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). $OP(1) \wedge MB07(1) \wedge MB14(1)$ 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{SHIFT} go to ground at the "normal" gate, so that only $SHL2 \wedge ADR01$ controls a set enable input to the LAR. Also, the ground \overline{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.

3.5.7.16 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.

During process word 12 the RTR instruction is placed 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 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). $OP(1) \wedge MB07(1) \wedge MB13(1)$ 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 $\text{SHR2} \wedge \text{ADR16}$ 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.

3.5.7.17 Clear the LINK (CLL) - The CLL instruction (744000) clears the LINK. The AC contents remain the same. During process word 12 the CLL instruction is placed in the MB, the op code portion in the IR, and the contents of the AC and LINK in the AR and ADRL. 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). $\text{OP}(1) \wedge \text{MB06}(1)$ 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.

3.5.7.18 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. (See Sections 3.5.7.3 and 3.5.7.17.)

3.5.7.19 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 SHLI 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. (See Sections 3.5.7.5 and 3.5.7.17.)

3.5.7.20 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. (See Sections 3.5.7.6 and 3.5.7.17.)

3.5.7.21 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.

During process word 12 the CLA instruction is placed 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 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.

3.5.7.22 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 complements the ADR to all 1s, and ACI(1) places all 1s in the AC. (See Sections 3.5.7.2 and 3.5.7.21.)

3.5.7.23 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 inclusively ORs the data set into 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. (See Sections 3.5.7.4 and 3.5.7.21.)

3.5.7.24 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 BIT00. LI(1) of the OPR process word transfers a 0 into the LAR from

cleared ADR00 and ACI(1) transfers all 0s into the AC, except in the case where 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 3.5.7.5 and 3.5.7.21.)

3.5.7.25 Load the Accumulator with 76XXXX (LAW) - The LAW instruction (76XXXX) loads a constant included in the instruction (bits 05-17) into the AC. The previous contents of the AC are lost and the LINK remains the same.

During process word 12 the LAW instruction is placed 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 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 and 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.

3.5.8 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 (4 μ s) fetch cycle. Bits 06-11 and 12-13 in the instruction select the I/O device and device operating mode for the program-controlled data transfer, as described in Section 3.8. The remaining bits (14-17) command the IOT operations.

During fetch STROBE SAL, SAR 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 3-18. 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 with STROBE SAL, SAR (Section 3.6.3) 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 above. 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.

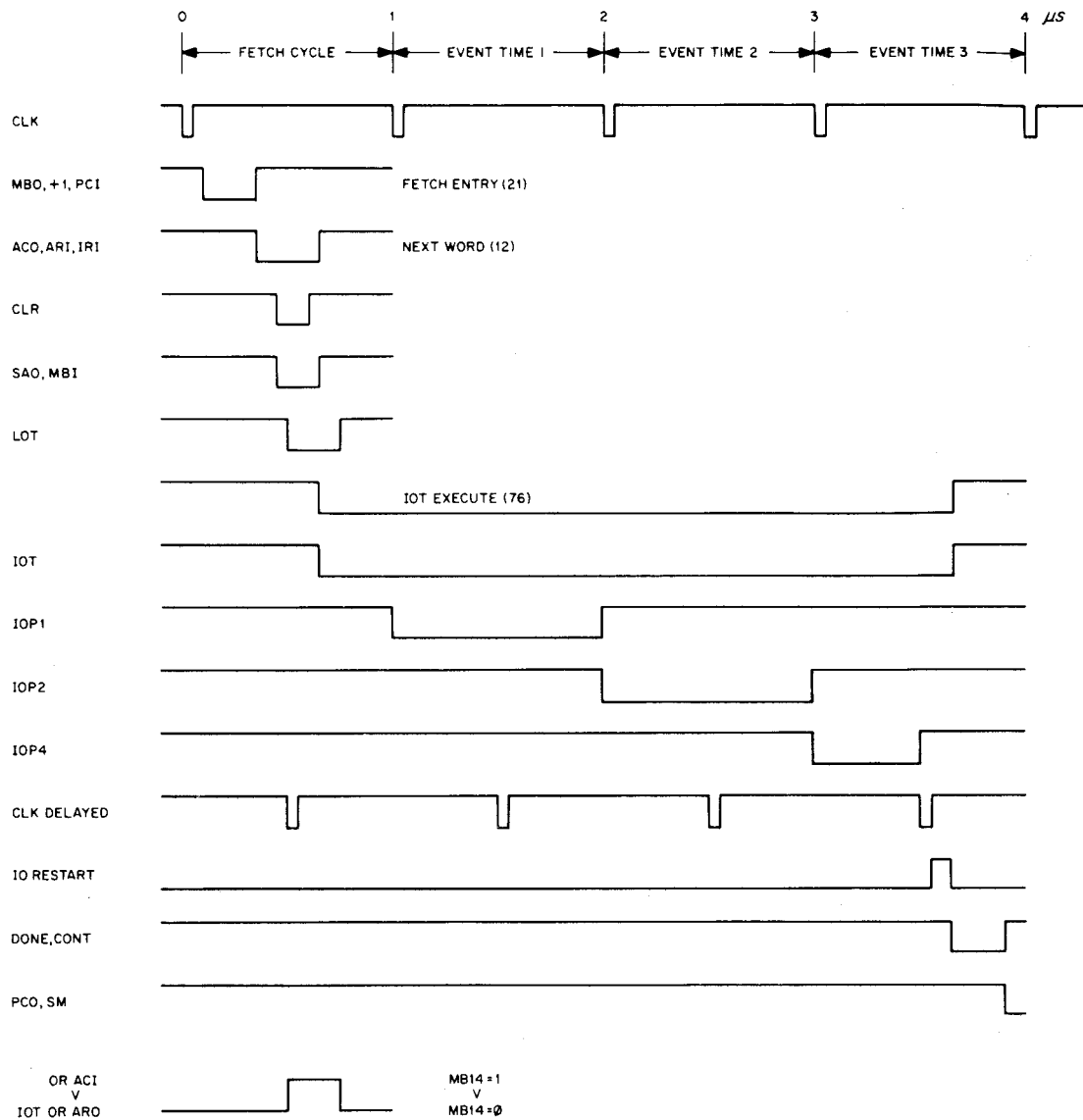


Figure 3-18 IOT Timing

MB14(0) is programmed for an output data transfer, generating IOT OR ARO at NAND gate R111-F10U. IOT OR ARO sets the ARO and the IO 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, Section 3.5.1). 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 above 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 nothing more than 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 MB06-13 into bus drivers B213, from which they emerge as device select levels DS00-05 and SD00-01. These device 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 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(1), to the I/O control via the CP-I/O interface. In Figure 3-18, 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 μ s 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 CLK pulse (IO CLK POS) also resets the pulse counter IO0, IO1 to 00. The second CLK 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 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 500 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 500 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/IO interface. This pulse restarts the CM timing. At the same time the IOP4 flip-flop resets if set on the earlier IO1(1) signal.

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(1), 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 themselves and the devices which they control, refer to I/O control, Section 3.8, basic I/O device control, Section 3.9, and any supplied I/O option manuals.

3.6 CORE MEMORY SYSTEM

The basic PDP-9 is equipped with a DEC Type MC70B Memory System having a 2-1/2D organization and a two-port data transfer capability. One port is connected to the CP/memory interface to service the central processor, and the other to the DMA channel for fast data transfers to/from optional I/O devices. The system can store 8,192 18-bit words, using a 13-bit address register. The necessary wiring for the addition of a 19th-bit DEC Type MP09A Memory Parity option is also included for use with a 19-bit optional core stack. Memory capacity can be extended in 8,192 word-increments up to a maximum of 32,768 words by adding DEC Type MM09A, MM09B, and MM09C expansion packages. Each package contains its own 13-bit address register and control logic. The Memory Extension Control KG09A is also required for extended memory systems. The following discussions cover the operation of the basic 8K system, but are equally applicable to extended systems. For information on extended memory, memory parity, and DMA options, refer to the respective option manuals. The DMA channel is described briefly in Section 3.8.3.

3.6.1 Organization

Figure 3-19 shows the elements which constitute the 8K memory system. The memory buffer register (MB), instruction register (IR), and program counter (PC) are part of the central processor, whereas the memory address register (MA) is included in the memory package. The read and write timing signals originating in the control and timing circuits are synchronized with the CP operations. Both read and write operations take place in one computer cycle, permitting random access to any memory cell within 1 μ s. The address selectors and diode matrices act as the current drivers for selected X and Y core drive lines.

3.6.2 Core Array and Field Selection

The 2-1/2D memory system derives its name from the magnetic core arrangement and the core driving scheme. The core array is two-dimensional (planar) as for a simple 2D system. In the pure 2D and 3D systems, the X axis is normally the address-word decoding dimension, and Y axis the bit decoding dimension. In the 2-1/2D system the Y axis serves a dual function, as half of the address word dimension during read and write operations, and as the data bit dimension during write operations.

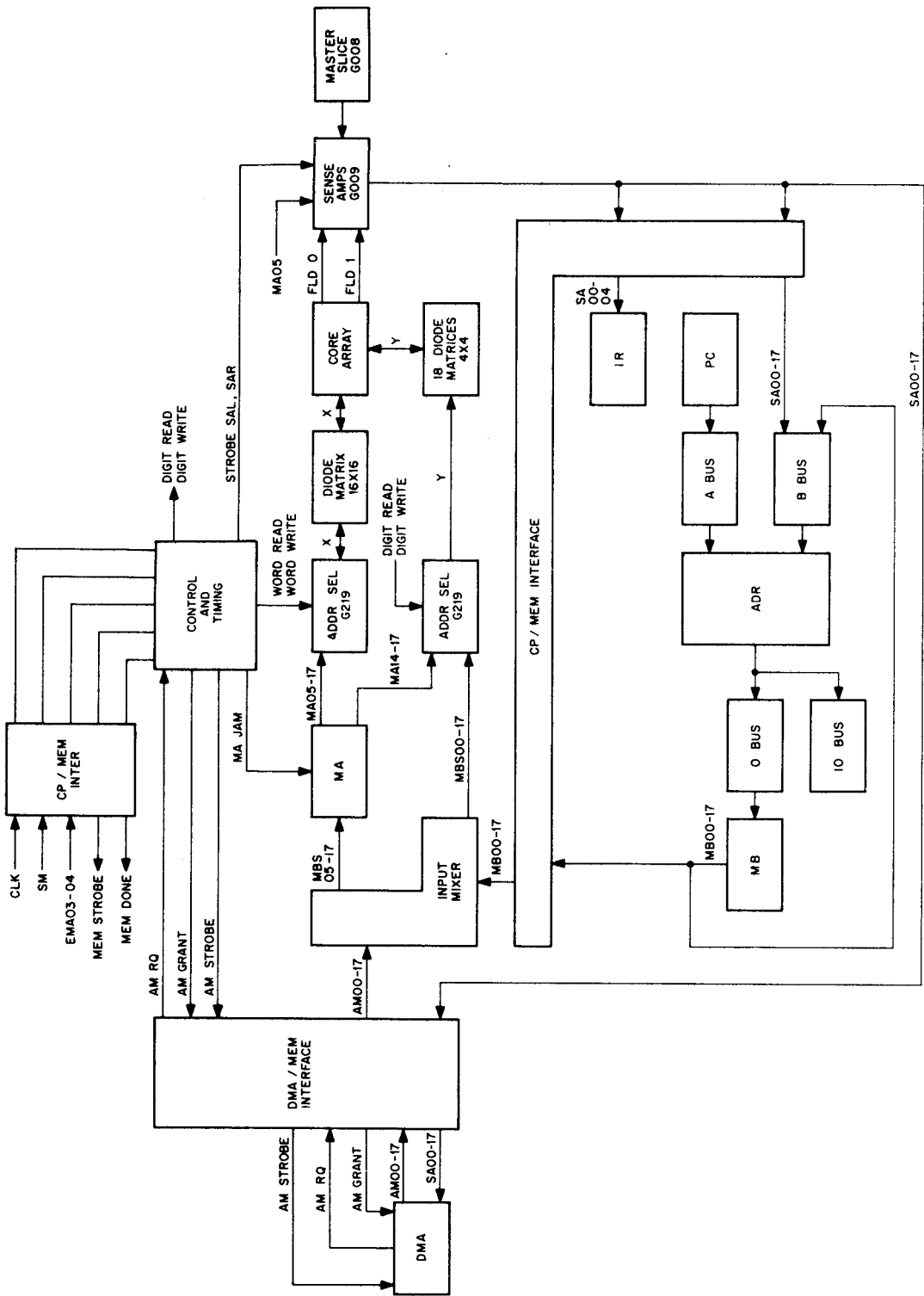


Figure 3-19 Core Memory System Block Diagram

As shown in Figure 3-20, the magnetic core array for each bit position consists of two 256 x 16 core planes, or 4096 cores each. The Y-axis lines are shown vertically, terminating at drive selector switches at each end. The X-axis lines intersect the cores horizontally through one field of 18 core planes, then return through the second field to the selector switches. With this arrangement, the 256 X lines drive 512 rows of 288 core columns. To select 1 core of 4096 in a plane, 1 of 256 X lines and 1 of 16 Y lines are selected to produce coincident half-currents through the intersected core. Note that the selected X-axis line intersects one row of cores in each field. However, the direction of X- and Y-line half-currents are coincident in one field, but anticomponent in the other. Current direction in the X-axis line depends on both the read/write control levels from the control and timing circuits and on the high-order field bit (MA05) in the address selectors. MA05 (0) controls the field 0 read/write current directions, and MA05 (1) controls the field 1 read/write current directions. Use of MA05 in this manner reduces the necessary word decoding matrix from 16 x 32 to 16 x 16, thus eliminating 16 bipolar switches. Current direction in the Y-axis line depends merely on the read/write control levels.

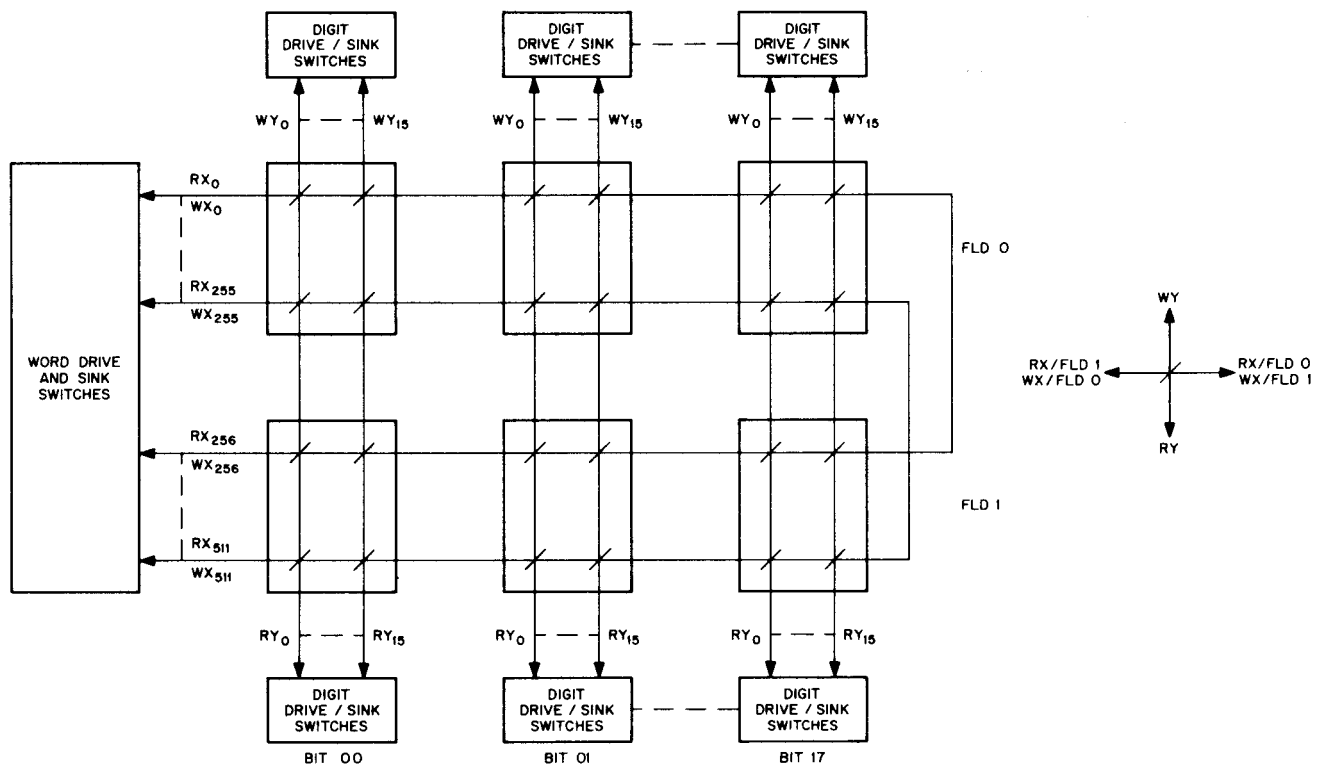


Figure 3-20 Planar Array Wiring

Figure 3-21 shows the wiring for one bit position and the sense lines coming from both fields to the corresponding sense amplifier. The half-current in the selected X line is coincident with the

half-current in the selected Y line in one field only, inducing a flux change in the intersected core. Read half-currents produce the flux change in such a direction as to cause a core in the 1 state to change to the 0 state. A core already in the 0 state remains at 0. When a core changes from 1 to 0, a voltage is induced in the sense winding, and the associated sense amplifier digitizes the voltage as a 1. The master slice control sets the operating threshold of the sense amplifier to prevent sensing of noise, and a strobe pulse from the control and timing circuits allows sampling of the output when the signal/noise ratio is greatest.

Write half-currents produce a flux change in the opposite direction, so that a core in the 0 state changes to the 1 state. During write, only the data bits which are binary 1s permit writing 1s into the respective cores. Data bits in the 0 state prevent the selection of Y half-currents. Thus, in this case, the inactive Y lines replace the write inhibit lines commonly used in 3D systems.

3.6.3 Control and Timing

The memory control and memory timing circuits are shown on drawings MC1 and MC2 respectively. The circuits include B310 and B360 delay modules adjusted for precisely timed output pulses for application to control flip-flops B213. The flip-flops in turn control the core drive-selection circuits in the address selectors G219.

For CP access to core memory, the memory read/write cycle concurs with a computer fetch, defer, execute, or IA0 cycle as dictated by the control memory processes in the CP. The read/write cycle and the computer cycle are synchronized to start on a computer CLK pulse by an SM(1) level originating in the control memory processes.

For DMA access to core memory, the memory read/write cycle and the DMA cycle are synchronized to start on a computer CLK pulse by an AM RQ(1) level originating in the optional Adapter/Multiplexer DM09A in the DMA channel. When an AM RQ is present at the current CLK time, the CP continues its access for the current cycle, then relinquishes it to the DMA on the next CLK pulse.

Figure 3-22 is the timing diagram for one CP-accessed read/write cycle. In this discussion, all intermediate delays are approximations, since they are adjustable to achieve the overall input/output timing precision. CLK triggers a 50-ns delay in B310-F36U on drawing MC2. The delayed output pulse is inverted in B602-E34 for a negative POST CLK pulse. POST CLK triggers a second 50-ns delay in B310-E36L for SYNC CLK, and a third delay (100 ns) in B360-D35L for SM(1) gating at B104-C37U.

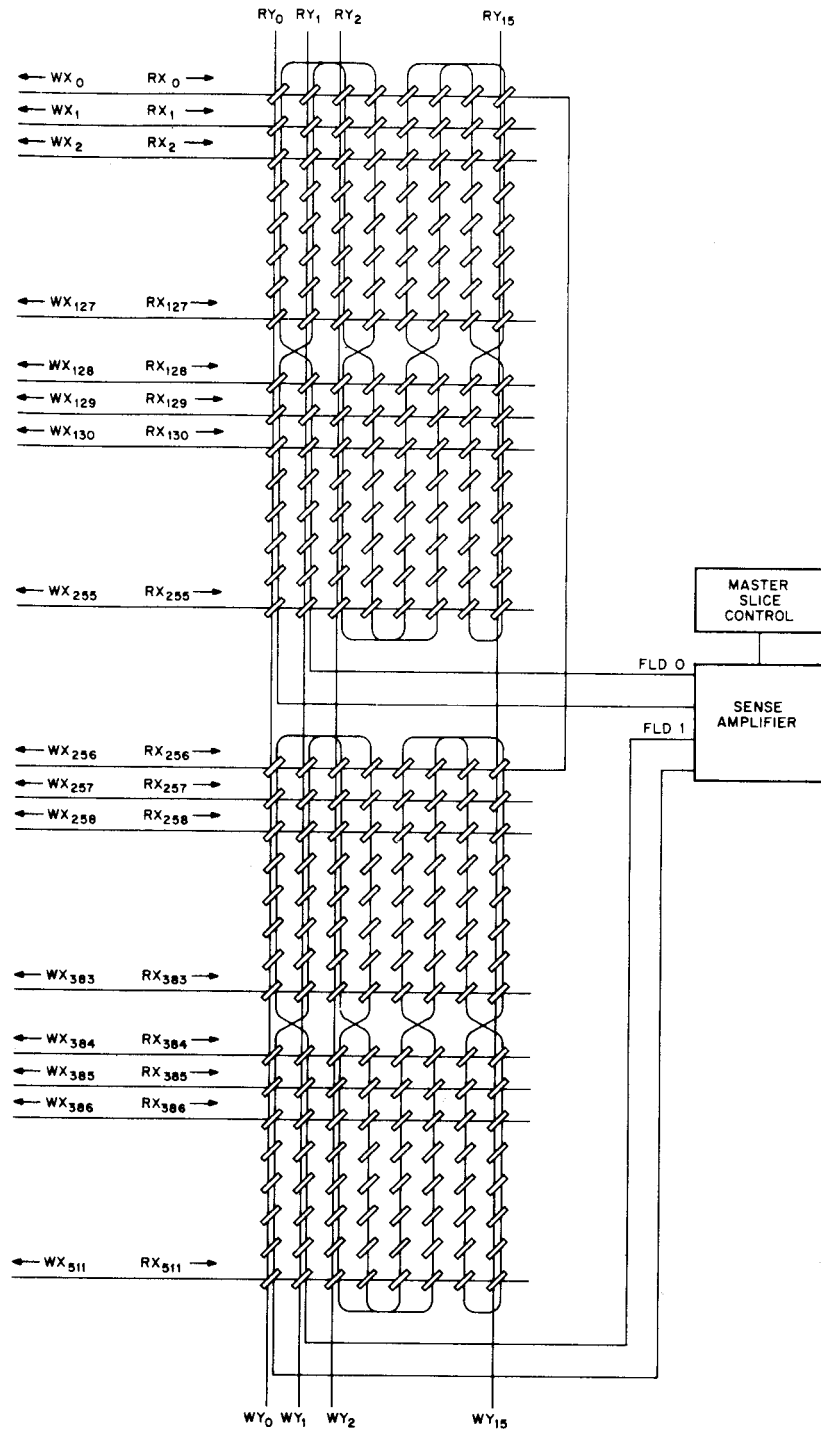


Figure 3-21 Core Wiring for One Bit Position

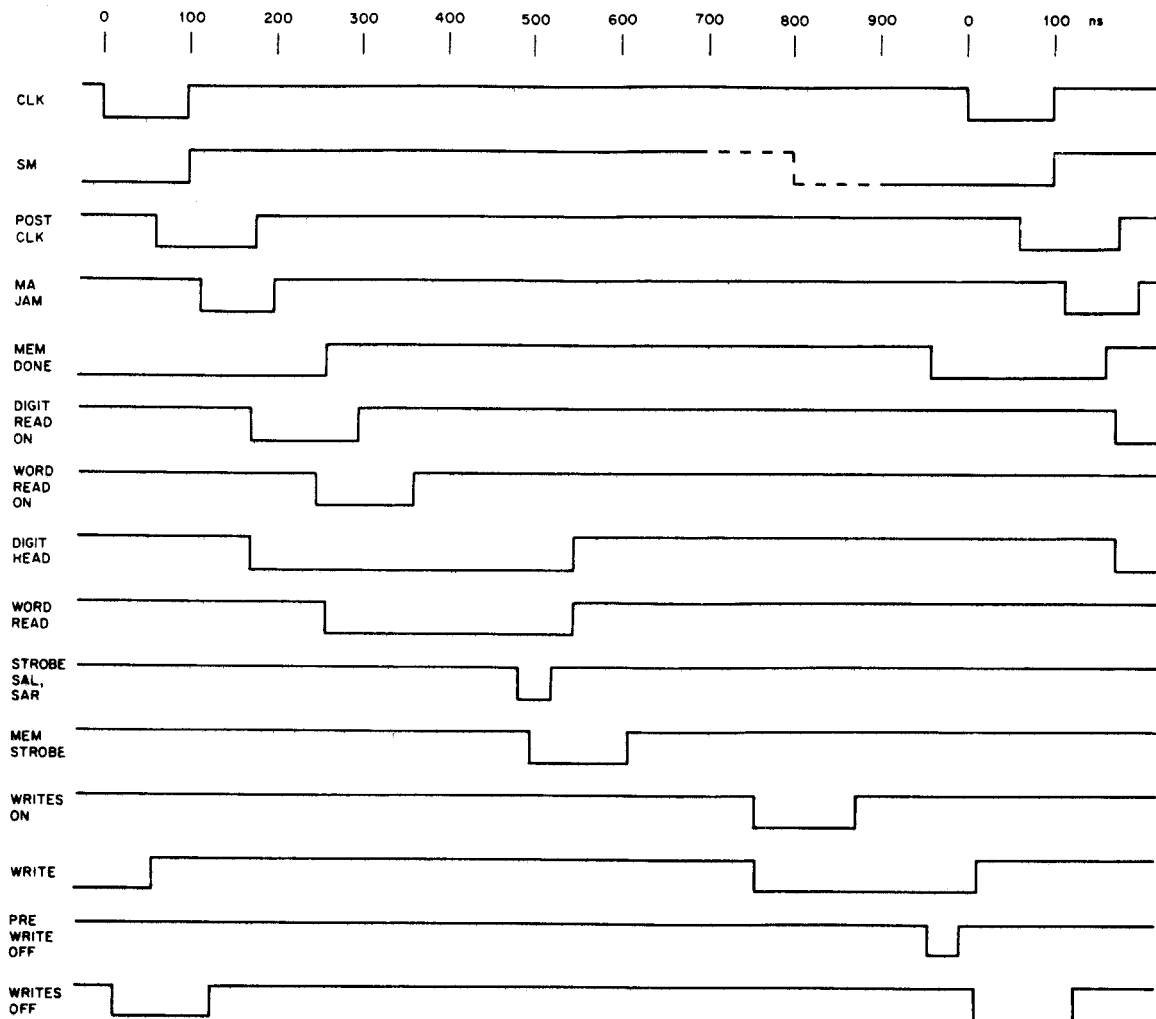


Figure 3-22 Core Memory Read/Write Timing

POST CLK strobcs the MODE flip-flop B213-D28 on drawing MC1(2). Assuming no prior AM RQ (1) from the DMA channel, AM SYNC(1) is negative at the jam input gate, and the flip-flop therefore sets for CP access to core memory. MODE(1) conditions the input mixer gates to the incoming address from the MB, drawing MC3. The address was placed in the MB at MEM DONE time of the previous memory cycle, as described later. The address bits MBS05-17 out of the input mixer are sent to the MA register. SYNC CLK generates MA JAM DIGIT and MA JAM WORD on MC1(1) to strobe the address into the MA. POST CLK delayed 100 ns in B360-D35L places a ground at the emitter of inverter B104-C37V, drawing MC2.

