

**PDP-11/05 computer  
manual**

1st Printing August 1972  
2nd Printing September 1972  
3rd Printing January 1973

Copyright © 1972, 1973 by Digital Equipment Corporation

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

Page

## PART I-COMPUTER DESCRIPTION

### CHAPTER 1 COMPUTER COMPONENTS

1.1	Introduction	1-1
1.2	Computer Components	1-1
1.2.1	KD11-B Processor	1-1
1.2.2	Core Memory	1-2
1.2.2.1	Memory Organization	1-3
1.2.2.2	Memory Specifications	1-3
1.2.3	Power Supply	1-3
1.2.4	Backplane	1-4
1.3	ME11-L Core Memory System	1-5
1.4	Extension Mounting Box	1-5

### CHAPTER 2 UNIBUS

2.1	Introduction	2-1
2.2	Unibus Structure	2-1
2.2.1	Bidirectional Lines	2-2
2.2.2	Master-Slave Relationship	2-2
2.2.3	Inter-Locked Communication	2-2
2.3	Peripheral Device Organization and Control	2-3
2.4	Unibus Control Arbitration	2-3
2.4.1	Priority Transfer Requests	2-4
2.4.2	Processor Interrupts	2-5
2.4.3	Data Transfers	

## CONTENTS (Cont)

	Page
<b>CHAPTER 3 UNPACKING AND INSTALLATION</b>	
3.1	Introduction 3-1
3.2	Unpacking 3-1
3.3	Mechanical Description 3-3
3.4	Installation 3-10
3.4.1	Mounting Computer on Installed Slides 3-10
3.4.2	Securing Computer to Cabinet Rack 3-10
3.4.3	Installation of I/O Cables 3-11
3.5	Interchangeable Peripheral Slots 3-11
3.6	Side and Top Cover Installation 3-11
3.7	AC Power Supply Connection 3-12
3.7.1	Connecting to Voltages Other Than 115V 3-12
3.7.2	Quality of AC Power Source 3-12
3.8	Cabinet Power Control 3-12
3.9	Installation Certification 3-13
3.10	Warranty Service (Domestic Only) 3-14
<b>CHAPTER 4 OPERATION</b>	
4.1	Introduction 4-1
4.2	Power Switch Operation 4-1
4.3	Function Switches 4-1
4.4	Address/Data Switches 4-3
4.5	Console Indicators 4-3
4.6	Operation Console 4-3
4.6.1	LOAD ADDRess Switch 4-5
4.6.2	EXAM Switch 4-5
4.6.3	DEPosit Switch 4-6

## CONTENTS (Cont)

	Page	
4.6.4	ENABLE/HALT Switch	4-6
4.6.5	START Switch	4-7
4.6.6	CONTINUE Switch	4-7
4.7	Unconditional Computer and Unibus Initialization	4-8
4.8	Loading Programs from Paper Tape	4-8
4.8.1	The Bootstrap Loader	4-9
4.8.1.1	Loading the Loader Into Memory	4-10
4.8.1.2	Loading Bootstrap Tapes	4-11
4.8.1.3	Bootstrap Loader Operation	4-13
4.8.2	The Absolute Loader	4-14
4.8.2.1	Loading the Loader Into Memory	4-15
4.8.2.2	Loading Absolute Tapes	4-15
4.8.3	Memory Dumps	4-17
4.8.3.1	Operating Procedures	4-17
4.8.3.2	Output Formats	4-18
4.8.3.3	Storage Maps	4-19
4.9	Installation Testing	4-19
4.10	SCL Baud Rate Adjustment	4-20

## PART II - KD11-B PROCESSOR

### CHAPTER 1 GENERAL DESCRIPTION

1.1	Introduction	1-1
1.2	KD11-B Definition	1-1
1.3	KD11-B and the Unibus	1-2
1.4	KD11-B As an Instruction Interpreter	1-3
1.5	KD11-B Print Set	1-4
1.5.1	Medium and Large Scale Integration Circuit Representations	1-7

## CONTNETS (Cont)

	Page	
1.5.2	Microprogram Documentation	1-7
1.5.3	Read-Only Memory (ROM) Maps	1-8
CHAPTER 2 INSTRUCTION SET		
2.1	Introduction	2-1
2.2	Addressing Modes	2-1
2.2.1	Introduction	2-1
2.2.2	Instruction Timing	2-4
2.3	PDP-11/05 Instructions	2-4
CHAPTER 3 CONSOLE DESCRIPTION		
3.1	Introduction	3-1
3.2	General Description	3-1
3.2.1	Address/Data Register Logic	3-1
3.2.2	Control Switch Logic	3-3
3.3	Detailed Description	3-3
3.3.1	Multiplexer	3-4
3.3.2	Clock	3-5
3.3.3	Counter	3-6
3.3.4	Display Buffer and Driver	3-10
3.3.5	Control Switches and Logic	3-10
3.3.5.1	Normal Operating Mode	3-12
3.3.5.2	Panel Lock Mode	3-14
3.3.5.3	Power Loss During Operation	3-14
CHAPTER 4 DETAILED DESCRIPTION		
4.1	Introduction	4-1
4.2	ROMs As Generalized Gates	4-1

## CONTENTS (Cont)

	Page	
4.3	KD11-B Data Path, Simplified Description	4-2
4.3.1	Data Path (DP) Detailed Description	4-2
4.3.2	DP Data Polarities	4-3
4.3.3	Data Path Control (DPC)	4-4
4.3.4	The A MUX1	4-6
4.3.5	The ALU	4-6
4.4	Scratch Pad Storage Register	4-7
4.4.1	Scratch Pad Address Multiplexer (SPAM)	4-7
4.4.2	Processor Status Word Register	4-7
4.4.3	The Constants Generator	4-8
4.4.4	The Console Switch Register	4-11
4.4.5	Serial Communications Line	4-11
4.5	B-Leg Storage Register	4-11
4.5.1	Byte Instructions	4-11
4.5.2	Instruction Register (IR) and IR Decode	4-12
4.6	Data Path Control and Clocking	4-13
4.7	Unibus Control	4-15
4.7.1	DATI Timing	4-16
4.7.2	DATI Operation	4-16
4.7.2.1	DATIP Operation	4-18
4.7.2.2	DATIP Logic	4-19
4.7.3	DATO	4-20
4.7.4	Byte Operations	4-20
4.7.5	Bus Errors	4-21
4.8	Internal Unibus Addresses	4-22
4.9	Bus Requests (BR)	4-23
4.10	Non-Processor Requests (NPR)	4-28
4.11	Serial Communications Line Description (SCL)	4-28
4.12	Line Clock	4-31
4.13	Power Fail	4-31

## CONTENTS (Cont)

	Page
<b>CHAPTER 5 MICROPROGRAM CONTROL</b>	
5.1	Introduction 5-1
5.2	Microprogrammed Control CVS Conventional Control 5-1
5.3	Control Store 5-2
5.4	Branching Within Microroutines 5-8
5.5	Microprogram Flow 5-11
5.5.1	Flow Chart Notation 5-13
5.5.2	Traps and Interrupts 5-22
5.5.3	Console Functions 5-23
5.6	Microprogram Symbolic Listing 5-26
5.7	Microprogram Binary Listing 5-27
5.8	Microprogram Cross Reference Listing 5-28
<b>CHAPTER 6 MAINTENANCE</b>	
6.1	Introduction 6-1
6.2	Diagnostics 6-1
6.3	Types of Failures 6-2
6.4	Suggested Equipment 6-2
6.5	Procedures 6-3
6.6	Adjustments 6-4
6.7	KD11-B Print Function Table 6-5
6.8	External Clock Inputs 6-5
6.9	KM11 Maintenance Panel 6-8
6.10	Using KM Maintenance Panel 6-11
6.11	Console Maintenance 6-12



## CONTENTS (Cont)

Page

### PART III - MM11-K, MM11-L MEMORIES

#### CHAPTER 1 GENERAL DESCRIPTION

1.1	Introduction	1-1
1.2	General Description	1-1
1.2.1	Physical Description	1-2
1.2.2	Specifications	1-2
1.2.3	Functional Description	1-7
1.2.3.1	G110 Control Module	1-7
1.2.3.2	G231 Drive Module	1-9
1.2.3.3	H213 or H214 Stack Module	1-10
1.2.4	Basic Memory Operations	1-10
1.2.4.1	Data In (DATI) Cycle	1-11
1.2.4.2	Data In, Pause (DATIP) Cycle	1-11
1.2.4.3	Data Out (DATO) Cycle	1-11
1.2.4.4	Data Out, Byte (DATOB) Cycle	1-12

#### CHAPTER 2 DETAILED DESCRIPTION

2.1	Introduction	2-1
2.2	Core Array	2-1
2.3	Memory Operation	2-3
2.4	Device and Word Selection	2-5
2.4.1	Memory Organization and Addressing Conventions	2-8
2.4.2	Device Selector	2-10
2.4.3	Word Selection	2-14
2.4.3.1	Word Address Register and Gating Logic	2-14
2.4.3.3	Drivers and Switches	2-21

## CONTENTS (Cont)

	Page	
2.4.3.4	Word Address Decoding and Selection Sequence	2-24
2.5	READ/WRITE Current Generation and Sensing	2-26
2.5.1	Read/Write Operations	2-26
2.5.2	X- and Y-Current Generators	2-29
2.5.3	Inhibit Driver	2-31
2.5.4	Sense Amplifier	2-33
2.5.5	Memory Data Register	2-33
2.6	Stack Discharge Circuit	2-34
2.7	DC LO Circuit	2-36
2.8	Operating Model Selection Logic	2-37
2.9	Control Logic	2-40
2.9.1	Timing Circuit	2-40
2.9.2	Slave Synchronization (SSYN) Circuit	2-47
2.9.3	Pause/Write Restart Circuit	2-50
2.9.4	Strobe Generating Circuit	2-52
2.9.5	Data In (DATI) Operation	2-54
2.9.6	Data In Pause (DATIP) Operation	2-57
2.9.7	Data Out (DATO) Operation	2-58
2.9.8	Data Out Byte (DATOB) Operation	2-59
CHAPTER 3 MAINTENANCE		
3.1	Introduction	3-1
3.2	Preventive Maintenance	3-1
3.2.1	Initial Procedures	3-1
3.2.2	Checking Output of Current Generator	3-2
3.3	Corrective Maintenance	3-2
3.3.1	Strobe Delay Check and Adjustment	3-3
3.3.2	Corrective Maintenance Aids	3-3
3.4	Programming Tests	3-11

## CONTENTS (Cont)

	Page
3.4.1 Address Test Up (MAINDEC-11-DIAA)	3-11
3.4.2 Address Test Down (MAINDEC-11-DIBA)	3-11
3.4.3 No Dual Address Test (MAINDEC-11-DICA)	3-12
3.4.4 Basic Memory Patterns Test (MAINDEC-11-DIDA)	3-12
3.4.5 Worst-Case Noise Test (MAINDEC-11-DIGA)	3-12

## PART IV - POWER SUPPLY

### CHAPTER 1 GENERAL DESCRIPTION

1.1 Introduction	1-1
1.2 Physical Description	1-2
1.2.1 Power Control	1-2
1.2.2 Power Chassis Assembly	1-3
1.2.3 DC Regulator Module	1-3
1.2.4 DC Cable	1-8
1.2.5 AC Cable	1-8
1.3 Specifications	1-9

### CHAPTER 2 DETAILED DESCRIPTION

2.1 Introduction	2-1
2.2 AC Input Circuit	2-1
2.3 DC Regulator Module Operation	2-4
2.3.1 Generation of $\pm$ Raw DC	2-6
2.3.2 LTC L Circuit	2-6
2.3.3 BUS AC LO and BUS DC LO L Circuits	2-7
2.3.4 +15V Regulator Circuit	2-9
2.3.5 +5V Regulator Circuit	2-10
2.3.6 -15V Regulator Circuit	2-12

## CONTENTS (Cont)

	Page
CHAPTER 3 MAINTENANCE	
3.1	Introduction 3-1
3.2	Adjustments 3-1
3.3	Circuit Waveforms 3-2
3.4	Troubleshooting 3-2
3.4.1	Troubleshooting Rules 3-6
3.4.2	Troubleshooting Hints 3-6
3.4.3	Troubleshooting Chart 3-7
3.5	Parts Identification 3-9

### TABLES

#### PART I

4-1	Significance of Address/Data Indicators 4-4
4-2	Bootstrap Loader Instructions 4-9
4-3	Memory Bank Assignments 4-10

#### PART II

2-1	Addressing Modes 2-3
2-2	Addressing Times 2-4
2-3	Single Operand Instructions 2-6
2-4	Double Operand Instructions 2-10
2-5	Program Control Instructions 2-13
3-1	Scan Address Signal Generation 3-4
3-2	Counter States 3-9
4-1	Utilization of SP 4-7
4-2	Scratch Pad Address Sources through SPAM 4-8
4-3	Contents of the Constants Generator (E025) ROM 4-9
4-4	Unibus Addresses 4-22

## TABLES (Cont)

		Page
4-5	Trap Priorities	4-25
5-1	KD11-B Control Store Fields	5-3
5-2	Microprogram Branches (BUT)	5-8
5-3	Flow Notation Glossary	5-15
6-1	Test Equipment and Tools	6-2
6-2	Baud Rate Adjustment	6-5
6-3	Engineering Drawing Print, List and Functions	6-6
6-4	KM-1 and KM-2 Overlay Designations	6-10

## PART III

1-1	MM11-K and L Memory Specifications	1-2
2-1	Addressing Functions	2-7
2-2	Enabling Signals for Word Register Gating	2-17
2-3	Word Address Decoding Signals	2-25
2-4	Selection of Bus Transactions	2-38
2-5	Generation of Memory Operating Signals	2-39

## PART IV

1-1	Power Supply Specifications	1-9
3-2	Troubleshooting Chart	3-7

## ILLUSTRATIONS

Figure No.	Title	Page
PART 1-COMPUTER DESCRIPTION		
1-1	Module Utilization Diagram For Configuration 1 (16K)	1-6
1-2	Module Utilization Diagram For Configuration 2 (8K)	1-7
1-3	Computer Backplane Connector and Pin Designations	1-8
3-1	Computer Packaging	3-2
3-2	Computer Mounting Box	3-4
3-3	Computer Box With Top Cover Removed	3-5
3-4	Computer Box With Top and Side Covers Removed	3-6
3-5	Computer Chassis (showing both peripheral cables and the Unibus)	3-7
3-6	Mounting Box Without Modules	3-8
3-7	Rear of Computer Showing Cable Strain Reliefs	3-9
3-8	Typical Cabinet Power Control System Wiring Diagram	3-13
4-1	Console Illustrating Switch Movements	4-2
4-2	Loading and Verifying the Bootstrap Loader	4-12
4-3	Loading Bootstrap Tapes Into Memory	4-13
PART 2-KD11-B PROCESSOR		
1-1	KD11-B With Interconnections to Memory and Peripherals	1-2
1-2	KD11-B Processor Block Diagram	1-4
1-3	Instruction Interpreter Block Diagram	1-4
1-4	Typical Small Scale Integrated Circuit Representations	1-5
1-5	DPF LOAD IRL Signal	1-6
1-6	ALV, MSI Circuit Type 74181 Representation	1-7
1-7	E068 ROM Map Example	1-9

ILLUSTRATIONS (Cont)		
Figure No.	Title	Page
2-1	Addressing Mode Instruction Formats	2-2
2-2	PDP-11 Instruction Formats	2-5
3-1	Console Functional Block Diagram	3-2
3-2	Console Clock, Schematic and Timing Diagram	3-5
3-3	Counter, Simplified Logic Diagram	3-7
3-4	Display Buffer and Driver, Simplified Logic Diagram	3-11
3-5	LED Driver Circuit	3-11
3-6	Control Switches and Bounce Buffers, Logic Diagram	3-13
4-1	1024-Bit and 256-Bit ROMs	4-1
4-2	32 x 8 ROM used as Generalized Gate	4-3
4-3	KD11-B Simplified Data Path Block Diagram	4-4
4-4	KD11-B Detailed Block Diagram	4-5
4-5	Byte Format for Shifting Instructions	4-13
4-6	KD11-B Processor Clock Phasing	4-14
4-7	KD11-B Basic Oscillator	4-14
4-8	DATI and DATO Timing	4-17
4-9	Unibus Address Decoding	4-24
4-10	Bus Request (BR) Timing	4-27
4-11	Double Buffering Data Flow	4-29
4-12	Bus AC LO and Bus DC LO Timing Diagram	4-32
5-1	Control Store Word Bit and Field Format	5-3
5-2	KD11-B Simplified Flow Diagram	5-12
5-3	Excerpt from Microprogram Flow (K-NL-KD11-B-1)	5-14
5-4	CMP #15, CHAR (022767) Simplified Flow Diagram	5-17
5-5	Excerpt of (K-WL-KD-11-B-2) Microprogram Symbolic Listing	5-29
5-6	Excerpt of Microprogram Binary Listing (K-W-KD11-B-3)	5-30
6-1	KM11 Maintenance Module, KD11-B Overlays	6-9

## ILLUSTRATIONS (Cont)

Figure No.	Title	Page
<b>PART 3-MM11-K, MM11-L MEMORIES</b>		
1-1	Component Side of G110 Control Module	1-4
1-2	Component Side of G231 Drive Module	1-5
1-3	Component Side of 8K H214 Stack Module	1-6
1-4	MM11-K, L Memory Block Diagram	1-8
2-1	Three-Wire Memory Configuration	2-2
2-2	Hysteresis Loop for Core	2-4
2-3	Three-Wire 3D Memory, Four Mats Shown for a 16-Word - 4-Bit Memory	2-6
2-4	Device and Word Address Selection Logic, Block Diagram	2-8
2-5	Memory Organization for 8K Words	2-9
2-6	Address Assignments For Three Banks of 8K Words Each	2-11
2-7	Jumper Configuration For A Specific Memory Address	2-13
2-8	Device Decoding Guide	2-15
2-9	Type 8251 Decoder, Pin Designation and Truth Table	2-18
2-10	Decoding of Read/Write Switches and Drivers Y4-Y7	2-19
2-11	Switch or Driver Base Drive Circuit	2-20
2-12	Y-Line Selection Stack Diode Matrix	2-22
2-13	Typical Y-Line Read/Write Switches and Drivers	2-23
2-14	Interconnection of Unibus, Data Register, Sense Amplifier, and Inhibit Driver	2-27
2-15	Y-Current Generator and Reference Voltage Supply	2-30
2-16	Sense Amplifier and Inhibit Driver	2-31
2-17	Type 7528 Dual Sense Amplifiers With Preamplifier Test Points	2-34
2-18	Stack Discharge Circuit	2-35
2-19	DC LO Circuit, Schematic Diagram	2-37
2-20	Basic Timing and Control Signal Functions	2-41
2-21	TWID H and TNAR H Control Logic	2-44



## ILLUSTRATIONS (Cont)

Figure No.	Title	Page
2-22	Generation of MSEL RESET L	2-46
2-23	Slave Sync (SSYN) Circuit	2-48
2-24	Pause/Write Restart Circuit	2-51
2-25	Strobe Generating Circuit	2-53
2-26	Flow Chart For Memory Operation	2-56
3-1	Strobe Pulse Waveform	3-3
3-2	Troubleshooting Chart	3-5
3-3	MM11-K Sense/Inhibit Waveforms	3-7
PART 4-POWER SUPPLY		
1-1	Power Chassis Assembly (With DC Regulator Module)	1-4
1-2	Power Supply Assembly (With DC Regulator Module Removed)	1-5
1-3	DC Regulator Module (Top View)	1-6
1-4	DC Regulator Module (Bottom View in Mounting Box)	1-7
2-1	Detailed AC Interconnection Diagram	2-2
2-2	115V Connections - Simplified Schematic Diagram	2-3
2-3	230V Connection Diagram	2-3
2-4	Regulator Module Block Diagram	2-5
2-5	Rectifier and LTC L Circuits	2-7
2-6	BUS AC LO and BUS DC LO Circuits	2-8
2-7	+15V Regulator Circuit	2-9
2-8	+5V Regulator Circuit	2-10
2-9	-15V Regulator Circuit	2-13
3-1	+5V Regulator Circuit Waveforms	3-3
3-2	-15V Regulator Circuit Waveforms	3-4
A	INTEGRATED CIRCUIT (IC) DESCRIPTIONS	A-1
B	DETAILS OF KD11-B IR DECODE	B-1
C	COMPUTER CONNECTORS	C-1
D	INTERFACE CIRCUITS AND HARDWARE	D-1



## FOREWORD

This manual describes the PDP-11/05 and PDP-11/10 Computers. The PDP-11/05 and PDP-11/10 are electrically identical. The PDP-11/05 is specified for the Original Equipment Manufacturer (OEM) market and the PDP-11/10 is specified for the end user market.

The PDP-11/05 is available in two versions: one provides a maximum of 8K words of core memory and the other provides a maximum of 16K words of core memory. The PDP-11/10 is available only with a maximum of 8K words of core memory.

This manual is divided into four parts.

Part 1	Computer Description
Part 2	KD11-B Processor
Part 3	MM11-K, MM11-L Memories
Part 4	Power Supply

Chapter outlines of each part are shown below.

### Part 1 COMPUTER DESCRIPTION

Chapter 1	Computer Components
Chapter 2	Unibus
Chapter 3	Unpacking and Installation
Chapter 4	Operation

### Part 2 KD11-B PROCESSOR

Chapter 1	General Description
Chapter 2	Instruction Set
Chapter 3	Console Description
Chapter 4	Detailed Description
Chapter 5	Microprogram Control
Chapter 6	Maintenance

**Part 3 MM11-K, MM11-L MEMORIES**

Chapter 1	General Description
Chapter 2	Detailed Description
Chapter 3	Maintenance

**Part 4 POWER SUPPLY**

Chapter 1	General Description
Chapter 2	Detailed Description
Chapter 3	Maintenance

A bound volume of engineering drawings is supplied with each computer.

The following related documents are valuable as references.

PDP-11/05, 11/10 Processor Handbook  
PDP-11 Peripherals and Interfacing Handbook  
PDP-11 Paper-Tape Software Programming Handbook  
(Document No. DEC-11-GGPB-D)

PART 1  
Computer Description

## CHAPTER 1 COMPUTER COMPONENTS

### 1.1 INTRODUCTION

This chapter briefly describes the major components of the computer. It includes module utilization diagrams for both computer configurations and a backplane connector and pin designation diagram.

### 1.2 COMPUTER COMPONENTS

The computer consists of a mounting box, console, processor, core memory, prewired backplane, power supply, fans, and interconnecting cables. The processor is contained on two modules, and each 4K or 8K memory is contained on three modules.

#### 1.2.1 KD11-B Processor

The processor modules are M7260 Data Paths and M7261 Control Logic and Microprogram. They are hex height modules and measure 8 1/2 in. long x 15 in. high. A hex height module contains six edge-connectors (A-F).

All the processor functional components are contained on these modules, as shown below.

The M7260 Data Path Module contains:

- Data path logic
- Processor status word logic
- Auxiliary arithmetic logic unit control
- Instruction register and decoding logic
- Serial communications line interface

The M7261 Control Logic and Microprogram Module contains:

- Internal address detecting logic
- Stack control logic
- Unibus control logic
- Priority arbitration logic
- Unibus drivers and receivers
- Micro branch logic
- Micro program counter
- Control store logic
- Power fail logic
- Line clock
- Processor clock

The serial communications line (SCL) interface is directly connected to the desired serial communications device. It can operate at speeds of 110-300 baud and is program compatible with the KL11 Teletype Control Interface option. The SCL is compatible with the LA30 DECwriter at 30 characters per second, the VT05 Alphanumeric CRT Display Terminal at 30 characters per second, and the Teletype Model 33 ASR at 10 characters per second.

The line time clock (LTC) allows the program to measure time by sensing the 50 Hz or 60 Hz ac line frequency. This clock is program compatible with the KW11-L Line Time Clock option.

The line time clock and the serial communications line interface are not connected to the Unibus. They use an internal bus and can be addressed only by the processor and the console.

### 1.2.2 Core Memory

The PDP-11/05 is available in two versions: one provides a maximum of 8K words of core memory and the other provides a maximum of 16K words of core memory. The PDP-11/10 is available only with a maximum of 8K words of core memory. A separate add-on core memory system (ME11-L) is available to provide an additional 8K, 16K, or 24K words of core memory. A PDP-11/05 or PDP-11/10 processor provides program control for a maximum of 32K words of memory; therefore, the self-contained memory plus the ME11-L must not be greater than 32K words.

1.2.2.1 Memory Organization - The memory is organized in 16-bit words consisting of two 8-bit bytes. The bytes are identified as low and high. The memory contains 8192 words or 16,384 bytes; therefore, 16,384 locations are assigned. The address locations are specified as 6-digit octal numbers. The 16,384 locations for the 8K memory are designated 000000 through 037777.

Each byte is addressable and has its own address location: low bytes are even numbered and high bytes are odd numbered. Words are addressed at even numbered locations only, and the high (odd) byte is automatically included. Consecutive words are therefore found in even numbered addresses.

The PDP-11 address word contains 18 bits A <17:00>, which provides the capability of addressing 262,144 (256K) locations (bytes) or 131,072 (128K) words. The basic processor provides 16 bits (A <15:00>) of address information, which handles 65,536 (64K) bytes or 32,768 (32K) words. During an addressing operation, if bits A <15:13> are all 1s, bits A <17:16> are forced to 1s, which relocates the last 8K bytes (4K words) to become the highest locations accessed by the bus. These top 4,096 word locations are reserved for peripheral and register addresses, and the user therefore has 28,672 (28K) words of memory to program.

1.2.2.2 Memory Specifications - The core memory is a read/write, random access, coincident current type with a cycle time of 900 ns and an access time of 400 ns. It is organized in a 3D, 3-wire planar configuration. Word length is 16 bits, and the memory is offered in two word capacities: model MM11-K contains 4096 words and model MM11-L contains 8192 words. Each memory is contained on three modules called the control, driver, and stack. For the MM11-K memory, the stack module is H213; H214 is the stack module for the MM11-L memory. The G110 Control Module and the G231 Driver Module are the same for both models.

### 1.2.3 Power Supply

The power supply consists of a dc regulator module, transformer, and fan, mounted in a chassis. It is installed in the computer mounting box. The power supply converts 115V or 230V, 47-63 Hz line voltage to three regulated dc voltages that are used by the processor, memory, and optional modules. The regulated voltages are:



+5V at 17A  
-15V at 6A  
+15V at 1A

An associated component, the power control, provides the ac line voltage to the power supply and cooling fans. The power control is installed in the rear panel of the computer mounting box. It consists of a line cord, circuit breaker, and output connector. A model is available for each of the two line voltages (115V or 230V), as shown below.

<u>Power Control Part Number</u>	<u>Rating</u>
BC05H	7A at 115V/47-63 Hz
BC05J	4A at 230V/47-63 Hz

The power supply provides three additional outputs. Signal LTC L is the Line Time Clock signal that drives the line time clock. The BUS AC LO L and BUS DC LO L signals actuate the processor power fail-auto restart circuitry.

#### 1.2.4 Backplane

The backplane is a connector assembly into which the computer modules are plugged. It provides interconnections between the Unibus, processor, memory, and optional modules. The interconnections are made via a printed circuit board and wirewrapped pins that are part of the backplane assembly.

There are two versions of the computer (8K memory and 16K memory). The backplane is wired differently for each version. As a result, the modules must be installed in specific locations as shown in Figure 1-1 for the 16K version and Figure 1-2 for the 8K version. These illustrations show the backplane as viewed from the module side. The slots are numbered 1 through 9 from top to bottom, and the connectors are lettered A through F from left to right.

Configuration 1 is the 16K version (Figure 1-1). Unibus M930 Terminator Modules are installed in slots A2-B2 and A5-B5. If other peripherals are to be connected to the computer, the terminator module in slot A2-B2 must be replaced with a BC11A Unibus cable, and a terminator module must be installed in the last device in the system. Slot C1-F1 provides the only space for a small peripheral controller. If this slot is not used, a G727 Grant Continuity Module must be installed in slot D1. If a small peripheral controller is

to be installed, the G727 module must be removed first. Slots A1 and B1 are wired for the KM11 Maintenance Module. The core memories (3 modules each) are physically interchangeable as systems.

Configuration 2 is the 8K version (Figure 1-2). Unibus M930 Terminator Modules are installed in slots A3-B3 and A5-B5. If required, a BC11A Unibus cable can be installed in place of the terminator in slot A3-B3. Slots C1-F1, C2-F2, C3-F3, and C4-F4 can be used for small peripheral controllers. Slot A1-B1 is wired for a DF11 Communications Line Adapter that provides signal conditioning for communications devices using signals that are not TTL compatible. Slots A2 and B2 are wired for the KM11 Maintenance Module.

Figure 1-3 shows the backplane connector block configuration as viewed from the wirewrap pin side. The pin arrangement for each connector block is identical. It represents the total pins (36) available on the double-sided edge connector of a single height module. Connector A1 is shown in detail.

### 1.3 ME11-L CORE MEMORY SYSTEM

Additional core memory is available for the computer in the ME11-L self-contained add-on core memory system. The basic ME11-L consists of an 8K MM11-L memory and power supply installed in a mounting box. It is expandable to 16K words or 24K words maximum by adding one or two more MM11-L memories. The ME11-L uses the same backplane construction as the computer. Nine slots are provided, and they are wired to accommodate three MM11-L memories. These core memories (3 modules each) are physically interchangeable as systems and as individual modules within a system for troubleshooting purposes. If only one memory is used, the modules must be installed in the three bottom slots (numbers 7, 8, and 9).

### 1.4 EXTENSION MOUNTING BOX

Additional interface logic for the computer is installed in an extension mounting box identical to those used for the rest of the PDP-11 Family. A rack-mounted box (BA11-ES) or a tabletop box (BA11-EC) can be used. The mounting box contains cooling fans, filter, and power cord. Space is provided to install six system units and an H720 Power Supply. Details of the extension mounting box, system units, and H720 Power Supply are included in the PDP-11 Peripherals and Interfacing Handbook.

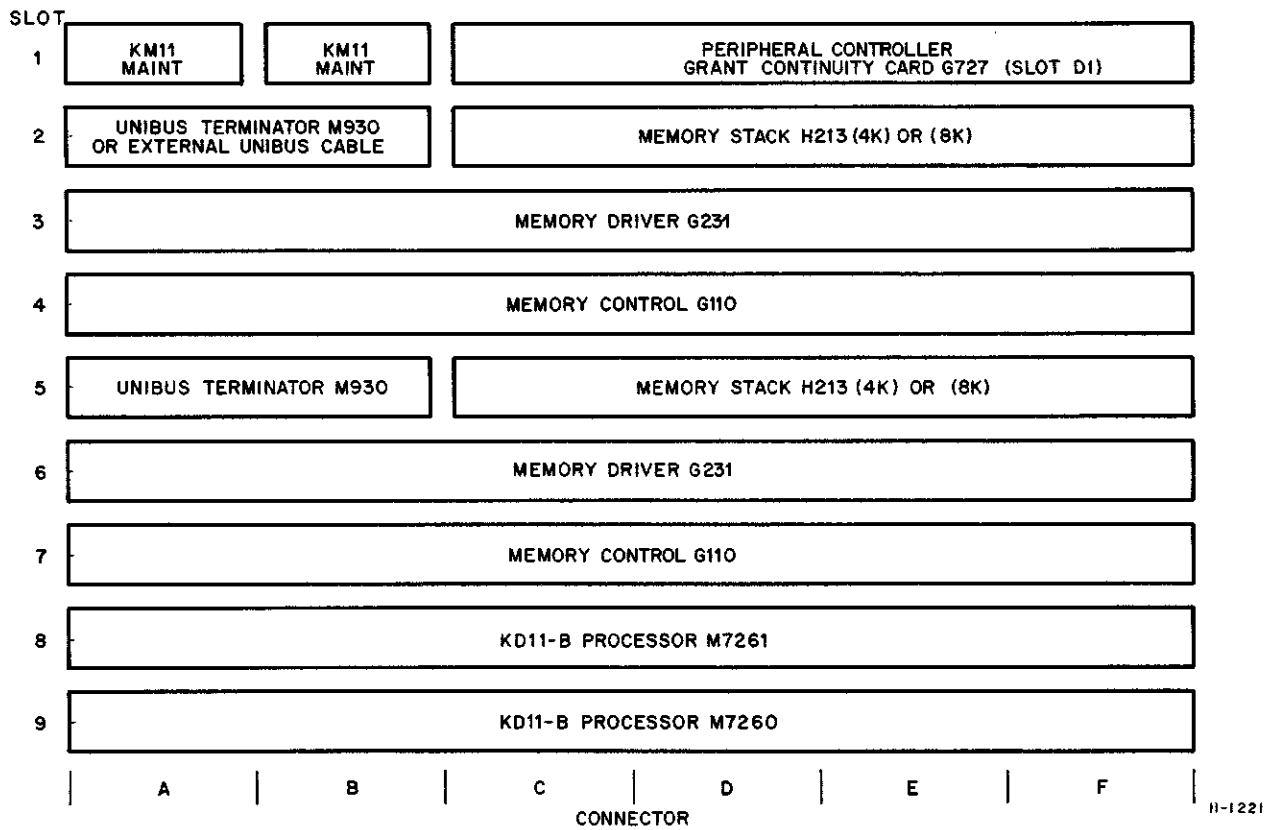
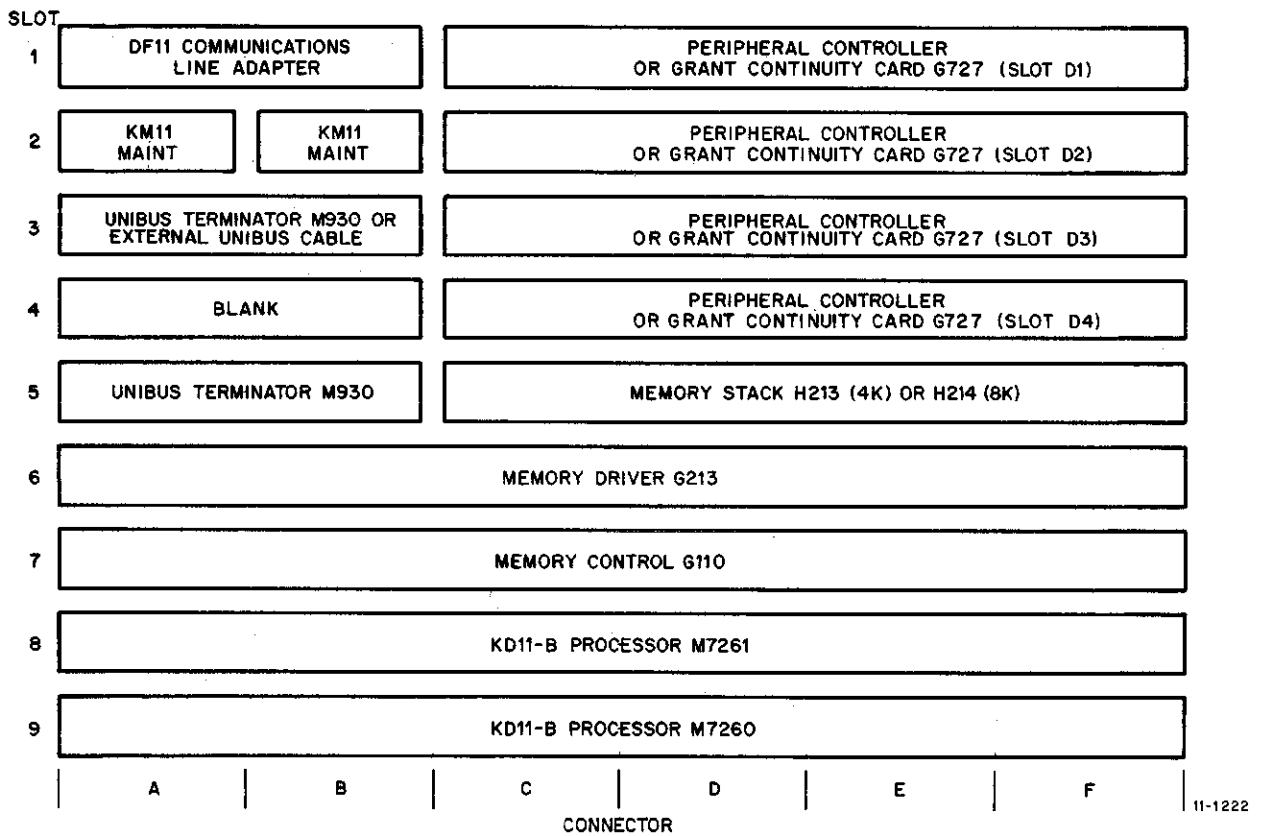
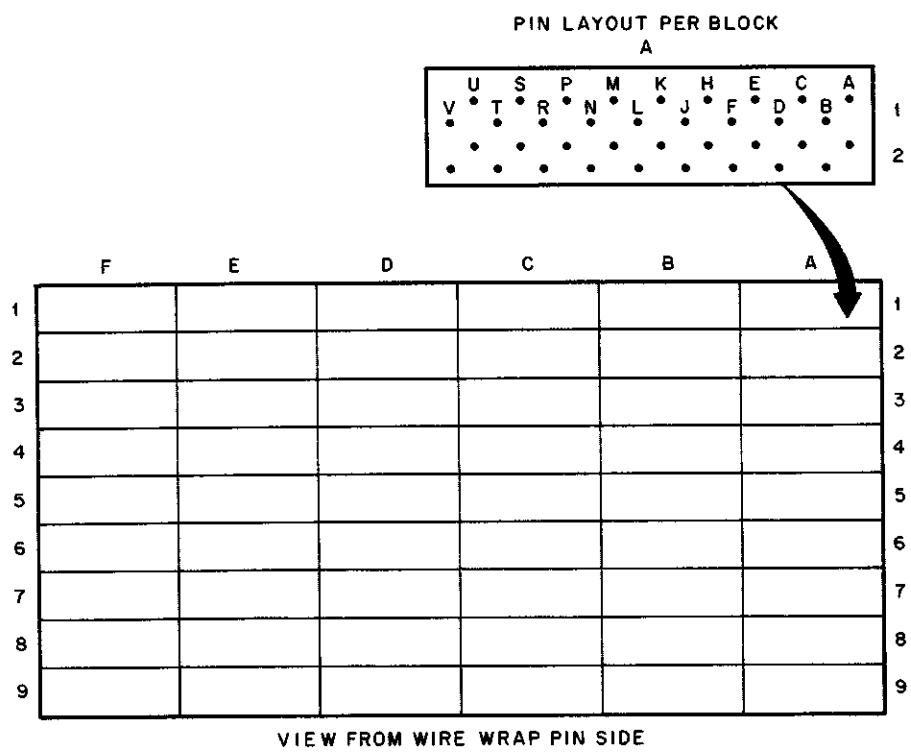


Figure 1-1 Module Utilization Diagram For Configuration 1 (16K)



11-1222

Figure 1-2 Module Utilization Diagram For Configuration 2 (8K)



11-1220

Figure 1-3 Computer Backplane Connector and Pin Designations

## CHAPTER 2 UNIBUS

### 2.1 INTRODUCTION

This chapter describes in general the operation of the Unibus.

The following documents, in conjunction with this manual, will aid the reader in understanding interface techniques and the overall PDP-11 system.

- a. PDP-11/05-11/10 Processor Handbook
- b. PDP-11 Peripherals and Interfacing Handbook
- c. Digital Logic Handbook

All communication between PDP-11 system components is through the high-speed Unibus. The Unibus operational concepts are vital to the understanding of the hardware and software implications of the Unibus.

### 2.2 UNIBUS STRUCTURE

The Unibus is a single, common path that connects the processor, memory and all peripherals. Addresses, data, and control information are transmitted along the 56 lines of the bus.

Every device on the Unibus employs the same form of communication; thus, the processor uses the same set of signals to communicate with memory and with peripheral devices. Peripheral devices also communicate with the processor, memory, or other peripheral devices via the same set of signals.

All instructions applied to data in memory can be applied equally well to data in peripheral device registers, enabling peripheral device registers to be manipulated by the processor with the same flexibility as memory. This feature is especially powerful, considering the capability of PDP-11 instructions to process data in any memory location as though it were an accumulator.

### 2.2.1 Bidirectional Lines

Most Unibus lines are bidirectional, allowing input lines to also be driven as output lines. This is significant in that a peripheral device register can be either read or can be used for transfer operations. Thus, the same register can be used for both input and output functions.

### 2.2.2 Master-Slave Relationship

Communication between two devices on the bus is a master-slave relationship. During any bus operation, one device has control of the bus. This device, the bus master, controls the bus when communicating with another device on the bus, the slave. A typical example of this relationship is the processor, as the master, transferring data to memory, as slave. Master-slave relationships are dynamic. The processor, for example, passes bus control to a disk. The disk, as master, then communicates with a slave memory.

The Unibus is used by the processor and all I/O devices; thus, a priority structure determines which device gains control of the bus. Consequently, every device on the Unibus capable of becoming bus master has an assigned priority. When two devices capable of becoming bus master have identical priority values and simultaneously request use of the bus, the device that is electrically closest to the bus receives control.

### 2.2.3 Interlocked Communication

Communication on the Unibus is interlocked between devices. Each control signal issued by the master device must be acknowledged by a response from the slave to complete the

transfer. Consequently, communication is independent of the physical bus length and the response time of the master and slave devices. The maximum transfer rate on the Unibus, with optimum device design, is one 16-bit word every 400 ns or 2.5 million 16-bit words per second.

### 2.3 PERIPHERAL DEVICE ORGANIZATION AND CONTROL

Peripheral device registers are assigned addresses similar to memory; thus, all PDP-11 instructions that address memory locations can become I/O instructions, enabling data registers in peripheral devices to take advantage of all the arithmetic power of the processor.

The PDP-11 controls devices differently than most computer systems. Control functions are assigned to a register address, and then the individual bits within that register can cause control operations to occur. For example, the command to make the paper-tape reader read a frame of tape is provided by setting a bit (the reader enable bit) in the control register of the device. Instructions such as MOV and BIS may be used for this purpose. Status conditions are also handled by the assignment of bits within this register, and the status is checked with TST, BIT, and CMP instructions.

### 2.4 UNIBUS CONTROL ARBITRATION

The Unibus is capable of performing two basic and parallel tasks in order to allow transfers by multiple peripherals at maximum speed. The first is the actual transfer of data between the current bus master and its addressed slave. The second is the selection of the next bus master, the peripheral which, as soon as the bus becomes free, will be allowed to assert control. It is important to note that the granting of future mastership is in no way influenced by either the current master or its method of obtaining the bus. It is this fact which allows these functions to be performed in parallel and allows transfers on the bus at a maximum rate.



### 2.4.1 Priority Transfer Requests

To gain mastership of the Unibus, a peripheral must first make a request to the processor for the bus and then wait for its selection. The processor contains the logic necessary to arbitrate these requests because normally there are several requests pending at any given time.

There are two classes of requests: Bus Requests and Non Processor Requests. A Bus Request (BR) is simply a request by a peripheral to obtain control of the Unibus with the understanding by the processor that the peripheral may end its use of the bus with a processor interrupt. An interrupt is a command to the processor to begin executing a new routine pointed to by a location selected by a device. A Non Processor Request (NPR) is similarly a request for the bus, but with the exception that it may not interrupt the processor. Since the granting of an NPR cannot affect the execution of the processor, it can occur during or between instructions. BRs however, by possibly causing execution to be diverted to a totally new routine, can only be granted between instructions. In this way, NPRs are assigned priority over any BR.

Between Bus Requests, there are four levels of priority created by four separate requests lines. These are assigned priority levels 4 through 7. BR4 is the lowest and BR7 is the highest. These levels are associated with the program controlled priority level of the processor controlled by bits 7, 6, and 5 of the processor status register. Only BRs on a priority level higher than the level of the processor are eligible for receiving a bus grant. Thus, during high priority program tasks, all or selected Unibus requests (hence interrupts) can be inhibited by raising the level of the processor priority.

Another form of priority arbitration occurs through the system configuration. When the processor grants a request, the grant travels along the bus until it reaches the first requesting device which terminates the grant. Therefore, along the same grant line, the device electrically nearest the processor has the highest priority. Also note that in the KD11-B, the internal line clock is logically the last device on BR6, and the serial communication line interface is logically the last device on BR4.

After a requesting device receives a bus grant it asserts its selection as next bus master until the bus is free, thus inhibiting other requests from being granted. When the bus becomes free, the selected device asserts control of the bus and relinquishes its selection as next bus master so that the priority arbitration among pending requests may continue.

#### 2.4.2 Processor Interrupts

After gaining control of the bus through a BR, a device can perform one or more transfers on the bus and/or request a processor interrupt. This is typically requested after a device has completed a given task, for instance typing a character or completing a block data transfer through NPRs. If a peripheral wishes to interrupt the processor, it must assert the interrupt after gaining control of the bus but before relinquishing its selection as next bus master. Thus the processor knows that it still shouldn't continue to fetch the next instruction, but must wait for the interrupt to be completed. Along with asserting the interrupt, the device asserts the unique memory address, known as the interrupt vector address, containing the starting address of the device service routine. Address vector +2 contains the new PSW to be used by the processor when beginning the service routine. After recognizing the interrupt, the processor reads the vector address and saves it in an internal register. It then pushes the current PSW and program counter onto the stack and loads the new PC and PSW from the vector address specified. The service routine is then executed.

#### NOTE

These operations are performed automatically and no device polling is required to determine which routine to execute.

The device service routine can cause the processor to resume the interrupted process by executing the return from interrupt, RTI, instruction which pops the top two words from the processor stack and transfers them back to the PC and PS registers.