SM(1) from control memory triggers 50-ns delay B310-E36U. The delayed output pulse is NORed for a negative input at inverter B104-C37V. As a protective measure against short-circuiting the address selectors, this input holds negative to turn on the inverter only if the previous cycle's WRITE

operation was turned off at the appointed time. That is, the DIGIT WRITE and WORD WRITE flip-flops must have been reset by WRITES OFF of the previous cycle to hold the B104 input at the negative level. If any of the WRITE flip-flops should become defective before turn-off, it would remain at -3V, pulling the input at B104-C37U to ground via NOR gates in B31. This prevents the inadvertent application of both read and write currents to the address selectors at the same time.

Under safe conditions the inverter turns on and its collector goes to ground, connected in common to the emitter at B104-C37N. B104-C37N conducts if the memory bank selection circuits supply a negative level to the base. For memory bank selection the extended memory address bits EMA03-04 or AM EMA03-04 from the pertinent option provide MAS03-04. The states of MAS03-04 must agree with the preset positions of the bank selection switches SW3-SW4. These switches are located at the rear of each memory bank. They are preset to designate the particular memory bank assignment. For the basic 8K memory bank, SW3-SW4 are both set to the ground or 00 positions (down). In memory bank 01, SW3-SW4 are set to 01, etc.

Addressing memory bank 00, MAS03-04 are also at 00. Therefore, the B104 inverters at E35 and F30 are all at cut-off, supplying the negative level to the base at B104-C37P. Other banks are inhibited (deselected) by the same MAS03-04 bits at 00, applied to their respective bank selection circuits.

Delayed SM(1), delayed POST CLK, and the bank selection circuits therefore interact to produce DIGIT READ ON, 180 ns after CLK. The leading edge of DIGIT READ ON sets the DIGIT READ DRIVE and DIGIT READ SINK flip-flops B213-D16, C16. These flip-flops in the set states condition the address selectors (digit drivers) to select and drive 18 digit current lines, drawings MC4(1) through MC4(9). Approximately 80 ns after DIGIT READ ON, WORD READ ON occurs on drawing MC2 to set the WORD READ flip-flop, drawing MC1(2). WORD READ(1) detects the state of MA05 on drawing MC1(1) to condition the address selectors (word drivers) to select and drive one word-read current line, drawing MC5. The direction of current depends on the MA05 state for memory field selection.

Delays in B360-E30L and B310-EF31EL generate a PRE-STROBE pulse on drawing MC2 320 ns after DIGIT READ ON. PRE-STROBE generates MEM STROBE, STROBE SAR, and STROBE SAL at the left side of the drawing, MEM STROBE restarts the control memory timing chain, drawing KC16, to extract the third process word of the computer cycle. STROBE SAL and STROBE SAR gate bits 00-08 and 09-17 respectively of the current-driven memory word out of the sense amplifiers, drawing MC6. The sense amplifier outputs SA00-17 go to the B bus via the CP/memory interface, and to the DMA channel via the DMA/memory interface.

If the computer is in a fetch cycle, SA00-17 contains an instruction word. The entire word is placed in the MB via gating processes through the B bus, ADR, and O bus, while the op code portion SA00-04 is gated into the IR from the CP/memory interface. The second process word (12) of the fetch cycle provides the gating processes (Section 3.5.1).

If the computer is in a defer cycle, SA00-17 contains an effective address as detected by the op code bit IR04 in the previously fetched instruction. The effective address reaches the MB via the same path, but the op code from the fetched instruction is retained in the IR for execution gating in the coming execute cycle.

If the computer is in an execute cycle, SA00-17 contains an operand addressed by the fetched instruction.

In an IA0 cycle the contents of SA00-17 are blocked from the MB and are therefore lost. The instructions using the IA0 cycle for execution don't care about these contents (Section 3.5.4).

A READS OFF pulse occurs at W612-F32D, drawing MC2, 60 ns after STROBE SAR, clearing the DIGIT READ DRIVE, DIGIT READ SINK, and WORD READ flip-flops simultaneously. In resetting, these flip-flops apply a negative level via NOR gates to the base of an inverter, B104-C37K. A pulse occurs at delay B360-F33L 200 ns after READS OFF to ground the emitter at B104-C37J. The collector goes to ground to generate WRITES ON at W612-F35D. Note that the emitter grounding by the READ(0) condition simulates the protective measure described earlier before READS ON was allowed to turn on.

WRITES ON sets the DIGIT WRITE DRIVE, DIGIT WRITE SINK, and WORD WRITE flip-flops. The WORD WRITE(1) level detects the MA05 bit as for WORD READ to condition the same word current line, since the MA contents have not changed. Current flows in the reverse direction, however. The DIGIT WRITE flip-flops, on the other hand, enable only those digit current lines whose address selectors receive 1s from the MB register bits, available as MBS00-17 from the input mixer. Digit write currents are also in the reverse direction, writing 1s into the cell specified by the MA contents. Cores in the cell whose lines are deselected remain at 0.

If the computer is in a fetch cycle, the MB contains the instruction word retrieved during the read half-cycle. Thus the word is now restored undisturbed to its original location in memory. In a defer cycle the MB contains the effective address retrieved during the read half-cycle, in which case the address may have been incremented by 1 for auto-indexing operations. In an execute cycle the MB contains an operand retrieved during the read half-cycle. The operand may have been modified, by manipulation in the CP, before the write half-cycle began (note the 260 ns pause between MEM STROBE and the beginning of WRITE). In an IA0 cycle the contents of the MB are entirely new to the addressed memory location, replacing the contents retrieved during read.

WRITES ON also triggers a delay at B360-F34L to produce PRE-WRITE OFF at B360-F34N, 200 ns later. This PRE-WRITE OFF pulse sets MEM DONE on drawing MC1(2). (PRE-WRITE OFF also issues an AM GRANT to the DMA channel if an AM RQ came in before SYNC CLK time at the outset.)

MEM DONE occurs just 40 ns before the next CLK pulse and remains until WORD READ ON recurs. MEM DONE goes to the 1 \rightarrow MBI gating on drawing KC19(2). Here it is Nanded with RUN(1), SM(1), and a Δ MB level which is produced if any one of four conditions is present: PCO(1), CAL(1),

ARO(1), or EXT(1). For example, if the current memory cycle occurs during computer execute, the BGN word in the execute process contains PCO(1) which gates the next address from the PC into the MB for the coming fetch cycle. If the current memory cycle occurs during fetch, and the fetched instruction is CAL, then CAL(1) is present to gate address 00020 into the MB for the coming IAO cycle (Section 3.5.6.3). ARO(1) is present during manual key operations from the console, and EXT(1) is present during program breaks for similar address changes.

These are all cases where the current address in the MB (and MA) must be changed, in preparation for the next memory cycle. Δ MB and the Nanded levels result in RQ MBI, producing $1 \rightarrow$ MBI. The $1 \rightarrow$ MBI level sets the MBI sense flip-flop in the control memory. MBI(1) then gates the new address from the PC, AR, etc., into the MB. MA JAM DIGIT and MA JAM WORD occur shortly thereafter to transfer the new address into the MA via the input mixer.

In other instances, at MEM DONE time of most instruction fetches, for example, the instruction word placed in the MB remains undisturbed to retrieve the operand during the coming execute cycle. MA JAM transfers the address portion of the instruction word into the MA for the operand retrieval.

PRE-WRITE OFF triggers a delay in B310-F36L to produce WRITES OFF 60 ns later. The WRITES OFF pulse resets the DIGIT WRITE DRIVE, DIGIT WRITE SINK, and WORD WRITE flip-flops. Although the asserted WRITE levels have overlapped the next CLK pulse, they are turned off before MA JAM occurs, so that the MA contents do not change until the WRITE cycle is completed.

On drawing MC2 a PK CLR pulse (power clear-key initialize clear) from the manual control logic KC10(1) immediately generates READS OFF and WRITES OFF pulses when the computer system is first turned on, shut down, or when manual key operations are initiated (Section 3.7). The READS OFF and WRITES OFF pulses reset the READ or WRITE flip-flops if set, protecting against loss of stored memory information should a spurious memory cycle result from switching transitions.

3.6.4 Address Selectors G219

The double-height Type G219 Address Selector Modules perform memory address selection by decoding the address in the MA and turning on X-Y drive currents in response to read/write levels from the memory control circuits. Each module contains eight logic switches, operating in pairs and forming one coordinate of a 4 x 4 matrix for digit decoding, and a 16 x 16 matrix for word decoding. The switch pair connects either a positive drive current source or a negative sink to one end of the associated drive line, as determined by a read or a write level from the memory control circuits. The other end of the drive line is connected to an identical, but complementary, switch pair in a second module, which forms the second coordinate of the matrix. Thus, for a read operation in a given memory field, drive current flows through the line from a drive switch in the first module to a sink switch in the second. For a write operation, drive current flows from a drive switch in the second module to a sink switch in the first.

3.6.4.1 Digit Drive Selection - The 18 Y-axis digit drivers shown in drawing MC4 each consist of a 4 x 4, two-diode per line matrix driven by a 4 x 4 switch matrix. The address selector modules form the switch matrix. Figure 3-23 is a simplified schematic of the drive selector circuits for bit 0, line Y_0 . Selection of Y_0 in all other digit drivers is identical, and occurs simultaneously. Selection of other Y lines is likewise identical, using other switch pairs in the address selectors. In Control and Timing, Section 3.6.3, the memory read cycle starts with a DIGIT READ level. This level is applied to the bit 0 address selectors at BV and EV. For Y_0 decoding, address bits MA14-17 are all 0. DIGIT READ MA16(0) and MA17(0) enable the input diode AND gate to inverter Q1 in AB07, while DIGIT READ, MA14(0) and MA15(0) enable the input AND gate to inverter Q2 in EF07. AB07-Q1 and EF07-Q2 turn on AB07-Q3 and EF07-Q4, via their respective pulse transformers. EF07-Q4 connects the positive side (+V DIGIT 00 RES) of the floating supply via its collector to W015-J, and AB07-Q3 connects the negative side (-V DIGIT 00) to W015-D. Thus the read current flows from the EF07-Q4 emitter through line Y_0 and its memory cores, to the collector of AB07-Q3. The clamping diodes at the Q4 emitter and Q3 collector clamp the positive voltage to -7V and the negative voltage to -30V. A resistor connected to +V DIGIT 00 RES limits the Y_0 line current to 360 mA (Section 3.6.6). WORD READ current through a selected X-axis line (Section 3.6.4.2) intersects one core on line Y_0 in each digit driver to complete the selection of an 18-core memory address.

When DIGIT WRITE occurs, the negative write sink switch EF07-Q3 and positive drive switch AB07-Q4 become active, but only if bit 0 is a 1. If 0, then MBS00 is 0; therefore the input diode AND gates are inhibited and no current flows through Y_0 . For a 1 in bit 0, the gates enable, and current flows through Y_0 to the write sink switch EF07-Q3. Current flow in this direction causes MBS00(1) to be written back into the memory at the same address. WORD WRITE current through a selected X line intersects the Y_0 lines in each digit driver to complete the memory address selection.

3.6.4.2 Word Drive Selection - The X-axis word drivers, drawing MC5, consist of a 16 x 16, two-diode per line matrix driven by a 16 x 16 switch matrix. Eight address selector modules form the switch matrix. Figure 3-24 is a simplified schematic of the drive selection circuits for line X_0 . Line X_0 is selected by decoding logic and switch pairs in the address selectors located at HJ23 and HJ27. The address bits from the MA register, MA06-13, are 0 for X_0 decoding. Additionally, MA05 determines the direction of read/write current for selection of X_0 cores in either memory field 0 or memory field 1. On drawing MC1(1), the WORD READ and WORD WRITE levels from the memory control circuits are gated with MA05 in inverter modules B104 at F22 and F23. At each location, the four inverters operate as two NOR gates in series.

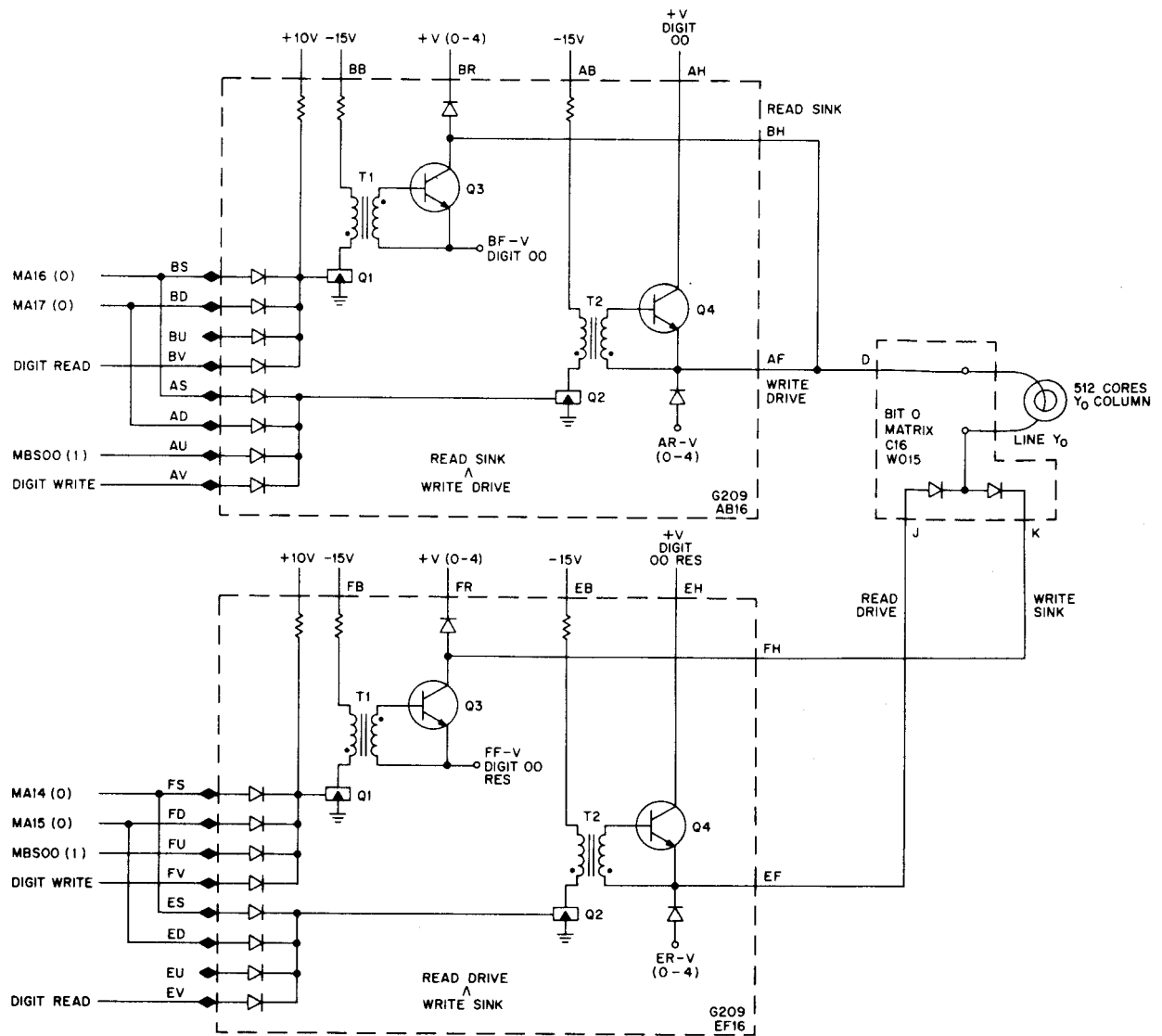


Figure 3-23 Digit Drive Selection, Bit 0, Line Y_0

For a WORD READ operation in memory field 0, MA05 is 0 and the WORD READ flip-flop in the memory control circuits is set. Therefore, at location F22 MA05(1) and WORD WRITE(1) are at ground, cutting off their respective inverters. MA05(0) and WORD READ(1) are negative levels, but their respective inverters are also cut off because their emitters are open. This results in a clamped, WORD READ(1) \wedge MA05(0) output (-3V) to the word driver. In Figure 3-24, WORD READ(1) \wedge MA05(0) enables the input diode gates to the switches HJ23-Q2 and HJ27-Q1, in conjunction with the complements of the MA inputs. HJ23-Q4 and HJ27-Q3 turn on via their pulse transformers. HJ23-Q4 connects the positive supply side (+V WORD) to the diode matrix and HJ27-Q3 connects the negative side (-V WORD RES) to the matrix. Thus current flows through the X_0 line from HJ23-Q4, through the X_0 cores

in both fields, and through diode HJ22-D to the sink at HJ27-JH. Bias resistors in the W017 hold the cathodes and anodes of the diodes in unselected columns at $-V$ and $+V$ respectively. Hold-down resistors in the W016 hold all unselected rows (sinks) at approximately $V/2$, Figure 3-24. This means that the two diodes associated with each line in an unselected row and column will be biased off. A resistor connected to $+V$ WORD RES limits the X_0 line current to 380 mA (Section 3.6.6). Current flow in this direction is the READ current for the X_0 row of cores in field 0. DIGIT READ current through a selected Y-axis line in each digit driver intersects the X_0 cores to complete memory cell selection (Section 3.6.4.1).

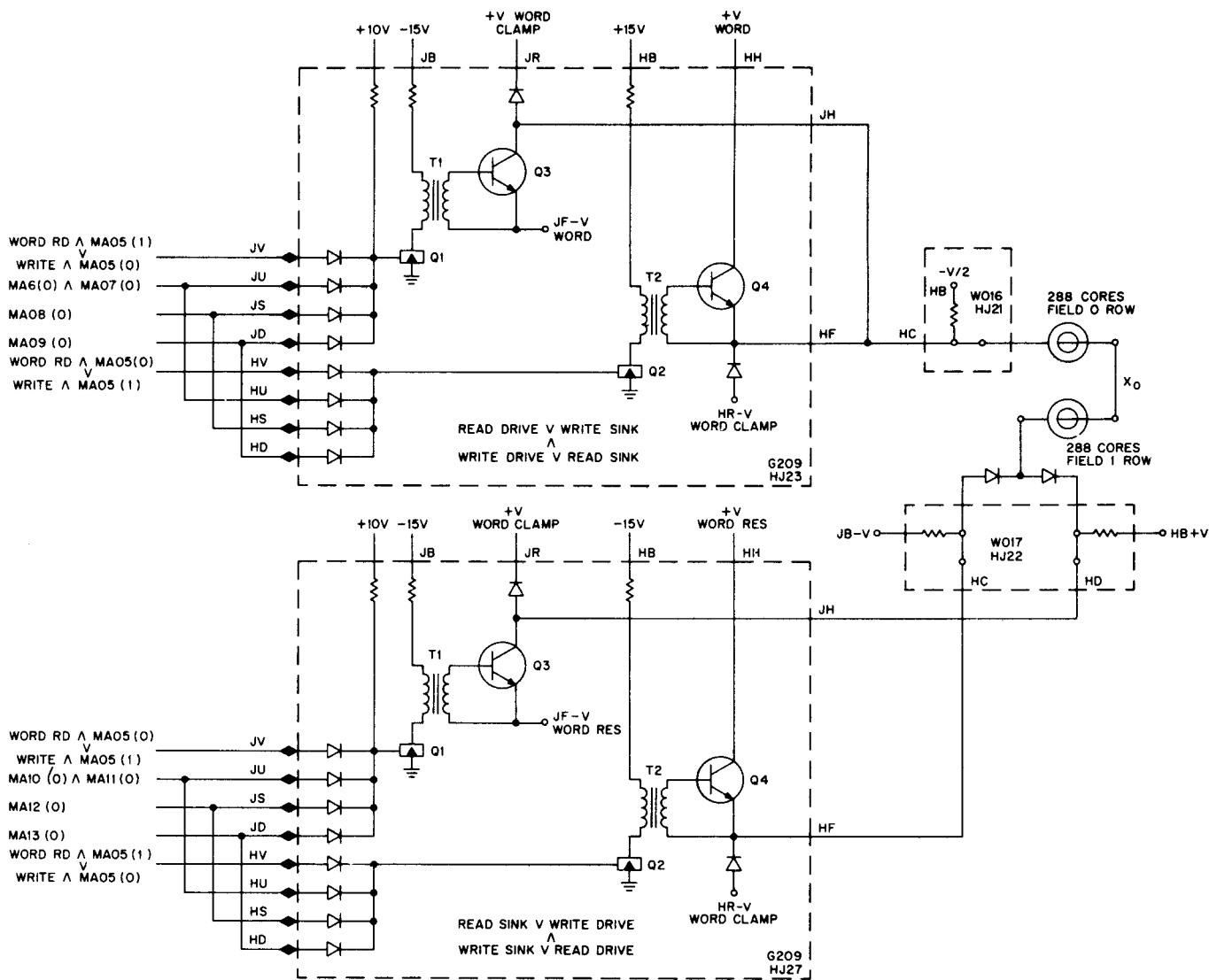


Figure 3-24 Word Drive Selection, Line X_0

Current flow in this direction may also be the WRITE current for the X_0 cores in field 1. Referring back to the MA05 gating at F22, note that this module also produces a negative level for conditions of MA05(1) and WORD WRITE(1). In this case MA05(1) and WORD WRITE(1) turn on their inverters. This applies a ground level to the emitters of the MA05(0) and WORD READ(1) inverters, but their bases are now at ground, keeping them cut off and producing the negative WORD WRITE(1) \wedge MA05(1) level. This level is applied to the same address selector switches in Figure 3-24 as for WORD READ(1) \wedge MA05(0). Current flows through the X_0 line in the same direction. However, current through the selected Y lines is now in the reverse direction (DIGIT WRITE) so that a WRITE operation takes place in memory field 1.

In the case of WORD READ(1) and MA05(1), or WORD WRITE(1) and MA05(0), the inverter module output at F22 becomes grounded and the module at F23 becomes active. At F22, a negative MA05(1) level will turn on its inverter, grounding the emitters of the MA05(0) and WORD READ(1) inverters. The accompanying WORD READ(1) level will then turn on its inverter forcing the module output to ground. The same is true for conditions of MA05(0) and WORD WRITE(1).

The module at F23 produces the negative WORD READ(1) \wedge MA05(1) level or the WORD WRITE(1) \wedge MA05(0) level to activate the other switches in HJ23 and HJ27.

3.6.5 Core Sensing Circuits

The basic PDP-9 memory system contains 18 Type G009 Sense Amplifiers, one Type G008 Master Slice Control, and two Type G010 Sense Amplifier Selectors. The sense amplifier is a two-stage, two-input dc amplifier. The two first stages are paralleled into a single second stage whose output is rectified and then sliced. The slice output is strobed to produce a sense amplifier output. The selected first stage corresponds to the field being read, denoted by MA05. MA05 selects the appropriate first stage through a G010 Sense Amplifier Selector, which is a combination bus driver and level converter (ground to -3V and -3V to -6V). A -3V level out of G010 selects while -6V deselects the appropriate sense amplifier first stage. Two G010 modules are used, one for bits 00-08 and one for bits 09-17. The sense amplifier outputs go to W612 pulse amplifiers which fire when 1s are read out of the associated memory cores. The W612 pulse amplifiers standardize the pulses for input to the MB and the IR. The master slice control supplies closely adjusted reference voltages to the clamping and comparator stages in the sense amplifiers. Drawing MC6 shows the connections of these modules to the memory system.

During a read operation, the signals induced in the sense windings of a core plane by cores changing states are on the order of 40 mV. In planes where a core does not change state, some noise is generated (a few millivolts). The sense winding of each plane is connected to a sense amplifier which samples the voltage induced in the winding and raises the voltage to a level capable of triggering a pulse amplifier W612 if a 1 is read out of core. At the time of the STROBE SAR, SAL pulse from the memory control circuits, the core signal is compared in the sense amplifier with a preset reference level

from the master slice control. An output pulse occurs only if the amplified signal exceeds the reference level at STROBE SAR, SAL time. Signals resulting from a change of core state meet this condition, whereas amplified noise does not. Thus the sense amplifiers generate output pulses only when reading binary 1s in the associated planes.

The sense amplifier contains a two-input, two-stage (second stage common) dc amplifier, a rectifying slicer, and a gated pulse amplifier. Inputs from either memory field 0 or memory field 1 are gated in by the MA05 bit at the G010 modules. The first stage of the dc preamplifier is a difference amplifier using two transistors as a matched pair. The sense winding of the associated core field is connected between the transistor bases. Signals induced in the sense winding are amplified as differential signals at the collectors. The output is further amplified by the second preamplifier stage (also a difference amplifier), then applied to the rectifying slicer. If the signal level at the slicer exceeds the slice level from the master slice control, the output gate is enabled. (The output of this gate is referred to as the slice window.) The STROBE SAR, SAL pulse is applied to a second output gate. This pulse is precisely timed to occur when the ratio of the read 1 signal to the read 0 signal is at maximum. The output gate produces a positive pulse which is inverted and standardized (320 ns) in a pulse amplifier W612 for application to the MB and IR via the CP/memory interface.

Master Slice Control G008 contains three Zener diode voltage reference networks, each supplying an emitter-follower output to the sense amplifiers. The 1ST STAGE CLAMP is fixed; the 2ND STAGE CLAMP must be set to +6V referenced to -15V; and the SLICE LEVEL is adjusted to about -4.2V referenced to +10V.

3.6.6 Current Sources

All power required for operation of the core memory system comes from the +10, -15V supply and the -30V supply (Section 3.1.4). The +10 and -15V outputs are used for transistor logic power and the -30V supply is used for stack drive current.

In the core memory system a G804 control module regulates the -30V supply and prevents operation of the central processor if any supply exceeds the limits defined in the overall system specification. The G805 modules deliver positive drive (-7V) and negative sink (-30V) voltages to the complementary pairs of transistor switches in the address selectors G219. The selected switch pair connects the voltages across the appropriate core drive line, and the voltage differential creates the current through the line. The voltage applied to one switch is connected via a 56-ohm resistor in resistor board G622, limiting the drive current to approximately 360 mA. This end of the drive line is connected either to the positive drive or the negative sink, depending on the intended direction of current flow.

Five resistors on each of eight G622 resistor boards accommodate all digit drive and word drive switch pairs. The interconnections of the G804, G805, and G622 modules are shown on drawing MC7.

3.6.6.1 Control Module G804 - This module accepts the +V and -V outputs from negative regulator modules, G805-HJ07, HJ05, HJ03, HJ01. Zener diode D21 establishes the reference level for differential amplifiers Q6-Q8 and the voltage adjustment rheostat R12. D21 keeps the combined line and load regulation over the full range of voltage adjustment to 0.25%. The differential amplifier output at the collector of Q8 controls the current through transistors Q7 and Q9, whose emitter circuit delivers the control voltage to the series regulator in G805 via terminal HL, HK (AL, AK).

Additionally, 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 differential amplifiers cause the output voltage of the negative regulator modules to decrease. The temperature tracking coefficient is approximately -0.5% per degree C, using the positive-coefficient thermistor shown.

Other sensing circuits in the control module supply an O.K. SIGNAL to the central processor upon sensing that the transistor logic voltages and the negative regulator voltages are correct. The O.K. SIGNAL remains negative as long as the voltages remain within 3V of their designated values. If not, the level goes to ground, resetting the RUN flip-flop and generating power clear pulses in the central processor. This prevents the occurrence of spurious memory cycles and consequent disruption of stored information, particularly during initial power turn-on (Section 3.7.2).