#### 2.4.3 Data Transfers

After asserting control of the Unibus, the device does not release control until it has completed either one or more data transfers or an interrupt. Typically, only one transfer is

completed each time the device gains control of the bus because few single devices can give or receive information at the maximum Unibus rate. Holding the bus for multiple transfers inhibits other devices from using the bus.

A transfer is initiated by the master device asserting a slave address and control signals on the bus and a master or address validity signal. The appropriate slave recognizes the valid address, reads or writes the data, and responds with a transfer complete signal. The master, recognizes the transfer complete, sends or accepts data and drops the address validating signal. It then can assert a new address and repeat the process or release control of the bus completely.

The importance of this type of structure is that it enables direct device to device transfers without any interaction from the central processor. An NPR device, such as a high speed CRT display, can gain fast access to the bus and transfer data at high rates while refreshing itself from memory without slowing down the processor.

## CHAPTER 3 UNPACKING AND INSTALLATION

### 3.1 INTRODUCTION

The computer is shipped ready to operate in either a protective box or a 19-inch cabinet. Unless required by peripherals, there are no special shipping mounts internal to the computer. Prior to final electrical testing each computer is thermal cycled, vibrated, and subjected to mechanical shock with all modules in place.

Basic computers are shipped in the package illustrated in Figure 3-1. Sufficient hardware is included in the shipping carton to rack mount the computer.

### 3.2 UNPACKING

The basic computer should be carefully removed from its box. Slide mounts are attached to the computer, but mounting screws are packed in a bag located in the same box. Also included is one 83600 SCL (Serial Communication Line) cable and two keys for the console lock. The 83600 SCL cable has a Berg 127009-0, 40-pin connector on one end that matches the SCL output connector on the computer. The other end of the 83600 SCL cable terminates in a Mate-N-Lock 1209340 that matches that used on the VT05, LA30, and ASR Teletype Model 33.

If the computer was ordered as a system with options requiring small peripheral controllers, the controllers may be inside the computer box. Small peripheral controllers are used to interface options such as line printers and paper-tape reader/punches as well as to implement programmable clocks and bootstrap loaders.

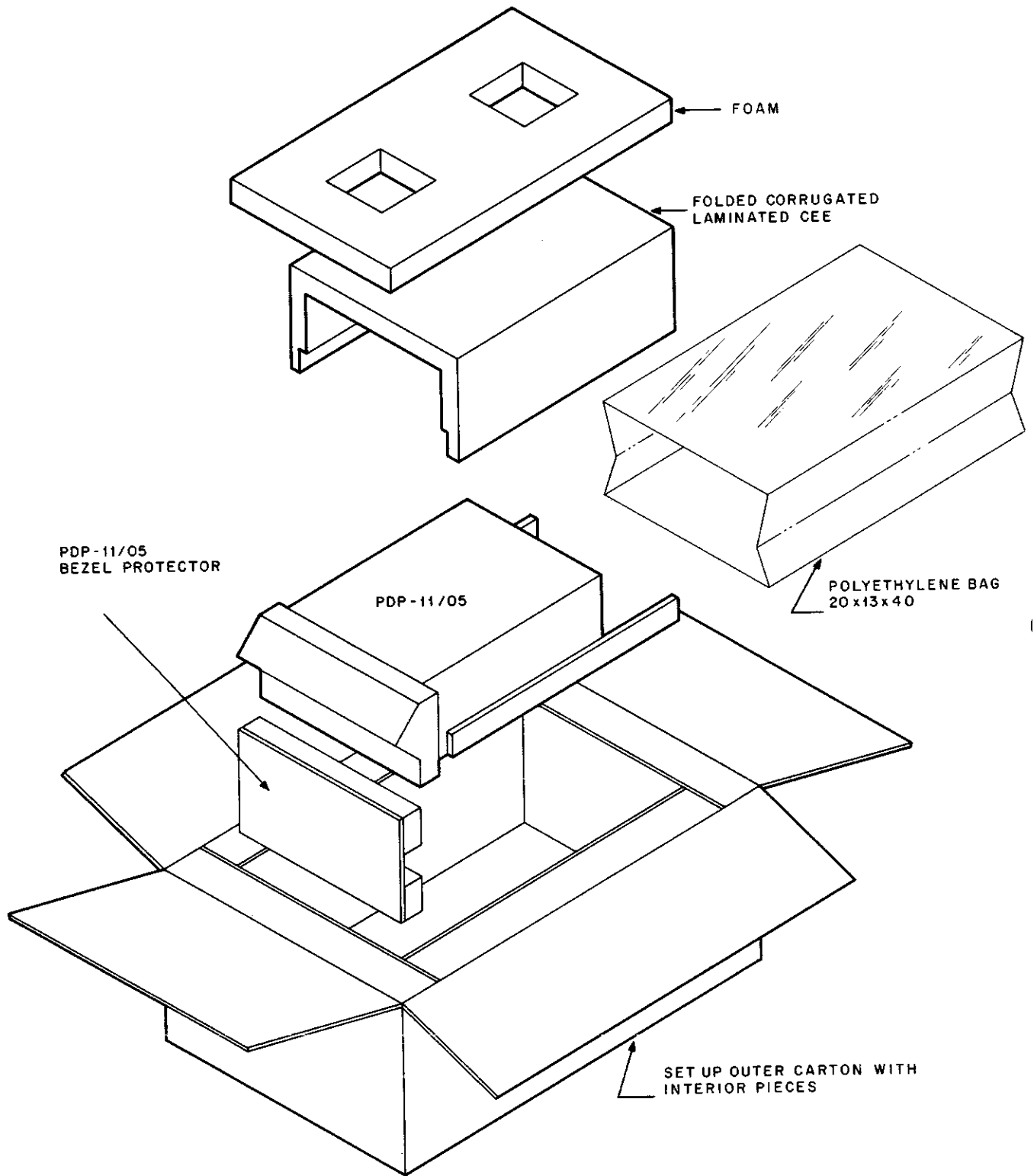


Figure 3-1 Computer Packaging

After removing the computer from its package, it should be inspected for damage. It is advisable to save the packing carton in case it is necessary to return the unit for service.

### 3.3 MECHANICAL DESCRIPTION

Figure 3-2 illustrates the 5-1/4 by 19 by 20 inch computer mounting box, including rack-mountable slide and console. The removable top cover of the mounting box is fastened by four Cam-Lock screws. The removable side panel is fastened by four Phillips-head screws.

Figure 3-3 shows the mounting box with the top cover removed. The backplane unit divides the power supply from the memory and processor side of the mounting box. The internal SCL cable runs from the backplane under the power supply unit to the rear of the mounting box.

Figure 3-4 depicts the mounting box with top cover and side panel off, and the processor and memory modules plugged in. In this case, the computer is a Configuration 2 machine, using an MM11-L, 8K memory unit. Three small peripherals are shown with the external cables attached. A G727 Grant Continuity Card is in the top peripheral slot and a M930 Unibus Terminator Card is in slot A3, B3 (Figure 3-6 also). In Figure 3-5, the Unibus cable is in place, replacing the Unibus terminator card.

Figure 3-6 shows the mounting box without any modules. The path of the console cable is under the M7260 Processor Module. The cable then comes over to plug into the top of the M7260. The module guides aid in inserting the modules into the proper slots.

Figure 3-7 is a rear view of the mounting box with attached rack-mountable slides. If the computer contains any peripheral controllers outside the mounting box, the Unibus is extended from under the top cover. The power control circuit breaker protects the power supply from overload. It is rated at 7A for 110 V units or 4A on 230 V. The SCL connector and ac remote power control connectors are also shown.

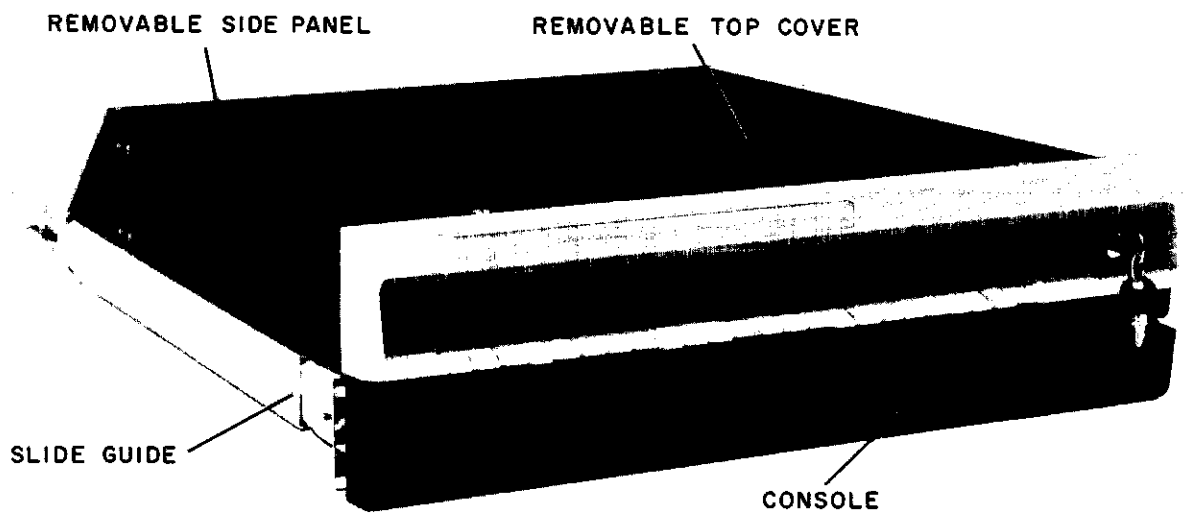


Figure 3-2 Computer Mounting Box

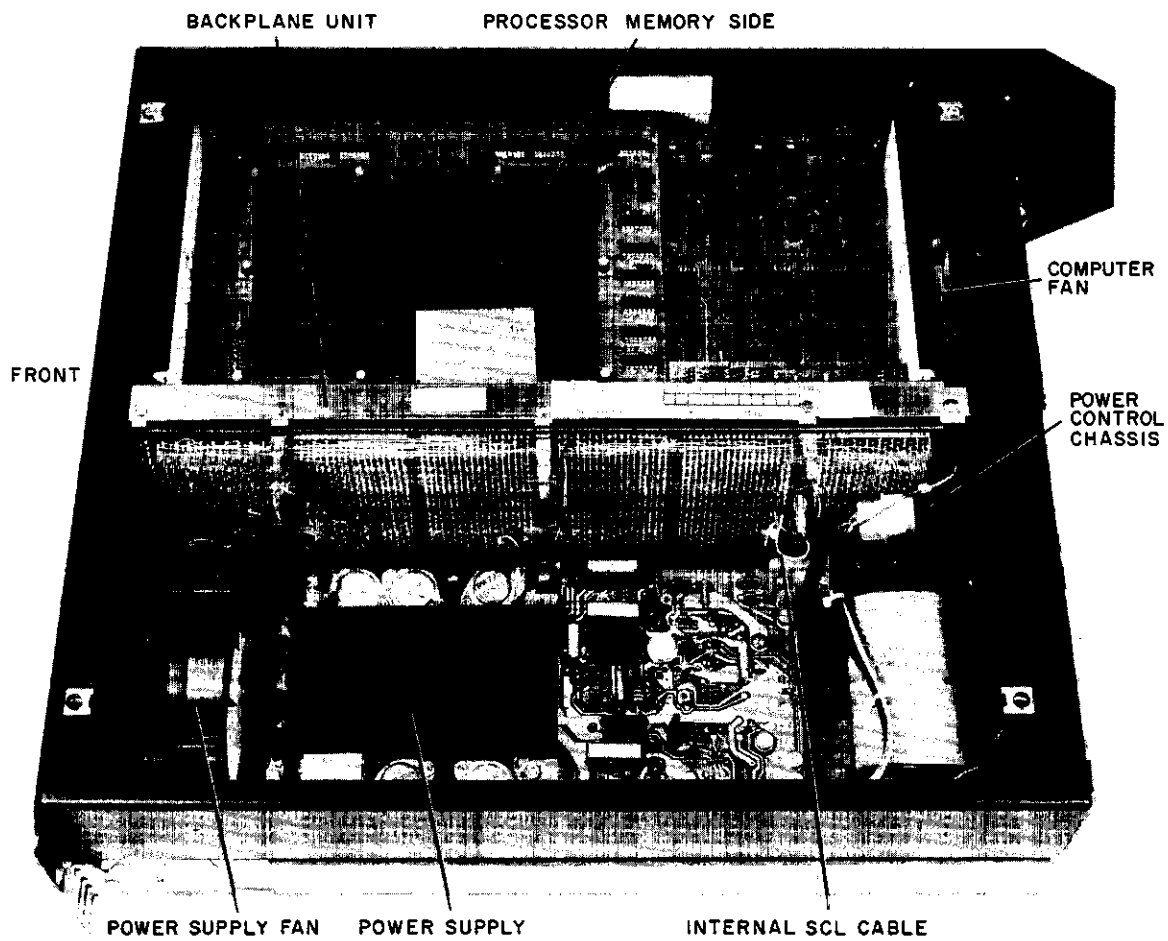


Figure 3-3 Computer Box With Top Cover Removed



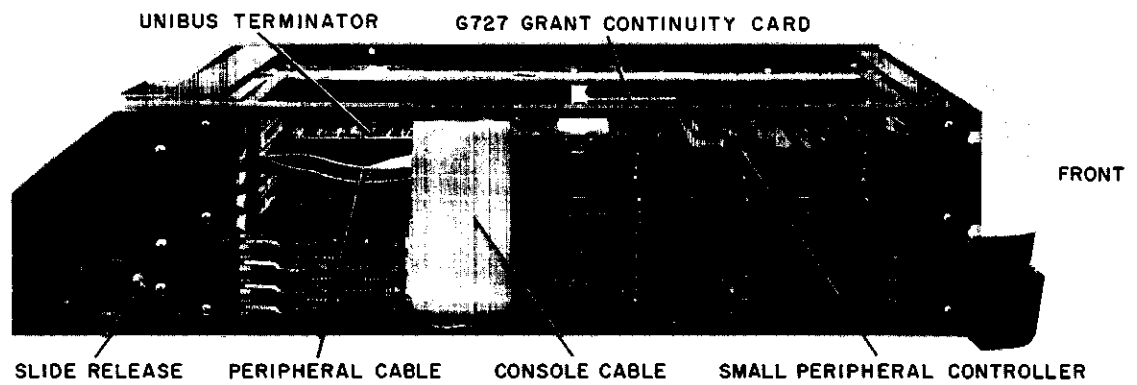


Figure 3-4 Computer Box with Top and Side Covers Removed

UNIBUS CABLE AND UNIBUS CABLE CLAMP

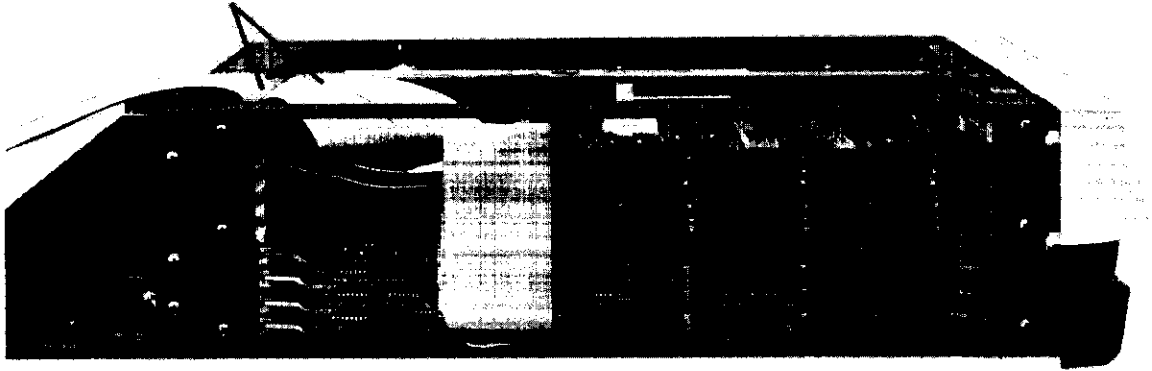


Figure 3-5 Computer Chassis (showing both peripheral cables and the Unibus)

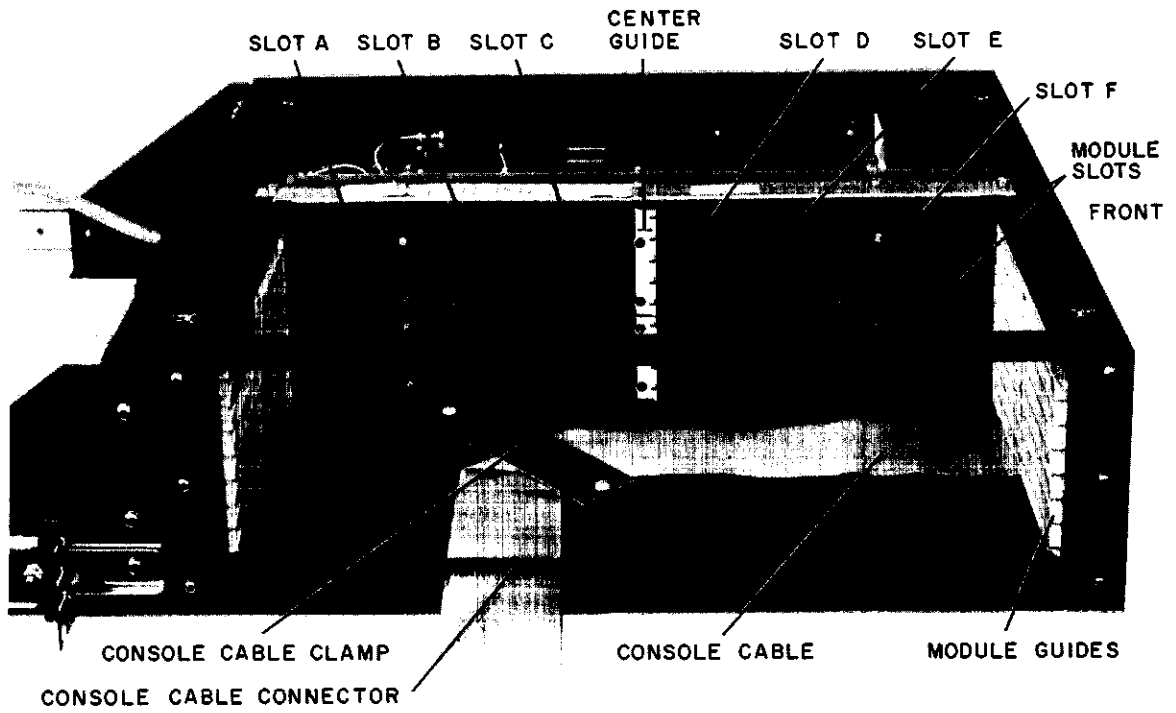


Figure 3-6 Mounting Box Without Modules

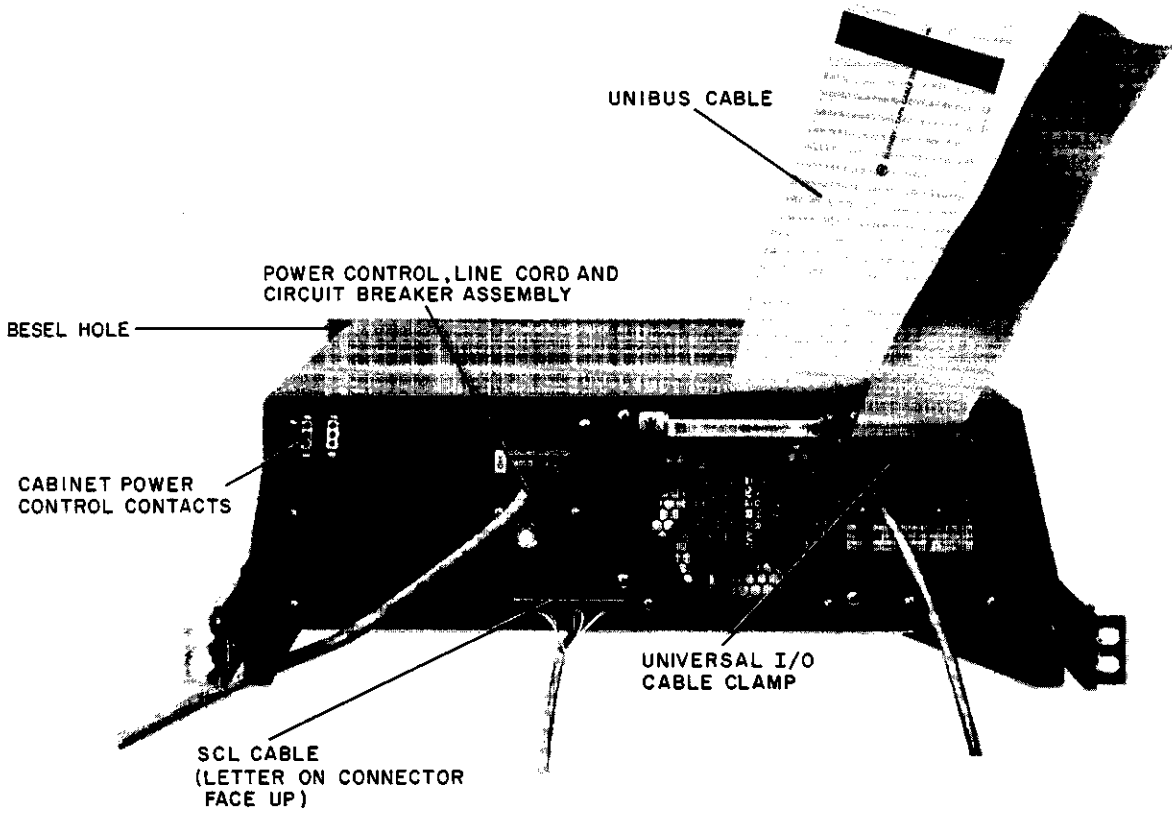


Figure 3-7 Rear of Computer Showing Cable Strain Reliefs

## 3.4 INSTALLATION

The computer mounts in a standard 19-inch wide by 20-inch deep equipment bay. The computer is mounted on slides for easy service. To mount the unit, first attach the fixed portion of the slides to the cabinet; the fixed portion of the slides can be removed from the computer by actuating the slide release shown in Figure 3-4.

Be sure to mount the slides so that the fixed guides are parallel and level with the ground.

### 3.4.1 Mounting Computer on Installed Slides

Once the slide guides have been securely fastened in the cabinet using all eight screws, lift the computer and slide it carefully onto the slide guides until the slide release locks. Carefully lift the slide release and push the computer fully into the rack, being careful not to tear any existing cabling.

The computer should then be fully extended until the slide release locks. As shown in Figure 3-4, the panel on the module side of the computer should be removed to permit installation of I/O cables and the Unibus if required. The panel is removed by loosening and removing four Phillips-head screws.

### 3.4.2 Securing Computer to Cabinet Rack

If the rack-mounted computer is used in a moving environment, it must be secured to the cabinet rack to prevent the machine from moving on its slides. This option, if desired, is implemented as follows:

1. Remove the console bezel from the computer by removing the four screws at the rear of the bezel, being careful not to tear the cable that connects the console and processor.
2. Drill the partial 7/32 inch holes at each top inside corner of the bezel through from the rear of the bezel (Figure 3-7).
3. Counter-bore the 7/32 inch holes at the front of the console bezel 1/2 inch in diameter.
4. Replace the console bezel.

5. Use two 10-32 x 2 inch Phillips-head screws and two Tinnerman nuts to secure the computer to the cabinet rack through the bezel holes at the desired rack position.
6. To make the 10-32 x 2 inch Phillips-head screws captive, notch a 1/8 inch long segment in each 10-32 x 2 inch Phillips-head screw just above the threads and insert a 1/8 I.D. O-Ring in each notch.

### 3.4.3 Installation of I/O Cables

Flat and round I/O cables should be fed through the Universal I/O cable clamp shown in Figure 3-6 for strain relief. They should then be connected to the appropriate small peripheral controllers. Note that the strain relief clamp prevents tension on the cables from damaging the connector block inside the computer. The wide Unibus cable, if required, should be folded as shown in Figure 3-5 and routed over and through a clamp attached to the top of the fan as shown in Figure 3-7. Note that there is a guide extending from the fan that prevents the Unibus cable from blocking air flow to the computer.

As shown in Figure 3-4, systems in which the Unibus is terminated in the computer box must have an M930 Terminator Card in slot A3-B3 as well as slot A5-B5.

## 3.5 INTERCHANGEABLE PERIPHERAL SLOTS

Note that the four peripheral slots in Configuration 2 are identical; therefore, it is possible to arrange the small peripheral controllers for the best mechanical convenience. For instance, if it is necessary to diagnose a failure in a small peripheral controller, it may be convenient to place the selected option in the top slot where its components will be exposed.

## 3.6 SIDE AND TOP COVER INSTALLATION

Figures 3-4 and 3-5 show the computer ready for installation of the side cover. Note that the console cable is folded into a flat loop in order to clear the side cover. Attached to the side cover is the continuation of the left-hand slide. All four 8-32 screws that hold the cover in place should be inserted and tightened securely. The top cover can now be installed using the four Cam-Lock screws.

### 3.7 AC POWER SUPPLY CONNECTION

Computers designed for use on 115 Vac circuits are equipped with a three-prong connector, which when inserted into a properly wired 115 Vac outlet grounds the case of the computer. It is unsafe to operate the computer unless the case is grounded because normal leakage current from the power supply flows into metal parts of the chassis.

If the integrity of the ground circuit is questionable, the user is advised to measure the potential between the computer case and a known ground with an ac voltmeter.

#### 3.7.1 Connecting to Voltages Other than 115V

The computer will operate at voltages ranging from 95V to 135V and from 190V to 270V (47 Hz - 63 Hz), providing the proper power control is attached to the computer. The computer is ordered for nominal voltages of 115V or 230V. The standard three-prong connector for 115V is identical with that found on most household appliances. A standard three-prong connector is also used for 230V.

On installations outside of the United States or where the National Electrical code does not govern building wiring, the user is advised to proceed with caution.

#### 3.7.2 Quality of AC Power Source

The computer is a complex electronic device. Computer systems consisting of CPU, memory, and peripherals are often sensitive to the interference present on some ac power lines. If a computer system is to be installed in an electrically "noisy" environment, it may be necessary to condition the ac power line. DEC Field Service engineers can assist customers in determining if their ac line is satisfactory.

### 3.8 CABINET POWER CONTROL

Provisions have been made for the computer switch to operate a cabinet power control. This feature permits the computer key lock switch to control the power supply for peripherals

attached to the computer (refer to Part 4). The power control contacts are closed when the key lock switch is in the POWER or PANEL LOCK positions. The wiring diagram for a typical cabinet power control systems is shown in Figure 3-8. The power control contacts of the computer may be used to switch a maximum of 230V at 4A.

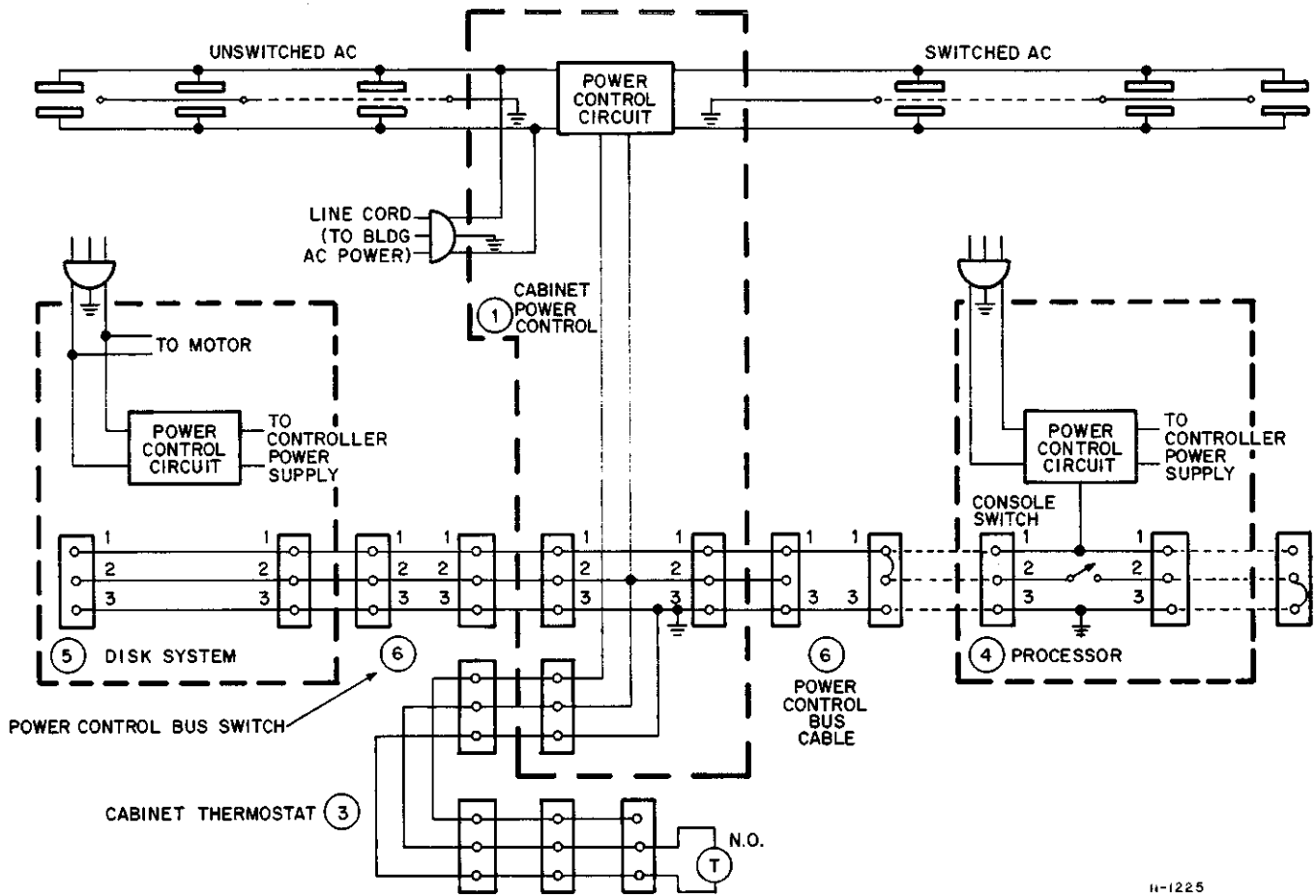


Figure 3-8 Typical Cabinet Power Control System Wiring Diagram

### 3.9 INSTALLATION CERTIFICATION

Once the computer has been installed, it is seriously recommended that a system diagnostic be run to ensure that the equipment operates correctly and that installation has been properly performed. Because system configurations widely vary, no one diagnostic will completely exercise all the attached devices.



It is recommended that the MAINDEC User's Manual that comes with the diagnostic package be consulted for the appropriate diagnostic to be run, depending upon the attached devices. The MAINDEC User's Manual lists the devices that each diagnostic will exercise. The three system exercisers presently available are T17 System Exerciser (MAINDEC-11-DZKAP) for relatively small systems, General Test Program (MAINDEC-11-DZQGA) for medium to large systems, and Communications Test Program (MAINDEC-11-DZQCA) for communications-oriented systems. At least one of the above diagnostics and, if appropriate, the other two, should be used to verify system operation.

Once the diagnostic is selected, the respective diagnostic write-up should be consulted for specific operating instructions. If the user is not familiar with console operation and/or procedures for loading paper tapes, he should read Part 1, Chapter 4, Operation of this manual.

### 3.10 WARRANTY SERVICE (Domestic Only)

If the machine is still covered under the 30 day return-to-factory warranty, and it is desired to return it for factory service, the following procedure should be used. If the machine is no longer on warranty, the local DEC Field Service office should be contacted.

1. Call the Maynard, Massachusetts Repair Depot, Telephone 617-897-5111, X4079 or X2135.
2. The caller will receive an RA (Return Authorization) number, which must appear on the shipping label of the package being returned.
3. Package the machine in an equivalent shipping container, similar to the one the computer arrived in. If possible, use the original computer shipping container.
4. Send the machine to the following address:

Digital Equipment Corporation  
146 Main Street  
Maynard, Massachusetts 01754  
Atten: Depot Repair, Bldg. 21-4

RA # XXXX

## 4.1 INTRODUCTION

This chapter assumes that the computer is installed and connected to the ac power line. It is also assumed that the reader has access to the appropriate diagnostic materials, and a copy of the absolute loader paper tape. It is further assumed that the user is using paper tapes to load software and diagnostics. For systems that have mass storage services, i.e., Disks or DEC tape, the user should refer to the appropriate software manuals for mass storage operating systems.

## 4.2 POWER SWITCH OPERATION

The key-lock power switch shown in Figure 4-1 has three positions.

OFF	-	Fully counterclockwise
POWER	-	90° clockwise from OFF
PANEL LOCK	-	180° clockwise from OFF

In the OFF position, ac power is removed from the primary of the computer power supply, and the cabinet power control contacts are open-circuited. In the other two positions, the ac power is applied to the computer power supply and the cabinet power control contacts are short-circuited. In the POWER position, the console function switches (the right six switches in Figure 4-1) are fully operative. In the PANEL LOCK position, the console function switches have no effect on the computer's operation. PANEL LOCK is used to secure a running computer from mischievous tampering.

## 4.3 FUNCTION SWITCHES

The right six switches in Figure 4-1 are called function switches. They are listed below in order of their appearance from left-to-right.

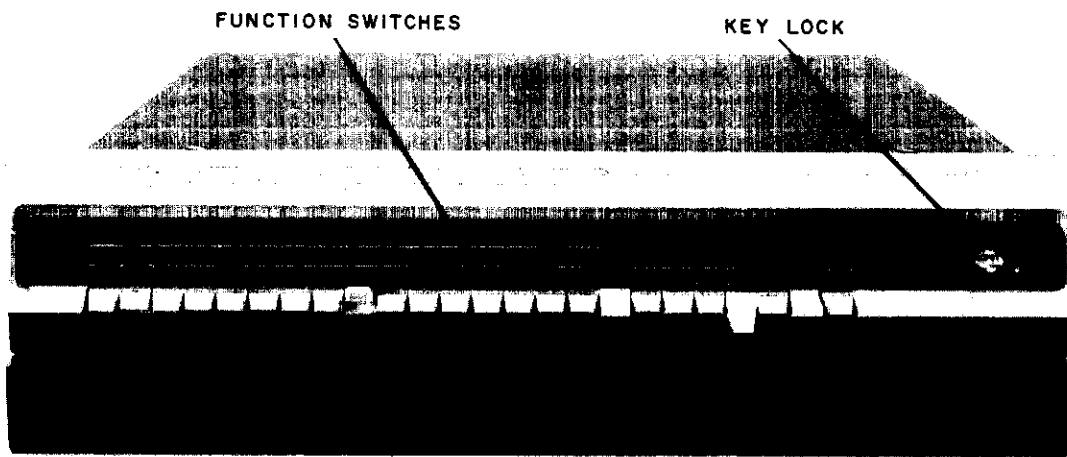


Figure 4-1 Console Illustrating Switch Movements

1. LOAD ADDRESS
2. EXAMINE
3. CONTINUE
4. ENABLE/HALT
5. START
6. DEPOSIT

Function switches 1 through 5 are actuated by being depressed as shown by the ENABLE/HALT switch in Figure 4-1. The DEPOSIT switch must be lifted for actuation. All of the function switches with the exception of ENABLE/HALT are spring loaded and return to their rest state when released.

#### 4.4 ADDRESS/DATA SWITCHES

The 16 ADDRESS/DATA switches are to the left of the function switches in Figure 4-1. These two position switches represent a manually set flip-flop register with up position representing a logical 1 and the down position a logical 0. The ADDRESS/DATA switches may be used in conjunction with the function switches or in conjunction with a program stored in the computer's memory. The ADDRESS/DATA switches are often referred to as "the Switch Register" in DEC documentation. In Figure 4-1, the contents of the Switch Register is equal to  $200_8$  because bit 7 is set to a 1 and all others are set to a 0.

#### 4.5 CONSOLE INDICATORS

There are 17 indicators on the computer console. The contents of the 16 ADDRESS/DATA lights either represent a 16-bit Unibus address or the contents of a 16-bit Unibus Address. Note that the state of the ADDRESS/DATA lights is defined only when the computer RUN light is not illuminated.

#### 4.6 OPERATION CONSOLE

The following paragraphs describe the operation of the function switches. Table 4-1 indicates the meaning of the ADDRESS/DATA lights for all cases where the contents of these lights are defined.

Table 4-1  
Significance of ADDRESS/DATA Indicators

Action	Qualification	Information Displayed In ADDRESS/DATA Indicators
POWER ON	<ol style="list-style-type: none"> <li>1. ENABLE/HALT switch in HALT position</li> <li>2. ENABLE/HALT switch in ENABLE position</li> </ol>	<ol style="list-style-type: none"> <li>1. Contents of location (24)<sub>8</sub></li> <li>2. Undefined. Depends on contents of memory</li> </ol>
LOAD ADDRESS	LOAD ADDRESS switch depressed	Contents of Switch Register
EXAMINE	<ol style="list-style-type: none"> <li>1. EXAM switch depressed</li> <li>2. EXAM switch released</li> </ol>	<ol style="list-style-type: none"> <li>1. Unibus address that is to be examined.</li> <li>2. Contents of Unibus address that was examined.</li> </ol>
DEPOSIT	<ol style="list-style-type: none"> <li>1. DEP switch raised</li> <li>2. DEP switch released</li> </ol>	<ol style="list-style-type: none"> <li>1. Unibus address that is to be deposited.</li> <li>2. Contents of Switch Register which is the data deposited.</li> </ol>
RUN LIGHT ON		Undefined
PROGRAM HALT	<ol style="list-style-type: none"> <li>1. ENABLE/HALT switch in HALT position</li> <li>2. HALT instruction executed</li> <li>3. Double bus error which is two successive attempts to access non-existent memory or improper odd byte address.</li> </ol>	<ol style="list-style-type: none"> <li>1. Address of instruction to be executed when CONT switch is actuated.</li> <li>2. Same as 1.</li> <li>3. Contents of Program Counter (R7) at time when double bus error occurred.</li> </ol>
PROGRAM EXECUTION	<ol style="list-style-type: none"> <li>1. START switch depressed</li> <li>2. CONT switch depressed</li> </ol>	<ol style="list-style-type: none"> <li>1. Address of Last Load address</li> <li>2. Address of instruction to be executed.</li> </ol>

#### 4.6.1 LOAD ADDRESS Switch

Depressing the LOAD ADDRESS switch when the computer is halted causes the contents of the Switch Register to be stored in a temporary register within the computer. This data is also displayed in the ADDRESS/DATA lights for verification. The LOAD ADDRESS operation:

1. Selects a Unibus address for a subsequent EXAM operation.
2. Selects a Unibus Address for a subsequent DEPOSIT operation.
3. Selects the starting address of a program.

#### 4.6.2 EXAM Switch

The EXAM switch permits the display of the contents of a cell in the Unibus address space in the ADDRESS/DATA lights. To examine a 16-bit cell, first select the appropriate address in the Switch Register and depress the LOAD ADDRESS switch. Then depress and release the EXAM switch.

The contents of the selected address will then be displayed in the ADDRESS/DATA lights.

Several features are built into the examine function to aid in programming the computer.

1. While the EXAM switch is depressed, the address to be examined is displayed. The data itself is displayed when the switch is released.
2. If the EXAM switch is repeatedly depressed, the Unibus address is incremented by two on each\* depression. This permits the examination of a list of addresses without repeated LOAD ADDRESS operations.
3. If an attempt is made to examine non-existent memory, it is necessary to perform the initialize operation explained in Paragraph 4.7.
4. Only full words are displayed in the ADDRESS/DATA lights; thus, bit 0, the byte address bit, is ignored when using the EXAM switch with the following exception. Note that the general registers are located on byte addresses. Therefore, when examining the general registers, address bit 0 is recognized and the increment feature is modified such that sequential registers may be examined by repeated use of the EXAM switch.

Note that the EXAM switch has no effect while the computer is in the RUN state or when the key operated power switch is in the PANEL LOCK position.

\*The Unibus address is incremented by one when examining general registers.

### 4.6.3 DEPOSIT Switch

The physical operation of the DEP switch requires that it be lifted for actuation. The DEP switch permits the contents of the Switch Register to be deposited in a Unibus address, which is typically specified by a previous LOAD ADDRESS operation. To deposit the instruction BRANCH SELF (777<sub>8</sub>) in location 200<sub>8</sub>, first set the Switch Register to 200<sub>8</sub> as shown in Figure 4-1 and actuate the LOAD/ADDRESS switch. Set the Switch Register to 777<sub>8</sub> then lift and release the DEPOSIT switch.

Several additional features are built into the deposit function:

1. While the DEP switch is actuated, the Unibus address to be effected is displayed in the ADDRESS/DATA lights. When the switch is released, the data deposited is displayed for verification.
2. If the DEP switch is repeatedly depressed, the Unibus address is incremented by two on each\* depression. This permits the depositing of an entire program with only one LOAD ADDRESS operation.
3. If an attempt is made to deposit into non-existent memory, it is necessary to perform the initialize operation explained in Paragraph 4.1.
4. All deposit operations affect full 16-bit words. Bit 0 of the address is used only when depositing into general registers. Otherwise, bit 0 of the address is ignored on deposit operations.

\*The Unibus address is incremented by one when depositing into general registers.

### 4.6.4 ENABLE/HALT Switch

Place the ENABLE/HALT switch in the HALT position (Figure 4-1); the computer will halt at the end of the current instruction, providing the switch is not in the PANEL LOCK position. All interrupts and traps will be executed prior to halting. This switch may be used in conjunction with the CONT switch to step through programs (Paragraph 4.6.6). With the ENABLE/HALT switch in the ENABLE position, programs may be executed once started by:

1. The START switch.
2. The CONTINUE switch.
3. The Auto-Restart power-up sequence.

#### 4.6.5 START Switch

The sequence for starting a program from the console is as follows:

1. Set the starting address of the program in the Switch Register.
2. Depress the LOAD ADDRESS switch.
3. Position the ENABLE/HALT switch in the ENABLE position.
4. Depress and release the START switch.

While the START switch is depressed, the following actions occur.

1. An initialize signal is generated on the Unibus. This initialize signal serves to reset all peripherals.
2. The Program Status Word is reset to zero.
3. The program counter, R7, is loaded with the last address loaded with the LOAD ADDRESS switch.

When the START switch is released, program execution begins with the instruction contained in the location specified by R7 and the RUN light is turned on. If the ENABLE/HALT switch is in the HALT position, the computer remains in the HALT state following the release of the START switch.

Observe the following cautions when using the start switch:

#### CAUTIONS

1. If the keylock is not in the PANEL LOCK position, depressing the START switch while a program is running initializes the computer system and restarts the program.
2. It is good practice to precede every program START with a LOAD/ADDRESS operation.
3. A program should not be started at an odd address. If a program is started at an odd address the first fetch operation will be aborted and an odd address trap will be attempted. If the stack pointer, R6, is not properly set up, the program in memory may be destroyed.

#### 4.6.6 CONTINUE Switch

The CONTINUE switch is used to continue a program without altering the program counter, R7, or the machine state. To continue a halted program, depress and release the CONTINUE switch. The program is resumed when the CONTINUE switch is released.



The CONT switch is used with the ENABLE/HALT switch to step through programs one instruction at a time. If the CONTINUE switch is actuated while the ENABLE/HALT switch is in the HALT position (Figure 4-1), a single instruction will be executed. Note that interrupts are serviced in single instruction mode. In single step mode, the address of the next instruction to be executed is displayed in the lights.

#### 4.7 UNCONDITIONAL COMPUTER AND UNIBUS INITIALIZATION

Unconditional initialization of the computer system most often arises due to an attempt to Examine from or Deposit into non-existent memory from the console. However, a peripheral or processor error may occur that can only be overcome by initializing the system from the console. The procedure is simply to depress the START switch with the ENABLE/HALT switch in the HALT position.

#### 4.8 LOADING PROGRAMS FROM PAPER TAPE

When the computer is first received, the content of its memory is not defined (it "knows" absolutely nothing, not even how to receive paper-tape input). However, the computer can accept data when toggled directly into core using the console switches. The Bootstrap Loader program is the first program to be loaded, and therefore must be toggled into core. The Loaders described in this section facilitate the loading of programs from the either low- or high-speed paper-tape readers. The low-speed reader is part of the Teletype Model 33 ASR and is operated via the SCL. The high-speed reader is DEC part number PC-11.

The Bootstrap Loader program instructs the computer to accept and store in core data that is punched on paper tape in bootstrap format. The Bootstrap Loader is used to load very short paper-tape programs of  $162_8$  16-bit words or less (primarily the Absolute Loader and Memory Dump Programs). Programs longer than  $162_8$  16-bit words must be assembled into absolute binary format using the PAL-11A Assembler and loaded into memory using the Absolute Loader.

The Absolute Loader (Paragraph 4.8.2) is a system program that enables data punched on paper-tape in absolute binary format to be loaded into any available memory bank. It is used primarily to load the paper-tape system software (excluding certain subprograms) and object programs assembled with PAL-11A.

The loader programs are loaded into the upper most area of available memory so that they will be available for use with system and user programs. When writing programs, the locations used by the loaders should not be used without restoring their contents; otherwise, the loaders will have to be reloaded because the object program will have altered them.

Memory dump programs are used to print or punch the contents of specified areas of memory. For example, when developing or debugging user programs it is often necessary to get a copy of the program or portions of memory. There are two dump programs supplied in the paper-tape software system: DUMPIT, which prints or punches the octal representation of all or specified portions of memory; and DUMPAB, which punches all or specified portion of memory in absolute binary format suitable for loading with the Absolute Loader.