On the module schematic, transistor Q10 goes into conduction if the -30V supply exhibits a 3V differential between the -7V and -30V outputs, as governed by the 3V reference levels set up in diode packages E3/E4. Q10 turns on Q11, which places MEM O.K. at ground. Likewise, transistor Q1 conducts if the -15V and +10V supply voltages are outside the specified limits. This causes the LOGIC O.K. level to go to ground. MEM O.K. and LOGIC O.K. are tied together as PWR O.K. which goes to the clock and run logic, drawing KC10, via the CP/memory interface.

3.6.6.2 Negative Regulator G805-- Series regulator Q1 accepts the control voltage 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 23V, so that the lower end is at -7V and the higher end sees the full -30V supply voltage. The five 20- μ F 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 resistor boards G622 and the address selectors as shown on the interconnecting diagram, drawing MC7.

3.6.6.3 Resistor Boards G622 - Eight resistor boards are connected between the negative regulator outputs and the address selectors as shown on drawing MC7. Each resistor has a parallel 470-pF peaking capacitor which offsets the inductance of the line and thus presents a better current rise time. The common sides of the resistors, tied to either -7V or -30V, are decoupled to ground by a 20 μ F, 50V capacitor.

3.7 MANUAL CONTROL

3.7.1 Primary Power Distribution

The primary power cable connects to plug P1 on the Power Control 841A, Figure 2-5 and drawing CS-841-A-1. The 841A unit is secured to one of the two 709 Power Supplies in the basic PDP-9 cabinet. Indicators I1 (WHT) and I2 (RED) illuminate when primary power is present at the input plug P1. To apply primary power to the supplies and other components of the system, circuit breaker CB1 must be turned on and the LOCAL/REMOTE switch S1 must be in the LOCAL position. The circuit breaker protects the primary power source against computer overloads.

Further, for system turn-on the console POWER switch must be turned ON, and the maintenance panel switch must be in the NORMAL position. In the ON position the console POWER switch completes the neutral line path from CB1 via the LOCAL position of S1 to the return side of relay K1. As shown in drawing IC-9-0-1, the twisted pair from the 841A BRN terminals go first to terminals on the maintenance panel, then to the console POWER switch, drawing CS-9-0-3.

Under these conditions relay K1 energizes, supplying primary power to the RED, WHT output terminals, and to four accessory output connectors P2 through P5. The terminal outputs go to both 709 Power Supplies, and to the fans in the three fan housings. Two of the four accessory output connectors are used to cable primary power to the KSR-33 Teletype unit and the PC03 Paper Tape Punch, both supplied with the basic PDP-9 system.

Free-standing cabinets housing expanded PDP-9 system options usually contain their own 841As and dc power supplies. These units can be interconnected by taking an accessory output from the P2-P5 connectors of one 841A to P6-P7 of the next, and supplying a separate source of primary power to P1. S1 on the additional 841As should remain in REMOTE, and the BRN terminals jumpered. With this configuration, it is possible to enable or disable primary power to individual cabinets by means of the circuit breaker CB1 in each 841A.

One exception is the primary power cabling to the TC09 DECTape Control option. This option receives its power from an auto-transformer tap at one 709 Power Supply in the basic PDP-9 cabinet. This power is available when the basic cabinet's 841A is energized as above.

3.7.2 Power Supplies

The 709 Power Supplies, drawing CS-709-0-1, are dual 50-cycle/60-cycle units designed for ease in conversion to any of the input voltage 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.

The +10, -15V transistor logic voltages are wired from the supplies to terminals on the fan housings as shown on drawing IC-9-0-1. Here they are switched through marginal check switches to the logic modules in each computer wing (Section 3.7.6).

The marginal checking, 25 Vac output from a secondary tap of one supply is adjustable through a Variac between the supply's ORN (25 Vac) and BRN (MC AC) terminals. The Variac is mounted on the marginal check panel (Section 3.7.6). The Variac's output comes back to the 709 supply for rectification, and the resulting plus and minus voltages are wired through the +10 MC and -15 MC positions of the -15 MC/OFF/+10 MC switch on the marginal check panel, to the open output terminals of the second power supply, then to terminals on the fan housings. Here they are switched through marginal check switches to the logic modules in each wing (Section 3.7.6).

A direct -30V output is used for paper-tape punch power from one of the supplies. In addition to this direct output, a relay-controlled -30V source is used by the core memory system. The relay (K1) is energized when a voltage-sensing circuit in core memory's G804 control module senses that the +10, -15V outputs have stabilized. Approximately five seconds of initial warm-up are required before the large capacitors in the 709 supplies charge to their full stabilized values. Thus the relay control protects core memory against erratic core switching at power turn-on. Likewise, at power turn-off, the G804 senses the 709 capacitor discharge and releases the relay to remove the -30V source.

The supply also contains a zero-current, 10 Vac line used to trigger the real-time clock located in the computer's I/O control wing (Section 3.9.4). This line is taken from the 10 Vac tap of the transformer secondary, to the terminals at the maintenance panel, then through the I/O/console interface to the real-time clock.

An auto-transformer effect is also achieved at a primary tap which supplies up to 5A at 120 Vac independent of input voltage. This tap in one 709 supply powers the supply's cooling fan and the TOTAL HOURS meter on the marginal check panel. The tap in the second supply powers the supply's cooling fan and the optional TC09 DECTape Control in another cabinet.

3.7.3. Main Clock

The main clock R409-J22 on drawing KC10(1) is the crystal-controlled, master timing device in the computer system. The clock operates continuously at a 1-mc rate, supplying 100-ns pulses at 1- μ s intervals. Although operating continuously, the application of its resulting CLK POS and CM CLK pulses to the computer system is controlled by the RUN flip-flop. The CLK POS and CM CLK pulses primarily synchronize the core memory and control memory cycles, and the I/O devices where synchronous operation is necessary. Ungated positive CLK pulses are used elsewhere in the system, independent of RUN and where synchronism is unnecessary. Gated CLK POS pulses are also inverted at W612-E02D for negative CLK pulses to the system.

3.7.4 Manual Controls

The following sections discuss the clock and run logic of drawing KC10 in accordance with the console operating functions, drawings CS-9-0-3 and CS-9-0-4. Timing diagrams for the major key functions supplement the text. ("key" refers to the telephone-type toggle switches on the console.) Refer also to the basic key flowdrawing KC6 and the key timing diagram of drawing KC11. The latter is typical of all key functions.

3.7.4.1 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 resets 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 generate IND CLK and PK CLR pulses at pulse amplifiers S602-J25K and W612-H32N, and CM STROBE A, C, D on drawing KC16. IND CLK produces CM STROBE B. The CM STROBES clear all CM sense flip-flops on drawing KC19 because CM CURRENT is absent.

When the 709 Power Supplies reach their stabilized states, the PWR CLR 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(1). 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 8 μ s to 60 ms. The five-position REPEAT SPEED 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 $B(1) \wedge A(0)$ condition generates IND EN at R111-J26PN. IND EN goes to the wiper arm of the console REGISTER DISPLAY switch, drawing CS-9-0-3, 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). On the next REPT CLK pulse C sets. 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 $A(0) \wedge B(1)$; IND EN applies PCD, ACD, ARD, or MQD to the appropriate-jam input gate if the respective register is selected at the REGISTER display switch, drawing CS-9-0-3.

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 to the REGISTER indicators (Section 3.7.5.1).

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

3.7.4.2 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. Figure 3-25 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 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- μ s negative KEY DLY.

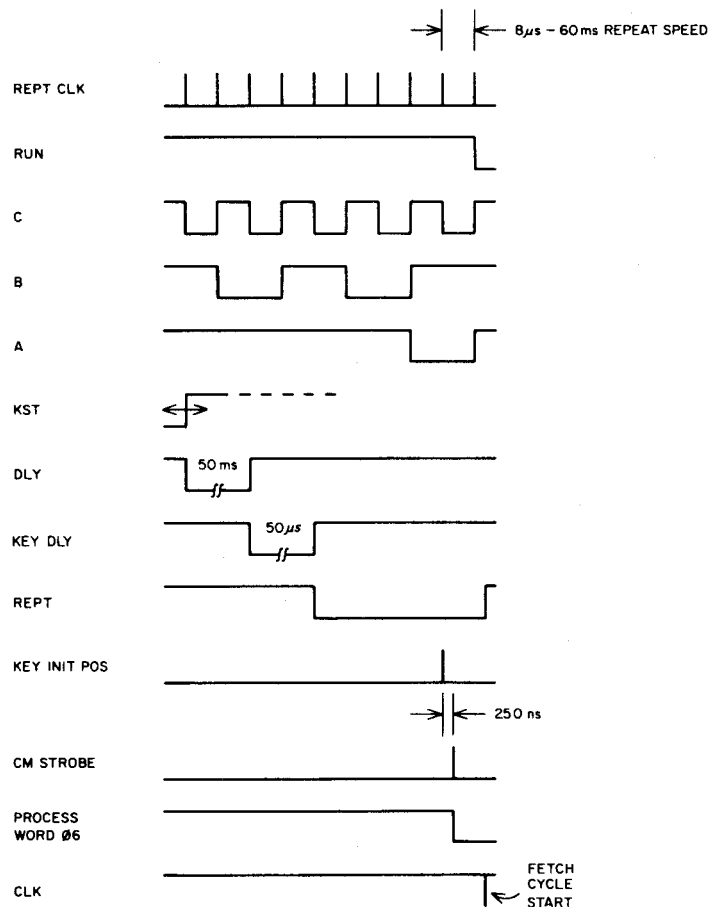


Figure 3-25 Program START Timing

The 50- μ s delay period allows sufficient time for settling of switch contact bounce and the 50- μ s 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- μ s 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 upon recovery sets the REPT flip-flop conditioned by \overline{KRI} (READ-IN key inactive). REPT(1) conditions the DCDset gate of flip-flop A, which sets on the next reset of C and 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) \wedge \overline{KCT} . The \overline{KCT} level is derived from the inactive CONTINUE key. KEY INIT POS starts the CM timing chain, drawing KC16, and generates PK CLR, drawing KC10(1). The timing chain issues CM CURRENT and CM STROBE (Section 3.4).

CM CURRENT enables the CM address selectors, drawing KC17, to decode the address in the CMA register. Recalling that KEY INIT POS has generated PK CLR, these two pulses act together to clear the CMA 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 KCI0(1). On drawing KC17, the cleared CMA levels CMA00(0), CMA01(0), and CMA02(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 IO 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 and A. As C resets, it sets the RUN flip-flop in conjunction with the stored A(1) condition at the input DCD gate. RUN(1) disables the REPT CLK, resets and holds A and B in the 0 state, and enables CLK POS pulses from the mainclock. CLK POS pulses are further inverted at pulse amplifiers B602-H33N and W612-E02 for negative CM CLK and CLK pulses respectively. Since the main clock becomes operational immediately at power turn-on, the first CLK POS pulse can appear at any time during the 1- μ s interval following RUN(1). 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. At CLK time, MA JAM in core memory places the address in the MA, the CM timing starts to extract process word 21, and the machine thus starts execution of the program.

RUN(1) also sets the SEN flip-flop on drawing KC10(2). SEN(1) monitors the states of PCO and ARO at the PCOS and AROS flip-flops throughout the running program. These flip-flops thus reflect the PCO and ARO states continuously (Section 3.7.4.4).

3.7.4.3 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, Section 3.7.4.12.

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(1). The other three inputs to the gate determine when the computer can be stopped. In order to stop, the console must be unlocked (LOCK, Section 3.7.6), 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(2). 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 produced from IND CLK on the next REPT CLK pulse.

3.7.4.4 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, and 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(2), if PCOS is set, $A(1) \wedge B(0) \wedge KCT(B)$ 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 (see 3.7.4.3 and 3.7.4.9).

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, inhibits the REPT CLK, and conditions the DCD gate of pulse amplifier S603-J23F, producing CLK POS pulses. 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 (Section 3.5.1).

3.7.4.5 DEPOSIT - The upper, spring-loaded DEPOSIT position of the DEPOSIT/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/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 SAL, SAR 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 function normally starts from a computer stop condition. Figure 3-26 shows the timing for setting up a DEPOSIT, DEPOSIT NEXT, EXAMINE, or EXAMINE NEXT operation; all of these require the same set-up conditions. KDP on drawing KC10(1) sets the REPT flip-flop after the 50- μ s 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) \wedge IND CLK condition. KEY INIT POS generates PK CLR, drawing KC10(1), 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 key produces level KIOA5 on drawing KC10(1). CMA00(0), CMA01(0), and CMA02(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 only, the address does not get past the AR (see DEPOSIT NEXT).

STROBE SAR, SAL 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.

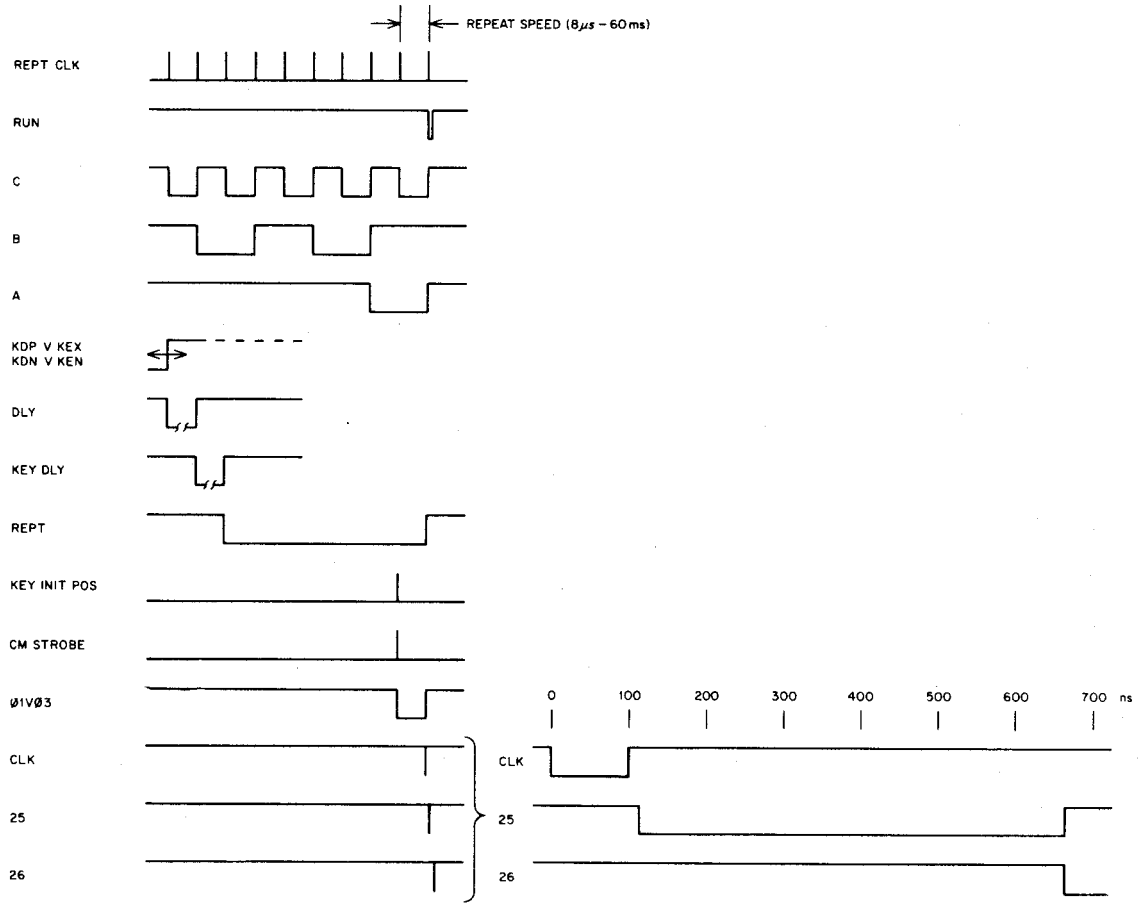


Figure 3-26 Initial Set-Up DEPOSIT or EXAMINE Timing

KEY(1) is NANDed with a KDPDN 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 Λ KDPDN 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 Λ KDPDN level on drawing KC19(3), also generates KDPDN V RI. The KDPDN V 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 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. From CONTINUE, process word 26 accomplishes the operations performed by the BGN word at the end of an instruction (Section 3.7.4.4). That is, it obtains the next address from the PC for the start of the next fetch cycle. For START, DEPOSIT NEXT, and EXAMINE, the address from the PC does not reach the MB and the CM address (21, fetch entry) changes appropriately.

3.7.4.6 DEPOSIT NEXT - Having deposited a single word in core memory with the DEPOSIT position of the DEPOSIT/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 function of Section 3.7.4.5 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(1) and in Figure 3-26, 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 CII7, 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. The entire process repeats for each depression to DEPOSIT NEXT.

3.7.4.7 EXAMINE - The upper spring-loaded EXAMINE position of the EXAMINE/EXAMINE NEXT key transfers a single word from core memory to the MB for automatic 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 function is normally started from a computer stop condition, e.g., following a DEPOSIT/DEPOSIT NEXT operation. On drawing KC10(1) and in Figure 3-26, 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, below.

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. The computer stops until made to proceed from CONTINUE, 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.

3.7.4.8 EXAMINE NEXT - Having examined a single word in core memory with the EXAMINE position of the EXAMINE/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 function of Section 3.7.4.7 are unnecessary and are in fact inhibited.

On drawing KC10(1) and in Figure 3-26, 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. IN CLR and CLR derived from the KEY(1) bit of process word 25 sets 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.

3.7.4.9 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 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. 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 3-27). 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/IO 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 strobcs 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 in the manner described for a programmed IOT0104 instruction, Section 3.9.2. RDR COUNT pulses timed with the appearance of the tape channel 8 holes by a RDR INDEX clock circuit (see Section 3.9.2) step the RDR 1 and RDR 2 flip-flops; the first RDR COUNT sets flip-flop RDR 1. RDR 1(1) strobcs the first line of tape into the reader buffer, RB00-05. The second RDR

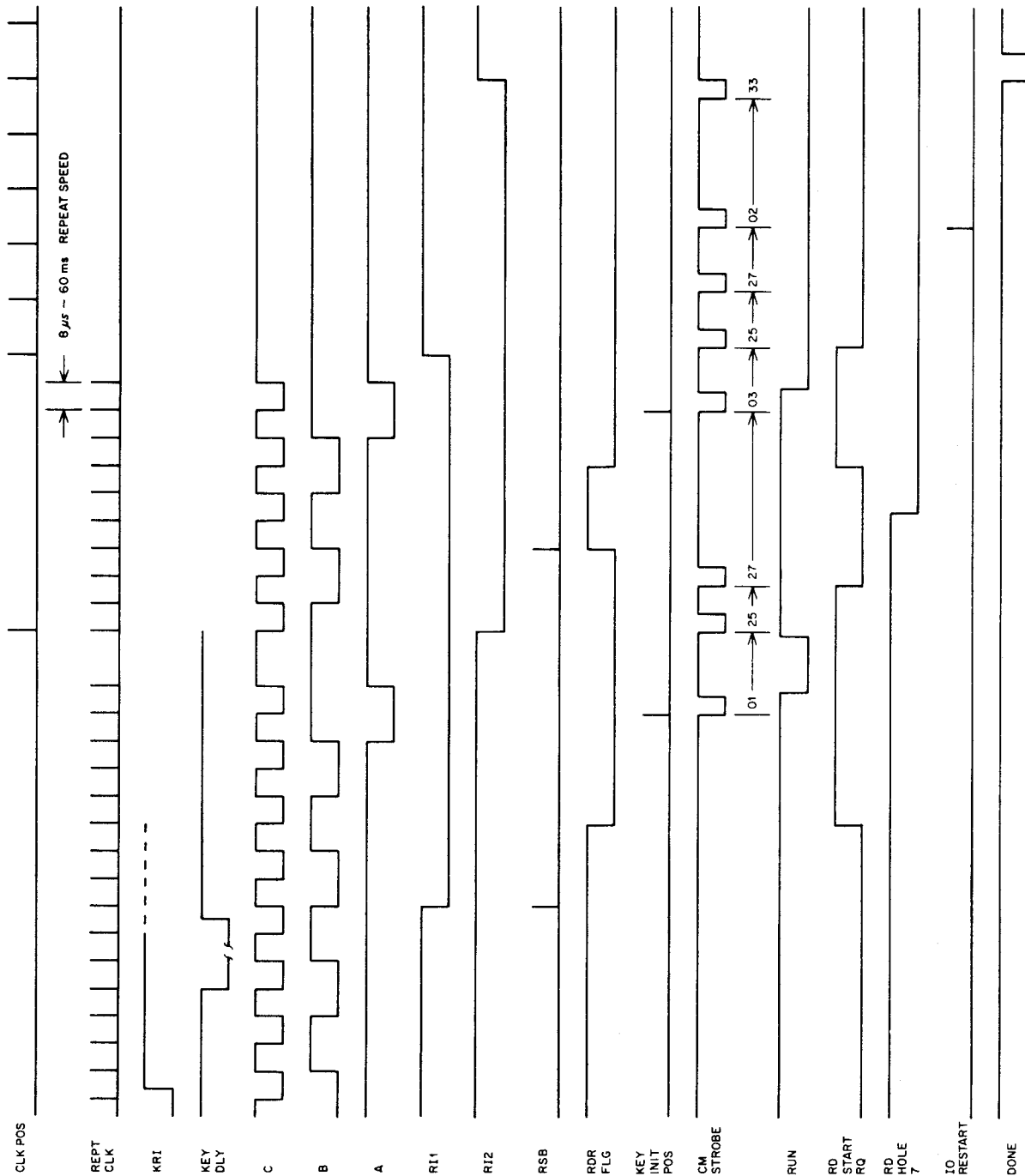


Figure 3-27 READ IN Mode Timing

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) from the negation side of the flip-flop is buffered at S107-E06N, drawing KD8, 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 C, therefore, 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 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 operation, Section 3.7.4.5. 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) \wedge 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. 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 RI2(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 μ s 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 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 3-27 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-A17, 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 as usual 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 μ s 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 μ s 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 MEM DONE on drawing KC19(2). RQ MBI generates $1 \rightarrow$ 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.

3.7.4.10 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(1). Unlike the gated negative CLK and CLK POS pulses, the positive CLK pulses are always present independently 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 CLK FLG, CLK RQ, and 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, CLK EN, 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(1) 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, ultimately generating KEY INIT POS and setting RUN (see START, 3.7.4.2). The ensuing CLK POS pulse resets RUN shortly thereafter, in conjunction with KIOA5 and the negation states of RD HOLE 7, 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.

3.7.4.11 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(1), 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, Section 3.7.4.2, CONTINUE, Section 3.7.4.4, READ IN, Section 3.7.4.9, and with built-in maintenance provisions, Section 3.7.7 REPEAT SPEED is a five-position rotary switch, drawing CS-9-0-3, which selects various capacitors to tune the REPT CLK frequency for speeds ranging from 8 μ s to 60 ms. 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 (8 μ s). Otherwise entire lines of characters may be skipped during the interval between RUN reset and RUN set.

3.7.4.12 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 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. The CONTINUE key can then be used to step the instructions one cycle at a time, with each CONTINUE depression.

The CONTINUE 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 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 as determined by the REPEAT SPEED switch. This eliminates the necessity for depressing the CONTINUE key for each advance.

3.7.4.13 CLK Switch - The latched down position of the CLK switch allows the program to enable the real-time clock. The down position provides a ground SW CLK level to the I/O control logic, drawing KD3(2). Here SW CLK is NORed at R111-J04U, then NANDed with CLK EN(1) at R111-J04H. CLK EN(1) occurs only when a clock-select IOT0004 sets the CLK EN flip-flop S202-J08. The ground level out of R111-J04H conditions the DCD set gate to the CLK RQ flip-flop S202-J07. Shmitt trigger W501-J05, comprising the real-time clock, strobes the gate to set the flip-flop. Once the CLK RQ flip-flop is set, the real-time clock function is effective through completion. The function cannot be disrupted if the CLK switch is turned off accidentally. See Section 3.9.4.

3.7.4.14 Other Controls - Other console switches control the operation of optional equipment. These controls are discussed in individual option manuals.

3.7.5 Display Indicators

3.7.5.1 REGISTER Indicator - The 12-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.

EAE - step counter contents (not wired)

RDR - reader buffer contents

TTI - keyboard buffer contents

STATUS - flags of I/O devices assigned to status check facility

API - on/off, interrupt request, and priority level status of eight priority levels

I/O ADDR - 15-bit address word in DCH or API operations

I/O BUS - 18-bit data word on I/O bus from/to any device

DPY - 9-bits each of X, Y buffers of optional Type 34H Display

The REGISTER DISPLAY switch and the REGISTER indicator are enabled by IND EN only in the computer stop condition, as detailed in Section 3.7.4.1. The IND EN level from drawing KC10(I) goes to the REGISTER DISPLAY switch wiper, drawing CS-9-0-3, 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 the 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 3-28, 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-9-0-4. 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, D18 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.

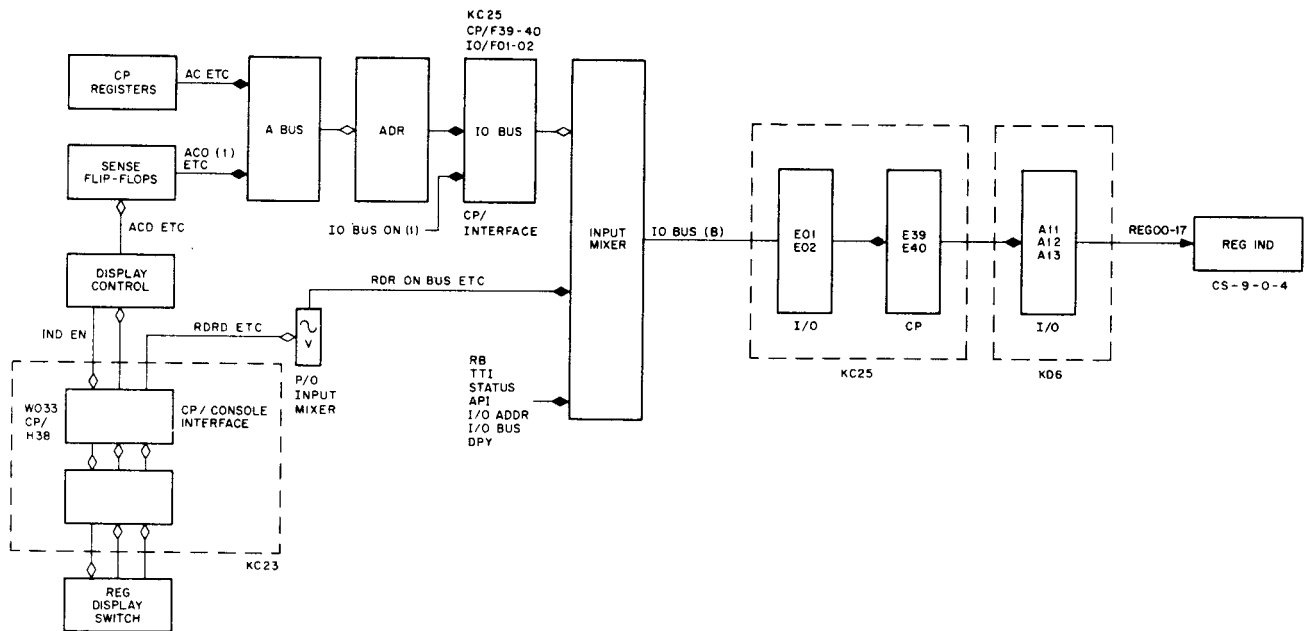


Figure 3-28 REGISTER DISPLAY Signal Paths

3.7.5.2 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 and displayed in the computer stop condition. The LINK flip-flop is shown on drawing KC15.