#### 4.8.1 The Bootstrap Loader

The Bootstrap Loader should be loaded (toggled) into the highest memory bank. The locations and corresponding instructions of the Bootstrap Loader are listed in Table 4-2 and explained below.

Table 4-2  
Bootstrap Loader Instructions

Location	Instruction
xx7744	016701
xx7746	000026
xx7750	012702
xx7752	000352
xx7754	005211
xx7756	105711
xx7760	100376
xx7762	116162
xx7764	000002
xx7766	xx7400
xx7770	005267
xx7772	177756
xx7774	000765
xx7776	YYYYY

In Table 4-2, xx represents the highest available memory bank. For example, the first location of the Loader would be one of the following, depending of memory size, and xx in all subsequent locations would be the same as the first.

Note in Table 4-3 that the contents of location xx7766 should reflect the appropriate memory bank in the same manner as the preceding locations.

Table 4-3  
Memory Bank Assignments

Location	Memory Bank	Memory Size
017744	0	4K
037744	1	8K
057744	2	12K
077744	3	16K
117744	4	20K
137744	5	24K
157744	6	28K

The contents of location xx7776 (YYYYYY in the Instruction column of Table 4-2) should contain the device status register address of the paper-tape reader to be used when loading the bootstrap formatted tapes. Either paper-tape reader may be used, and each is specified as follows:

Teletype Paper-Tape Reader	-	177560
High-Speed Paper-Tape Reader	-	177550

4.8.1.1 Loading the Loader Into Memory - With the computer initialized for use as described in Paragraph 4.7 toggle in the Bootstrap Loader as explained below.

1. Set xx7744 in the Switch Register (SR) and press LOAD ADDRESS (xx7744 will be displayed in the indicators).
2. Set the first instruction, 016701, in the SR and lift DEPOSIT (016701 will be displayed in the indicators).

NOTE

When depositing data into consecutive words, the DEPOSIT automatically increments the address to the next word.

3. Set the next instruction, 000026, in the SR and lift DEPOSIT. Continue to deposit subsequent instructions.
4. Deposit the desired device status register address in location xx7776, the last location of the Bootstrap Loader.

It is a good programming practice to verify that all instructions are stored correctly. This is done by proceeding at step 6 below.

6. Set xx7744 in the SR and press LOAD ADDRESS.
7. Press and release EXAMine (the octal instruction in location xx7744 will be displayed in the indicators so that it can be compared to the correct instruction, 016701). If the instruction is correct, proceed to step 8, otherwise go to step 10.
8. Press EXAMine. The instruction of the location displayed in the ADDRESS/DATA indicators with the switch depressed will be displayed when the switch is released. Compare the indicator contents to the instruction at the proper location.
9. Repeat step 8 until all instructions have been verified or go to step 10 whenever the correct instruction is not displayed.

Whenever an incorrect instruction is displayed, it can be corrected by performing steps 10 and 11.

10. With the incorrect instruction displayed in the ADDRESS/DATA register, set the correct instruction in the SR and lift DEposit. The contents of the SR will be deposited in the location displayed with the key lifted.
11. Press EXAMine to ensure that the instruction was correctly stored.
12. Proceed at step 9 until all instructions have been correctly verified.

The Bootstrap Loader is now in core. The procedures above are illustrated in the flow chart of Figure 4-2.

#### 4.8.2 Loading Bootstrap Tapes

Any paper-tape punched in bootstrap format is referred to as a bootstrap tape and is loaded into memory using the Bootstrap Loader. Bootstrap tapes begin with about two feet of special bootstrap leader code (ASCII code 351, not blank leader tape as is required by the Absolute Loader).

With the Bootstrap Loader in memory, it will load the bootstrap tape into memory starting anywhere between location xx7400 and location xx7743, i.e.,  $162_8$  words. The paper-tape input device used is that which is specified in location xx7776. Bootstrap tapes are loaded into memory as explained below.

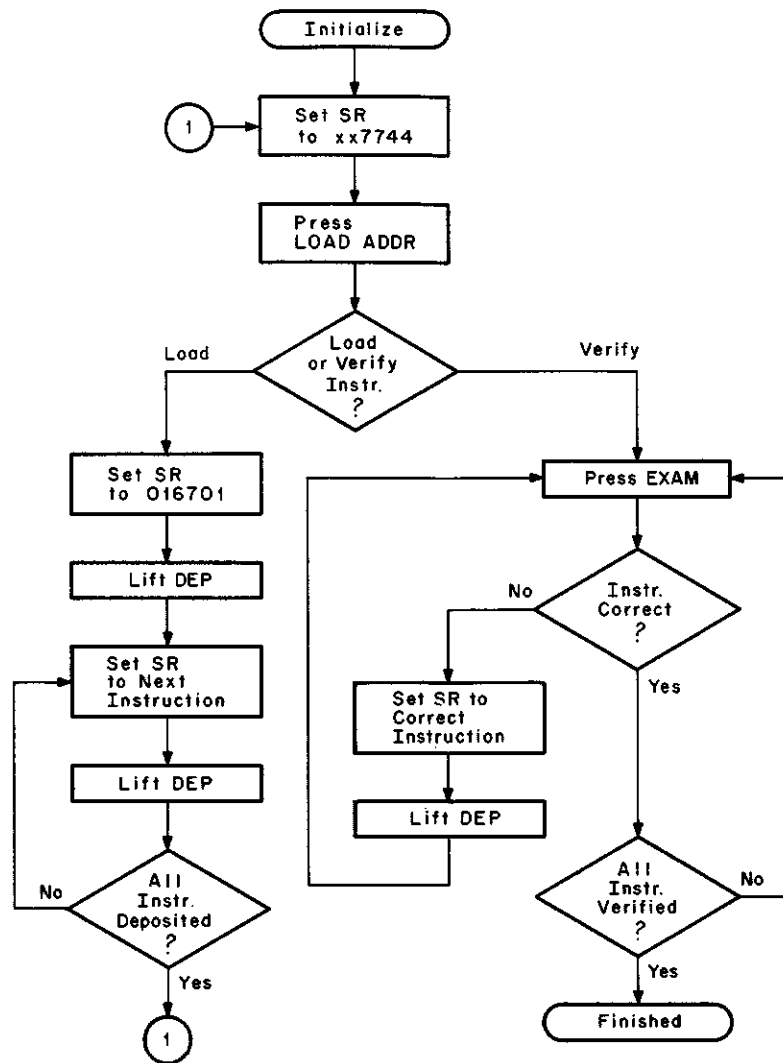


Figure 4-2 Loading and Verifying the Bootstrap Loader

1. Set the ENABLE/HALT switch to HALT.
2. Place the bootstrap tape in the specified reader with the special bootstrap leader code over the reader sensors (under the reader station).
3. Set the SR to xx7744 (the starting address of the Bootstrap Loader) and press LOAD ADDR.
4. Set the ENABLE/HALT switch to ENABLE.
5. Press START. The bootstrap tape will pass through the reader as data is being loaded into memory.
6. The bootstrap tape stops after the last frame of data (Figure 4-5) has been read into memory. The program on the bootstrap is now in memory.

The procedures above are illustrated in the flowchart of Figure 4-3.

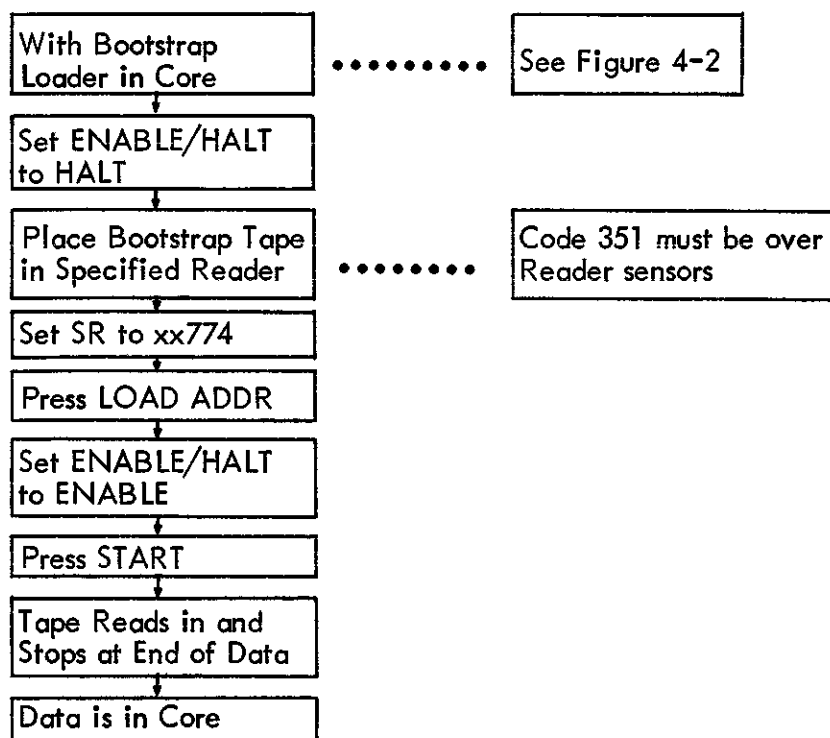


Figure 4-3 Loading Bootstrap Tapes Into Memory

If the bootstrap tape does not read in immediately after depressing the START switch, it is due to any one of the following reasons:

1. Bootstrap Loader not correctly loaded.
2. Using the wrong input device.
3. Code 351 not directly over the reader sensors.
4. Bootstrap tape not properly positioned in reader.

4.8.1.3 Bootstrap Loader Operation - The Bootstrap Loader source program is shown below. The starting address in the example denotes that the Loader is to be loaded into memory bank zero (a 4K system).

```

000001          R1=%1          ;USED FOR THE DEVICE
                                ;ADDRESS
000002          R2 %2          ;USED FOR THE LOAD AD-
                                ;DRESS DISPLACEMENT
017400          LOAD=17400     ;DATA MAY BE LOADED NO
                                ;LOWER THAN THIS
017744          .-17744       ;START ADDRESS OF THE
                                ;BOOTSTRAP LOADER
017744 016701  START:  MOV DEVICE,R1  ;PICK UP DEVICE ADDRESS,
000026          ;PLACE IN R1
017750 012702  LOOP:   MOV #.-LOAD+2,R2 ;PICK UP ADDRESS
000352          ;DISPLACEMENT
017754 005211  ENABLE: INC@ R1       ;ENABLE THE PAPER TAPE
017756 105711  WAIT:   TSTD@R1      ;READER
                                ;WAIT UNTIL FRAME
017760 100376          BPL WAIT      ;IS AVAILABLE
017762 116162  MOVB 2 (R1),LOAD (R2) ;STORE FRAME READ
000002          ;FROM TAPE IN MEMORY
017400
017770 005267          INC LOOP+2     ;INCREMENT LOAD ADDRESS
177756          ;DISPLACEMENT
017774 000765  BRNCH: BR LOOP       ;GO BACK AND READ MORE
                                ;DATA
017776 000000  DEVICE: 0           ;ADDRESS OF INPUT DEVICE

```

The Bootstrap Loader source program is a brief but fairly complex example of the PAL-11A Assembly Language. Explanations of the program and PAL-11A are found in the PDP11 Paper-Tape Software Programming Handbook, DEC-11-GGPB-D.

#### 4.8.2 The Absolute Loader

The Absolute Loader is a system program that enables data punched on paper-tape in absolute binary format to be loaded into any memory bank. It is used primarily to load the paper-tape system software (excluding certain subprograms) and object programs assembled with PAL-11A. The major features of the Absolute Loader include:

1. Testing of the checksum on the input tape to ensure complete, accurate loads.
2. Starting the loaded program upon completion of loading without additional user action, as specified by the .END in the program just loaded.
3. Specifying the load address of position independent programs at load-time rather than at assembly time, by using the desired Loader switch register option.

4.8.2.1 Loading the Loader Into Memory - The Absolute Loader is supplied on punched paper-tape in bootstrap format. Therefore, the Bootstrap Loader is used to load the Absolute Loader into memory. It occupies locations xx7474 through xx7743, and its starting address is xx7500. The Absolute Loader program is 72<sub>10</sub> words long, and is loaded adjacent to the Bootstrap Loader.

4.8.2.2 Loading Absolute Tapes - Any paper-tape punched in absolute format is referred to as an absolute tape, and is loaded into memory using the Absolute Loader. When using the Absolute Loader, there are two methods of loading available: normal and relocated.

A normal load occurs when the data is loaded and placed in memory according to the load addresses on the object tape. It is specified by setting bit 0 of the Switch Register to 0 immediately before starting the load. There are two types of relocated loads.

- a. Loading to continue from where the loader left off after the previous load.

This type is used, when the object program being loaded is contained on more than one tape. It is specified by setting the Switch Register to 000001 immediately before starting the load.

- b. Loading into a specific area of memory.

This is normally used when loading position-independent programs. A position-independent program is one that can be loaded and run anywhere in available memory. The program is written using the position-independent instruction format. The type of load is specified by setting the Switch Register to the LOAD ADDRESS and adding 1 to it (i.e., setting bit 0 to 1).

Optional Switch Register settings for the three types of loads are listed below.

<u>Type of Load</u>	<u>Switch Register</u>	
	<u>Bits 1-14</u>	<u>Bit 0</u>
Normal	(ignored)	0
Relocated - continue loading where left off	0	1
Relocated - load in specified area of memory	nnnn (specified address)	1



The absolute tape may be loaded using either of the paper-tape readers. The desired reader is specified in the last word of available memory (xx7776). The input device status word may be changed at any time prior to loading the absolute tape. With the Absolute Loader in memory as explained in Paragraph 4.8.1.2, absolute tapes are loaded as explained below.

1. Set the ENABLE/HALT switch to HALT.  
To use an input device other than that used when loading the Absolute Loader, change the address of the device status word (in location xx7776) to reflect the desired device, i.e., 177560 for the Teletype reader or 177550 for the high-speed reader.
2. Set the SR to xx7500 and press LOAD ADDR.
3. Set the SR to reflect the desired type of load.
4. Place the absolute tape in the proper reader with blank leader tape directly over the reader sensors.
5. Set ENABLE/HALT to ENABLE.
6. Press START. The absolute tape will begin passing through the reader station as data is being loaded into memory.

If the absolute tape does not begin passing through the reader station, the Absolute Loader is not in memory correctly. Therefore, reload the loader and start over at step 1 above. If it halts in the middle of the tape, a checksum error occurred in the last block of data read in.

Normally, the absolute tape will stop passing through the reader station when it encounters the transfer address as generated by the statement .END, denoting the end of a program. If the system halts after loading, check that the low byte of R0 is 0.\* If so, the tape is correctly loaded. If not 0, a checksum error has occurred in the block of data just loaded, indicating that some data was not correctly loaded. The tape should be reloaded starting at step a above.

When loading a continuous relocated load, subsequent blocks of data are loaded by placing the next tape in the appropriate reader and pressing the CONTINUE switch.

The Absolute Loader may be restarted at any time by starting at step 1.

\* To read R0, LOAD ADDRESS 177700 and press EXAM.

### 4.8.3 Memory Dumps

A memory dump program is a system program that enables the contents of all or any specified portion of memory to be dumped (print or punch) onto the Teletype printer and/or punch, line printer, or high-speed punch. There are two dump programs available in the paper-tape software system:

- a. DUMPIT, which dumps the octal representation of the contents of specified portions of memory onto the teleprinter, low-speed punch, high-speed punch, or line printer.
- b. DUMPAB, which dumps the absolute binary code of the contents of specified portions of memory onto the low-speed punch or high-speed punch.

Both dump programs are supplied on punched paper tape in bootstrap and absolute binary formats. The bootstrap tapes are loaded over the Absolute Loader as explained in Paragraph 4.8.1.3. The absolute binary tapes are position independent and may be loaded and run anywhere in memory as explained in Paragraph 4.8.2.2. DUMPIT and DUMPAB are very similar in function, and differ primarily in the type of output they produce.

4.8.3.1 Operating Procedures - Neither dump program will punch leader or trailer tape, but DUMPAB will always punch ten blank frames of tape at the start of each block of data dumped.

The operating procedures for both dump programs follow:

1. Select the dump program desired and place it in the reader specified by location xx7776 (Paragraph 4.8.1).
2. If a bootstrap tape is selected, load it using the Bootstrap Loader, Paragraph 4.8.1.2. When the computer halts go to step 4.
3. If an absolute binary tape is selected, load it using the Absolute Loader (Paragraph 4.8.2.2), relocating as desired.  
Place the proper start address in the Switch Register, press LOAD ADDRESS and START. (The start addresses are shown in Paragraph 4.8.3.3.)
4. When the computer halts, enter the address of the desired output device status register in the Switch Register and press CONTINUE (low-speed punch and teleprinter = 177564; high-speed punch = 177554; line printer = 177514).

5. When the computer halts, enter in the Switch Register the address of the first byte to be dumped and press CONTInue. This address must be even when using DUMPIT.
6. When the computer halts again enter in the Switch Register the address of the last byte to be dumped and press CONTInue. When using the low-speed punch, set the punch to ON before pressing CONTInue.
7. Dumping will now proceed on the selected output device.
8. When dumping is complete, the computer will halt.

If further dumping is desired, proceed to step 5. It is not necessary to respecify the output device address except when changing to another output device. In such a case, proceed to the second paragraph of step 3 to restart.

If DUMPAB is being used, a transfer block must be generated as described below. If a tape read by the Absolute Loader does not have a transfer block, the loader will wait in an input loop. In such a case, the program may be manually initiated. However, this practice is not recommended because there is no guarantee that load errors will not occur when the end of the tape is read.

The transfer block is generated by performing step 5 with the transfer address in the Switch Register, and step 6 with the transfer address minus 1 in the Switch Register. If the tape is not to be self-starting, an odd-numbered address must be specified in step 5 (000001, for example).

The dump programs use all eight general registers and do not restore their original contents. Therefore, after a dump the general registers should be loaded as necessary prior to their use by subsequent programs.

4.8.3.2 Output Formats - The octal output from DUMPIT is in the following format:

```
xxxxxx>YYYYYY YYYYYY YYYYYY YYYYYY YYYYYY YYYYYY
```

Where xxxxxx is the address of the first location printed or punched, and YYYYYY are words of data, the first of which starts at location xxxxxx. This is the format for every line of output. There are only eight words of data per line, but there can be as many lines as needed to complete the dump.

The output from DUMPAB is in absolute binary, as explained in Paragraph 4.8.2.3.

4.8.3.3 Storage Maps - The DUMPIT program is 87 words long. When used in absolute format, the storage map is as shown in Figure 4-4.

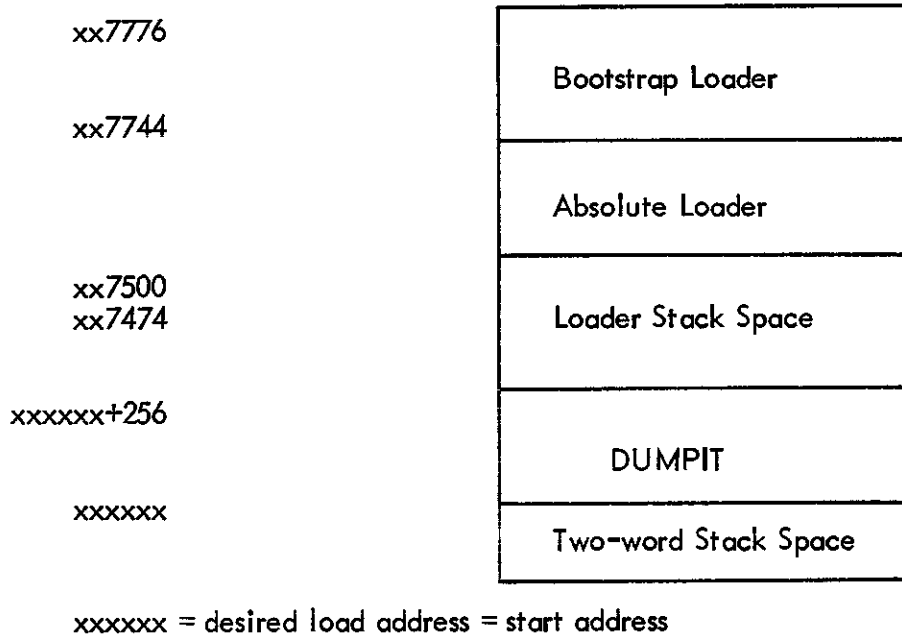


Figure 4-4 Absolute Format

When used in bootstrap format the storage map is as shown in Figure 4-5.

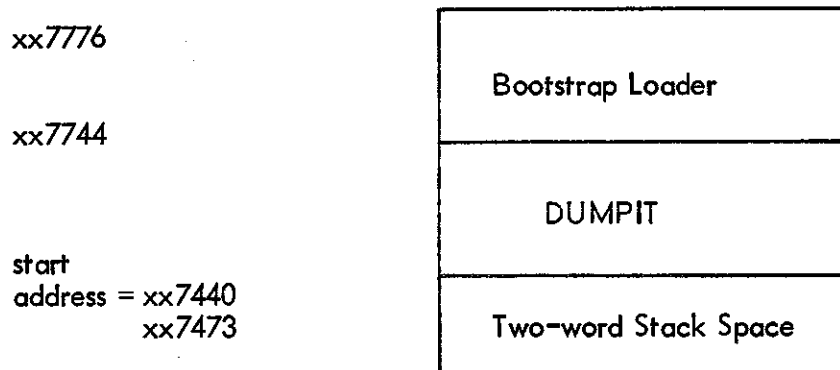


Figure 4-5 Bootstrap Format

## 4.9 INSTALLATION TESTING

There exists a hierarchy of diagnostics for testing the processor, memory, and DEC manufactured peripherals. A summary of these diagnostics and the recommended order of running them is contained in their respective program listings supplied with each diagnostic and

packaged in each PDP-11/05 software kit.

#### 4.10 SCL BAUD RATE ADJUSTMENT

The SCL baud rate is adjusted at the factory to a nominal 110 baud. It may be necessary to adjust the SCL baud rate in the field in order for the equipment to operate properly. The range of adjustment is 110 to 300 baud. The best and most convenient method of adjusting the SCL clock when there is a printing terminal attached to the SCL line is as follows:

1. Load in the diagnostic test T-17 using the Absolute Loader.
2. LOAD ADDRESS 200<sub>g</sub>.
3. Set the Switch Register equal to 376<sub>g</sub>.
4. Adjust the potentiometer on the bottom processor module, M7260, until a consistent type-out appears.
5. Turn the potentiometer counterclockwise until the typeout fails.
6. Then turn the potentiometer clockwise until the type-out again fails counting the turns.
7. Finally, turn the potentiometer counterclockwise until the setting is mid-way between the failure points.

Alternatively, it is possible to observe the SCL clock output by placing the M7260 module on extenders and probing E8406 with an oscilloscope. The clock frequency should be adjusted by turning the potentiometer on the M7260 module to equal 16 times the desired baud rate. For instance, the SCL clock frequency for 110 baud is 1760 Hz with a period of 568  $\mu$ s. Note that the SCL clock is designed to operate terminals. For critical or high-speed applications, the SCL may be driven by an external clock. The maximum baud rate is 10 kHz which requires a clock rate of 160 kHz. DEC standard diagnostics are not guaranteed at SCL baud rates exceeding 300 baud.

PART 2  
KD11-B Processor

## CHAPTER 1 GENERAL DESCRIPTION

### 1.1 INTRODUCTION

Part II provides both general and detailed descriptions of the KD11-B processor and its console, a description of the PDP-11 instruction set, a description of the KD11-B microprogram, and maintenance information. The various chapters of Part II are outlined below:

- Chapter 1 - General Description
- Chapter 2 - Instruction Set
- Chapter 3 - Console Description
- Chapter 4 - Detailed Description
- Chapter 5 - Microprogram Control
- Chapter 6 - Maintenance

The general description of the KD11-B consists of defining the processor and illustrating its use with its peripherals and the Unibus. The KD11-B processor print set found in the Engineering Drawing Manual is often referenced in the KD11-B logic description.

### 1.2 KD11-B DEFINITION

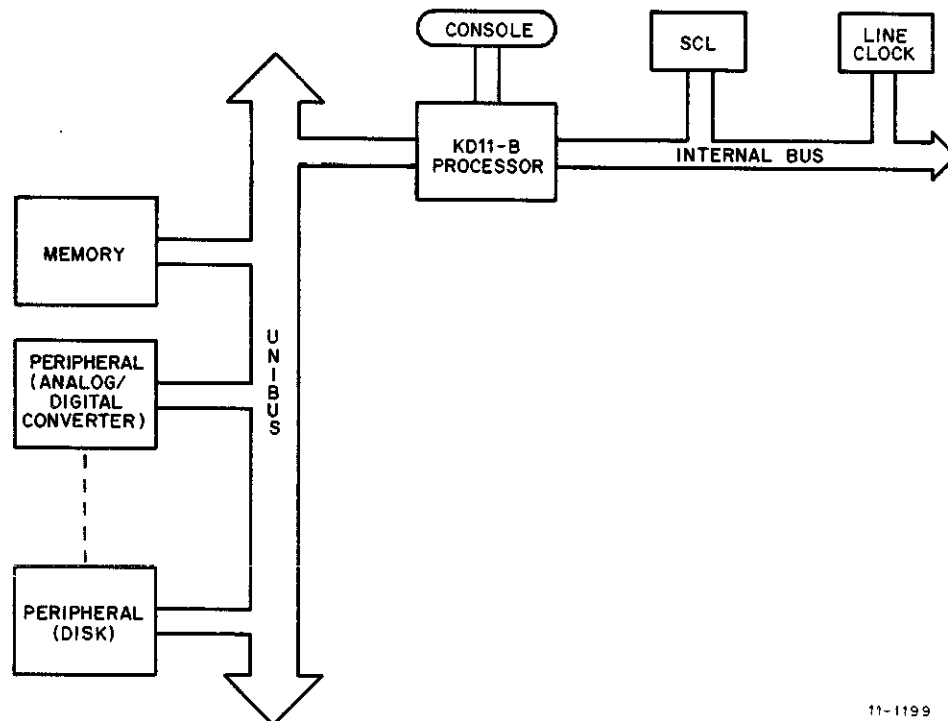
The KD11-B is program compatible with the KA11 used in the PDP-11/15 and PDP-11/20, although the KD11-B executes instructions somewhat more slowly. The instruction set of KD11-B is described in detail in Chapter 2 along with some slight differences between the KD11-B and the KA11 (PDP-11/20).

Physically the KD11-B consists of two 8-1/2 x 15 inch modules, the M7260 and M7261. Each module contains approximately 100 dual-in-line integrated circuits of the 14-, 16-, and 24-pin variety. There is one MOS-LSI 40-pin integrated circuit used on the M7260. This MOS circuit is the Serial Communication Line (SCL) receiver and transmitter. All other integrated circuits used on the KD11-B are bipolar. The connections between the two modules occur through the backplane.

The KD11-B programmer's console interfaces to the processor via a 40-conductor cable that is attached to the M7260 module. The console is described in detail in Chapter 3.

### 1.3 KD11-B AND THE UNIBUS

The processor is interfaced with memory and most peripherals by the Unibus as shown in Figure 1-1. The KD11-B is capable of arbitrating both Bus Requests (BR) and Non-Processor Requests (NPR) as they are asserted onto the Unibus by the connected peripherals.



11-1199

Figure 1-1 KD11-B With Interconnections to Memory and Peripherals



The line clock and the serial communications line (SCL) do not interface with the processor via the Unibus in the traditional PDP-11 sense; both connect to the KD11-B through an internal bus. For most programs, these peripherals are indistinguishable from their appearance on other PDP-11 implementations. In other words, the program may access the line clock and the serial communications line by using instructions that move data to and from the Unibus address specified for these peripheral options in the PDP-11 Peripheral and Interfacing Handbook. These Unibus addresses are as follows:

1. Line Clock Status Register Address = 177546
2. SCL Receiver Status Register Address = 177560
3. SCL Receiver Buffer Register Address = 177562
4. SCL Transmitter Status Register Address = 177564
5. SCL Transmitter Buffer Register Address = 177566

However, it is not possible for the line clock and the serial communications line (SCL) to be addressed by any devices attached to the Unibus other than the KD11-B processor. For example, it is not possible to perform NPRs to the serial communications line from another peripheral such as the DECTape unit.

The serial communications line input/output is available for connection to such devices as the LA30 DECwriter, the VT05 CRT Terminal, or the Model 33 ASR Teletype. These SCL input/output signals interface at the fingers of the processor's M7260 module via a Berg connector located on the rear of the computer chassis as shown in Chapter 3 of Part I.

#### 1.4 KD11-B AS AN INSTRUCTION INTERPRETER

Figure 1-2 illustrates the division of the KD11-B into Unibus control and instruction interpreter. This division is significant because in the KD11-B the Unibus Control is implemented as a block of logic that is relatively independent of the rest of the processor.

In Figure 1-3, the instruction interpreter is further divided into a Data Path (DP), a Data Path Control (DPC), and a Control Store (CS). Whenever power is applied to the computer, the DPC continually executes a program that is stored in the CS. All instructions, interrupt sequences, and console functions are performed by the DPC when executing

a microprogram contained in the CS. The Unibus control and the DP are facilities used by the DPC in the course of performing its tasks. The program contained in the CS is referred to as the microprogram.

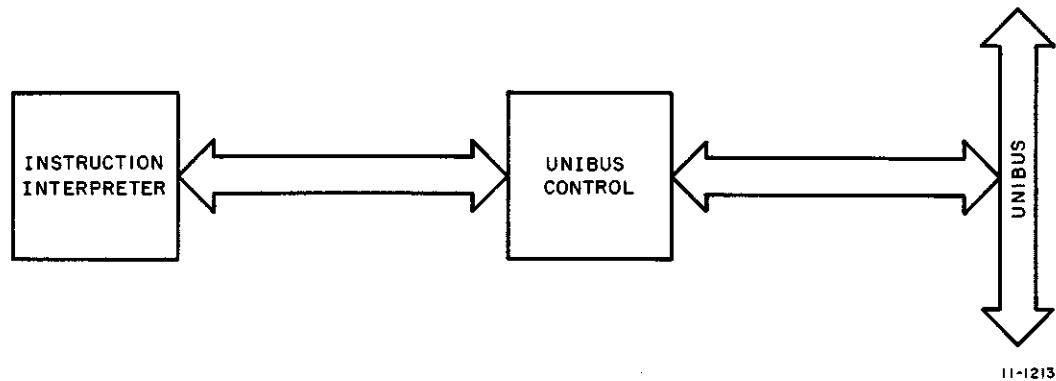


Figure 1-2 KD11-B Processor Block Diagram

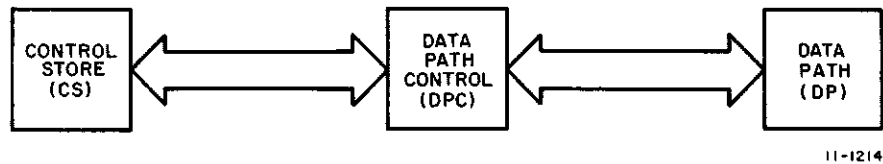


Figure 1-3 Instruction Interpreter Block Diagram

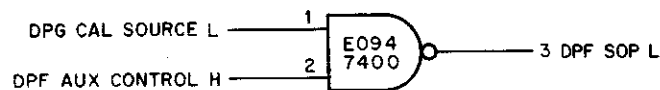
### 1.5 KD11-B PRINT SET

Throughout the remainder of Part II frequent reference will be made to the drawings in the KD11-B print set located in the Engineering Drawing Manual (refer to Table 6-1 in Part II). Each print with its respective engineering drawing number is listed as follows:

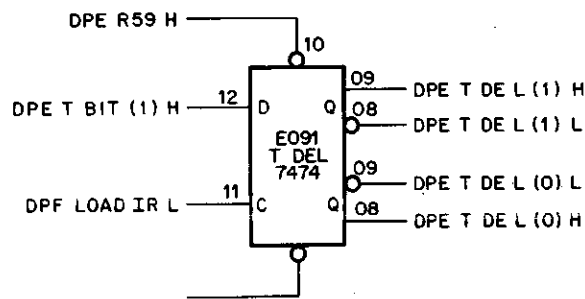
1. The Data Path (M7260): D-CS-M7260-0-01 (9 sheets, DPA to DPH1)

2. The Control Module (M7261): D-CS-M7261-0-01 (11 sheets, CONA to CONJ)
3. The Console: D-CS-5409766-0-1
4. Microprogram Flow Listing: K-MP-KD11-B-1
5. Microprogram Symbolic Listing: K-MP-KD11-B-2
6. Microprogram Binary Listing: K-MP-KD11-B-3
7. Microprogram Cross Reference Listing: K-MP-KD11-B-4
8. Read-Only Memory Maps (ROM): K-RL-M7260-8, K-RL-M7261-8

For this discussion, the prints are referenced by the designations DPA through DPH for the M7260, and CONA through CONF for the M7261. As a general rule, all small scale integrated circuits are shown as individual logic equivalent gates or flip-flops, with pin numbers designated. Figure 1-4 shows an example of a positive NAND gate and a D-type flip-flop.



a) 2 Input Positive Nand Gate.



b) Typical 7474 Flip-Flop

11-1198

Figure 1-4 Typical Small Scale Integrated Circuit Representations

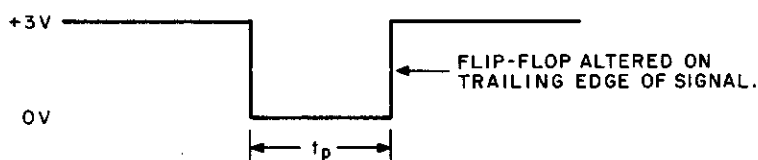
The "E094" contained within the gate (Figure 1-4) indicates the physical location of the dual-in-line integrated circuit on the appropriate module. Integrated circuits pins are referenced using the notation E09403. The first three digits after the "E" refer to the

location of the integrated circuit on the module, and the next two digits are the pin number on that integrated circuit. The prefix of the output signal, in this case DPF, indicates the print name on which the gate appears; the prefix on names of input signals indicates the print page from which the input signal originates, i.e., DPG, DPF. The particular gate illustrated in Figure 1-4 appears on drawing F (D-CS-M7260-0-01, sheet 7). The gate appearing on print DPF is physically located on the M7260 module; however, the input signals come from prints DPG and DPF, and therefore the input signals originate on the M7260 module. Note that signal names with the prefix CONC might originate on CONC or CONC1. Similarly, signal names with the prefix DPH might originate on DPH or DPH1.

Figure 1-4 depicts a typical 7474 flip-flop that appears on drawing DPE (D-CS-M7260-0-01, sheet 6). Several important points are shown: the name of the flip-flop (TDEL); the print it appears on (DPE); and its physical location (E091). Four possible output signal names are available from the flip-flop's two physical outputs:

Physical Output	Signal Names
Q	DPE TDEL (1)H DPE TDEL (1)L
$\bar{Q}$	DPE TDEL (0)L DPE TDEL (0)L

To clock a 7474 flip-flop there must be a pulse input of some duration ( $t_p$ ) to the clock pin. The clock signal for the 7474 flip-flop is shown in Figure 1-5. Note that the signal DPF LOAD IR L is a negative-going pulse. Since the 7474 flip-flop is clocked on the rising edge of a signal, the flip-flop T DEL is clocked on the trailing edge of DPF LOAD IR L.

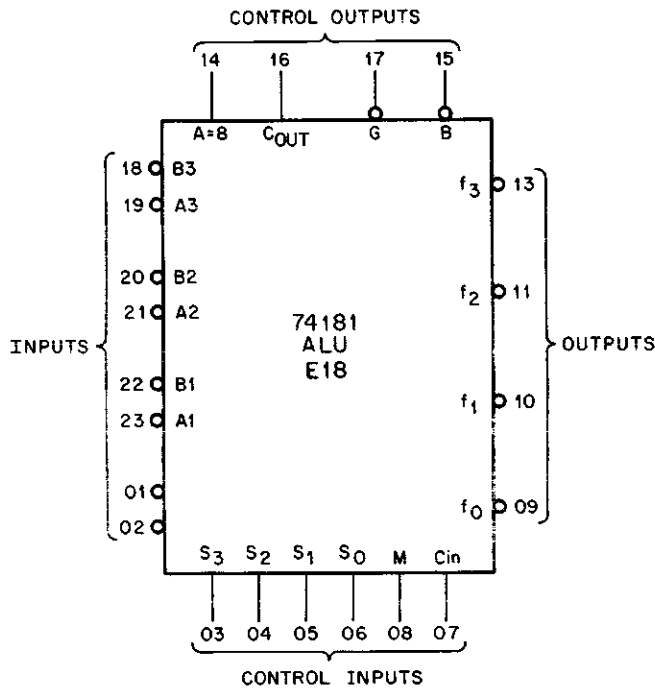


11-1201

Figure 1-5 DPF LOAD IRL Signal

### 1.5.1 Medium and Large Scale Integrated Circuit Representations

MSI and LSI integrated circuits (Figure 1-6) are represented in the KD11-B print set as rectangles with inputs on the left and outputs on the right. Control lines often enter the IC from the bottom. The functional description of the KD11-B MSI and LSI ICs is contained in Appendix A.



11-1197

Figure 1-6 ALU, MSI Circuit Type 74181 Representation

### 1.5.2 Microprogram Documentation

The microprogram is documented at three levels in the print set. The first level is the Microprogram Flow Listing (K-MP-KDN-B-1). At this level, the microprogram is described in terms of register transfers. The Microprogram Symbolic Listing (K-MP-KD11-B-2) shows how the microprogram accomplishes each step. (References in the microprogram listing are symbolic, e.g., scratch pad address = R7). The binary equivalent is shown in the Microprogram Binary Listing (K-MP-KD11-B-3), which actually shows the binary contents of each word of the microprogram. The Microprogram Cross Reference Listing lists the microprogram by address (K-MP-KD11-B-4). The microprogram is discussed in detail in Chapter 5.

### 1.5.3 Read-Only Memory (ROM) Maps

Figure 1-7 is a typical ROM map listing. ROM map listings for the ROMs used in the KD11-B processor are provided in the Engineering Drawing Manual (K-RL-M7260-8 and K-RL-M7261-8).

```

/( =Y8 (PIN #9) CONA INT TRAN SYNC L
+/( =Y7 (PIN #7) CONA REG ADDR L
++/( =Y6 (PIN #6) CONA RECEIVE L
+++/( =Y5 (PIN #5) CONA TRANSMIT L
++++/( =Y4 (PIN #4) CONA LOAD MODEM PSW L
+++++/( =Y3 (PIN #3) CONA LOAD L CLK PSW L
++++++/( =Y2 (PIN #2) CONG SP WRITE L
+++++++/( =Y1 (PIN #1) CONG LOAD PSW L
+++++++
OCTAL
ADDRESS DATA
000 0 00000 11111111 377
001 1 00001 11111111 377
002 2 00010 11111111 377
003 3 00011 11111111 377
004 4 00100 01111110 176 PSW .TRAN OUT BA=177776
005 5 00101 11111111 377 PSW .TRAN OUT,BAR
006 6 00110 01111011 173 LCLK .TRANOUT
007 7 00111 11111111 377 LCLK .TRANOUT,BAR
010 8 01000 00111101 075 GR<R0:R17> .TRANOUT BA=1777XX
011 9 01001 10111111 277 GR<R0:R17> .TRANOUT,BAR
012 10 01010 01111111 177 ODD BYTE (LCLK/TK/TP)
013 11 01011 11111111 377
014 12 01100 11111111 377
015 13 01101 11111111 377
016 14 01110 01111111 177 SWR .TRANOUT BA=177570
017 15 01111 11111111 377 SWR .TRANOUT,BAR
020 16 10000 01010111 127 TKS .TRANOUT BA=177560
021 17 10001 11011111 337 TKS .TRANOUT,BAR
022 18 10010 01100111 147 TPS .TRANOUT BA=177564
023 19 10011 11101111 357 TPS .TRANOUT,BAR
024 20 10100 01011111 137 TKB .TRANOUT BA=177562
025 21 10101 11011111 337 TKB .TRANOUT,BAR
026 22 10110 01101111 157 TPB .TRANOUT BA=177566
027 23 10111 11101111 357 TPB .TRANOUT,BAR
030 24 11000 11111111 377
031 25 11001 11111111 377
032 26 11010 11111111 377
033 27 11011 11111111 377
034 28 11100 11111111 377
035 29 11101 11111111 377
036 30 11110 11111111 377
037 31 11111 11111111 377
+++++
+++++/( A(PIN #10) IS CONA TRAN OUT L
+++/( B(PIN #11) IS Y3 OF F025
++/( C(PIN #12) IS Y2 OF F025
+/( D(PIN #13) IS Y1 OF F025
/( E(PIN #14) IS Y4 OF F025

```

Figure 1-7 E068 ROM Map Example

## CHAPTER 2 INSTRUCTION SET

### 2.1 INTRODUCTION

The KD11-B is defined by its instruction set. The sequences of processor operations are selected according to the instruction decoding. This chapter contains tables that describe the PDP-11 instructions and instruction set addressing modes. A detailed description of the instruction set and addressing modes is contained in the PDP-11/05-11/10 Handbook. Instruction set differences between the PDP-11/05-11/10 and PDP-11/20 are listed in Table 2-8.

### 2.2 ADDRESSING MODES

#### 2.2.1 Introduction

Data stored in memory must be accessed and manipulated. Data handling is specified by a PDP-11 instruction (MOV, ADD etc.) which usually indicates:

- a. The function (operation code).
- b. A general purpose register for locating the source operand and/or a general purpose register for locating the destination operand.
- c. An addressing mode (to specify how the selected register(s) is to be used).

A large portion of the data handled by a computer is usually structured (in character strings, in arrays, in lists etc.). Thus, the PDP-11 is designed to handle structured data efficiently and flexibly. The general registers may be used with an instruction in any of the following ways:

- a. As accumulators. The data to be manipulated resides within the register.
- b. As pointers. The contents of the register are the address of the operand, rather than the operand itself.
- c. As pointers that automatically step through core locations. Automatically stepping forward through consecutive core locations is termed autoincrement addressing; automatically stepping backwards is termed autodecrement addressing. These modes are particularly useful for processing tabular data.

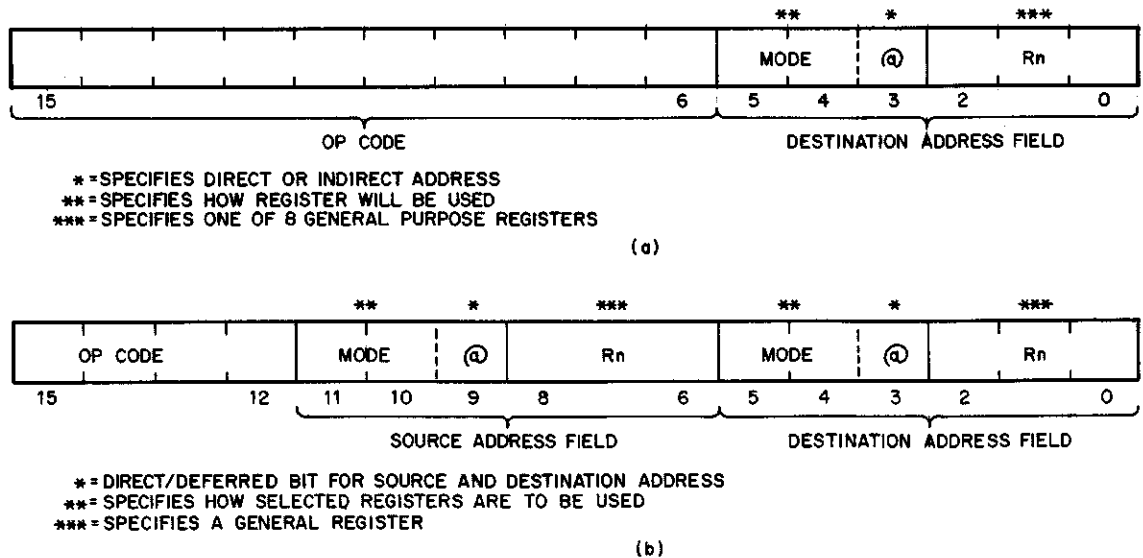


- d. As index registers. In this instance, the contents of the register and the word following the instruction are summed to produce the address of the operand. This allows easy access to variable entries in a list.

PDP-11s also have instruction addressing mode combinations that facilitate temporary data storage structures for convenient handling of data that must be frequently accessed. This is known as the "stack".

In the PDP-11 any register can be used as a "stack pointer" under program control; however, certain instructions associated with subroutine linkage and interrupt service automatically use Register 6 as a "hardware stack pointer". For this reason, R6 is frequently referred to as the "SP".

Two types of instructions utilize the addressing modes: Single operand and double operand. Figure 2-1 shows the formats of these two types of instructions. The addressing modes are listed in Table 2-1.



11-1227

Figure 2-1 Addressing Mode Instruction Formats

Table 2-1  
Addressing Modes

Binary Code	Name	Assembler Syntax	Function
DIRECT MODES			
000	Register	Rn	Register contains operand.
010	Autoincrement	(Rn) +	Register contains address of operand. Register contents incremented after reference.
100	Autodecrement	-(Rn)	Register contents decremented before reference register contains address of operand.
110	Index	X(Rn)	Value X (stored in a word following the instruction) is added to (Rn) to produce address of operand. Neither X nor (Rn) are modified.
DEFERRED MODES			
001	Register Deferred	@Rn or (Rn)	Register contains the address of the operand.
011	Autoincrement Deferred	@(Rn) +	Register is first used as a pointer to a word containing the address of the operand, then incremented (always by 2; even for byte instructions).
101	Autodecrement	@-(Rn)	Register is decremented (always by two; even for byte instructions) and then used as a pointer to a word containing the address of the operand.
111	Index Deferred	@X(Rn)	Value X (stored in a word following the instruction) and (Rn) are added and the sum is used as a pointer to a word containing the address of the operand. Neither X nor (Rn) are modified.
PC ADDRESSING			
010	Immediate	#n	Operand follows instruction.
011	Absolute	@#A	Absolute address follows instruction.
110	Relative	A	Address of A, relative to the instruction, follows the instruction.
111	Relative Deferred	@A	Address of location containing address of A, relative to the instruction follows the instruction.