3.7.5.3 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.

3.7.5.4 INSTRUCTION Indicator - The five INSTRUCTION indicator lamps continually display the contents of the IR. The first four lamps indicate the op code IR00-03 of the instruction being executed. The fifth lamp illuminates when the 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.

3.7.5.5 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 extinguishing the lamp.

3.7.5.6 CLK Indicator - The CLK indicator illuminates when an IOT0044 instruction (CLON) enables the real-time clock. The CLK switch must be turned on (down) before the IOT instruction can enable the clock. On drawing KD3(2), the decoded instruction sets the CLK EN flip-flop. CLK EN(1) goes through the CP/IO interface and CP/console interface to its indicator driver transistor in the console, then to its indicator lamp. The indicator remains illuminated until an IOT0004 instruction (CLOF) resets CLK EN. A subsequent CLON instruction can enable the clock if the CLK switch is still on.

3.7.5.7 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.

3.7.5.8 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.

3.7.5.9 REPT Indicator - The REPT indicator illuminates when the REPT flip-flop is set. From drawing KC10(1) 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.

3.7.5.10 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.

3.7.5.11 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 IO/console interface to the DATA indicator driver transistor in the console, then to the indicator.

3.7.5.12 Other Indicators - Other console indicators pertain to optional equipment. These are discussed in individual option manuals.

3.7.6 Marginal Check Panel

The marginal check panel (Figure 2-2) contains a Variac control, a voltage selector and a voltmeter for the application of either variable marginal supply voltages or the standard +10, -15V supply outputs from the 709 Power Supply to the PDP-9 system. A TOTAL HOURS meter on the panel indicates the cumulative elapsed time of system POWER ON operations.

Drawing CS-9-0-5 includes the marginal check panel wiring. In the upper right corner, the 25VAC IN from the 709 power supply is adjustable at the Variac and is sent back to the supply for rectification (Section 3.7.2). The rectified outputs, adjustable by the Variac from 0 to ± 20 vdc, are applied to the +MC IN, -MC IN terminals at switch S1 in the marginal check panel. Concurrently, the other switch input terminals receive the fixed, standard +10, -15V supply outputs.

Switch S1 is a six-deck, three-position rotary wafer switch. In the OFF position the fixed supply voltages go through the wiper arms to the open output terminals of the second supply, then to the fan housings; the voltmeter is inactive. The marginal check switches on the fan housings thus receive the fixed supply voltages for application to the modules.

In the -15 MC position, the +MC IN terminal is grounded through S1A. The -MC IN goes through S1C to the marginal check switches on the fan housings via the second supply's open output terminals, and to one side of the voltmeter through S1D. The fixed +10V input goes through S1B to the second supply's open output terminals, then to marginal check switches. The fixed -15V input is removed from the marginal check switches, but is applied to the other side of the voltmeter. Therefore the -15 MC selection places the minus marginal voltage and the fixed plus voltage at the disposal of the marginal check switches. The voltmeter reads the difference between the fixed -15V supply output and the minus marginal voltage adjusted by the Variac. For example, if the marginal voltage is adjusted to -18V by clockwise rotation of the Variac, the meter reads +3V, to the right of center zero. If the Variac is adjusted counterclockwise for an output of -12V, the meter reads -3V, to the left of center zero.

In the +10 MC position, S1 reverses the applied voltages, placing plus marginal voltage and fixed -15V at the marginal check switches. The meter reads the difference between the adjusted plus marginal voltage and the fixed +10V supply output. If the Variac is adjusted clockwise for a marginal

output of +13V, the meter reads +3V to the right of center zero. A +7V adjustment is indicated by -3V to the left of center zero. See Chapter 4 for marginal check switch operations.

Just below the Variac circuitry on the drawing, the TOTAL HOURS meter receives primary power from the auto-transformer of one 709 Power Supply. The primary power becomes available only when the console POWER switch is turned ON, Section 3.7.1. Thus the TOTAL HOURS meter clocks the lapsed time of actual power-on operations, providing a numerical read-out to the nearest tenth of an hour from 00000.0 to 99999.9 hours.

3.7.7 Maintenance Panel

The maintenance panel (Figure 2-2) contains a six-deck, five-position rotary wafer switch that locks and unlocks all console controls, exercises the computer registers with a built-in test program for maintenance purposes, and simulates the console DEPOSIT/EXAMINE functions.

Drawing CS-9-0-5 includes the maintenance panel wiring. An etched circuit board mounted inside the panel connects the control lines to/from the switch S1. A mating connector W033 takes the control lines from the etched board to the IO/console interface connector W033-A06 in the I/O control section, drawing KD6. At A06 the LOCK line controls the use of both the real-time clock and the Power Failure Detection Option KP09A, installed in the I/O wing. All lines are also jumpered from A06 to F03, from which they are cabled to the CP/IO interface connector F38 for use in the CP, drawing KC25.

Other terminals on the etched board accept and distribute control lines from/to the Power Control 841A, the 709 Power Supply, the console, and the marginal check panel as shown on drawing IC-9-0-1.

The 10 Vac real-time clock trigger from the 709 Power Supply to the etched board goes unswitched from the board to the I/O control section via the mating connector. All other lines are switched in accordance with the functions explained below.

3.7.7.1 NORMAL - The NORMAL position of S1 allows all console controls to function. On drawing CS-9-0-5 one (BRN) terminal in the lower right corner of the etched board connects the primary power neutral line from 841-CB1 to the console POWER switch. In the ON position the POWER switch closes the line and routes it back to the second BRN terminal on the board. From here it goes directly to 841-K1, energizing the relay to activate the 709 Power Supplies.

The upper (BLU) terminal on the etched board receives -15V from the 709 supply via the marginal check panel for console keys. This voltage goes directly from the terminal to one side of all console keys. In the NORMAL position of S1, section S1D applies a ground level to the other side of the keys from pin N of the etched board's connector. Pin N is grounded at the IO/console interface

end, A06. The ground connection is taken from the BLK terminal on the board to the keys. The console keys are thus enabled, allowing them to function.

S1C connects -15V from the BLU terminal to the MK (manual key enable-disable) input line via pin K of the etched boards' connector. MK at -15V goes to the I/O control section where it disables the auto restart feature of the Power Failure Detection Option KP09A, then comes back through the CP/I/O interface connector F38 to one side of all console switches, allowing the switches to function.

In the CP, $\overline{\text{LOCK}}$ allows the RUN flip-flop to reset, drawing KC10(1).

3.7.7.2 LOCK - The LOCK position of S1 electrically locks all console controls, rendering them ineffective. Manipulation of any console control cannot disrupt a program in progress.

In the LOCK position S1E bypasses the console POWER switch to close the neutral line to 841-K1. As long as S1 is LOCKed the system is energized and the POWER switch is ineffective.

S1D removes the ground assertion levels from the console keys and substitutes -15V negation levels, disabling the keys.

S1C grounds the MK output to the mating I/O connector at pin K. The ground level enables the auto restart feature of the power failure detection option in the I/O section and disables the console switches.

S1A grounds the LOCK level at pin M of the I/O connector. LOCK at ground in the I/O control section performs the function of the console CLK switch, allowing the program to enable the real-time clock, drawing KD3(2). In the CP, $\overline{\text{LOCK}}$ at ground prevents the reset of the RUN flip-flop, drawing KC10(1).

3.7.7.3 MAINT - The MAINT position of S1 circulates a self-incrementing count through all active CP registers to verify proper register operation and transfer path operation. To initiate the test program the I/O RESET key must be depressed to initialize the count at 0, the console REPT switch must be turned on (up), S1 turned to MAINT, and the console START key latched (up), in that order. The program then increments and circulates at a rate determined by the REPEAT SPEED switch.

The incrementing count can be observed at the console's MEMORY BUFFER indicator and/or the REGISTER indicator. An observed disparity between MEMORY BUFFER and REGISTER indicators can be isolated to a specific register or transfer path by turning off the REPT switch, then selecting and observing each register in turn using the REGISTER DISPLAY switch.

In the MAINT position, S1A connects a ground assertion level (KMT) from the etched board connector pin N to pin D. This KMT level (key maintenance) goes to the I/O interface and back to the CP where it produces KIOA3, KIOA4, and KIOA5 on drawing KC10(1).

SIC connects a -15V MK level to the console switches, and SID connects a ground KEYS level to the console keys, as for NORMAL operations. Thus the console controls are enabled for the MAINT function.

On drawing KC10(1) the KST level from the latched START key initiates the START function, Section 3.7.4.2. In this case, however, KEY INIT POS causes the CM STROBE to extract the KMT process word from CM location 07, because of the KIOA3, 4, 5 levels addressing the CM address gates. Process word 07 contains ARO, +1, MBI, CONT, and CMA22.

ARO(1) gates the contents of the AR (initially zero) onto the A bus, the A bus contents go directly through the ADR, and NOSH places them on the O bus. Process +1(1) produces CI17 on drawing KC14. CI17 initiates a carry at ADR17 to increment the contents as they pass through the ADR. MBI(1) gates the incremented O bus contents into the MB. CONT(1) allows the CM STROBE to retrigger the CM timing chain, drawing KC16.

The next REPT CLK pulse sets RUN as for program START. RUN(1) holds flip-flops A and B in the reset states, disables the REPT CLK, and allows the application of one CLK, CLK POS, and CM CLK pulse to the system. In the absence of an SM(1) bit in the 07 process word, the CLK pulse cannot start a core memory cycle, nor can CM CLK retrigger the CM timing. CLK POS resets RUN, in order to inhibit more CLK pulses, re-enable the REPT CLK, and initiate stepping of flip-flops A, B, C. The REPT flip-flop cannot reset on CLK POS because the REPT switch is on (up).

In the meantime, the CM STROBE triggered by CONT(1) extracts process word 22. Process word 22 contains MBO, ACI, ARI, PCI, MQI, ADSO, and CMA07. MBO(1) gates the MB contents onto the B bus, the B bus contents go directly through the ADR, and NOSH places them on the O bus. ACI(1), ARI(1), PCI(1), and MQI(1) gate the O bus contents into the respective registers.

ADSO(1) is used for checking out the transfer path from the console ADDRESS switches to the input mixer, I/O bus (B), and O bus. This process inclusive-ORs the 15-bit contents of the ADDRESS switches with the contents of the ADR at the O bus. ADSO(1) produces ADSO(G) to the ADDR SW03-17 bits into the input mixer, drawing KD7, where they are buffered at I/O bus (B). ADSO(1) also produces LIO on drawing KC13. LIO gates the I/O bus (B) contents, and NOSH the ADR contents, onto the O bus. If corresponding bits are both 0s, the result is 0. If corresponding bits are both 1s or are different, the result is 1.

The ADDRESS switches of course may be set to all 0s, in which case the inclusive-OR function effectively results in a direct transfer of the ADR bits onto the O bus.

This completes the functions of process word 22. Now the system waits for the re-enabled REPT CLK pulses to step flip-flops A, B, C, through their cycle. Another KEY INIT POS ensues to trigger the CM chain, and RUN later sets to gate another CLK, CLK POS, and CM CLK pulse to the

system. The CM STROBE produced by KEY INIT POS extracts process word 07 to repeat the incrementing and gating functions. The test program thus continues as long as the maintenance panel switch S1 is in MAINT, the START key is latched, and the REPT switch is on.

3.7.7.4 EXAMINE - The EXAMINE position of S1 simulates the console EXAMINE function. The operator first loads the address of the core memory location from which the data word is to be read, then turns S1 to EXAMINE. The data word is read into the MB and observed at the MEMORY BUFFER indicator as explained in Section 3.7.4.7.

With the REPT switch turned on (up) and the EXAMINE position of S1 selected, the EXAMINE function repeats. In this case, the same word is read into the MB and the MEMORY BUFFER indicator with each repeat of the function unless the ADDRESS switches are manipulated.

In the EXAMINE position S1D connects a -15V KEYS level to the etched board's BLK terminal, disabling the console keys. S1C connects a -15V MK level to the console switches via the etched board connector's pin K, enabling the switches. S1A connects a ground KXDM level (key examine/deposit maintenance) to the clock and run logic, drawing KC10(1). KXDM performs the same functions as KEX to extract process word 01 from control memory.

3.7.7.5 DEPOSIT - The DEPOSIT position of S1 simulates the console DEPOSIT function. The operator first loads the core memory address into the ADDRESS switches, the data word into the DATA switches, then turns S1 to DEPOSIT. The data word is then written into the core memory location specified by the ADDRESS switches, as explained in Section 3.7.4.5.

With the REPT switch on (up) and the DEPOSIT position of S1 selected, the DEPOSIT function repeats. In this case the same word is deposited in the same core memory location with each repeat of the function unless the ADDRESS and/or the DATA switches are manipulated.

In the DEPOSIT position, S1D connects a -15V KEYS level to the etched board's BLK terminal, disabling the console keys. S1C connects a -15V MK level to the console switches via etched board connector pin K, enabling the switches. S1B connects a ground KDPM level (key deposit maintenance) to the clock and run logic, drawing KC10(1). S1A connects a ground KXDM level to the clock and run logic as for the EXAMINE position of S1. KXDM and KDPM perform the same functions as KDP on drawing KC10(1) to extract process word 01.

3.8 I/O CONTROL

The I/O control section offers flexibility for increasing the capability and efficiency of the PDP-9 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 User Handbook, Document F-95, for typical installations of expanded PDP-9 systems.

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 25,000 wps) are normally connected to the I/O bus for program-controlled transfer operations. High-speed devices (25,000-333,000 wps) are normally connected to the I/O bus for data channel (DCH) transfer operations. Extremely-high-speed devices (up to 1,000,000 wps) are normally connected to the memory bus for direct memory access (DMA) channel operations. The DMA channel is considered an I/O channel since the DMA-interfaced devices are also interfaced to the I/O bus for programmed initialization.

High-speed and critical-use devices may also be connected to the I/O bus in an optional automatic priority interrupt (API) system. The devices connected into the DCH, DMA, and optional API channels are multiplexed for servicing from the same bus connections.

The optional API system and a DMA multiplexer option are described in separate manuals.

3.8.1 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, Sections 3.2.2.2. and 3.5.8. 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 CP's AC or AR register.

All program-controlled devices are chain-connected in parallel at the I/O bus. The number of devices is limited only by the modulo-64, six-bit 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-select codes are allocated to the basic PDP-9's paper tape reader, paper tape punch, teletype, program interrupt, I/O status check, and real-time clock operations. Data transfers to and from the standard devices take place via the input mixer rather than the I/O bus. See Section 3.9.

3.8.1.1 I/O Bus Connections - The I/O bus for program-controlled data transfers terminates at module connector receptacles H800 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 H800 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 H800 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 feet 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-foot limitation. For example, devices using the program interrupt facility could be grouped in the secondary chain without affecting other devices in the primary.

3.8.1.2 Block Diagram Discussion - Figure 3-29 shows the interface for program-controlled data transfers and Figure 3-30 shows the typical logic contained in a device. 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 issues a skip request to the central processor via the I/O control section. 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 instruction gates the data from the AR onto the I/O bus via the A bus and ADR, and stokes 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 stokes 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 W103 Device Selector 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 μ s intervals following the 1 μ s 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 μ s fetch cycle and an extended execute period of 3 μ s whether or not the 3 μ s period is fully needed to execute the instruction. This is shown in Figure 3-18 and explained in Section 3.5.8. 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- μ s execute period.

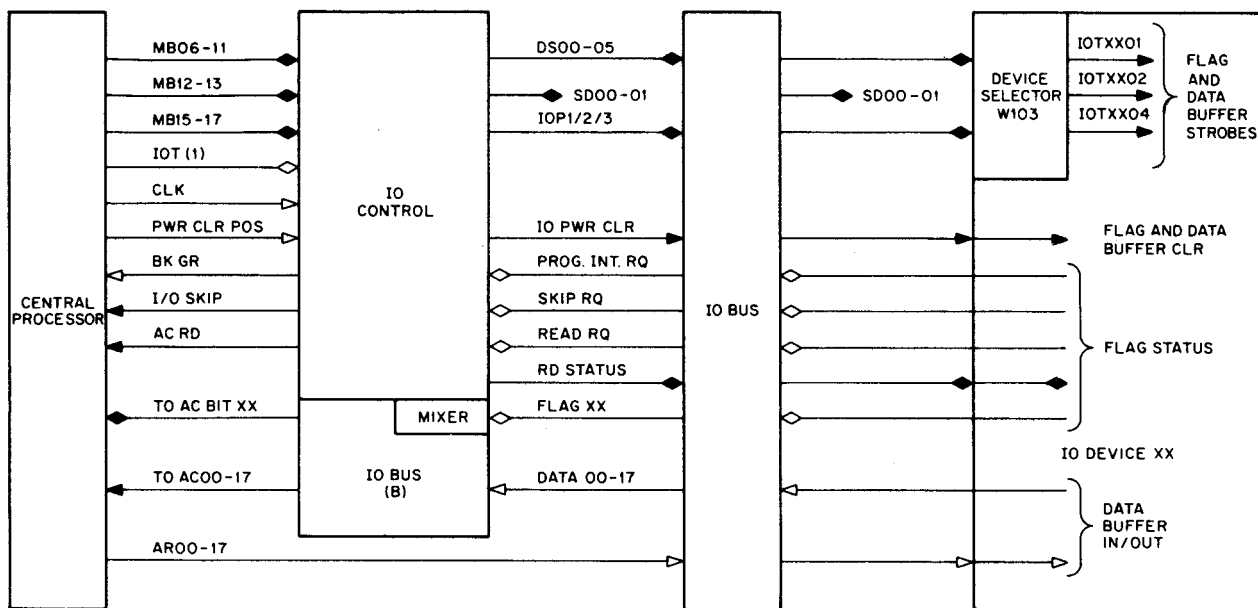


Figure 3-29 Program-Controlled I/O Interface

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 transfer, 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 I. If more than one device requests a program interrupt simultaneously, the computer honors the first flag sensed in the I/O search routine. Succeeding devices wait for the completion of previous interrupts. The computer returns to the main program after it services all interrupts in turn.

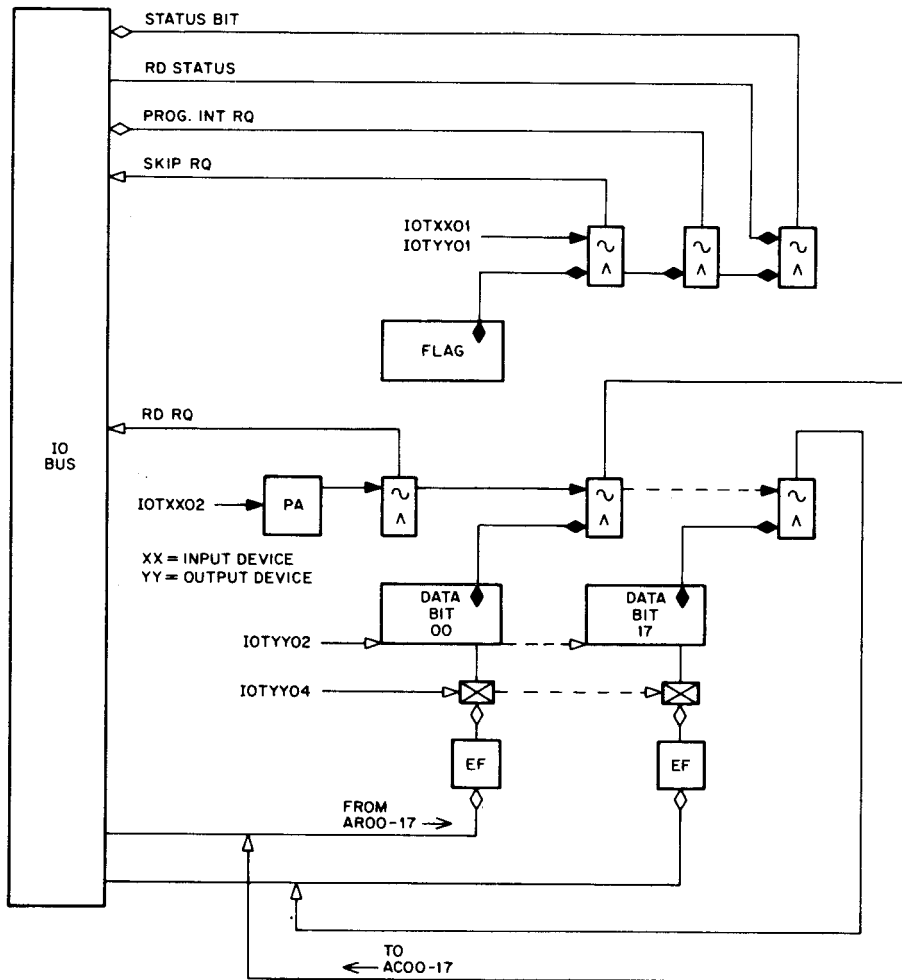


Figure 3-30 I/O Device Control Logic

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.

3.8.1.3 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 3-31 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 SD00-05 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.

3.8.1.4 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, Section 3.5.8. 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 SD00-05. These enable the input gates at device selector XX; 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(1). 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 sets IO0 for a count of 10. IO CLK(B) pulses are derived on drawing KD3(3) from IO CLK POS pulses at S107-J21D. In Figure 3-18 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 flip-flop in the set state gates on the IOTXX02 pulse in the device selector.

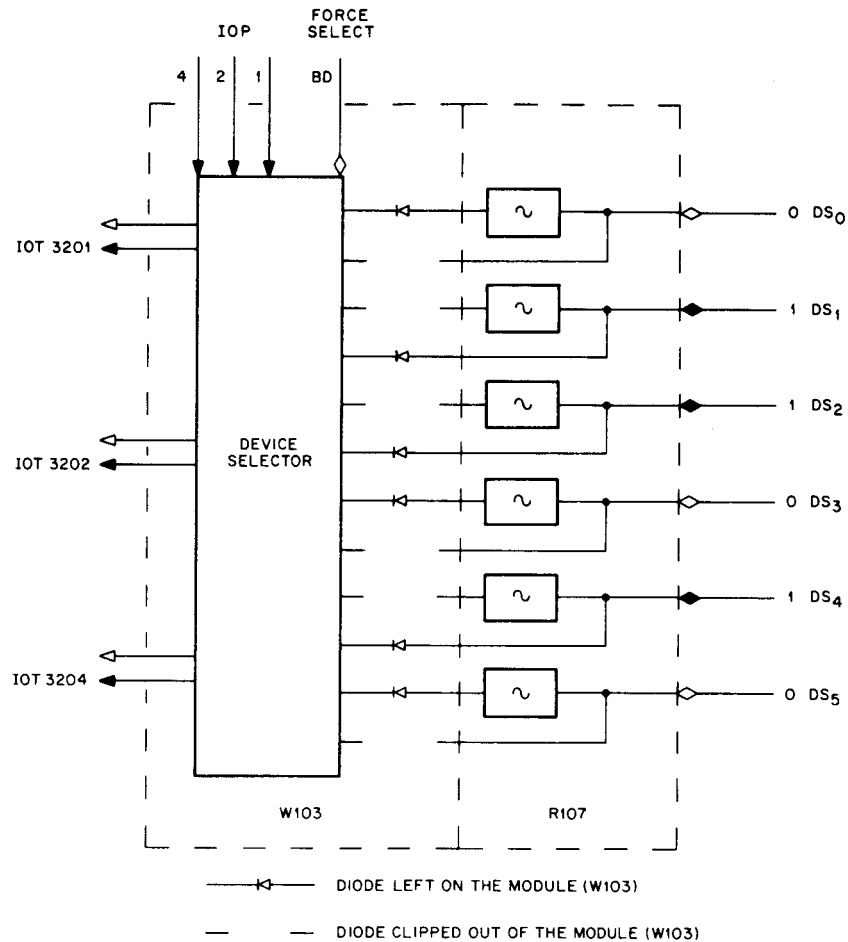


Figure 3-31 Device Selector W103, Schematic Diagram

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 DYLD at R002-F20N for AC RD, 500 ns after IO CLK POS triggers delay B301-H22D.

AC RD goes to the CM sense flip-flops, drawing KC19(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 500 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, Section 3.5.8. 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.

3.8.1.5 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 3-30 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, Section 3.5.8. 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.

3.8.1.6 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 to produce the IO SKIP pulse 500 ns after IO CLK POS in pulse amplifier W612-F18D, drawing KD3(3). 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.

3.8.1.7 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 them 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 interrupt enable logic is contained in the I/O control, drawing KD3. IOT00042 is detected during fetch to start the IOP generation as in Section 3.8.1.4. 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 SD00P level. SD00P conditions the DCD set gate to the PIE flip-flop, drawing KD3(2).

Since MB06-11 are 00, DS00-05 are at ground on drawing KD3(1). These ground levels and IOT(1) produce the negative 0XEN and 00XXEN(B) levels. The 00XXEN(B) level waits for the IO0, IO1 pulse counter and MB16(1) to set the IOP2 flip-flop. IOP2(1) and 00XXEN(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 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.

Further, IO CLK(B) pulses derive from the main clock's CLK POS pulses on drawing KD3(2). Here an AM SYNC BUS(0) signal goes to ground and disables IO CLK POS pulses if the DMA channel is in a current program break. Thus the PI facility is lowest in priority among all program break segments (DMA, DCH, RTC, API).

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 (DCH SYNC(1), Section 3.8.2). 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 3-32 IO SYNC POS sets 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 NORed 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, MEM DONE from core memory, and RUN(1). 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.