Rn = Register

X, n, A = next program counter (PC) word (constant)

### 2.2.2 Instruction Timing

The PDP-11 is an asynchronous processor in which, in many cases, memory and processor operations are overlapped. The execution time for an instruction is the sum of a basic instruction time and the time to determine and fetch the source and/or destination operands. Table 2-2 shows the addressing times required for the various modes of addressing source and destination operands. All times stated are subject to  $\pm 10\%$  variation.

Table 2-2  
Addressing Times

Addressing Format			Time ( $\mu\text{s}$ )	
Mode	Description	Symbolic	Source *	Destination **
0	register	R	0	0
1	register deferred	@R or (R)	0.9	2.4
2	auto-increment	(R) +	0.9	2.4
3	auto-increment deferred	@(R) +	2.4	3.4
4	auto-decrement	-(R)	0.9	2.4
5	auto-decrement deferred	@-(R)	2.4	3.4
6	indexed	$\pm X$ (R)	2.4	3.4
7	indexed deferred	@ $\pm X$ (R) or @(R)	3.4	4.7

\* For Source time, add  $1.3\mu\text{s}$  for odd byte addressing.

\*\* For Destination time, modify as follows.

1. Add  $1.3\mu\text{s}$  for odd byte addressing with a non-modifying instruction.
2. Add  $2.4\mu\text{s}$  for odd byte addressing with a modifying instruction.
3. Subtract  $1.2\mu\text{s}$  for a non-modifying instruction.

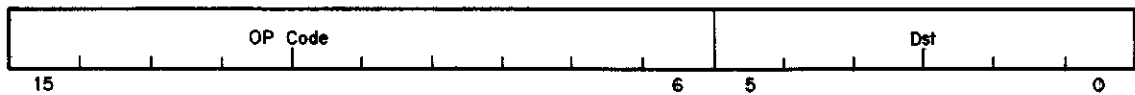
### 2.3 PDP-11/05 INSTRUCTIONS

The PDP-11 instructions can be divided into five instructions groupings:

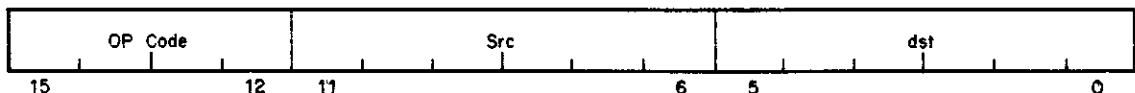
1. Single Operand Instructions (shifts, multiple precision instructions, rotates)
2. Double Operand Instructions (arithmetic and logical instructions)
3. Program Control Instructions (branches, subroutines, traps)
4. Operate Group Instructions (processor control operations)
5. Condition Codes Operators (processor status word bit instructions)

Tables 2-3 through 2-7 list each instruction, including byte instructions for the respective instruction groups. Figure 2-2 shows the six different instruction formats of the instruction set, and the individual instructions in each format.

1. Single Operand Group (CLR,CLRB,COM,COMB,INC,INCB,DEC,DECB,NEG,NEGB,ADC,ADCB,SBC,SBCB,TST,TSTB,ROR,RORB,ROL,ROLB,ASR,ASRB,ASL,ASLB, JMP, SWAB)



2. Double Operand Group (BIT,BITB,BIC,BICB,BIS,BISB,ADD,SUB)

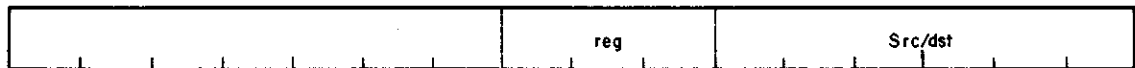


3. Program Control Group

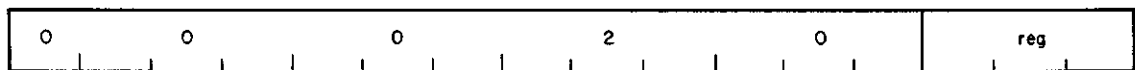
a. Branch (all branch instructions)



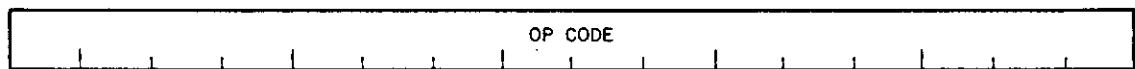
b. Jump To Subroutine (JSR)



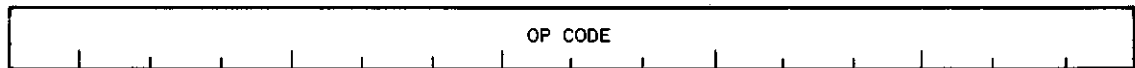
c. Subroutine Return (RTS)



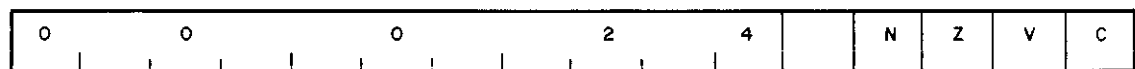
d. Traps (break point, IOT, EMT, TRAP)



4. Operate Groupe (HALT, WAIT, RTI, RESET)



5. Condition Code Operators (all condition code instructions)



11-1226

Figure 2-2 PDP-11 Instruction Formats

Table 2-3 Single Operand Instructions

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
CLR CLRB 3.4 $\mu$ s	0050DD* 1050DD	$(dst) \leftarrow 0$	N: cleared Z: set V: cleared C: cleared	Contents of specified destination are replaced with zeroes.
COM COMB 3.4 $\mu$ s	0051DD 1051DD	$(dst) \leftarrow \bar{n} (dst)$	N: set if most significant bit of result is 0. Z: set if result is 0. V: cleared. C: set.	Replaces the contents of the destination address by their logical complement (each bit equal to 0 set and each bit equal to 1 cleared).
INC INCB 3.4 $\mu$ s	0052DD 1052DD	$(dst) \leftarrow (dst) + 1$	N: set if result is less than 0. Z: set if result is 0. V: set if $(dst)$ was 077777. C: not affected.	Add 1 to the contents of the destination.
DEC DECB 3.4 $\mu$ s	0053DD 1053DD	$(dst) \leftarrow (dst) - 1$	N: set if result is less than 0. Z: set if result is 0. V: set if $(dst)$ was 100000. C: not affected	Subtract 1 from the contents of the destination.
NEG NEGB 3.4 $\mu$ s	0054DD 1054DD	$(dst) \leftarrow \bar{\bar{dst}}$	N: set if result is less than 0. Z: set if result is 0. V: set if result is 100000. C: cleared if result is 0.	Replaces the contents of the destination address by its 2's complement. Note that 100000 is replaced by itself.
ADC ADCB 3.4 $\mu$ s	0055DD 1055DD	$(dst) \leftarrow (dst) + C$	N: set if result is less than 0. Z: set if result is 0. V: set if $(dst)$ is 077777 and C is 1. C: set if $(dst)$ is 177777 and C is 1.	Adds the contents of the C-bit into the destination. This permits the carry from the addition of the low order words/bytes to be carried into the high order result.

\* DD = destination (address mode and register)

†  $(dst)$  = destination contents

Table 2-3 Single Operand Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
SBC SBCB 3.4 $\mu$ s	0056DD 1056DD	(dst) $\leftarrow$ (dst) -C	N: set if result is less than 0. Z: set if result is 0. V: set if (dst) was 100000. C: cleared if (dst) is 0 and C is 1.	Subtracts the contents of the C-bit from the destination. This permits the carry from the subtraction of the low order words/bytes to be subtracted from the high order part of the result.
TST TSTB 3.4 $\mu$ s	0057DD 1057DD	(dst) $\leftarrow$ (dst)	N: set if result is less than 0. Z: set if result is 0. V: cleared. C: cleared.	Sets the condition codes N and Z according to the contents of the destination address.
ROR RORB 3.4 $\mu$ s	0060DD	(dst) $\leftarrow$ (dst) rotated right one place.	N: set if high order bit of the result is set. Z: set if all bits of result are 0. V: loaded with the exclusive OR of the N-bit and the C-bit as set by ROR.	Rotates all bits of the destination right one place. The low order bit is loaded into the C-bit and the previous contents of the C-bit are loaded into the high order bit of the destination.
ROL ROLB 3.4 $\mu$ s	0061DD 1061DD	(dst) $\leftarrow$ (dst) rotated left one place.	N: set if the high order bit of the result word is set (result <0); cleared otherwise. Z: set if all bits of the result word = 0; cleared otherwise. V: loaded with the exclusive OR of the N-bit and C-bit (as set by the completion of the rotate operation).	Rotate all bits of the destination left one place. The high order bit is loaded into the C-bit of the status word and the previous contents of the C-bit are loaded into the low order bit of the destination.

Table 2-3 Single Operand Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
ASR ASRB 3.4 $\mu$ s	0062DD 1062DD	(dst) $\leftarrow$ (dst) shifted one place to the right.	C: loaded with the high order bit of the destination. N: set if the high order bit of the result is set (result <0); cleared otherwise. Z: set if the result = 0; cleared otherwise. V: loaded from the ex- clusive OR of the N-bit and C-bit (as set by the completion of the shift operation). C: loaded from low order bit of the destination.	Shifts all bits of the destination right one place. The high order bit is replicated. The C-bit is loaded from the low order bit of the destination. ASR performs signed division of the destination by two.
ASL ASLB 3.4 $\mu$ s	0063DD 1063DD	(dst) $\leftarrow$ (dst) shifted one place to the left.	N: set if high order bit of the result <0); cleared otherwise. Z: set if the result = 0; cleared otherwise. V: loaded with the ex- clusive OR of the N-bit and C-bit (as set by the completion of the shift operation). C: loaded with the high order bit of the destina- tion.	Shifts all bits of the destination left one place. The low order bit is loaded with a 0. The C-bit of the status word is loaded from the high order bit of the destination. ASL performs a signed multiplication of the desti- nation by 2 with overflow indication.

Table 2-3 Single Operand Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
JMP 1.0 $\mu$ s	0001DD	PC $\leftarrow$ dst	Not affected.	JMP provides more flexible program branching than provided with the branch instruction. Control may be transferred to any location in memory (no range limitation) and can be accomplished with the full flexibility of the addressing modes, with the exception of register mode 0. Execution of a jump with mode 0 will cause an "illegal instruction" condition. (Program control cannot be transferred to a register.) Register deferred mode is legal and will cause program control to be transferred to the address held in the specified register. Note that instructions are word data and must therefore be fetched from an even-numbered address. A "boundary error" trap condition will result when the processor attempts to fetch an instruction from an odd address.
SWAB 4.3 $\mu$ s	0003DD	Byte 1/Byte 0 Byte 0/Byte 1	N: set if high order bit of low order byte (bit 7) of result is set; cleared otherwise. Z: set if low order byte of result = 0; cleared otherwise. V: cleared. C: cleared.	Exchanges high order byte and low order byte of the destination word (destination must be a word address).



Table 2-4 Double Operand Instructions

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
MOV MOVB 3.7 $\mu$ s 3.1 $\mu$ s mode 0	01SSDD* 11SSDD	(dst) $\leftarrow$ (src) †	N: set if (src) < 0; cleared otherwise. Z: set if (src) = 0; cleared otherwise. V: cleared. C: not affected.	Word: Moves the source operand to the destination location. The previous contents of the destination are lost. The source operand is not affected. Byte: Same as MOV. The MOVB to a register (unique among byte instructions) extends the most significant bit of the low order byte (sign extension). Otherwise, MOVB operates on bytes exactly as MOV operates on words.
CMP CMPB 3.7 $\mu$ s	02SSDD 12SSDD	(src) - (dst) [in detail, (src) + ~ (dst) + 1]	N: set if result < 0; cleared otherwise. Z: set if result = 0; cleared otherwise. V: set if there was arithmetic overflow; that is, operands were of opposite signs and the sign of the destination was the same as the sign of the result; cleared otherwise. C: cleared if there was a carry from the most significant bit of the result; set otherwise.	Compares the source and destination operands and sets the condition codes, which may then be used for arithmetic and logical conditional branches. Both operands are unaffected. The only action is to set the condition codes. The compare is customarily followed by a conditional branch instruction. Note that unlike the subtract instruction the order of operation is (src) - dst, not (dst) - (src).

\* SS = source (address mode and register).

† (src) = source contents.

Table 2-4 Double Operand Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
BIT BITB 3.7 $\mu$ s	03SSDD 13SSDD	$(src) \wedge (dst)$	N: set if high-order bit of result set; cleared otherwise. Z: set if result = 0; cleared otherwise. V: cleared. C: not affected.	Performs logical AND comparison of the source and destination operands and modifies condition codes accordingly. Neither the source nor destination operands are affected. The BIT instruction may be used to test whether any of the corresponding bits that are set in the destination are clear in the source.
BIC BICB 3.7 $\mu$ s	04SSDD 14SSDD	$(dst) \leftarrow \sim (src) \wedge (dst)$	N: set if high order bit of result set; cleared otherwise. Z: set if result = 0; cleared otherwise. V: cleared. C: not affected.	Clears each bit in the destination that corresponds to a set bit in the source. The original contents of the destination are lost. The contents of the source are unaffected.
BIS BISB 3.7 $\mu$ s	05SSDD 15SSDD	$(dst) \leftarrow (src) \vee (dst)$	N: set if high order bit of result set; cleared otherwise. Z: set if result = 0; cleared otherwise. V: cleared. C: not affected.	Performs inclusive OR operation between the source and destination operands and leaves the result at the destination address; that is, corresponding bits set in the destination. The contents of the destination are lost.
ADD	06SSDD	$(dst) \leftarrow (src) + (dst)$	N: set if result 0; cleared otherwise. Z: set if result = 0; cleared otherwise. V: set if there was arithmetic overflow as a result of the operation; that is both operands were of the same sign and the	Adds the source operand to the destination operand and stores the result at the destination address. The original contents of the destination are lost. The contents of the source are not affected. Two's complement addition is performed.

Table 2-4 Double Operand Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
SUB 3.7 $\mu$ s	16SSDD	$(dst) \leftarrow (dst) - (src)$ in detail, $(dst) + \sim (src) + 1 (dst)$	<p>result was of the opposite sign; cleared otherwise.                      C: set if there was a carry from the most significant bit of the result; cleared otherwise.</p> <p>N: set if result &lt; 0; cleared otherwise.                      Z: set if result = 0; cleared otherwise.                      V: set if there was arithmetic overflow as a result of the operation, that is if operands were of opposite signs and the sign of the source was the same as the sign of the result; cleared otherwise.                      C: cleared if there was a carry from the most significant bit of the result; set otherwise.</p>	Subtracts the source operand from the destination operand and leaves the result at the destination address. The original contents of the destination are lost. The contents of the source are not affected. In double-precision arithmetic the C-bit, when set, indicates a "borrow".

Table 2-5 Program Control Instructions

NOTE: Condition Codes are Unaffected by these Instructions

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
BR 2.5 $\mu$ s	000400 †xxx	PC $\leftarrow$ PC + (2 x offset)	Unaffected	Provides a way of transferring program control within a range of -128 to + 127 words with a one word instruction. It is an unconditional branch.
BNE 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	001000 †xxx	PC $\leftarrow$ PC + (2 x offset) if Z = 0	Unaffected	Tests the state of the Z-bit and causes a branch if the Z-bit is clear. BNE is the complementary operation to BEQ. It is used to test inequality following a CMP, to test that some bits set in the destination were also in the source, following a BIT, and generally, to test that the result of the previous operation was not zero.
BEQ 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	001400 †xxx	PC $\leftarrow$ PC + (2 x offset) if Z = 1	Unaffected	Tests the state of the Z-bit and causes a branch if Z is set. As an example, it is used to test equality following a CMP operation, to test that no bits set in the destination were also set in the source following a BIT operation, and generally, to test that the result of the previous operation was zero.
BGE 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	002000 †xxx	PC $\leftarrow$ PC + (2 x offset) if N $\vee$ V = 0	Unaffected	Causes a branch if N and V are either both clear or both set. BGE is the complementary operation to BLT. Thus, BGE always causes a branch when it follows an operation that caused addition to two positive numbers. BGE also causes a branch on a zero result.

† xxx = Offset, 8 bits (0-7) of instruction format.  
R = register (linkage pointer).

Table 2-5 Program Control Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
BLT 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	002400 †xxx	$PC \leftarrow PC +$ (2 x offset) if $N \vee V = 1$	Unaffected	Causes a branch if the exclusive OR of the N- and V-bits are 1. Thus, BLT always branches following an operation that added two negative numbers, even if overflow occurred. In particular, BLT always causes a branch if it follows a CMP instruction operating on a negative source and a positive destination (even if overflow occurred). Further, BLT never causes a branch when it follows a CMP instruction operating on a positive source and negative destination. BLT does not cause a branch if the result of the previous operation was zero (without overflow).
BGT 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	003000 †xxx	$PC \leftarrow PC +$ (2 x offset) if $Z \vee (N \vee V) = 0$	Unaffected	Operation of BGT is similar to BGE, except BGT does not cause a branch on a zero result.
BLE 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	003400 †xxx	$PC \leftarrow PC +$ (2 x offset) if $Z \vee (N \vee V) = 1$	Unaffected	Operation is similar to BLT but in addition will cause a branch if the result of the previous operation was zero.
BPL 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	100000 †xxx	$PC \leftarrow PC +$ (2 x offset) if $N = 0$	Unaffected	Tests the state of the N-bit and causes a branch if N is clear. BPL is the complementary operation of BMI.
BMI 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	100400 †xxx	$PC \leftarrow PC +$ (2 x offset) if $N = 1$	Unaffected	Tests the state of the N-bit and causes a branch if N is set. It is used to test the sign (most significant bit) of the result of the previous operation.

Table 2-5 Program Control Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
BHI 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	101000 †xxx	$PC \leftarrow PC +$ (2 x offset) if $C = 0$	Unaffected	Causes a branch if the previous operation caused neither a carry nor a zero result. This will happen in comparison (CMP) operations as long as the source has a higher unsigned value than the destination.
BLOS 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	101400 †xxx	$PC \leftarrow PC +$ (2 x offset) if $C \vee Z = 1$	Unaffected	Causes a branch if the previous operation caused either a carry or a zero result. BLOS is the complementary operation to BHI. The branch occurs in comparison operations as long as the source is equal to or has a lower unsigned value than the destination. Comparison of unsigned values with the CMP instruction to be tested for "higher or same" and "higher" by a simple test of the C-bit.
BVC 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	102000 †xxx	$PC \leftarrow PC +$ (2 x offset) if $V = 0$	Unaffected	Tests the state of the V-bit and causes a branch if the V-bit is clear. BVC is complementary operation to BVS.
BVS 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	102400 †xxx	$PC \leftarrow PC +$ (2 x offset) if $V = 1$	Unaffected.	Tests the state of V-bit (overflow) and causes a branch if the V-bit is set. BVS is used to detect arithmetic overflow in the previous operation.
BCC BHIS 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	103000 †xxx	$PC \leftarrow PC +$ (2 x offset) if $C = 0$	Unaffected	Tests the state of the C-bit and causes a branch if C is clear. BCC is the complementary operation to BCS.
BCS BLO 1.9 $\mu$ s no branch 2.5 $\mu$ s branch	103400 †xxx	$PC \leftarrow PC +$ (2 x offset) if $C = 1$	Unaffected	Tests the state of the C-bit and causes a branch if C is set. It is used to test for a carry in the result of a previous operation.

Table 2-5 Program Control Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
JRS 3.8 $\mu$ s	004RDD	(tmp) $\leftarrow$ (dst) (tmp is an internal processor register) $\downarrow$ (SP) $\leftarrow$ reg (push reg contents onto processor stack) reg $\leftarrow$ PC PC holds location following JSR; this address PC $\leftarrow$ (tmp), now put in (reg).	Unaffected	<p>In execution of the JSR, the old contents of the specified register (the "LINKAGE POINTER") are automatically pushed onto the processor stack and new linkage information placed in the register. Thus, subroutines nested within subroutines to any depth may all be called with the same linkage register. There is no need either to plan the maximum depth at which any particular subroutine will be called or to include instructions in each routine to save and restore the linkage pointer. Further, since all linkages are saved in a reentrant manner on the processor stack, execution of a subroutine may be interrupted, and the same subroutine reentered and executed by an interrupt service routine. Execution of the initial subroutine can then be resumed when other requests are satisfied. This process (called nesting) can proceed to any level.</p> <p>JSR PC, dst is a special case of the PDP-11 subroutine call suitable for subroutine calls that transmit parameters.</p>
RTS 3.8 $\mu$ s	00020R	PC $\leftarrow$ (reg) (reg) $\leftarrow$ SP $\uparrow$	Unaffected	<p>Loads contents of reg into PC and pops the top element of the processor stack into the specified register.</p> <p>Return from a non-reentrant subroutine is typically made through the same register that was used in its call. Thus, a subroutine called with a JSR PC, dst exits with a RTS PC, and a subroutine called with a JSR R5, dst may pick up parameters with addressing modes (R5) +, X (R5), or @X (R5) and finally exit, with an RTS R5.</p>

Table 2-5 Program Control Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
(No mnemonic) 8.2 μs	000003	↓ (SP) ← PS ↓ (SP) ← PC PC ← (14) PS ← (16)	N: loaded from trap vector Z: loaded from trap vector V: loaded from trap vector C: loaded from trap vector	Performs a trap sequence with a trap vector address of 14. Used to call debugging aids. The user is cautioned against employing code 000003 in programs run under these debugging aids.
IOT 8.2 μs	000004	↓ (SP) ← PS ↓ (SP) ← PC PC ← (20) PS ← (22)	N: loaded from trap vector Z: loaded from trap vector C: loaded from trap vector	Performs a trap sequence with a trap vector address of 20. Used to call the I/O executive routine IOX in the paper-tape software system, and for error reporting in the disk operating system.
EMT 8.2 μs	104000	↓ (SP) ← PS ↓ (SP) ← PC PC ← (30) PS ← (32)	N: loaded from trap vector Z: loaded from trap vector V: loaded from trap vector C: loaded from trap vector	All operation codes from 104000 to 104377 are EMT instructions and may be used to transmit information to the emulating routine (e.g., function to be performed). The trap vector for EMT is at address 30; the new central processor status (PS) is taken from the word at address 32.  CAUTION: EMT is used frequently by DEC system software and is therefore not recommended for general use.