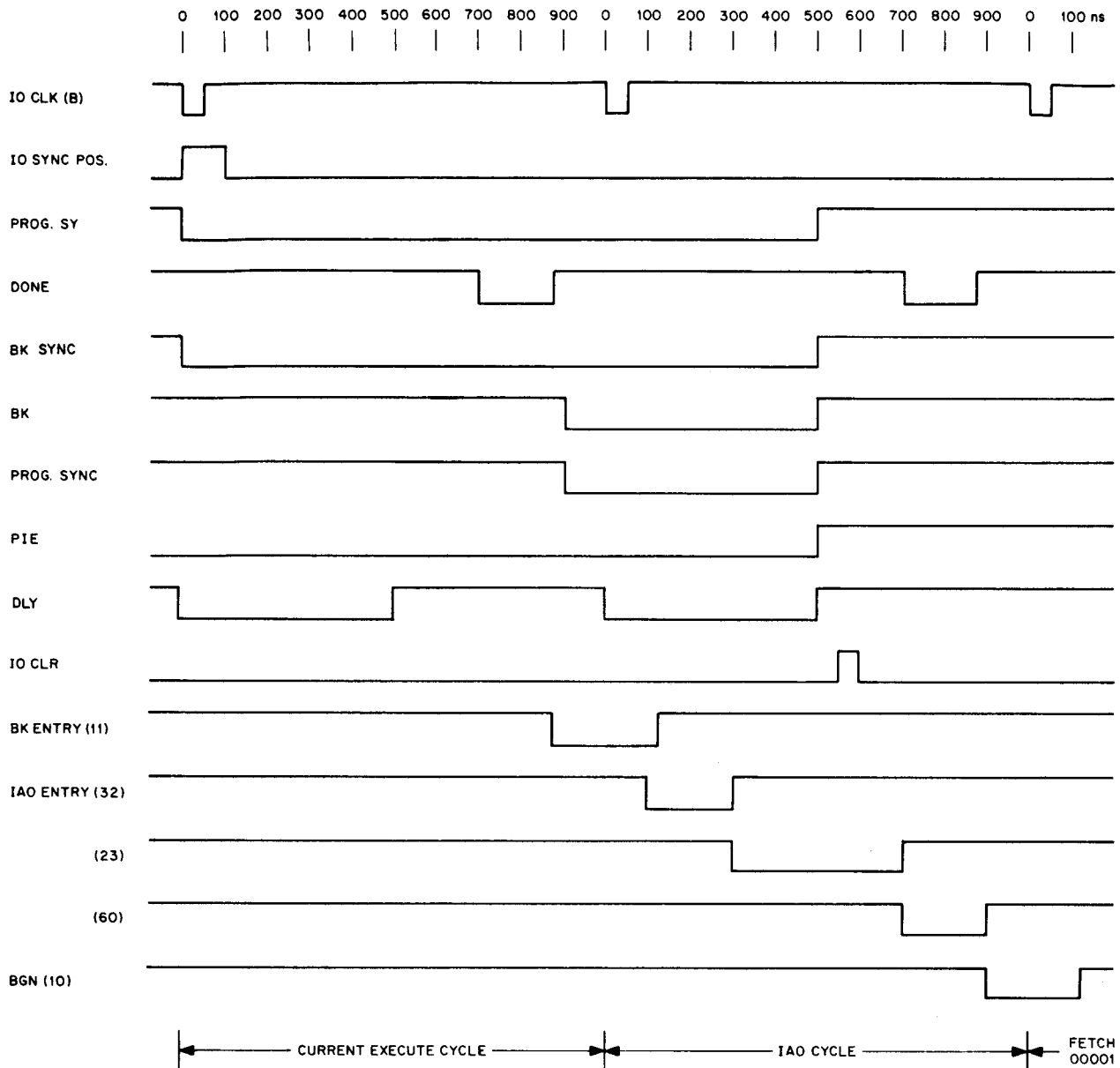


Figure 3-32 Program Interrupt Timing

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 (Section 3.5.6.3) 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 EXT(1) inhibits the setting of the CAL flip-flop.

Returning to the I/O control logic, the CLK pulse that initiated the IA0 cycle develops another IO 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 500 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 S603-J10K to generate a ground-going PIE(0) 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 reenable 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 I 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 I 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 I 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 (Section 3.5.2) 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 (Section 3.5.2), 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/IO 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 SD00 ground level.

3.8.1.8 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 as in Section 3.8.1.4. In this case device 03 is the KSR 33 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 3-3.

On drawing KD3(1), DS00-02 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, 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). 500 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), as in Section 3.8.1.4 to gate the contents from I/O bus(B) to the O bus and into the AC.

Table 3-3
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 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

3.8.2 Data Channel Transfers

Four data channels (DCH) provide a relatively 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 used also for program-controlled transfers.

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

<u>Channel Registers</u>		<u>Device</u>
00030	00031	0
00032	00033	1
00034	00035	2
00036	00037	3

Four additional devices can be attached to the existing configuration; appropriate channel register addresses must be assigned.

These registers must be initialized by the program before the device may begin transferring data. The first register of a pair is a word count (WC) register which is 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.

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

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 unique WC register on the secondary I/O bus; the BK SYNC causes the computer to exit the program and to go into the BK entry process as for program interrupt, Section 3.8.1.7.

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 recognizes the BK as a DCH break, and therefore goes into 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 into the MB. From here the contents are incremented by 1 and examined. If the contents (word count) have incremented to 0, an IO OFLO signal goes to the device, telling it to shut down operations 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 reinitializes 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. 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 Device Selector W103 to turn on an

IOTXX02 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 merely 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 blocked 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 IOTXX04 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. Moreover, a series of IOT instructions is non-interruptable. The entire series must be completed; the DCH request causes a break on completion of the instruction following the last IOT. For maximum operating efficiency, the program must be planned with the device transfer rates and the break entry point in mind. At 50,000 wps, a one-word transfer takes place in $20 \mu\text{s}$ ($1/50 \times 10^{-3}$). The worst-case delay between breaks is about $30 \mu\text{s}$. As device transfer rates approach the maximum 333,000 wps, it becomes less efficient to hold a device in "ready" while executing extended instructions above. It is better to avoid programming these instructions where DCH breaks are anticipated.

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 (3.8.2.8).

3.8.2.1 I/O Bus Connections - The DCH uses the data lines, IO SYNC, RD RQ, and IO PWR CLR lines of the primary I/O bus, 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 (Section 3.8.1.1) apply to the secondary. Of the 15 IO ADDR lines shown, the DCH uses the six least significant for the four assigned pairs of channel addresses.

The basic PDP-9 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.

3.8.2.2 Multiplexer W104 - Figure 3-33 is the logic diagram for the W104 module. The device flag and IO SYNC pulse sets the REQ flip-flop. The REQ flip-flop when set 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.

3.8.2.3 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 only where no AM SYNC (DMA) is present, drawing KD3(2), and 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, Figures 3-33 and 3-34. 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 negation 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, RTC, PI, or optional API operation is in progress. The DCH, therefore, cannot interrupt any of these current operations.

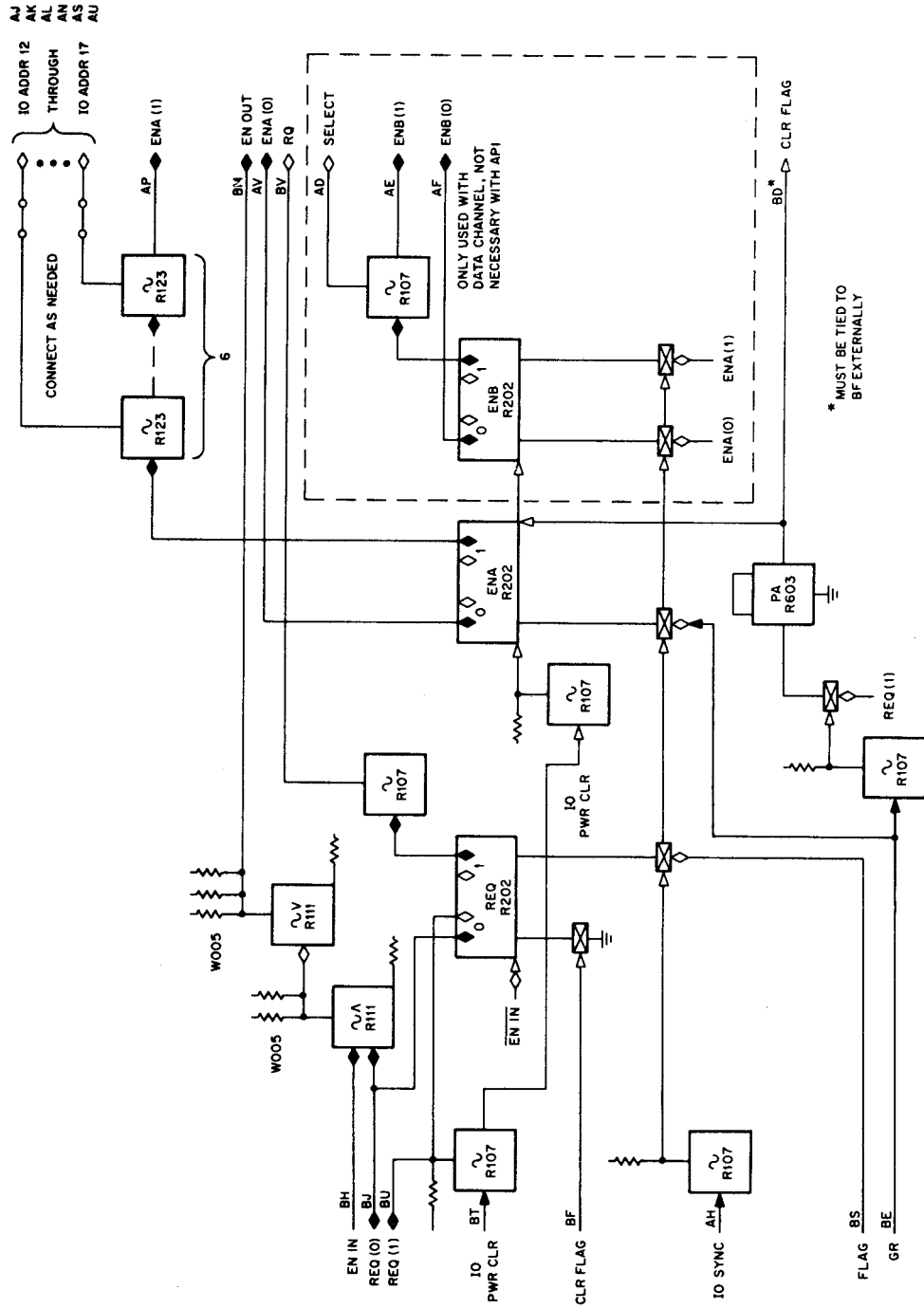


Figure 3-33 Multiplexer W104, Logic Diagram

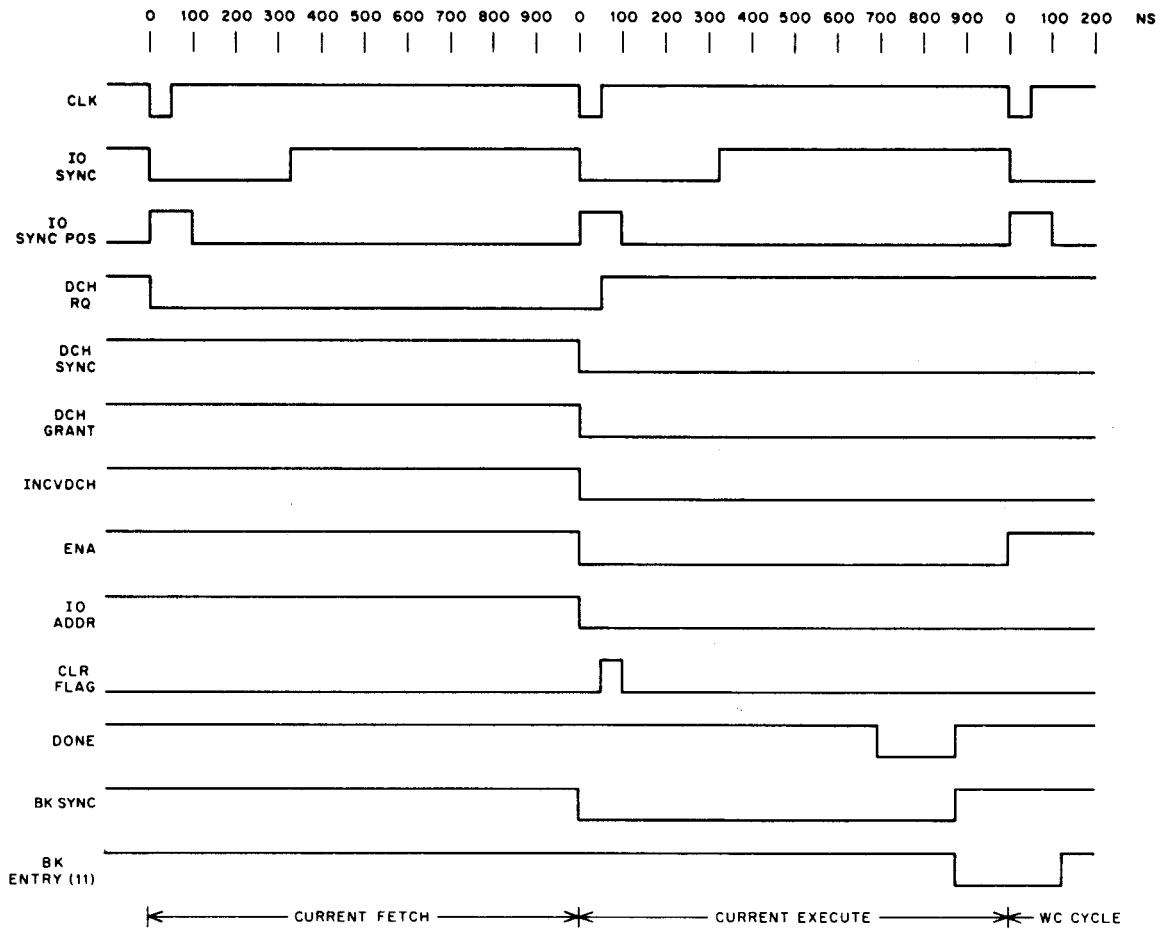


Figure 3-34 DCH Break Synchronization

IO SYNC POS strobes a DCD gate conditioned by DCH RQ to set the DCH SYNC flip-flop. DCH SYNC(1) holds the CLK SYNC and optional 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- μ s DCH BK DLY, generates DCH GRANT on drawing KD3(1), and INC V 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 for IO ADDR(B). The DCH BK DLY illuminates the console DATA indicator, Section 3.7.5.11.

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 Figure 3-34.

BK SYNC goes to the CP via the CP/IO 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 $1 \rightarrow$ MBI in conjunction with SM(1), MEM DONE from core memory, and RUN(1). $1 \rightarrow$ 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 V 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.

3.8.2.4 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 3-33. ENB(1) sends a SELECT level to the W103 Device Selector in the DCH device. This level is applied to W103-BD, Figure 3-31, bypassing the device select code input gates to force selection of the device.

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 $1 \rightarrow$ 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. $BK0(1) \wedge BK1(0)$ 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 places them on the O bus. As the contents pass through the ADR they are incremented by 1. MBI(1) gates the contents into the MB. CONT(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), BK1(0), and the CM STROBE triggered by MEM STROBE produce OFLO on drawing KC14. OFLO goes through the CP/IO 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 (Section 3.8.1.7) 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), $\overline{INC MB}$, and BK1(0) boost this address to 14. The negative $\overline{INC MB}$ level is present because the add-to-memory capability (Section 3.8.2.8), and the real-time clock (Section 3.9.4) 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 $1 \rightarrow MBI$ in conjunction with SM(1), MEM DONE, and RUN(1). $1 \rightarrow MBI$ 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.

3.8.2.5 CA Cycle - SM(1) and CLK start the control memory and the core memory for the CA cycle. CM STROBE produced by $SM(1) \wedge CM CLK$ extracts process word 37, which contains DCH and CMA13. DCH(1) steps the BK counter to 11 by setting BK1, with BK0 remaining set.

If the device intended to make an input data transfer it has placed a RD RQ and $\overline{WR RQ}$ 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 issues from 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 IOTXX02 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 $\overline{WR RQ(B)}$ on drawing KD3(2). Reset DCH SYNC removes the BK SYNC, INCV DCH and DCH GRANT levels, releases the reset hold on the CLK SYNC, and optional PRE API SYNC flip-flops, and the set hold on the DCH SYNC SAVE flip-flop.

BK0(1), BK1(1), and $+1 \rightarrow CA\ INH$ produce DCH INX on drawing KD3(3). The $+1 \rightarrow CA\ INH$ 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.

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).

3.8.2.6 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 $\overline{WR RQ}$ on drawing KC19(3) prevent CLR from setting SAO. Thus, the STROBE SAL, SAR 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 CONT(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 500 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.

3.8.2.7 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 INCLR 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 KD3(2). Reset DCH SYNC removes the BK SYNC, INC V 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 IOTXX04 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 500-ns delay multivibrator B301-H22. 500 ns later the DLY recovers ($\overline{\text{DLY}}$) to trigger IO RESTART. 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 $\overline{\text{DLY}}$ also resets IOP4, and DCH SYNC SAVE as for the input transfer. IO SYNC derived from the next CLK pulse resets ENB.

3.8.2.8 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, an ADOF level from the ADRL appears at the DATA OFLO flip-flop, drawing KC15, if the sum exceeds $\pm 2^{17} - 1$, or ± 377777 . An EAE STROBE DLYD pulse from control memory sets the flip-flop in conjunction with SAO(1), ARO(1), and BK1(1)B. These are all present during process word 36 in the first data cycle. EAE STROBE DLYD occurs approximately 150 ns after MEM STROBE triggers the chain to extract process word 17, or 50 ns before the word is extracted. The DATA OFLO(1) level is returned to the device over the I/O bus. See Section 3.5.6.1 for ADOF details.

3.8.3 DMA Channel Transfers

The direct memory access channel provides a direct path for data transfers between core memory and as many as to three extremely-high-speed devices. Any of the three devices may request and steal a core memory cycle from the central processor at any time, starting with the CLK pulse following the next core memory cycle. Each DMA device includes its own word count and address registers, similar to those described for DCH transfers, and the necessary register control logic. One DM09A Adapter/Multiplexer option or equivalent logic is also needed to establish device priority and transfer control. When a device steals a core memory cycle, the control memory processes are suspended for the 1- μ s cycle duration. The exception is the occurrence of a DMA request in the middle of an extended IOT or EAE execute period. Since these instructions do not require an operand from core memory, they continue to completion simultaneously with the DMA core memory cycle. Thus, in any case, the DMA request waits no longer than two cycles before it is honored, one cycle during which the request appears, and another during which the transfer is synchronized to start on the following CLK pulse.

Since suspension of control memory processes entails the suspension of all other forms of interrupts in the program, it follows that the DMA operation has highest priority.

3.8.3.1 Interface Connections - The central processor's MB outputs (MB00-17) are connected to receptacles D01, E01 in the CP section, drawing KC24. From there they go to D40, E40 in the core memory section. In core memory, they are applied to the input mixer gates, drawing MC3.

The adapter/multiplexer buffer register outputs (AM00-17) are connected to A32, B32 in the core memory section, drawing MC8. In core memory they are applied to the input mixer gates.

The core memory sense amplifier outputs (SA00-17) are connected to A35, B35, drawing MC8, and to A40, B40, drawing KC24. From A35, B35 they go to the adapter/multiplexer's buffer register, and from A40, B40 they go to A01, B01 in the CP section. The latter are applied to the B bus (SA00-17) and to the IR (SA00-04).

A ground AM RQ(1) level comes from the adapter/multiplexer when any of the three devices is ready for an input/output transfer. This level is connected to A29K in the core memory section, drawing MC8. From there it goes to the core memory AM SYNC flip-flop, drawing MC1(2), where it waits throughout the next memory cycle. The AM00-17 lines must contain the address of the data word to be accessed within 80 ns after the AM RQ(1) is honored by the memory control, i.e., 80 ns after memory control issues an AM GRANT to the adapter/multiplexer.

An AM STROBE pulse (120 ns negative) from core memory timing goes to the adapter/multiplexer via A29T to indicate that core memory is reading out the contents of the addressed location. For an output data transfer, the adapter/multiplexer will accept the data into its buffer register and will strobe it into the device's data register. The data may or may not be sampled by the device logic, so that the contents in the buffer register may or may not become modified for rewriting into core memory during the write half-cycle. A 200 ns gap between read and write exists in core memory to afford data sampling and modification.

For an input transfer the adapter/multiplexer ignores the data read out and places new data in its buffer register to be written into memory during write half-cycle.

An AM GRANT pulse (120 ns negative) from core memory control goes to the adapter/multiplexer via A29V to indicate that the core memory write half-cycle has been completed. The adapter/multiplexer uses the leading edge of the pulse to place a new address in its buffer register. MA JAM occurs in core memory after MEM DONE to place the address in the MA for the new core memory cycle.

Cabling between the core memory interface and the adapter/multiplexer must not exceed three feet. Total cabling between the core memory interface and three I/O devices must not exceed 50 ft.

3.8.3.2 Transfer Operations - Figure 3-35 is a block diagram of the DM09A Adapter/Multiplexer option, showing the connecting lines between core memory and the option, and between the option and the I/O devices. For a detailed description of the option's internal logic, refer to the option manual.

The adapter/multiplexer services the devices in a preset order of priority and routes the address and data into the PDP-9 core memory circuits.

When one or more devices generate a RQ IN level the adapter/multiplexer selects the higher priority device and sends a ground AM RQ(1) level to the AM SYNC flip-flop, drawing MC1(2).

AM RQ(1) occurs asynchronously during the current core memory cycle. Before the next CLK pulse appears, a PRE WRITE OFF pulse of the current cycle produces MEM DONE, as usual for this processor-accessed cycle (Figure 3-22). The CLK pulse produces POST CLK and SYNC CLK as one more processor-accessed cycle occurs. SYNC CLK sets AM SYNC, and AM SYNC(1) becomes AM SYNC(1)B and AM SYNC BUS(1). AM SYNC(1)B goes to the CM timing, drawing KC16, where it will prevent SM(1) of the processor-accessed cycle from restarting the CM on the next CLK pulse. 50 ns after this next CLK

pulse POST CLK resets the MODE flip-flop conditioned by AM SYNC(1). PRE-WRITE OFF sets MEM DONE and produces AM GRANT. AM GRANT lets the adapter/multiplexer place an address on address lines AM05-17.

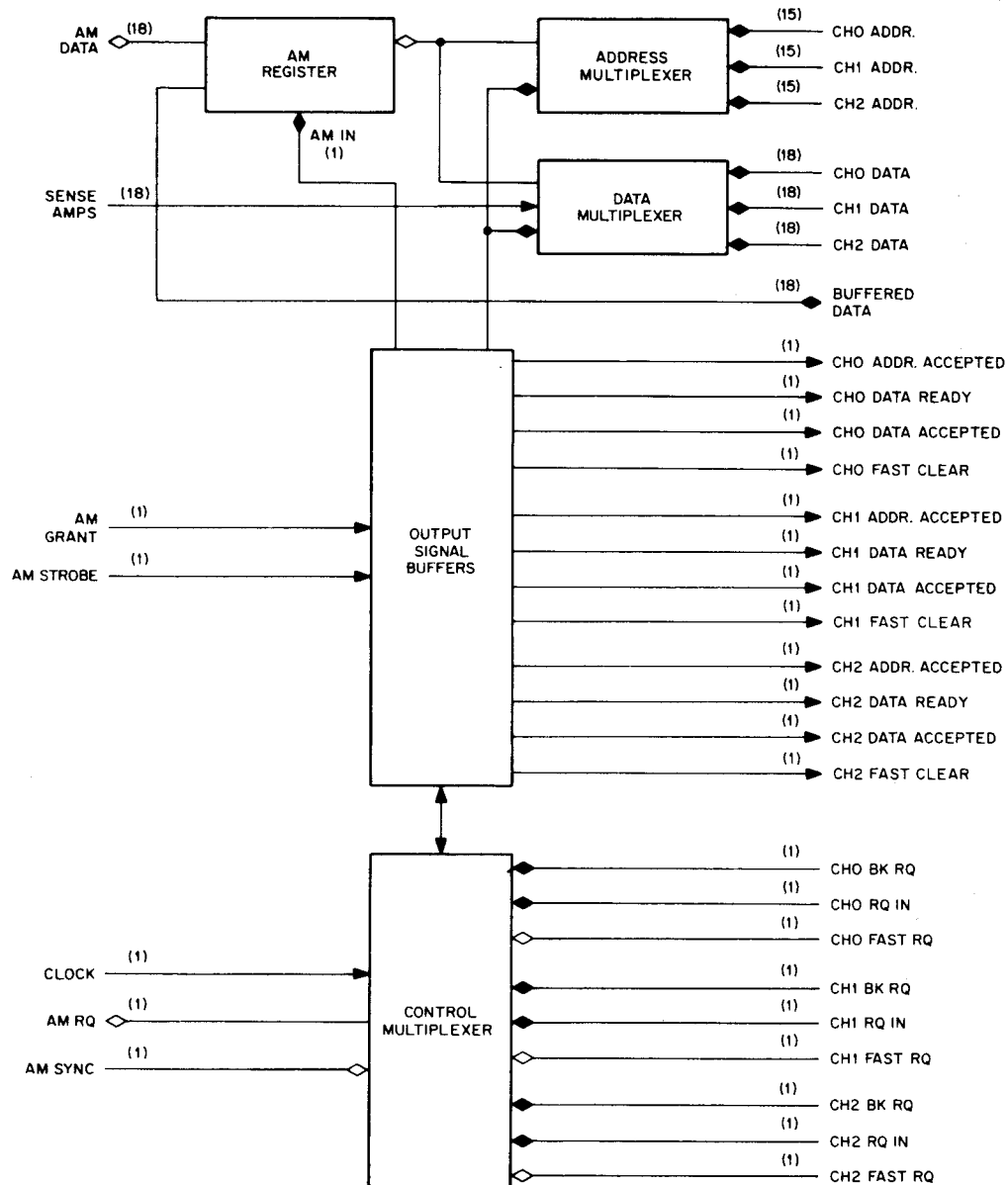


Figure 3-35 Adapter/Multiplexer DM09A, Block Diagram

MODE(0) gates the address lines AM05-17 from the adapter/multiplexer's AM register into the input mixer, drawing MC3. MA JAM gates the address MBS05-17 into the MA. The adapter/multiplexer sends an ADDR ACCEPTED signal to the requesting device.

The core memory read half-cycle begins for the DMA transfer. If an input transfer is specified by the device, the data read out during the read half-cycle is blocked at the AM register. The AM STROBE from core memory causes the adapter/multiplexer to gate the input data from the device into the AM register via the data multiplexer logic, so that the data is applied to the input mixer at the start of the core memory write half-cycle. The adapter/multiplexer sends a DATA ACCEPTED signal to the device. The data lines MBS00-17 go into the core memory's write drive selectors G219 for write bit driving.

If an output transfer is specified by the device, the data read out during the read half-cycle enters the AM register. The AM STROBE causes the adapter/multiplexer to issue a DATA READY level to the device, which then gates the buffered data into its data register.

The device logic may sample and modify the output data, during the 200 ns pause between core memory read and core memory write. The core memory write half-cycle rewrites the modified or unmodified data into the addressed location.

If another AM RQ(1) is issued by the device, the DMA process repeats on the next CLK pulse. AM GRANT at CLK time tells the adapter/multiplexer to gate the new address from the device into the input mixer.

3.8.4 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 progress, 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 an DBR/JMP I instructions.

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.

3.8.5 Transfer Priorities

From the foregoing paragraphs, the following descending order of interrupt priorities is established where simultaneous interrupt requests occur.

DMA
DCH
RTC
API
PI
Main Program

3.9 BASIC I/O DEVICE CONTROL

The basic PDP-9 includes four I/O devices as standard equipment: KSR-33 Teletype; PC02 Paper-Tape Reader; PC03-Paper Tape Punch; and Real-Time Clock.

The control logic and data buffer registers for the three transfer devices are included in the I/O control section of the computer. The real-time clock operates internally as a DCH device. The following sections are concerned primarily with this control logic. For details on the device mechanisms, refer to the appropriate manuals listed in Table 1-1.

3.9.1 Teletype

The KSR 33 Teletype Unit, model TS, is a keyboard send-receive unit supplied as standard equipment with the PDP-9. Operating at speeds up to 10 char/sec, the unit is jumpered internally for half-duplex operation with local hard copy provisions. Refer to the KSR 33 manual listed in Table 1-1 for complete details.

The teletype unit consists of two separate I/O devices--an input transfer device using the keyboard, and an output transfer device using the teleprinter. For input transfers, the operator uses the

keyboard as an ordinary typewriter. As a key is pressed, a serial code pulse train is set up in the keyboard generator portion of the KSR 33 which relates a given bit position to a given time interval. The serial code pulses are applied to the teletype control logic in the PDP-9's I/O control section. Here they are converted into parallel format in an input shift register. When the last code pulse is shifted into the register, a keyboard flag raises to cause a program interrupt. The program interrupt subroutine senses the status of the flag with an IOT skip instruction, then an IOT read instruction clears the flag and transfers the contents of the shift register into the AC via the input mixer and I/O bus (B). As the keyboard generates input characters, the teleprinter is also active, printing a hard page copy.

For output transfers the process is reversed. When the program is ready to print a character, an IOT load instruction transfers the contents of the AC into an output buffer via the AR and the I/O bus. In this case the character code is in parallel binary form, and the teletype control logic shifts the character one bit at a time out of the buffer and into the KSR 33 teleprinter circuits. When the last character bit is shifted out, the teleprinter flag raises to cause a program interrupt. The program interrupt subroutine senses the status of the flag with an IOT skip instruction, then an IOT load instruction clears the flag and loads another character into the buffer.

The serial code for each character is an 11-unit pulse train consisting of marks (current pulses) and spaces (no current). The first unit is always a start space, the next eight units comprise the character code of marks and spaces, and the last two contain a double stop mark.

At 10 char/sec the baud rate is 110; one unit therefore, extends over 9.09 ms. Input and output clocks operating at the baud rate synchronize the serial-parallel and parallel-serial conversions in the teletype control logic.

The 8-bit code used by the KSR 33 Teletype Unit is the American Standard Code for Information Interchange (ASCII) modified. This code is read in the reverse of the normal octal form used in the PDP-9 since bits are numbered from right to left with bit 1 having the most significance.

The KSR 33 can generate all ASCII codes except 340 through 374 and 376. Generally, codes 207, 212, 215, 240 through 337, and 377 are sufficient for teletype operation. The standard number of characters printed per line is 72. The sequence for proceeding to the next line is a carriage return followed by a line feed (as opposed to a line feed followed by a carriage return). Octal codes from 200 through 337, 375, and 377 are indicated in Table 3-4 with the associated ASCII character.

3.9.1.1 Keyboard Control - The Keyboard control logic is shown on drawing KD11(1). When a character key is pressed, 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) place the solenoid driver output W040-B33S at -15V, releasing the printer solenoid to enable the printing circuits for simultaneous printout.

The TTI CLK, enabled by TT IN ACT(1), is adjusted internally to issue 400 ns 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 at a pulse amplifier within the TTI CLK. TTI LOAD strobes the input gates of the TTI register. TT KBD IN(B) produced by the start space resets TTI00. All other bits remain at 1.

Successive TTI LOAD pulses appear synchronously with successive character units to shift each mark or space into the next low-order TTI bit position. On the eighth TTI LOAD, the start space gets into the least significant bit, TTI07, and the seventh code unit gets into TTI00. TTI07 at 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 TTI00, shifts the first into TTI07, 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 operations.

Note that a 0 in the TTI register corresponds to a mark and a 1 corresponds to a space. These will be complemented at the O bus to place a 1 for a mark and a 0 for a space in the AC when an IOT read instruction is issued.

Initially, the TTI register was set to 1s, producing TTI FULL. The negative TTI FULL level operates with $\overline{\text{TT KBD IN(B)}}$ 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 level sets 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). $\overline{\text{TT KBD IN(B)}} \wedge \text{TTI FULL}$ causes TT IN ACT to reset on the first TTI CLK pulse. Although the first TTI CLK generates a TTI LOAD, TTI00 remains at 1 because of the absence of TT KBD IN(B) at the reset gate.

KBD FLG(1) causes a ground PROG INT RQ at inverter R111-D39H. The PROG INT RQ goes to the I/O control logic, drawing KD3(2), to cause a program interrupt. When the CP honors the interrupt, the KSF instruction in the flag search subroutine determines that the keyboard has caused the interrupt, and jumps to the keyboard service routine. The search and service instructions pertaining to the keyboard are as follows:

IOT0301	KSF	Skip on keyboard flag
IOT0312	KRB	Clear keyboard flag and AC, then read the keyboard buffer into the AC

3.9.1.2 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) as usual (Section 3.5.8). On drawing KD3(1) the IOT(B) level derived from IOT(1) samples

the MB06-11 bits in the instruction to produce the appropriate DS00-05 levels, and IOT(1) further decodes DS00-02 to produce OXEN. OXEN samples DS03-05 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 sets IOT0302 and TT RDR RUN in conjunction with KBD SEL on drawing KD11(1). It also resets the KBD FLG.

IOT0302(1) generates an INT RD RQ BUS level on drawing KD3(3). This level is NORed at R111-F19UV 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 TTI00-07 into the input mixer gates, positions 10-17, 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.

3.9.1.3 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) as for KRB. 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 KBD SEL(B) sample the state of the KBD FLG. If the KBD FLG is set, the INT SKP RQ BUS goes to ground, triggering pulse amplifier W612-F18D on drawing KD3(3). The 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.

3.9.1.4 Clear Teleprinter Flag and Load the Buffer - Unlike keyboard control operations, teleprinter control is initiated by an IOT instruction. The TLS instruction (700406) is a combined IOT0402 and IOT0404 instruction which selects teleprinter operation, clears the T-PRNTR FLG, and loads the output buffer (TTO) register with an 8-bit binary character from the AC (AR) via the I/O bus. Printer selection enables the TTO clock to shift the bits out of the register serially and into the teletype printing circuits, including a start space and stop mark.

During the IOT fetch, the TLS instruction is decoded in the CP's op code detection circuits for IOT(1) as usual (Section 3.5.8). On drawing KD3(1), the IOT(B) level samples the MB06-11 bits in the instruction to produce the appropriate DS00-05 levels, and IOT(1) further decodes DS00-02 to produce OXEN. OXEN goes to the teleprinter control, drawing KD11(2), to sample DS03-05, 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 on drawing KD3(3) 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 TTO00-07 from the I/O bus. (The character is placed on the I/O bus from the AR during fetch, Section 3.8.1.5.)

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 because the keyboard is deselected. 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 turns on the first TTO LOAD pulse.

TTO OUT ACT(1) upon setting sets the TT LINE flip-flop. TT LINE(1) places the output of solenoid driver W040-B33S at -15V, releasing the printer select solenoid to cause 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 TTO00, and all character bits into the next low-order TTO bit position. 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 TTO07 and TTO EN(0) into TTO06. 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 ACT while setting the T-PRNTR FLG. The tenth TTO LOAD also resets TT LINE with TTO07(1), which was the initial TTO STOP state. Thus TT LINE(0) 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. 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.

3.9.1.5 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 W612-F18D on drawing KD3(3). The 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 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.

3.9.1.6 LINE/OFF/LOCAL Switch - The LINE/OFF/LOCAL switch on the KSR 33 controls the application of power to the motor and the logic connections to the I/O control section of the computer. In the LINE position the unit is energized and connected to the computer as an I/O device. In the LOCAL position the unit is energized for off-line operations only, disconnecting the I/O bus line signals by shorting out the incoming signals and placing a dummy load on the outgoing signals. The console POWER switch must be turned on to power the KSR 33 unit in both cases.

3.9.2 Tape Reader

A DEC Type PC02 Paper-Tape Reader and its control logic are standard equipment with the PDP-9. The tape reader and its companion PC03 Paper-Tape Punch are mounted together above the operator's console. Collectively they comprise the DEC Type PC09 Paper-Tape Reader/Punch system. The control logic for each is located in the I/O control section of the computer.

The tape reader photoelectrically senses the holes punched in 8-channel, 1-inch wide paper or Mylar tape at a maximum rate of 300 lines per second. Its mechanical operation is fully described in a separate document listed in Table 1-1.

IOT instructions under program control start the reader, load an 18-bit buffer register with the information read from the tape, sense the status of the register, and transfer the information from the register to the AC via the input mixer I/O bus (B). The instructions select the reader for operation in either the binary or the alphanumeric mode, as appropriate to the punched format.

Alternatively, the READ IN key on the operator's console can be used to enter data manually from the reader into core memory (Section 3.7.4.9). A manual FEED pushbutton above the reader's tape feed platform can be used to feed a tape through without reading. A ninth-channel photodiode senses

the sprocket-driven feed holes in the tape. If the end of the tape is sensed, special RDR NO TAPE logic stops the reader and issues a program interrupt request.

An IOT RSA instruction selects reader operation in the alphanumeric mode. Each RSA causes one line of 8-bit information to be read into bits 10-17 of the reader buffer (RB), as shown in Figure 3-36. The binary mode is selected by an IOT RSB instruction. One 18-bit binary word occupies three lines of tape, each line containing one 6-bit character. Each RSB instruction causes three successive lines of tape to be read into the appropriate bit positions of the RB. In the binary mode, channel 7 is never read (except during manual READ IN operations). Channel 8 must always be punched and read to gate each line into the respective RB positions, but is ignored as an information bit. The reader flag goes up with each line in the alphanumeric mode, and with every third line in the binary mode.

All IOT instructions associated with the reader are listed below. The logic functions that they perform follow.

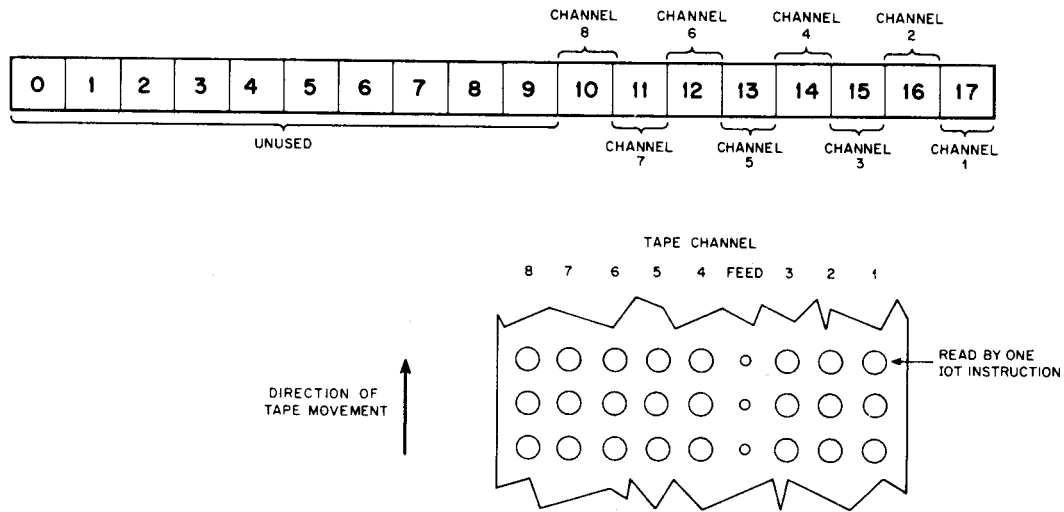
IOT0101	RSF	Skip on reader flag
IOT0102	RCF	Clear reader flag and inclusive OR the RB into the AC
IOT0112	RRB	Clear reader flag and AC, then transfer RB into AC
IOT0104	RSA	Clear reader flag and select reader in alphanumeric mode
IOT0144	RSB	Clear reader flag and select reader in binary mode

3.9.2.1 Select Reader Binary - The RSB instruction (700144) is decoded in the CP's code detection circuits for IOT (1) as usual (Section 3.5.8). On drawing KD3 (1) the IOT (B) level samples the MB06-11 bits in the instruction to produce the appropriate DS00-05 levels, and IOT (1) further decodes DS00-02 to produce 0XEN. 0XEN goes to the reader control, drawing KD9 (1) to sample DS03-05, resulting in RDR SEL and RDR SEL(B). RDR SEL conditions a DCD gate at pulse amplifier S603-D10T. The gate now waits for the start of the third IOT execute period, at which time the CLK pulse counter IO0 steps to 0, strobing the MB15(1) gate on drawing KD3(3) to generate the IOP4P pulse. IOP4P triggers the PA on KD9(1) to generate the IOT0104 command.

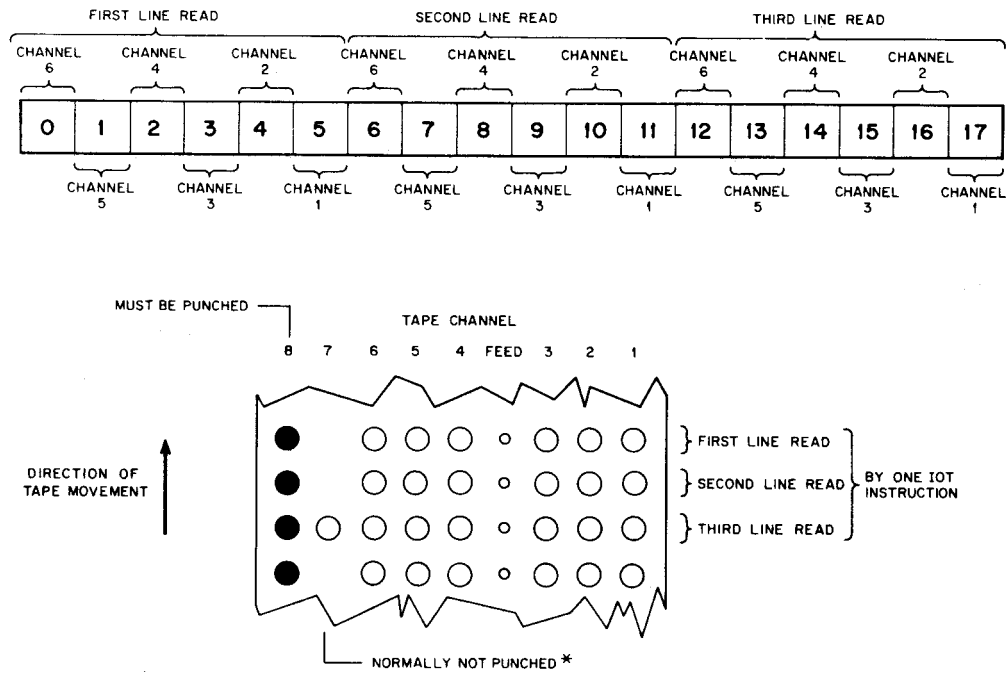
IOT0104 resets the RDR FLG, RDR 1, RDR 2, RDR ALPHA, and RB flip-flops, and sets the RDR RUN flip-flop. RDR ALPHA becomes reset because MB12 is 0, generating SD00P on drawing KD3(1).

RDR RUN(1) and RDR NO TAPE(0) are Nanded on KD9(1), and the grounded output is NORed to produce a negative RUN level. RUN is applied to NAND gate R111-E07UV in the RDR CLK circuits. Here the STOP DLY POS input is initially negative because the 45 ms STOP DLY is quiescent.

RUN and the negative STOP DLY POS enable R111-E07UV to condition the RDR INDEX input DCD gate and to trigger the 1 μ s GO DLY. When the GO DLY recovers, RDR GO sets the RDR CLK EN flip-flop.



a. Tape Format and Reader Buffer Register Bit Assignments in Alphanumeric Mode



b. Tape Format and Reader Buffer Register Bit Assignments in Binary Mode

Figure 3-36 Paper Tape and Reader Buffer Formats

*Note: In hardware readin mode, channel 7 must be punched in line 3 of last data word or reader will not stop.

RDR CLK EN(1) triggers the RDR CLK R401 and the clock accelerator G903. Initially the RDR CLK pulses occur at 5 ms intervals, but within 10 to 20 pulses the G903 accelerates the clock rate so that a RDR CLK pulse occurs every 1.67 ms. Each RDR CLK pulse strobes the conditioned DCD gate to generate RDR INDEX pulses.

The first RDR INDEX pulse sets the RDR PWR flip-flop and strobes the input gates of the RDR A, RDR B flip-flops. RDR A and RDR B act as a Gray code pulse counter whose outputs are used to drive the tape reader synchro motor along with PWR(B) derived from the RDR PWR flip-flop.

The RDR A and RDR B flip-flops step through a count of 00, 10, 11, 01, and back to 00 on successive RDR INDEX pulses. Their outputs go through S107 inverters on drawing KD9(2) to four dual solenoid driver modules W040 at the reader motor (drawing PC09-0-2). The solenoid driver modules actuate in pairs to drive two of four motor windings for motor rotation. The stepping RDR A and RDR B flip-flops sequentially select the pairs of solenoid drivers along with PWR(B) to drive two windings at a time for continuous motor rotation. This scheme requires that the RDR A and RDR B count be stepped twice to produce a tape movement equal to one character position. The RDR INDEX rate is thus twice the character rate (300 char/sec = 1 char/3.33 ms).

Assuming that a new tape has been loaded in the reader or that the reader has completed a previous read operation, the tape now contains nothing but feed holes which allows time for RDR CLK acceleration. The motor is driven by stepping RDR A and RDR B, but in the absence of punched holes in channel 8, the control logic prevents reading 0s into the RB, consequently preventing the RDR FLG from setting on three lines of blank tape.

Eventually the blank tape is exhausted and the first character is pulled into the reading position. At this stage RDR A and RDR B step to a count of either 00 or 11 (RDR A = RDR B). This indicates that the tape holes of the first character are in position above the photosensors.

With the first character in the read position, the RD HOLE 8P V ALPHA level is present at the DCD gate to the pulse amplifier S603-E08M, drawing KD9(1). Since the reader is reading a binary tape, RD HOLE 8 is always punched. The RD HOLE 8 level becomes RD HOLE 8P V ALPHA at S107 inverters on drawing KD9(2). This level conditions the DCD gate above. Simultaneously, the sensed data levels RD HOLE 1(B) through RD HOLE 6 (B) condition the input gates to RB00-05.

The next RDR INDEX pulse steps RDR A and RDR B to 10 (or 01), continuing motor rotation toward a between-character tape position. On drawing KD9(1), $RDR A(1) \wedge RDR B(0)$ or $RDR A(0) \wedge RDR B(1)$ produce the ground RDR COUNT level. RDR COUNT triggers pulse amplifier S603-E08M conditioned by RD HOLE 8P V ALPHA. The PA output sets the RDR 1 flip-flop.

RDR 1(1) strobes a DCD gate conditioned by RDR ALPHA(0) on drawing DK9(2). This gate triggers pulse amplifier S603-D10F. The PA pulse strobes the DCD set gates at the RB00-05 flip-flops,

placing 1s in all register positions conditioned by RD HOLE(B) levels. The RD HOLE(B) levels come from the tape hole photosensors via S107 inverters at W023-A17. Thus, a 1 in the RB denotes a punched hole; a 0 denotes a blank.

On the count of 11 (or 00), the drive motor pulls the next character into reading position. The photosensors read these out to the DCD gates at RB06-11.

The count of 01 (or 10) produces another RDR COUNT. RDR COUNT sets the RDR 2 flip-flop, and RDR 2(1) then triggers pulse amplifier S603-D10M in conjunction with RDR RUN(1). The PA output strobes the second character into RB06-11.

The count of 00 (or 11) again pulls the tape character into the reading position, and the RD HOLE levels sensed by the photodiodes condition the DCD input gates to the RB. The count of 10 (or 01) pulls the tape into the between-character position and produces another RDR COUNT.

This time RDR COUNT sets the RDR FLG in conjunction with conditioning levels RDR 2(1) and RDR RUN(1). RDR FLG(1) triggers pulse amplifier S602-C09K on drawing KD9(2) in conjunction with RDR RUN(1). The PA output pulse strobes the third character into RB12-17.

RDR FLG(1) also resets RDR RUN on KD9(1) and generates a PROG INT RQ on KD9(2) if the optional API system is disabled or not installed. The PROG INT RQ goes to KD3(2) to initiate a program interrupt.

The program now has 1.67 ms in which to enter a flag search and a service routine to transfer the contents of the RB into the AC, perform any desired operation on the contents, re-enable the PI facility, and issue another RSB to assemble the next word in the RB. If another RSB does not appear in the allotted time, the next RDR CLK pulse resets RDR CLK EN in conjunction with $\overline{\text{RUN}}$.

RDR CLK EN(0) disables the RDR CLK and triggers the 45 ms STOP DLY. The 45-ms-duration STOP DLY POS level at ground holds RDR CLK EN in the reset state and inhibits RDR GO triggering. STOP DLY upon recovery ($\overline{\text{STOP DLY}}$) resets the RDR PWR flip-flop. Resetting RDR PWR disables the reader motor, so that the motor starts to decelerate. The 45-ms delay prevents a late RSB from restarting the reader motor immediately. Although a late RSB sets RDR RUN, the consequent RUN level cannot enable the output gate at the STOP DLY for RDR GO until the delay has lapsed. This delay allows the reader motor to come to a complete stop. Upon delay recovery RDR GO sets RDR CLK EN and the reader restarts, with nothing more than an overall data rate reduction of about 25 char/sec. Otherwise without this delay a late RSB would cause erratic motor rotation and tape tearing.

3.9.2.2 Select Reader Alpha - The RSA instruction (700104) results in the IOT0104 command as for RSB, but now MB12 is 0, producing a ground level SD00 to set the RDR ALPHA flip-flop. RDR ALPHA(1) now provides the RD HOLE 8P V ALPHA conditioning level, since channel 8 may or may not be punched in the alphanumeric mode. When the RDR A and RDR B flip-flops step to $A \neq B$, RDR COUNT sets RDR 1

and the RDR FLG, conditioned by RDR ALPHA(1) and RDR RUN(1). RDR FLG(1) gates on the PA conditioned by RDR RUN(1) to strobe RD HOLE 1(B)-6(B) into RB17-12 as for the third RSB character. Additionally, RDR ALPHA(1) and the inverted PA pulse strobe RD HOLE 7(B) and RD HOLE 8 (B) into RB11, RB10. The RDR FLG issues a PROG INT RQ on every line read, therefore, as opposed to every three lines for RSB. Although set on successive RDR COUNT levels, RDR 1 and RDR 2 are superfluous to alphanumeric operation. All other logic discussed for RSB prevails.

3.9.2.3 Transfer to AC - The RCF instruction (700102) clears the RDR FLG, RDR 1, and RDR 2 flip-flops, inclusive-ORs the contents of the AC and the RB at I/O bus (B), then gates the result into the AC. The RRB instruction (700112) clears the AC in addition to the flip-flops above, then gates the contents of the RB into the AC.

During the IOT fetch for the former, the contents of the AC are transferred into the AR and MB14(0) is detected to set ARO and IO BUS ON. These levels place the AR contents on the I/O bus via the A bus and ADR at the beginning of IOT execute. During execute the contents of the RB are gated into the input mixer and placed on I/O bus (B), where they are ORed with the contents on the I/O bus.

For the latter, MB14(1) is detected to set ACI. ACI(1) places 0s in the AC from the inactive O bus, thus clearing the AC. The fetch cycle logic for both cases is explained in Section 3.5.8.

In both cases, MB06-11 are detected in the I/O control logic to produce RDR SEL, and when IO1 sets, signaling the start of the second IOT execute period, it produces IOP2P on drawing KD3(3) in conjunction with MB16(1). IOP2P sets IOT0102 in conjunction with RDR SEL on drawing KD9(1). IOT0102(1) triggers a pulse amplifier to produce a CLR RDR pulse. CLR RDR resets RDR FLG, RDR 1, RDR 2.

IOT0102(1) generates an INT RD RQ BUS level on drawing KD3(3). This level is NORed at R111-F19UV 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), IOT0102(1) generates RDR ON BUS. The RDR ON BUS level gates RB00-17 into the input mixer gates, whose outputs are NORed onto I/O bus (B). If the instruction is an RCF, the contents of the AC (via the AR) are also present at the output NOR gates from the I/O bus. If an RRB, the AC has been cleared and nothing appears on the I/O bus.

LIO gates the results on the I/O bus (B) onto the O bus, and ACI(1) gates them from the O bus into the AC.

3.9.2.4 Skip on Reader Flag - The RSF instruction (700101) senses the status of the RDR FLG. If the flag is set, the reader 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 RDR SEL on drawing KD9(1). When IOO sets, signaling the start of the first IOT execute period, it produces IOP1P in conjunction with MB17(1). IOP1P sets IOP1.

IOP(1) and RDR SEL(B) sample the state of the RDR FLG on drawing KD9(1). If the RDR FLG is set, the INT SKP RQ BUS goes to ground, triggering pulse amplifier W612-F18D on drawing KD3(3). The PA pulse is the IO SKIP signal which sets the SKIP flip-flop in the CP, drawing KC14. The RSF 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.

3.9.2.5 Reader No Tape Logic - During RSA or RSB operations ground RDR COUNT levels occur on alternate steps of RDR A and RDR B where $A \neq B$. $\overline{\text{RDR COUNT}}$ applied to the DCD gates of the RDR NO TAPE flip-flop on KD9(1) samples the FEED HOLE condition. Since the $A \neq B$ condition occurs only when the tape is in a between-characters position, the FEED HOLE level should be at ground. If negative at RDR COUNT time, it can only mean that the tape has torn. This places a ground conditioning level at the DCD set gate of the RDR NO TAPE flip-flop. When RDR COUNT turns off ($\overline{\text{RDR COUNT}}$) it sets RDR NO TAPE, generating $\overline{\text{RUN}}$ at R111-C08HJ. $\overline{\text{RUN}}$ stops the reader by resetting RDR CLK EN on the next RDR CLK pulse, thus entering the STOP DLY period as for normal shut-down procedures. RDR NO TAPE(1) generates a PROG INT RQ as though the RDR FLG was raised. An extended or unattended program should make use of the read status (IORS) facility to test the RDR NO TAPE status before each RSB or RSA.

3.9.2.6 FEED Button - Paper tapes usually contain two feet of leader for loading ease and any amount of trailer. These leaders and trailers contain nothing but feed holes. Excess leader as well as the trailer can be fed through to the take-up bin by depressing the FEED button above the loading platform. When pressed, it supplies a ground RDR FEED level to NOR gate R111-C08JH, drawing KD9(1), resulting in a negative RUN. RUN starts the reader as for programmed RSA and RSB operations, but the absence of RD HOLE 8P V ALPHA inhibits input strobing of the RB. When released, the FEED button removes the RDR FEED ground from the logic, turning on $\overline{\text{RUN}}$ to stop the reader.

3.9.3 Tape Punch

A DEC Type PC03 Paper-Tape Punch and its control logic are standard equipment with the PDP-9. The tape punch and its companion PC02 Paper-Tape Reader are mounted together above the operator's console. Collectively, they comprise the DEC Type PC09 Paper-Tape Reader/Punch system.

The control logic for each is located in the I/O control section of the computer.

The tape punch electromechanically punches holes in 8-channel, 1-inch wide paper or Mylar tape at a maximum rate of 50 lines per second. Its mechanical operation is fully described in a separate document listed in Table 1-1. A ninth-channel mechanism punches the feed holes.

IOT instructions under program control start the punch, load a buffer register with 6-bit binary or 8-bit alphanumeric information from the AC (AR) via the I/O bus, and sense the status of the flag. The instructions select the punch for operation in either the binary or the alphanumeric mode.

Alternatively, the manual FEED button on the punch panel may be used to drive the tape for punching of nothing but feedholes. The POWER button on the punch panel should be pressed ON in either case for punch operation. The OFF position of the button is connected to special PUN NO TAPE logic. The PUN NO TAPE condition can be sensed by the IORS facility (Section 3.8.1.8) when the tape runs out or when the POWER button is OFF. PUN NO TAPE thus provides a warning when approximately 2-in. of tape remains before running out.

An IOT PSA instruction selects the punch for alphanumeric operation. Each PSA causes one line of 8-bit information to be gated into the punch buffer (PB), and synchronizing circuits in the punch control logic use the buffer contents to drive the punch mechanism. An IOT PSB instruction selects the punch for binary operation, where each PSB causes one line of 6-bit information to be gated into the PB. In the binary mode three successive lines of information comprise one 18-bit binary word. Channel 7 is never punched (except for READ IN use, Section 3.7.4.9); channel 8 is always punched to indicate binary mode for the tape reader, but the bit is ignored as an information bit. The punch flag goes up with each line punched in either mode.

All IOT instructions associated with the punch are listed below. The logic functions that they perform follow:

IOT0201	PSF	Skip on punch flag
IOT0202	PCF	Clear punch flag
IOT0204	PSA	Clear flag and select punch in alphanumeric mode
or		
IOT0214	PSA	Clear AC and flag, and select punch in alpha mode (punch feed holes only)
IOT0244	PSB	Clear flag and select punch in binary mode

3.9.3.1 Select Punch Binary - The PSB instruction (700244) is decoded in the CP's op code detection circuits for IOT(1) as usual (Section 3.5.8). On drawing KD3(1) the IOT(B) level samples the MB06-11 bits in the instruction to produce the appropriate DS00-05 levels, and IOT(1) further decodes DS00-02 to produce 0XEN. 0XEN goes to the punch control, drawing KD10(1), to sample DS03-05, resulting in PUN SEL. PUN SEL conditions a DCD gate at pulse amplifier S602-D24K. The gate now waits for the

start of the third IOT execute period, at which time the CLK pulse counter IO0 steps to 0, strobing the MB15(1) gate on drawing KD3(3) to generate IOP4P. The IOP4P pulse triggers the pulse amplifier to generate the IOT0204 command.

IOT0204 resets the PUN FLG, sets the PUN ACT flip-flop, and strobes the input DCD gates of the punch buffer flip-flops PB10-17. Since PSB is an output transfer instruction, the contents of the AC were placed in the AR, then gated onto the I/O bus and IO BUS (B) during the IOT fetch cycle (Section 3.8.1.5). For the binary mode, IO BUS 12-17 contain the first 6-bit character of the 18-bit word to be punched. These are strobed into PB12-17 by IOT0204. Additionally, bit PB11 (hole 7) must always receive a 0 and bit PB10 (hole 8) must always receive a 1 for binary mode formatting. In the PSB instruction, MB12 is 1, producing a ground SD00P on drawing KD3(1). SD00P at ground on drawing KD10(1) causes PB11 to reset and PB10 to set regardless of the I/O bus levels.

PUN ACT(1), set by IOT0204, starts the punching synchronization by triggering the 3sec PUN PWR delay R303-CD27D. On the leading edge of the 3sec delay period PUN PWR(1) issues the PUN PWR ON level to energize the punch drive motor, drawing KD10(2), and also triggers a 1sec delay R302-C25V. The 1sec delay period allows the motor that much time to accelerate to proper operating speed. Upon recovery, the inverted PUN SPD level goes negative.

PUN ACT(1) NORed at R111-D26NP enables R111-D26HJ with the recovered PUN SPD to condition a DCD gate at 10-ms delay R302-C25M. The 10-ms delay is triggered by a PUN SYNC signal coming from a reluctance pickup coil at the punch motor shaft. When the pickup coil senses that the punch motor has rotated to a pre-adjusted punch position, the PUN SYNC signal results to trigger the delay. The delay output goes negative (PUN) for 10 ms, during which time the punching operations take place.

This negative PUN delay generates PUN LINE at S107-C19J. PUN LINE enables the solenoid driver W040 gates at the outputs of all PB bits which contain 1s. The affected solenoid drivers supply the drive current to the respective bit punch mechanisms.

PUN LINE also enables two additional solenoid drivers connected in parallel. These drivers supply enough current to punch the feed hole (NDX) and to advance the tape to the next line (FWD FD).

When the 10 ms punching period lapses, PUN reverts to $\overline{\text{PUN}}$, setting the PUN FLG and re-setting the PUN ACT flip-flop. PUN FLG(1) generates a PROG INT RQ for the I/O control logic. Although PUN ACT has reset, the 3sec PUN PWR delay remains to drive the punch motor at continuous operating speed. To punch the next line another RSB follows in the service routine honoring the PROG INT RQ (Section 3.8.1.7). Succeeding PSB instructions of course retrigger the 3sec PUN PWR delay before the previous delay runs out. When the last line is punched the last delay lapses to stop the motor.

3.9.3.2 Select Punch Alpha - The PSA instruction (700204) results in the IOT0204 command as for PSB, but now MB12 is 0, producing a negative $\overline{SD00P}$ level. The negative $\overline{SD00P}$ now gates IO BUS 10 and IO BUS 11 levels into PB10 and PB11 for alphanumeric formatting.

3.9.3.3 Skip on Punch Flag - The PSF instruction (700201) senses the status of the PUN FLG. If the flag is set, the punch 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 PUN SEL on drawing KD10(1). When IO0 sets, signaling the start of the first IOT execute period, it produces IOPIP in conjunction with MB17(1). IOPIP sets IOPI.

IOPI(1) and PUN SEL sample the state of the PUN FLG on drawing KD10(1). If the PUN FLG is set, the INT SKP RQ BUS goes to ground, triggering pulse amplifier W612-F18D on drawing KD3(3). The PA pulse is the IO SKIP signal which sets the SKIP flip-flop in the CP, drawing KC14. The PSF 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.

3.9.3.4 Clear Punch Flag - The PCF instruction (700202) clears the PUN FLG. MB06-11 are detected to produce PUN SEL as usual, and when clock pulse counter IO1 sets, signaling the start of the second IOT execute period, it produces IOP2P in conjunction with MB16(1). IOP2P strobes an input DCD gate to pulse amplifier S602-D24U, preconditioned by PUN SEL. The CLR PUN pulse from the PA resets the PUN FLG.

3.9.3.5 Clear AC, Clear Flag, and Select Punch - A PSA instruction microcoded with a 1 in MB14 will clear the AC and select the punch for alphanumeric mode. This instruction (700214) can be used to punch feed holes in tape leaders and trailers, and to space the punching of information for timing purposes. MB14(1) is detected in the CP's op code detection circuits to clear the AC during IOT fetch (Section 3.8.1.5).

3.9.3.6 POWER Button and TAPE Switch - The POWER pushbutton S1 on the punch's front panel should be pressed to ON before any punch operation. Although unrelated to the application of power to the punch, the OFF position applies a ground \overline{TAPE} level, drawing KD10(2), to the PUN NO TAPE gate R111-D21UV, drawing KD10(1). The resulting PUN NO TAPE level appears as the IO BUS 09 status bit when using the read status (IORS) facility (Section 3.8.1.8). Thus the OFF position has the same

effect as the TAPE switch S3. S3 is a microswitch which is mounted just ahead of the tape punches and whose contacts are separated by the threaded paper tape. If the tape tears, the contacts close.

3.9.3.7 FEED Button - The FEED pushbutton S2 on the punch's front panel can be depressed to punch feed holes in the tape. PUN FEED from the button triggers the 3 sec delay to produce PUN PWR ON as for a programmed IOT select instruction. The eventual PUN LINE level activates the FWD FD and NDX circuits to punch the feed holes and advance the tape. All other logic is disabled.

3.9.4 Real-Time Clock

The real-time clock (RTC) is incorporated in the I/O control section as standard equipment. The RTC permits the programming of long delays with a high degree of long term stability. RTC operations simulate the first cycle (WC) of DCH operations to increment a count held in preassigned memory location 00007. The circuits involve a filament transformer driving a Shmitt Trigger W501 module at the AC line frequency for an output of 60 (or 50) pulses per second, or one pulse every 16.7 (or 20.0) ms.

Normally, the program initializes the contents of location 00007 to the 2s complement of the desired pulse count. The RTC logic must be enabled by both the CLK switch on the console and by an IOT CLON instruction. Each RTC pulse thereafter interrupts the program for one computer cycle to increment the count in location 00007. A BGN process at the end of the cycle resumes program execution, until the next RTC pulse causes another break. When the count increments to 0, a CLK FLG in the I/O control section is raised. The CLK FLG is connected to the PROG INT RQ line in the I/O Control and to the optional API system. If the PI facility alone is enabled, the PROG INT RQ will cause a program interrupt break, Section 3.8.1.7. If both the PI and API facilities are enabled, the API facility has priority and causes an API break. The status of the CLK FLG can be checked by an IOT CLSF instruction or an IOT IORS instruction, Section 3.8.1.8. The IOT instructions associated with the RTC, then, are as follows.

IOT0001	CLSF	Skip on CLK FLG
IOT0004	CLOF	CLK OFF and clear CLK FLG
IOT0044	CLON	CLK ON and clear CLK FLG

On drawing KD3(1), IOT(B) derived from IOT(1) in the CP's op code detection circuits samples MB06-11 for DS00-05. For an IOT0044, MB06-11 are all 0s, and IOT(1) gates on 00XXEN(B) as a result. Also, IOT(B) and MB12(1) result in an SD00P ground level. MB12 is a 1 as indicated by the first octal code 4 in the IOT0044 instruction.

On drawing KD3(3), IOT(B) and the IO CLK(B) pulse derived at the start of the first IOT execute period sets IO0 for a count of 10 in the I/O pulse counter. The next IO CLK(B) sets IO1 for a count of 11, and the third resets IO0 for a count of 01.

IO0 in resetting strobes the DCD gate conditioned by MB15(1) for an IOP4P pulse. MB15 is a 1 as indicated by the second octal code 4 in the IOT0044 instruction. IOP4P sets the IOP4 flip-flop. IOP4(1) and 00XXEN(B) generate the IOT0004 pulse on drawing KD3(1).

IOT0004 sets the CLK EN flip-flop in conjunction with SD00P, and resets the CLK FLG if set, drawing KD3(2). CLK EN(1) and the SW CLK level from the console CLK switch conditions the DCD set gate of the CLK RQ flip-flop. The strobe pulse at the DCD gate comes from the Shmitt Trigger W501 acting as the RTC pulse source. The RT CLK signal that triggers the W501 is the 10VAC, 60 (or 50) cps signal that comes from the transformer in the 709 power supply.

The next pulse out of W501 after CLK RQ conditioning sets the flip-flop. The W501 pulse can occur at any time up to 16.7 ms after CLK RQ is conditioned. Meanwhile the program continues.

When CLK RQ sets, it conditions the DCD set gate of the CLK SYNC flip-flop. On the next IO CLK(B) pulse in the program an IO SYNC POS pulse occurs if no IOT, RTC, PI, or API operation has been initiated. If a DCH RQ has appeared in the meantime, the DCH SYNC flip-flop sets on the IO SYNC POS pulse, and in setting, holds the CLK SYNC flip-flop in the reset state. Thus, the DCH operation has priority over the RTC operation. In any case, the CLK SYNC flip-flop will set on the first IO SYNC POS pulse permissible.

CLK SYNC(1) holds PROG SY in the reset state to bar subsequent program interrupts and generates IO ADDR15(B), 16(B), and 17(B) levels on drawing KD5. These represent address 00007, the location of the RTC's pulse count register. CLK SYNC(1) also generates INC V DCH and BK SYNC on drawing KD3(2).

BK SYNC now waits for the DONE(1) bit in the execution process of the current program instruction. DONE(1) and BK SYNC extract the BK entry process word (11) as for DCH set-up operations, Section 3.8.2.3. BK(1) set by EXT(1) of BK entry resets CLK RQ in conjunction with CLK SYNC(1).

The BK entry process (11) replaces the BGN (10) process in the current execute cycle as for DCH synchronization. Whereas the BK entry process for the DCH places the DCH device's WC address in the MB, it now places address 00007 in the MB. The process also puts 0s in the IR as for the DCH.

The next computer CLK pulse and SM(1) of BK entry extract process word 34 and start the core memory as for the DCH to simulate the WC cycle. The core memory read cycle reads the contents of location 00007 into the MB while process word 34 increments the address 00007 and places it in the AR. For RTC operations, however, incrementing the address is inconsequential.

DCH(1) in process word 34 sets BK0 as for the DCH. BK0(1) and BK1(0) produce DCH INX for CI17 as described in Section 3.8.2.4. In the pause between memory read and memory write, the word count in the MB is incremented by 1 and recirculated in the MB through the ADR, as for DCH.

When the next CM STROBE occurs to extract the next process word, it samples the ADRA=0, ADRB=0 bus to test for overflow as for DCH operations. If the pulse count in the MB has incremented

to 0, then CM STROBE, ADR=0, etc., produce OFLO on drawing KC14. OFLO goes through the CP/IO interface to set the IO OFLO flip-flop, drawing KD3(2). IO OFLO(1) sets the CLK FLG in conjunction with CLK SYNC(1). CLK FLG(1) applies a PROG INT RQ level to the PROG SY flip-flop to initiate a program interrupt on the next IO SYNC POS if the PI facility is enabled and the API is disabled or not installed.

Process word 34 contains CMA10. For DCH operations, this address is boosted to 14 by DCH(1), INC MB, etc., on drawing KC17. For RTC operations, the address remains the same because $\overline{\text{INC MB}}$ is now at an inhibiting ground level. The ground level INC MB results from CLK SYNC(1) on drawing KD3(3).

Process word 10 (BGN) is therefore extracted on the CM STROBE to set up the MB for the resumption of the program. Also on this CM STROBE, INC MB and BK0(1) cause the DONE process bit to set, drawing KC19(1). DONE(1) B resets CLK SYNC. During the BGN period, the core memory write half-cycle is writing the incremented pulse count back into location 00007. On the next computer CLK pulse and SM(1) of BGN, the program resumes and BK0 resets.

If the pulse count has not incremented to 0, another W501 pulse occurring 16.7 ms later (minus the time expired) will again set the CLK RQ flip-flop to repeat the RTC operation. RTC interrupts will thus occur until resulting in an overflow or until the CLK EN flip-flop is reset by an IOT CLOF instruction. The CLOF instruction (IOT0004) resets CLK EN via the same logic used for CLON above. In this case, SD00 conditions the DCD reset gate. It also resets the CLK FLG.

The IOT CLSF instruction may be used to test the status of the CLK FLG. On drawing KD3(1), the IOT0001 instruction is sampled to produce 00XXEN(B) as for IOT0044 and IOT0004, but now MB17 is a 1, generating IOP1P on drawing KD3(3) when IO0 sets. IOP1P sets IOP1, and IOP1(1) samples the CLK FLG status with 00XXEN(B). If the CLK FLG is set, a ground INT SKP RQ BUS level results. INT SKP RQ BUS generates IOSKIP on drawing KD3(3). IOSKIP sets the SKIP flip-flop in the CP, drawing KC14. Upon completion of the third IOT execute period SKIP(1) and the BGN process word increments the contents in the PC as they are dumped in the MB. Thus, the program skips the next sequential instruction as explained in Section 3.8.1.6.

CHAPTER 4 MAINTENANCE

4.1 MAINTENANCE PHILOSOPHY

Basically, PDP-9 maintenance is directed to the module-replacement level. Down-time 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 occasional marginal checking to expose weakening conditions before they become malfunctions.

When a malfunction occurs, corrective maintenance is instituted to isolate the problem and to make proper adjustments or replacements. Primarily, this involves the use of diagnostic routines prepared on paper tape and designed to test the functional units of the system. Categorizing these primary areas does not imply complete independence. The procedures and techniques of periodic checking can aid and are indeed necessary in malfunction tracing; conversely, intermittent error conditions occurring during system operations can be caused to occur continuously by simulating marginal power conditions.

4.2 TOOLS AND TEST EQUIPMENT

Maintenance activities for the PDP-9 system require the standard test equipment and special materials listed in Table 4-1, plus standard hand tools, cleaners, test cables and probes.

Table 4-1
Tools and Test Equipment

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

Table 4-1 (cont)
Tools and Test Equipment

Equipment	Manufacturer	Designation
Hand-Operated Wire-Wrap Tool with a 26263 bit for 24 AWG Wire and 18840 Sleeve	Gardner-Denver	14H1C
Module Extender*	DEC	Type W980
Paint Spray Can*	DEC	DEC Black
Paint Spray Can*	DEC	DEC Red
Air Filter*	Research Products Corp.	EZ Clean 2-inch Type MV
Filter-Kote*	Research Products Corp.	By Name

*One supplied with the equipment.

4.3 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.

4.4 MARGINAL CHECK SWITCHES

In addition to the marginal check panel controls (Table 2-2 and Figure 2-2), maintenance tasks make use of marginal check (MC) switches mounted on the fan housings in each computer wing, Figure 4-1. These switches 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.

As shown in Figure 4-1, twelve sets of three switches each are designated 01-12. Each set applies the selected voltages to specific modules in the wing. Drawings KC9, MC17, and KD13 are the MC Configuration drawings for the CP, core memory, and I/O control wings respectively. The drawings designate by number the sets of MC switches which apply their voltages to the specific modules.

Of the three switches in each set, the FIXED/MC switch selects either the operating voltages or the marginal check voltages from the marginal check panel and feeds them to the +10 and -15 switches. The positions of the two switches thus labeled then route the voltages to the specific modules. The unmarked (off) positions of the switches remove all voltages from the specific modules.

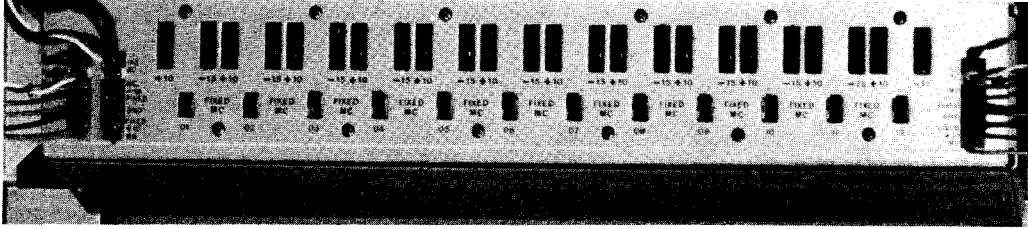


Figure 4-1 Marginal Check Switches

4.5 PREVENTIVE MAINTENANCE

Preventive maintenance consists of tasks performed periodically to ensure satisfactory operation. Performance of these tasks forestalls failures caused by progressive deterioration or minor damage. Data obtained from each task should be recorded in a log book. Analysis of this data indicates the rate of circuit deterioration and provides information for determining when components should be replaced to prevent failure of the system.

Preventive maintenance tasks consist of mechanical checks, i.e. cleaning and visual inspections; marginal checks, which aggravate border-line circuit conditions or intermittent failures for detection and/or correction; and checks of specific circuit elements such as power supplies, sense amplifiers, master slice control, and memory selectors. All maintenance schedules should be established by conditions at the installation site. The most important schedule is that of mechanical checks, which should be performed monthly or as often as required to allow efficient functioning of the air filters, for example, to avoid machine failures caused by overheating due to dirty filters. All other tasks should be performed on a regular schedule determined by the reliability requirements of the system. A typical recommended schedule is every 600 operating hours or every four months, whichever comes first.

4.5.1 Mechanical Checks

- a. Clean the exterior and the interior of the equipment cabinet using a vacuum cleaner or clean cloths moistened in nonflammable solvent.
- b. Remove and clean the air filters in each section of the computer. The filters slide out of the fan housings. Wash filters in soapy water and dry in an oven or by spraying with compressed gas. Spray each filter with Filter-Kote (Research Products Corporation, Madison, Wisconsin).
- c. Lubricate door hinges and casters with a light machine oil, wiping off excess oil.
- d. Repaint any scratched or corroded areas with the supplied paint spray can.

- e. Inspect all wiring and cables for cuts, breaks, fraying, wear, deterioration, kinks, strain, and mechanical security. Tape, solder, or replace any defective wiring or cable covering.
- f. 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.
- g. Inspect all module mounting panels to assure that each module is securely seated in its connector.
- h. Inspect power supply capacitors for leaks, bulges, or discoloration. Replace any capacitors giving these signs of malfunction.

4.5.2 Power Supply Checks

Perform the power-supply output checks in Table 4-2. 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 -15V 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 4-2
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	-12.5 to 15.0	22.0	0.7
Black (+) to Yellow (-)	-30	-28.5 to 35.0	10.0	0.9
Black (+) to Brown (-)	-30	-28.5 to 35.0	10.0	0.9

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 -20V 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, $\pm 1V$ (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 +20V. Record as above. Turn the control knob fully counterclockwise, set the selector switch to the OFF position, and disconnect the multimeter.

4.5.3 Core Memory Current Check

Measure the read/write currents in the core memory. These currents should equal the values specified on the memory array label (approximately 400 mA). This label indicates the optimum memory setting determined at the factory. Whenever possible this check should be performed at an ambient temperature of 25°C. Compensate measured read/write by subtracting 1 mA for every degree of ambient temperature above 25°C. Add 1 mA for each degree below 25°C. The memory current check and sense amplifier check that follows must not be performed when the equipment temperature is below 20°C.

Measure the read/write current using the oscilloscope and clip-on current probe at the read side of a fully selected drive line of the X or Y axis G219 Address Selector (refer to the G219 module schematic). Synchronize the oscilloscope with the appropriate negative transition of the DIGIT READ or WORD READ level. Set the read/write current to 400 mA or to the value specified on the memory array label by adjusting R12 in the G804 Control Module. Note that read/write currents are measured from baseline to peak, rather than peak to peak.

4.5.4 Sense Amplifier Check

The G009 Sense Amplifier modules are adjusted using marginal checking techniques. Perform the marginal checks using the Memory Checkerboard Program, 9A-D1AA-PH. Check and adjust each sense amplifier circuit so that approximately equal margins below and above +10V provide proper sense amplifier operation.

4.6 CORRECTIVE MAINTENANCE

Should a malfunction occur, the condition should be analyzed and corrected as indicated in the following procedures. No test equipment nor special tools are required for corrective maintenance other than a broad bandwidth 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 error by locating the fault to within a module through use of diagnostic routines, control panel troubleshooting, signal tracing, or aggravation techniques.
- c. Replace defective module or modules to get the system on-line.
- d. Log entry to record pertinent data.

Circuit troubleshooting to locate defective parts within a module and repairs to replace or correct the cause of the circuit malfunction can proceed after the system is again operable. Repaired modules should be subjected to validation tests to assure that the fault has been corrected.

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.

4.6.1 Marginal Checking with Diagnostic Programs

Diagnostic routines are programs designed to exercise or test specific functions within the computer system, and are available as perforated paper tapes in read-in mode format. A detailed description of the program contained on tape, procedures for using the program, and information on analyzing the program printout accompanies each tape (Table 1-3). These programs isolate the problem to a major functional unit but not to the individual module level. However, examination of the printout, observation of panel indicators, and knowledge of the contents of the logic diagrams will allow maintenance personnel to establish the particular fault.

The following listings briefly summarize the functions of each basic routine for testing the functions of the PDP-9 system. Other test routines are available for optional equipment. The programs are listed in the recommended execution sequence.

a. Memory Address Test - 9A-DOCA-PH

The Memory Address Test checks the memory system to ensure that all memory locations not occupied by the program in a given 8K 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 -9A-D1AA-PH

The Basic Memory Checkerboard Test checks and verifies the operational status of core memory by testing its ability to detect a 1 or 0 under maximum half-select noise conditions. The program is a means for a quick check of an 8K memory field, when the need arises. The test may be performed on a PDP-9 with 8,192 to 32,768 words of core memory. The program tests only the 8K (K = 1024) memory bank in which it is located. It is suggested that the more thorough Extended Memory Checkerboard Test be used on a PDP-9 equipped with the extended memory option.

c. Extended Memory Checkerboard Test - 9A-D1BA-PH

The Extended Memory Checkerboard Test checks and verifies the operational status of core memory by testing its ability to detect a 1 or 0 under maximum half-select noise conditions. The amount of core memory to be tested must be set into DATA Switches 14-17 by the operator. The program will test any memory configuration from 8K to 32K words, in 8K segments. Print-outs are provided for data error information.

d. Instruction Test Part I - 9A-DO1A-PH

Instruction Test Part I tests the PDP-9 OPR Instructions, singly and combined, and the memory reference instructions LAC, AND, SAD, TAD, XOR and CAL. Individual error halts are used to describe the failing tests.

e. Instruction Test Part II - 9A-DO2A-PH

Instruction Test Part II is a continuation of Part I, testing the following instructions and machine functions: DZM, DAC, ISZ, JMP, JMS, XCT, AUTOINDEX, INDIRECT ADDRESS, REAL TIME CLOCK, DBR and Program Interrupt. Individual error halts are used to describe the failing tests.

f. High Speed Reader Test - 9A-D2CB-PH

The High Speed Reader Test checks and verifies the operational status of the reader by testing the associated control logic and the reader mechanics. The program consists of two parts. Part one is a test tape generator which punches the test tapes used for part two, if needed. Part two is divided into five sections. The first section is a series of tests of the readers' IOT instructions. Section two tests the acceleration time, motor delay timing and the response of the control logic with no tape in the reader. Section three tests the reader's ability to read data from tape correctly using all control IOTs. Section four is a variable reader speed test, in which the operator varies the reader's speed with the DATA switches. Section five reads the tape with random block lengths and stalls between frames.

g. Punch Test - 9A-D2DB-PH

The Punch Test checks and verifies the operational status of the punch control logic, and the mechanical functions of the punch. A series of six tests are performed on the punch control, followed by nine tests on the punch itself. Provision is made to continuously loop in two of the six punch control tests, and any one of the punch data tests.

h. TTY Test - 9A-D2BA-PH

The TTY Test verifies the operational status of a KSR-33 or KSR-35 teletype and associated control logic.

The program consists of two separate parts. Part one tests the teleprinter control logic, the complete character set, carriage return, line feed, space, right hand margins, and a mechanical worst case. The mechanical worst case test provides a pattern for either a model 33 or 35 KSR. The appropriate pattern is selected by the operator with a DATA switch.

Part two tests the keyboard input control logic, character input/output, and ability to interrupt the teleprinter from the keyboard without losing the input character.

i. JMP Self Test - 9A-DODB-PH

JMP-Self Test checks the PDP-9 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.

j. JMP-Y Interrupt Test - 9A-DOEA-PH

The JMP-Y Interrupt Test determines if the PDP-9 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.

k. JMS-Y Interrupt Test - 9A-DOFA-PH

The JMS-Y Interrupt Test determines if the PDP-9 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.

l. ISZ Test - 9A-DOBA-PH

The ISZ test checks the operation of the ISZ instruction. Various checks of the ISZ instruction are made, including ISZ of 777777g to 0g 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.

m. Basic Exerciser - 9A-D7AB-PH

The Basic Exerciser exercises the CP, core memory and I/O devices associated with a basic PDP-9 configuration. Once initiated, the program performs tests on all OPR and memory reference instructions; tests on the adder; memory checkerboard patterns; tests on the real-time clock, punch, reader, teletype and program interrupt.

The Basic Exerciser contains a condensed version of the Instruction Test parts I and II, and a memory checkerboard test similar to the Basic Memory Checkerboard Test. These tests run continuously, and are interrupted by the punch, reader or teletype at a device rate. The real-time clock interrupts and suspends all operations at random time intervals. The instruction test or I/O device resumes operation after the clock interrupt has been serviced.

Nine DATA switch functions are provided to enable the operator to (1) inhibit the instruction and memory tests and run the real-time clock, program interrupt, and the punch, read, print sequence alone; (2) inhibit program interrupts and run the instruction and memory tests alone; (3) loop continuously in the adder test; (4) loop continuously in the memory checkerboard test; (5) inhibit program relocation; (6) inhibit the real-time clock, but continue testing with program interrupt and all other devices enabled; (7) run the instruction and memory tests, and the clock and punch with the read and print sequence inhibited; (8) run the reader, real-time clock and instruction and memory tests with the punch and teletype inhibited; (9) run the read and print sequence, real-time clock and instruction and memory tests with the punch inhibited.

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

The biased operating voltages at which circuits fail should be recorded in the maintenance log. 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.

Raising the operating voltages above +10V increases the transistor cutoff bias that the previous driving transistor must overcome; therefore low-gain transistors fail. Lowering the bias voltage below +10V reduces transistor base bias and noise rejection, thus providing a test to detect high-leakage transistors. Lowering this voltage also simulates high-temperature conditions (to check for thermal run-away). Raising and lowering the -15V supply increases and decreases the primary collector supply voltage for all modules and so affects output signal voltage.

Since the marginal voltages attainable vary for different circuit changes and/or system configurations, determine the expected marginal check voltages for a specific system from the initial factory tests 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.

CAUTION

Do not increase the -15V margin beyond $\pm 5V$. Failure to observe this precaution may cause serious damage to the logic elements.

Marginal check voltages are supplied to the various sections of the processor through connections made to the module connectors in 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. 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 consequently the mounting panels are color-coded as shown in drawing IC-9-0-1.

Marginal check and normal supply voltages are distributed to each module through the marginal check switches, Figure 4-1. There are two positions for the first switch: FIXED and MC. Therefore, specific modules may be marginally checked while all others maintain fixed voltage or no voltage through the positions of +10, -15 switches. In each set the second switch controls the +10V supply and the third controls the -15V supply.

To perform the checks:

- a. Assure that all marginal check switches are in the FIXED, +10, -15 positions (normal +10V and -15V power supplies are being used).
- b. Set the selector switch on the marginal check panel to the +10MC position.
- c. Adjust the output of the marginal check power supply so that the marginal check voltmeter indicates 0V.
- d. Refer to drawings KC9, MC17, and KD13, and set the appropriate FIXED/MC switches for the module(s) to be checked to the MC positions.
- e. Start computer operation in a diagnostic program or routine which fully utilizes the circuits in the modules to be tested.
- f. 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.
- g. 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.
- h. Return the FIXED/MC switches to the FIXED positions.
- i. Repeat steps d through h for other modules to be checked.
- j. Set the selector switch on the marginal check panel to the -15MC position and adjust the output until the marginal check voltmeter indicates 0V.
- k. Refer to drawings KC9, MC17, and KD13, and set the appropriate FIXED/MC switches for the modules to be checked to the MC positions.

- l. Repeat steps e through g. Return the FIXED/MC switches to the FIXED positions.
- m. Repeat steps j through l for each test.
- n. Set the marginal check panel selector switch to the OFF position.

4.6.2 System Troubleshooting

Begin troubleshooting by performing the operation in which the malfunction was initially observed, using the same program, and thoroughly check the program for proper control settings. Assure that the PDP-9, 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.

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. Proceed to the memory troubleshooting or CP troubleshooting procedures. 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 fault is intermittent, a form of aggravation test should be employed to locate the source of the fault.

4.6.2.1 Memory Troubleshooting - If the entire memory system fails, use a multimeter to check the outputs of the 709 power supplies. If the supply is defective, troubleshoot it and correct the cause of the trouble; then check the memory current and sense amplifier adjustments, Sections 4.5.3 and 4.5.4.

Refer to Section 3.6.4 and Figures 3-23, 3-24. Note that a core address is selected by a combination of drive and sink G219 switches. READ or WRITE levels gate all G219 selectors which generate and distribute the actual read/write current to specific cores. In each axis, selection of the two switches is accomplished by the bit configuration in the MA register. The actual read/write current pulses flow from the drive switch through a core matrix line, to the sink switch.

For a selected core a current spike can be observed on an oscilloscope, and a missing spike then represents a malfunctioning address. Before loading a Memory Address Test to find a specific address malfunction, check the read/write gating levels at every G219 module. A G219 module cannot select without the gating level. If the read/write currents are not as specified on the memory array labels, adjust G804 Control Module current according to Section 4.5.3.

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 G219 selectors that decode common bits of the failing addresses. Also check the associated resistor board and memory matrix module.

If an address is dropping bits, use the console DEPOSIT key and ADDRESS switches to simulate an address of all binary 1s. Then examine the appropriate G219 outputs to determine which bit position is not being set. Check the sense amplifier and resistor board for the associated bit.

If an address is picking up bits, simulate an all 0 address and proceed as above.

To locate the cause of a specific address failure, use an oscilloscope and current probe to trace read and write current while performing the Memory Address Test repetitively. Perform the Memory Checkerboard Test to troubleshoot all other memory conditions.

4.6.2.2 Logic Troubleshooting - If the instructions do not seem to be functioning properly, perform the Instruction Test, 9A-DO1A-PH. This test halts to indicate instructions that fail. When an instruction fails, as indicated by the console indicators when the program stops, or by the diagnostic printout that follows the error halt, consult the descriptive listing for 9A-DO1A-PH to obtain an interpretation that will localize the fault.

If the computer program interrupt system or the teletype teleprinter do not seem to be functioning properly, perform the TTY Test Program (9A-D2BA-PH). If the tape reader or punch operation is questionable, perform the Reader and Punch Test (9A-D2CB-PH) and 9A-D2DB-PH.

Refer to the appropriate documents (Table I-1) for detailed maintenance information on the teletype, tape reader, and tape punch.

4.6.2.3 Signal Tracing - 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. If an intermittent malfunction occurs, signal tracing must be combined with an appropriate form of aggravation test.

4.6.2.4 Aggravation Tests - Intermittent faults should be traced through aggravation techniques. Intermittent logic malfunctions are located by the performance of marginal check procedures as described in Section 4.6.1.

Intermittent failures caused by poor wiring connections can often be revealed by shaking modules while running a repetitive test program. Tapping a wooden rod held against the handles of a suspected panel of modules is a useful technique. By repeatedly starting the test program and shaking fewer and fewer modules, the malfunction can be localized to within one or two modules. After isolating the malfunction in this manner, check the security of the modules in the connector; check the module connector for wear or misalignment, and check the module wiring for cold solder joints or wiring kinks.

4.6.3 Circuit Troubleshooting

Basic functions and specifications for standard system modules used in the PDP-9 are presented in the Logic Handbook, C-105. Circuit schematics are included in Chapter 5. The following design considerations may also be helpful in troubleshooting standard modules:

a. Forward-biased silicon diodes are used in the same manner as Zener diodes, usually to provide a voltage differential of 0.75V. For instance, a series string of four diodes produces the -3 Vdc clamp voltage used in most modules.

b. An incoming pulse which turns off the conducting transistor amplifier changes the state of DEC flip-flops. Since these flip-flops use PNP transistors, the input pulse must be positive and must be coupled to the base of the transistor. Flip-flop modules that accept negative pulses to change the state invert this pulse by means of a normal transistor inverter circuit.

c. Fixed-length delay lines are extremely reliable and very seldom malfunction. However, if malfunctions occur, these delay lines should not be replaced on the printed-wiring board. In such cases return the entire module to DEC for repair.

4.6.3.1 In-Line Dynamic Tests - To troubleshoot a module while maintaining its connection within the system:

- a. De-energize the computer.
- b. Remove the suspected module from the mounting panel.
- c. Insert a W980 Module Extender into the mounting panel connector holding the suspected module.
- d. Insert the suspected module into the module extender. This makes all components and wiring points of the module accessible to test probes.
- e. Energize the computer and establish the program conditions desired for troubleshooting the module. Trace voltages or signals through the module, using a dc voltmeter or an oscilloscope.

4.6.3.2 In-Line Marginal Checks - The normal marginal checking method can test specific modules of questionable reliability, or further localize the cause of an intermittent failure which has been

localized to within one module row. The following checks are performed with the aid of a modified W980 Module Extender to marginal check an individual module. To modify an extender for these checks:

a. Disconnect module receptacle terminals A, B, and C from the male connecting terminals. This can be accomplished by cutting the printed wiring for these lines near the plug end and removing a segment of this wiring in each line.

b. Solder a 3-ft test lead to the printed wiring for terminals A, B, and C. Make this solder joint close to the receptacle end of the extender, on the receptacle side of the wiring break. Observe the normal precautions when making this connection to assure that excessive heat does not delaminate the printed-wiring board and that neither solder nor flux jumps conduction between lines.

c. Attach a spade lug, such as an AMP 42025-1 Power Connector, to the end of each test lead and label each lead to correspond to the A, B, or C terminal of the receptacle to which it is connected.

To marginal check a module within the computer:

a. De-energize the computer.

b. Remove the module to be checked from the module mounting panel; replace it with the modified extender, and insert the module in the extender.

c. If the +10V marginal check is to be performed, connect test lead A to the +10V (MC) orange connector terminal at the fan housing. Connect test lead B to the fixed -15V blue connector terminal and test lead C to black ground connector, keeping all MC switches in the FIXED, +10, -15, positions.

d. When performing the -15V marginal check, connect test lead A to the fixed +10V red connector, test lead B to the -15V (MC) green connector terminal, and test lead C to the black ground terminal, keeping all MC switches in the FIXED, +10, -15 positions.

e. Restore computer power, adjust the marginal check power supply to provide the nominal voltage output, and start operation of a routine which fully utilizes the module being checked. The procedures and routines suggested in Section 4.6.1 can be used as a guide to marginal checking modules.

f. Increase or decrease the output of the marginal check power supply until the routine stops, indicating module failure, and record each bias voltage at which the module fails. Also record the conditions of all console controls and indicators when a failure occurs. This information indicates the module input conditions at the time of the failure and aids in tracing the cause of the fault to a particular component part.

g. Repeat steps e and f for each bias voltage. If margins of $\pm 5V$ on the +10 Vdc supplies can be obtained, and the -15 Vdc supply can be adjusted between -7V and -18V without module failure, it is assumed a module is operating satisfactorily. If the module fails before these margins are obtained, use normal signal tracing techniques within the module to locate the source of the fault.

4.6.3.3 Static Bench Tests - Visually inspect the module on both the component and printed-wiring sides for short circuits in the etched wiring and for damaged components. If this inspection fails to confirm or reveal a fault, measure resistances with a multimeter.

CAUTION

Do not use the lowest or highest resistance ranges of the multimeter when checking semiconductor devices. The X10 range is suggested. Failure to heed this warning may result in damage to components.

Measure the emitter-collector, collector-base, and emitter-base resistances of transistors in both directions. A good transistor indicates an open circuit in both directions between collector and emitter. Normally, 50 to 100 ohms exist between the emitter and the base or between the collector and the base in the forward direction, and open-circuit conditions exist in the reverse direction. To determine forward and reverse directions, consider a transistor as two diodes connected back-to-back. In this analogy PNP transistors are considered to have both cathodes connected together to form the base, and both the emitter and collector assume the function of an anode. In NPN transistors the base is assumed to be a common anode connection, and both the emitter and collector are assumed to be the cathode.

Multimeter polarity must be checked before measuring resistances, since many meters (including the Triplet 630) apply a positive voltage to the common lead when in the resistance mode. Note that although incorrect resistance readings are a sure indication that a transistor is defective, correct readings give no guarantee that the transistor is functioning properly. A more reliable indication of diode or transistor malfunction is obtained by using one of the many inexpensive in-circuit testers commercially available.

Damaged or cold-solder connections can also be located using the multimeter. Set the multimeter to the lowest resistance range and connect it across the suspected connection. Poke the wires or components around the connection, or alternately rap the module lightly on a wooden surface, and observe the multimeter for open-circuit indications. Often the response time of the multimeter is too slow to detect the rapid transients produced by intermittent connections. Current interruptions of very short duration, caused by an intermittent connection, can be detected by connecting a 1.5V flashlight battery in series with a 1500-ohm resistor across the suspected connection. Observe the voltage across the 1500-ohm resistor with an oscilloscope, while probing the connection.

4.6.3.4 Dynamic Bench Tests - In general, return to DEC for repair or replacement any module which fails marginal in-line tests, or which is considered faulty for other reasons. Many modules require special equipment for dynamic testing, since the timing of pulse amplifiers and delay modules must be rigidly maintained within narrow limits. Dynamic tests, therefore, should be oriented only toward discovery of defective semiconductors. Dynamic tests may be carried out by means of a Type H901 Patchcord Mounting Panel connected to the computer power supply outputs by means of Type 914 Power

Jumpers. Simulated ground-level signals may then be applied to the module under test, using Type 911 Patchcords; an oscilloscope connected to terminals on the front of the Type H901 panel can monitor output terminals of the module under test.

4.6.4 Tape Reader Adjustments

4.6.4.1 Solenoid Drivers - Check the tape reader's solenoid driver modules W040, drawing PC09-0-2, periodically as follows:

- a. Connect oscilloscope lead to pin R of the first module.
- b. Depress the reader FEED button.
- c. While reader synchro motor is operating, the module output should change from 0 to -30V.
- d. Connect the scope lead to pin S of the module and check for the same output change.
- e. Repeat the checks for the three remaining modules.

4.6.4.2 Photo Amplifier - Adjust the trimpot on photo amplifier G904, drawing PC09-0-2, periodically as follows:

- a. Remove tape from the reader and place the tape hold-down bar in its down position.
- b. Set the REGISTER DISPLAY switch on the console to the STATUS position and depress the PROGRAM STOP and I/O RESET keys. The STATUS bit 08 in the REGISTER indicator should be illuminated, indicating a reader no tape status.
- c. Using the DEPOSIT key, ADDRESS and DATA switches, enter the following reader clock program into memory:

```
200/          LAW 16377
              DAC 300
              RSA
              RSF
              JMP .-1
              ISZ 300
              JMP .-1
              JMP 200
```

- d. Depress I/O RESET and START. The reader sprocket wheel should rotate freely.
- e. If sprocket wheel does not rotate freely, depress PROGRAM STOP. Turn the trimpot on G904 ccw a few turns, depress I/O RESET and START. Repeat this step until the reader sprocket turns freely under program control.
- f. With the program running and the sprocket wheel turning, adjust the pot on the G904 until the sprocket wheel stops.
- g. Turn the pot an additional half-turn cw. Depress PROGRAM STOP and observe the STATUS bit 08 in the REGISTER indicator. The indicator bit should be illuminated to indicate a no tape status.

- h. Thread a tape of alternate 1s and 0s in the reader.
- i. Connect the scope leads to any information channel at W023-A1. Calibrate the scope for 1 cycle = 10 CM.
- j. Depress I/O RESET and START. The observed waveform should be a very nearly symmetrical square wave.

4.6.4.3 Clock and Accelerator - Check and adjust the reader clock and accelerator modules whenever failing components necessitate removal or replacement of one of the interacting parts. Store the reader clock program as in Section 4.6.4.2.c and proceed as follows:

- a. Connect the scope lead to the RDR CLK output R401-E03D, drawing KD9(1).
- b. Calibrate the scope for 1 MS/CM, and turn the scope intensity up to observe the low-repetition-rate output pulse.
- c. Depress I/O RESET and START. The time interval between the fall of the first clock pulse and the rise of the second should be 5.6 ms. Adjust R401-E03 for the proper interval. It is necessary to start and stop the program a number of times to obtain an accurate adjustment since the G903 Accelerator will decrease the interval after 10 clock pulses. Use I/O RESET-START and PROGRAM STOP alternately.
- d. With the reader running at full speed, the interval should decrease to 1.66 ms. Adjust the accelerator G903-E04 for the proper interval.

4.6.4.4 Mechanical Adjustments - Refer to the tape reader maintenance manual listed in Table 1-1 for mechanical adjustments.

4.6.5 Module Repair

Module repair should be limited to the replacement of semiconductors. In all soldering and unsoldering operations, avoid placing excessive solder or flux on adjacent parts or service lines. When soldering semiconductor devices (transistors, crystal diodes, and metallic rectifiers) which may be damaged by heat, the following special precautions should be taken.

- a. Use a heat sink, such as a pair of pliers, to grip the lead between the device and the joint being soldered.
- b. Use a 6V soldering iron with an isolation transformer. Use the smallest soldering iron adequate for the work.
- c. Perform the soldering operation in the shortest possible time, to prevent damage to the component and delamination of the module etched wiring.

4.7 REPLACEMENT MODULES

DEC offers an optional spare modules kit (SP09A) containing one spare of each infrequent-use module in the PDP-9 plus two each of the widely used B131, B169, B213, G219, R111, S202, and W612 modules.

APPENDIX A
OPTION LIST

DEC offers a complete line of options for use with the basic PDP-9. Descriptions of particular options shipped per customer order are contained in separate option manuals. For future reference and design planning, all available options are listed here.

A1 MEMORY EXPANSION AND MEMORY OPTIONS

Memory Extension Control, Type KG09B*

Permits expansion of memory above 8,192 words.

8,192-Word Core Memory Module, Type MM09A/B/C

For expansion of memory in 8,192-word modules above the original 8,192 words. Requires Memory Extension Control, Type KG09B; up to three (3) Type MM09 modules may be added to a system, to bring memory size up to 32,768 words.

Memory Parity Option, Type MP09*

Adds 19th bit plane to PDP-9 memory stack and provides a parity check on all transfers to and from memory.

Type MP09A, first 8K stack

Type MP09B, each additional 8K stack

A2 CENTRAL PROCESSOR OPTIONS

Extended Arithmetic Element, Type KE09A*

Adds microcoded instructions including multiply, divide, normalize and long shifts to basic PDP-9.

Automatic Priority Interrupt, Type KF09A

Provides up to 32 unique channels and 4 hardware priority levels for nested interrupts and quick recognition of peripheral service requests. Also includes 4 software priority levels.

*These options are housed in the basic PDP-9 cabinet.

Power Failure Detection Option, Type KP09A*

Senses power line failures and provides interrupt in time to allow program to store all active registers. Provided with automatic restart.

Memory Protection Option, Type KX09A*

A3 INPUT/OUTPUT OPTIONS

Direct Memory Access Channel Adapter Multiplexer, Type DM09A

Provides second memory buffer register and multiplexed priority channels for up to three devices which use the direct memory access channel.

Input/Output Bus Adapter, Type DA09A

Provides interface to PDP-9 I/O Bus for devices designed to interface to PDP-7 Information Collector.

PDP-9/PDP-9 Interprocessor Buffer, Type DB99A

Controls and buffers the flow of information between two PDP-9s, using the data channel facilities of both central processors. Transfer rates of up to 250,000 18-bit words/second can be achieved without program intervention.

PDP-9/PDP-8 Interprocessor Buffer, Type DB98A

Controls and buffers the flow of information between one PDP-9 and one PDP-8, using the data channel facilities of both processors. Transfer rates in excess of 200,000 12-bit words/second can be achieved without program intervention.

PDP-9/PDP-7 Interprocessor Buffer, Type DB97A

Controls and buffers the flow of information between one PDP-9 and one PDP-7, using programmed transfers.

18-Bit Output Relay Buffer, Type DR09A

Buffer and 18 spdt relays actuated by computer command.

*These options are housed in the basic PDP-9 cabinet.

A4 MASS STORAGE DEVICES

DECtape Control, Type TC09

Provides control for up to eight DECtape Transports, Type TU55; Utilizes data channel facility for direct transfer to or from memory of DECtape information.

DECtape Transport, Type TU55

Single DECtape transports capable of holding 3-1/2 in. reels of mylar tape. Tape runs at 80 ips, holds up to 3,000,000 bits of information, and transfers at 15 kc character rate. Requires control, Type TC09.

Automatic Magnetic Tape Control, Type TC59

Controls up to eight IBM-compatible magnetic tape transports automatically. Buffers flow of information to or from computer via data channel facility. Can read or write in either BCD or binary mode.

Magnetic Tape Transport, Type TU20

• Reads and writes IBM-compatible tape at 45 ips and 200/556/800 bpi. Requires control, Type TC59.

Block Transfer Drum, Type RM09

Provides block transfers to and from drum via the direct memory access channel. Data transfers every 67.2 μ s, with 17.2 ms required for each 256-word block. Average access time is 8.65 ms. Includes buffers and controls, but requires DMA Multiplexer, Type DM09A.

Type RM09A, 32,768 words

Type RM09D 262,144 words

Type RM09B, 65,536 words

Type RM09E 524,288 words

Type RM09C 131,072 words

A5 PRINTERS AND PLOTTERS

Automatic Line Printer and Control, Type 647

Prints 64 characters, 120 columns per line. Includes single line buffer. Also available at extra cost with up to 160 columns, and at 1000 lines per minute.

Type 647A, 300 lpm

Type 647B, 600 lpm

Incremental Plotter and Control, Type 350

Draws and labels graphs, charts, and plots at high speed. Uses California Computer Products (Calcomp) or equivalent incremental recorder.

<u>With Calcomp Model</u>	<u>Step Size (Inches)</u>	<u>Speed (Steps/ Minute)</u>	<u>Paper Width (Inches)</u>
563	0.01	12,000	31
	0.005	18,000	31
565	0.01	18,000	12
	0.005	18,000	12

A6 DATA COMMUNICATIONS EQUIPMENT

Multi-Station Teletype Control, Type LT09A

Provides control for up to five Teletype stations.

Data Communication System, Type 680

Provides stand-alone data communications system that can be interfaced to PDP-9 via Inter-processor Buffer, Type DB98A.

A7 CARD INPUT EQUIPMENT

100 cpm Card Reader and Control, Type CR01E

Reads standard 80-column cards from a 430-card capacity bin at 100 cards per minute.

200 cpm Card Reader and Control, Type CR02B

Reads standard 80-column cards at 200 cards per minute in either alphanumeric or binary modes.

A8 CATHODE RAY TUBE DISPLAY OPTIONS

Precision CRT Display, Type 30D

Plots point-by-point data on a 16-in. cathode ray tube. Includes separately variable 10-bit X and Y coordinates and programmable intensity control.

Symbol Generator, Type 33

Plots symbols on a 5 x 7 dot matrix in one of four sizes on Type 30D Display. Average plotting time is 140 μ s.

Oscilloscope Display, Type 34H*

Plots point-by-point data on X and Y plotting scope such as Tektronix RM503 or Hewlett-Packard equivalent. Uses 10 bits per axis, has intensity control.

Programmed Buffered Display, Type 339

Includes DMA Multiplexer, Type DM09A; Light Pen, Type 370; and push-button control box. Character Generator, Type VC38

Provides programmable, 128-character set with average display time of 25 μ s per character.

Search Logic, Type VF38

Zoom Logic, Type VZ38

Slave Mode, Type VS38

Precision Incremental CRT Display, Type 340C

Plots points, lines, vectors, and characters on a 9-3/8 in. square raster with 1024 points per axis. Operates on cycle-stealing basis via DMA channel. Contains subroutine interface, Type 347, as standard equipment for manipulation of nonconsecutive display tables. Requires DMA multiplexer, Type DM09A to interface to PDP-9.

Character Generator, Type 342

Plots standard ASCII-code characters on a 5 x 7 dot matrix in one of four sizes on Type 340C Display. Average plotting time is 35 μ s per character. Two character sets are available.

Slave Display, Type 343

Slave display to Type 30D or Type 340 Displays. Includes 25 ft of remote cable.

High Speed Light Pen, Type 370

Uses fiber optic light pipe and photo-multiplier system for fast detection of displayed information.

*These options are housed in the basic PDP-9 cabinet.

A9 A-D AND D-A OPTIONS

General Purpose Analog-To-Digital Converter, Type 138E

Used to convert input analog voltages into digital numbers from 6 to 12 bits. Conversion time varies with accuracy and resolution from 9 to 35 μ s.

General Purpose Multiplexer Control, Type 139E

Permits up to 24 channels of analog signals to be read by analog-to-digital converter. Channels can be selected in sequence or by individual address. Prices of switches not included.

Multiplexer Expansion, Type AA03B

Allows Type 139E to be expanded to 64 channels.

A100 through A103 Switches for Multiplexer Control

Each module contains two multiplexer switches. For details of these switches and directions for their use, see Logic Handbook, C-105.

Digital-To-Analog Converters, Type AA01A

Three independent channels convert 12-bit digital numbers into corresponding analog signals. Output amplifiers are available on request.

A201 Operational Amplifier
(Requires Precision Power Supply H702.)

H702 Precision Power Supply

A10 19-INCH CABINETS

CAB-1B: with French doors front and back; without an indicator panel. With/without side panels.

CAB-9A: with full-length single doors front and back; without indicator panel. With/without side panels.

CAB-9B: with single doors front and back; with indicator panel. With/without side panels.

CAB-9C: with snap-on covers on the front and full-length single door on the rear; without indicator panel. With/without side panels.

CAB-9D: with snap-on covers on the front and full-length single door on the rear; with indicator panel. With/without side panels.

APPENDIX B
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 cell at location Y.
JMS Y	10	2	Jump to subroutine. The contents of the PC and the LINK, memory EXD mode, and memory protect mode status are deposited in memory cell Y. The next instruction is taken from cell Y + 1.
DZM Y	14	2	Deposit zero in memory. Zero is deposited in memory cell Y.
LAC Y	20	2	Load AC. The 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 (1s complement). The contents of Y are added to the contents of the AC in 1s complement arithmetic and the result is left in the AC.
TAD Y	34	2	2s complement add. The contents of Y are added to the contents of the AC in 2s complement arithmetic and the result is left in the AC.
XCT Y	40	1+	Execute. The instruction in memory cell Y is executed.
ISZ Y	44	2	Increment and skip if zero. The contents of Y are incremented by one in 2s complement arithmetic. If the result is zero, the next instruction is skipped.
AND Y	50	2	AND. The logical operation AND is performed between the 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 cell Y.

INPUT/OUTPUT TRANSFER INSTRUCTIONS

Mnemonic Symbol	Octal Code	Operation Executed
IOF	700002	<u>Program Interrupt</u> Interrupt off. Disable the PI facility.
ION	700042	Interrupt on. Enable the PI facility.
		<u>Real Time Clock</u>
CLSF	700001	Skip the next instruction if the clock flag is set to 1.
CLOF	700004	Clear the clock flag and disable the clock.
CLON	700044	Clear the clock flag and enable the clock.
		<u>Paper Tape Reader</u>
RSF	700101	Skip if reader flag is a 1.
RCF	700102	Clear reader flag, then inclusively OR the contents of reader buffer into the AC.
RRB	700112	Read reader buffer. Clear reader flag and AC, and then transfer contents of reader buffer into AC.
RSA	700104	Clear reader flag and select reader in alphanumeric mode. One 8-bit character is read into the reader buffer.
RSB	700144	Clear reader flag and select reader in binary mode. Three 6-bit characters are read into the reader buffer.

INPUT/OUTPUT TRANSFER INSTRUCTIONS (cont)

Mnemonic Symbol	Octal Code	Operation Executed
		<u>Paper Tape Punch</u>
PSF	700201	Skip if the punch flag is set to 1.
PCF	700202	Clear the punch flag.
PSA	700204	Clear the punch flag and punch a line of tape in alphanumeric mode.
PSA	700214	Clear AC and flag, and punch feed holes only.
PSB	700244	Clear the punch flag and punch a line of tape in binary mode.
		<u>I/O Equipment</u>
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.
		<u>Teletype Keyboard</u>
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.
		<u>Teletype Teleprinter</u>
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 (2s complement) the next instruction is skipped.
SZA	740200	Skip on zero accumulator. If the contents of the AC equal zero (2s 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 zero (2s 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 zero (2s 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.

OPERATE INSTRUCTIONS (cont)

Mnemonic Symbol	Octal Code	Operation Executed
RTR	742020	Rotate two right. The contents of the AC and the LINK are rotated two positions to the right.
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 76XXXX.