Table 2-5 Program Control Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
TRAP 8.2 $\mu$ s	104400 to 104777	↓ (SP) ← PS ↓ (SP) ← PC PC ← (34) PS ← (36)	N: loaded from trap vector Z: loaded from trap vector V: loaded from trap vector C: loaded from trap vector	Operation codes from 104400 to 104777 are TRAP instructions. TRAPs and EMTs are identical in operation, except that the trap vector for TRAP is at address 34.  NOTE: Since DEC software makes frequent use of EMT, the TRAP instruction is recommended for general use.

Table 2-6 Operate Group Instructions

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
HALT 1.8 $\mu$ s	000000		Not affected.	Causes the processor operation to cease. The console is given control of the processor. The console data lights display the address of the halt instruction plus two. Transfers on the Unibus are terminated immediately. The PC points to the next instruction to be executed. Pressing the CONTINUE key on the console causes processor operation to resume. No INIT signal is given.
WAIT 1.8 $\mu$ s	000001		Not affected.	Provides a way for the processor to relinquish use of the bus while it waits for an external interrupt. Having been given a WAIT command, the processor will not compete for bus use by fetching instructions or operands from memory. This permits higher transfer rates between device and memory, since no processor induced latencies will be encountered by bus requests from the device. In WAIT, as in all instructions, the PC points to the next instruction following the WAIT operation. Thus, when an interrupt causes the PC and PS to be pushed onto the stack, the address of the next instruction following the WAIT is saved. The exit from the interrupt routine (i.e., execution of an RTI instruction) will cause resumption of the interrupted process at the instruction following the WAIT.

Table 2-6 Operate Group Instructions (cont.)

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
RTI 4.4 $\mu$ s	000002	PC (SP) PSW (SP)	N: loaded from processor stack Z: loaded from processor stack V: loaded from processor stack C: loaded from processor stack	Used to exit from an interrupt or TRAP service routine. The PC and PSW are restored (popped) from the processor stack. If a trace trap is pending, the first instruction after the RTI will be executed prior to the next "T" Trap.
RESET 20 ms	000005	PC (SP) PSW (SP)	Not affected	Sends INIT on the Unibus for 20 ms. All devices on the Unibus are reset to their state at power up.

Table 2-7 Condition Codes Operators

Mnemonic/ Instruction Time	OP Code	Operation	Condition Codes	Description
CLC	000241			Set and clear condition code bits. Selectable combinations of these bits may be cleared or set together. Condition code bits corresponding to bits in the condition code operator (bits 0-3) are modified according to the sense of bit 4, the set/clear bit of the operator, i.e., set the bit specified by bit 0, 1, 2 or 3, if bit 4 is a 1. Clear corresponding bits if bit 4=0.
CLZ	000242			
CLN	000244			
CLV	000250			
Set all CC's	000261			
Clear all CC's	000262			
Clear V and C	000264			
No operation	000270			
No operation	000277			
2.5 $\mu$ s	000257			
	000243			
	000240			
	000260			

11-2-21

#### 2.4 INSTRUCTION SET DIFFERENCES

Table 2-8 lists the differences between the PDP-11/20 and PDP-11/05 instruction sets.

Table 2-8 PDP-11 Differences

11-2-22

PDP-11/15, PDP-11/20	PDP-11/05, PDP-11/10
<ol style="list-style-type: none"> <li>1. OPR %R, (R) +/- (R), source operand is %R after autoinc/autodec of DEST register where registers are the same.</li> <li>2. OPR %R, @-(R) / @ (R) + uses R after autodec/autoinc as source operand.</li> <li>3. MOV PC, LOC stores PC of INST +4 in LOC.</li> <li>4. SWAB does not change V.</li> <li>5. Program halt displays PC of halt instruction in ADDRESS lights. DATA lights display (RO)</li> <li>6. Register addresses (177700-177717) time out when used as a program address by the processor. Can be addressed under console operation.</li> <li>7. Byte ops to the odd byte of the PS cause odd address trap.</li> <li>8. The RESET instruction clears the RUN light such that program loops that make frequent use of RESET may not appear to be running.</li> <li>9. Power fail during RESET instruction is not recognized until after the instruction is finished. (70 ms). Too late, so don't use RESET. RESET instruction consists of 70 milliseconds pause with INIT occurring during first 20 milliseconds.</li> <li>10. The first instruction in an interrupt service routine is guaranteed to be executed. *Double BUS ERROR results in a HALT in the 11/20. TTRT named in 11/20 logic is unnamed in 11/20 instruction card, IR = 3.</li> </ol>	<p>OPR %R, (R) +/- R source operand is %R before autoinc/autodec of DEST, registers are the same.</p> <p>OPR %R, @-(R) / @ (R) + uses R before autodec/autoinc as source operand.</p> <p>MOV PC, LOC stores PC of INST +2 in LOC</p> <p>Swab clears V.</p> <p>Displays next PC.</p> <p>Register addresses (177700-177717) are valid program addresses when used by the processor.</p> <p>Byte ops to odd byte of PS do not trap. Not all bits may exist</p> <p>RESET does not clear the RUN light.</p> <p>Power fail immediately ends the RESET instructions and traps if an INIT is in progress (22 milliseconds). A minimum INIT of 300 ns occurs if the instruction aborted. Power Fail during RESET Fetch is fatal - no power down sequence.</p> <p>The first instruction in an interrupt routine will not be executed if another interrupt occurs at a higher priority level than was assumed by the first interrupt.</p>

Table 2-8 PDP-11 Differences (Cont.)

PDP-11/15, PDP-11/20	PDP-11/05, PDP-11/10
<p>11. Sequence of internal processor traps, external interrupts, HALT and WAIT:</p> <p>BUS ERROR Trap*              Odd Address              Data Time Out</p> <p>HALT Instruction for Console              Operation</p> <p>TRAP Instr's - illegal or Reserved              Instructions,              TRTT; IOT, EMT, TRAP</p> <p>TRACE TRAP - "T" bit of processor              status</p> <p>OVFL Trap - Stack Overflow</p> <p>PWR FAIL Trap - Power down</p> <p>CONSOLE BUS REQUEST -              Console Operation after              HALT switch</p> <p>UNIBUS BUS REQUEST -              Peripheral Request,              compated with Processor              Priority - usually an              Interrupt occurs.</p> <p>WAIT LOOP - Loop on a WAIT instruction in IR until an              interrupt allows exit. A CONSOLE BUS REQUEST              returns to this loop after being honored.</p>	<p>Sequences:</p> <p>BUS ERROR Traps          HALT Instruction          TRAP Instructions          OVFL Trap          PWR Fail Trap          UNIBUS BUS REQUESTS          CONSOLE STOP              (HALT switch)          WAIT LOOP</p>

## CHAPTER 3 CONSOLE DESCRIPTION

### 3.1 INTRODUCTION

This chapter provides a general description and a detailed description of the console logic. The general description is keyed to the block diagram level. The detailed description covers the theory of operation of the console logic.

The function and use of the console controls are covered in Part 1, Chapter 4.

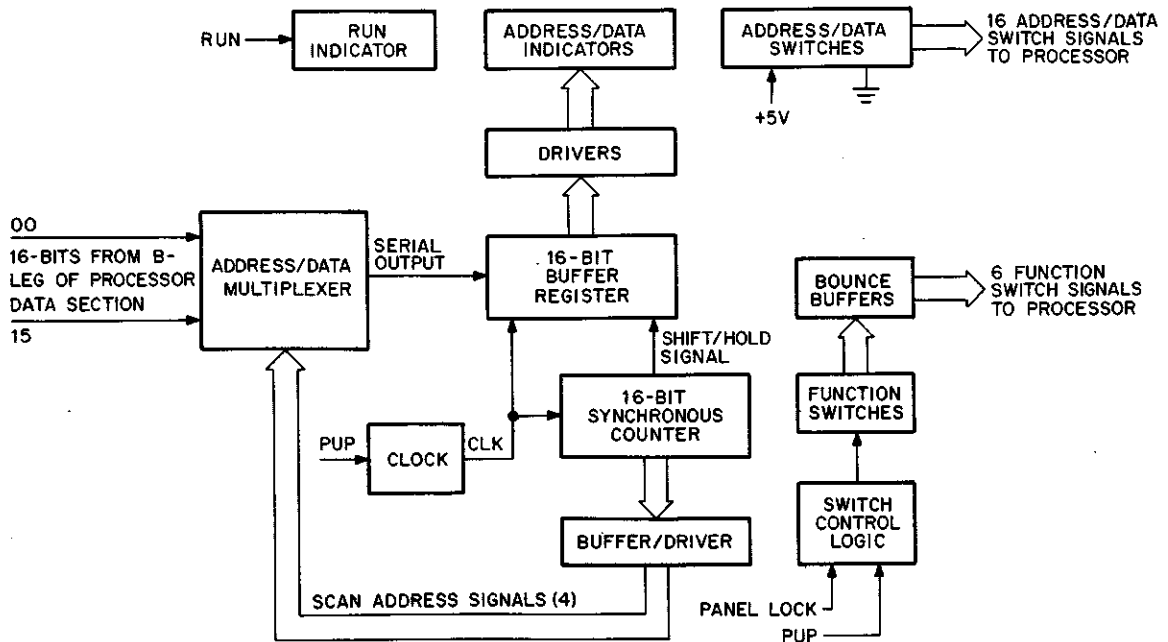
### 3.2 GENERAL DESCRIPTION

The console logic is divided into two sections: address/data register logic and control switch logic. All the console logic is contained on one printed circuit board, which also contains the switches and indicators.

#### 3.2.1 Address/Data Register Logic

During manual console operation, data and addresses are generated by positioning the 16 ADDRESS/DATA REGISTER switches. The switches are 2-position toggle type: the up position grounds the switch and provides a low signal to the processor logic; the down position provides a high signal to the processor logic by connecting the switch to +5V.

The address/data register logic samples the 16 bits (address or data) from the B-leg of the processor data section and displays them via the ADDRESS/DATA REGISTER indicators (Figure 3-1). The address/data multiplexer scans the processor 16-bit B-leg signals and provides a serialized output to the buffer register. The output of the register consists of



II-0954

Figure 3-1 Console Functional Block Diagram

16 signals that are buffered and sent to the 16 ADDRESS/DATA indicators. The buffer has two modes of operation, which are controlled by the SHIFT/HOLD signal from the 16-bit synchronous counter. In the shift (scan) mode, serialized data from a scan operation is shifted into the register: this operation takes 16  $\mu$ s. At the end of this time, the register enters the hold (display) mode for 240  $\mu$ s during which time the register contents are displayed. This process is continuous and a scan pulse display sequence takes 256  $\mu$ s. The information that is scanned (multiplexer input) remains stable for a long time compared to the 256- $\mu$ s cycle for the register; therefore, the multiplexer scans relatively stable information that can be displayed.

In addition to supplying the SHIFT/HOLD signal that controls the buffer register, the counter also generates the four scan address signals that select the multiplexer inputs.

The clock provides pulses to clock the counter and buffer register. It starts when power is applied and is self-sustaining thereafter.



### 3.2.2 Control Switch Logic

The six console control switches allow programming functions to be performed manually. They are: Load Address (LOAD ADRS), Examine (EXAM), Continue (CONT), Deposit (DEP) and START. The switches provide signals to the processor logic, which actually controls the functions.

A bounce buffer is connected across the output contacts of each switch to eliminate interruptions of the output signal due to contact bounce when the switch is activated. The bounce buffer is a latch constructed of two cross-coupled inverters.

The control switch logic senses a power up signal (PUP) and PANEL LOCK signal to ensure control switch lockout during the Panel Lock mode and to eliminate program interruption after a power interruption with the HALT/ENABLE switch left inadvertently in the HALT position during operation in the Panel Lock mode.

## 3.3 DETAILED DESCRIPTION

This section provides a detailed description of the console logic. Each major functional unit is discussed separately and with regard to its interrelation with other functional units.

Both detailed and simplified logic diagrams are used to support the text. The simplified logic diagrams are included in this chapter; however, the detailed logic diagrams are part of the print set that is supplied with each computer. Three drawings are referenced, and they are identified as D-CS-5409766-0-1, sheets 1, 2, and 3.

Sheet 1 - Display Buffer and Driver (C-1)

Sheet 2 - Control Keys (C-3)

Sheet 3 - Scan Control and Switch Register (C-2)

In this discussion, these drawings are referenced by the C-numbers located in the title box and shown above (C-1, C-2, and C-3).

### 3.3.1 Multiplexer

The multiplexer, located on the processor M7260 module, scans the 16 bits in the B-leg of the processor data section. The information on these lines can be data bits or address bits. It is serialized in the multiplexer and transmitted over the console cable to the buffer.

The multiplexer is a Type 74150 Data Selector/Multiplexer (1-of-16). It has 16 inputs ( $E_0$  through  $E_{15}$ ) and a single output. Four SCAN ADRS lines from the counter are the data select lines for the multiplexer: 4 bits give 16 unique combinations. A low strobe signal enables the selected input to the output; however, the signal is inverted at the output.

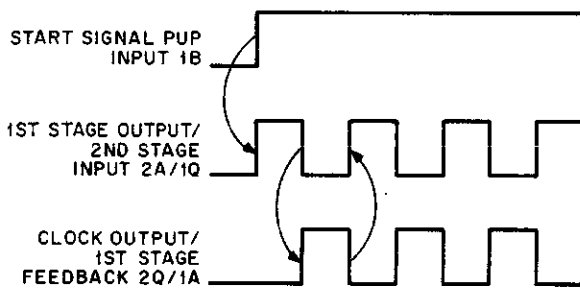
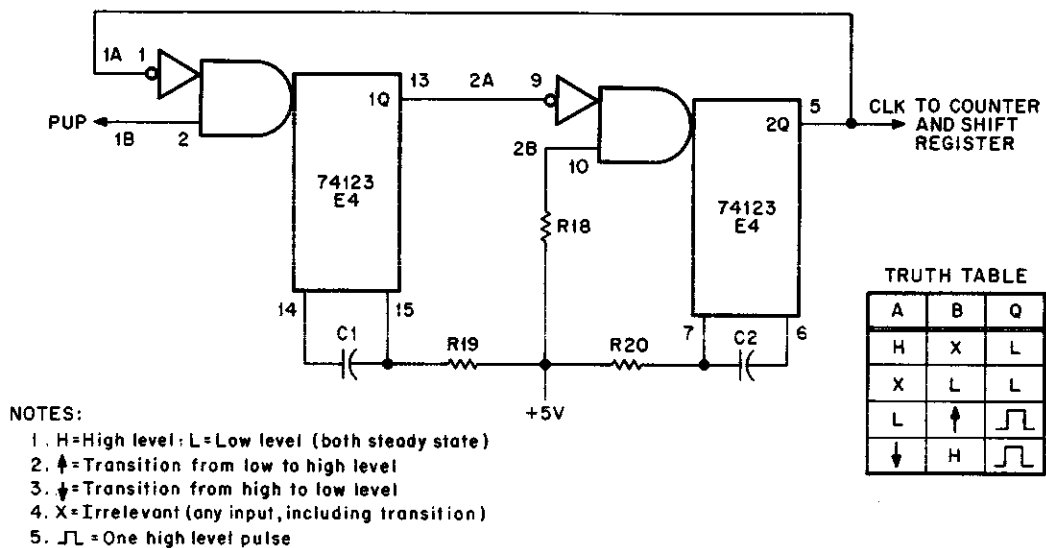
The four SCAN ADRS lines select the input lines on an equivalent number basis. For example, if the SCAN ADRS lines represent decimal 5, input 5 is selected and enabled to the serial output. The SCAN ADRS lines from the counter are inverted before being sent to the multiplexer. When the counter state is zero (0000), the SCAN ADRS lines indicate 15 (1111) and multiplexer input 15 is selected. This ensures that input 15 is shifted into the proper bit position in the buffer after a scan operation is complete. Table 3-1 shows the relationship between the counter state and SCAN ADRS signals.

Table 3-1  
Scan Address Signal Generation

Counter State	SCAN ADRS	MUX Line Scanned
0000 (0)	1111 ( $15_{10}$ )	15
0001 ( $1_{10}$ )	1110 ( $14_{10}$ )	14
0010 ( $2_{10}$ )	1101 ( $13_{10}$ )	13
⋮	⋮	⋮
1110 ( $14_{10}$ )	0001 ( $1_{10}$ )	1
1111 ( $15_{10}$ )	0000 (0)	0

### 3.3.2 Clock

The console clock provides pulses to clock the counter and shift register (drawing C-2). It is a simple oscillator that generates high level clock pulses. Two retriggerable monostable multivibrators (Type 74123) are connected back-to-back to form a simple oscillator (Figure 3-2). The Q output of each is used to trigger the other. The clock starts when power is applied to the processor and is self-sustaining thereafter.



11-0949

Figure 3-2 Console Clock, Schematic and Timing Diagram

One 74123 IC package (E4) contains two separate and identical units identified as 1 and 2. Output 1Q (pin 13) is connected to input 2A (pin 9) and output 2Q (pin 5) is fed back to input 1A (pin 1). The complementary  $\bar{Q}$  outputs are not used; the CLEAR inputs are not used either. Input 2B is held high by application of +5V via resistor R18; therefore, unit 2 can be triggered only by a high-to-low level transition at input 2A (see truth table in Figure 3-2). Input 1B (pin 2) is connected to signal PUP from the processor. This signal is low when power is off and is high when power is on. When PUP is low, the clock output is inhibited regardless of the state of input 1A. When PUP goes high during the power up sequence, it triggers the first high level pulse at output 1Q. The high-to-low level transition of this pulse triggers the first high level pulse at output 2Q (see timing diagram in Figure 3-2). Because both B-inputs are high, the feedback connection (2Q to 1A) allows each unit to trigger on the high-to-low transition at its A-input. This produces a continuous string of positive pulses (CLK signal) at output 2Q. Pulse generation is self-sustaining as long as PUP is high.

The counter is clocked on the low-to-high clock pulse transition and the shift register is clocked on the high-to-low clock pulse transition. The period between clock pulses allows time for the serial data from the multiplexer to settle down. This is important because the serial data is sent to the shift register via a cable connection.

### 3.3.3 Counter

The counter provides four scan address lines that are the data select lines for the data/address multiplexer (drawing C-2). It also provides a control signal (SHIFT DISPLAY) to the shift register, which places it in the Hold mode.

Two Type 74193 Synchronous 4-Bit Up/Down Counters (E6 and E8) are cascaded to provide an 8-bit counter (Figure 3-3). Cascading is accomplished by connecting the CARRY output (pin 12) of the first counter to the COUNT UP input (pin 5) of the second counter. The counter is used only in the Count-up mode; therefore, the COUNT DOWN input is disabled by connecting it to +5V, and the BORROW output is not used.

The preset feature is not used; thus, the LOAD input (pin 11) is disabled by connecting it to +5V. The CLEAR input is not used so that the counter cannot be forced to 0 by an outside signal.

When the clock starts, the counter starts counting through its 256 states. It counts continuously as long as the clock is running.

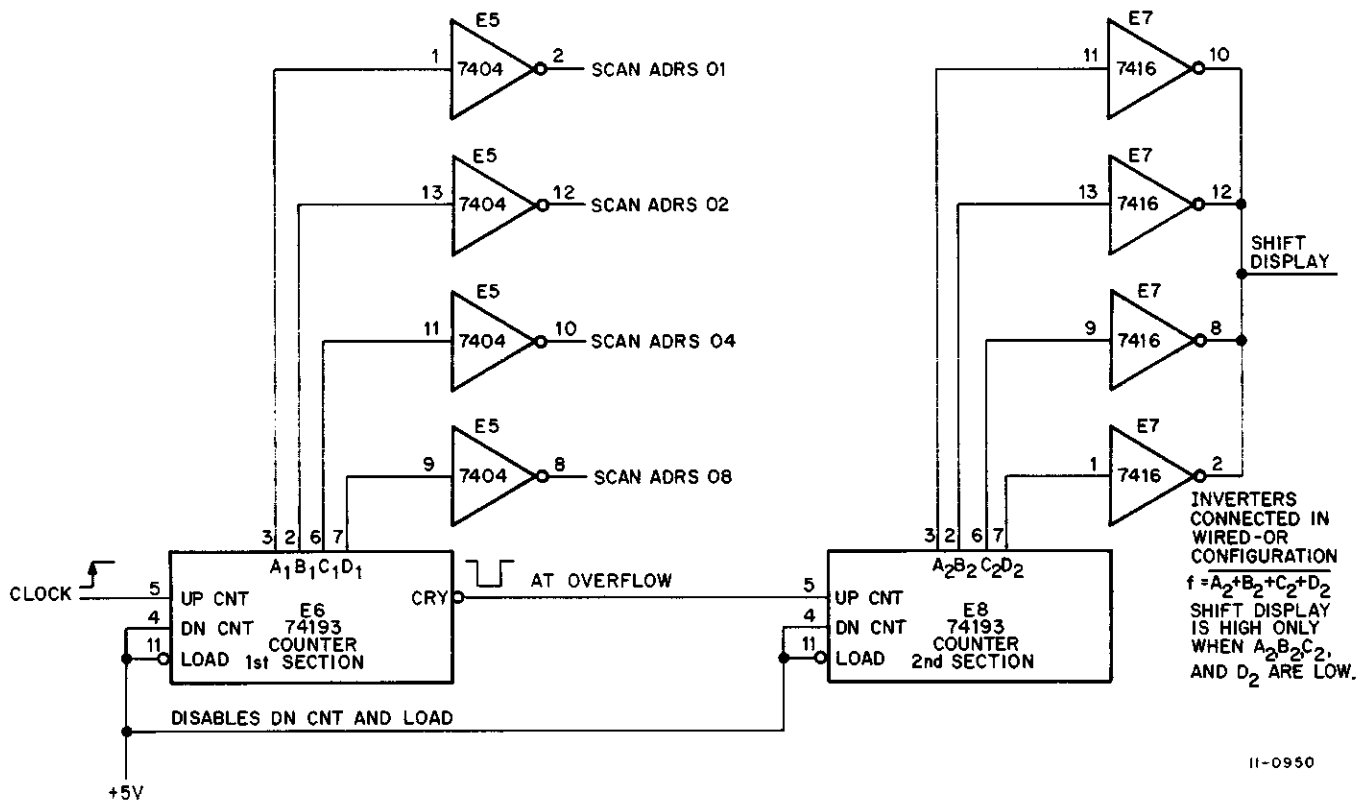


Figure 3-3 Counter, Simplified Logic Diagram

Two modes of operation occur during one complete counting sequence (256 states) before overflow (all 0s) occurs and the sequence repeats. Output A of the first 4-bit section (E6) is the least significant bit; output D<sub>2</sub> of the second 4-bit section (E8) is the most significant bit. The first section advances from 0 through 15 (16 counts), overflows (goes to 0), and starts over. At overflow, a pulse from the CARRY output of the first section is sent to

the COUNT UP input of the second section, which increments the second section by 1. After 16 overflows, the counter is full (all 8 bits are 1s) which represents  $255_{10}$  or 256 counts. The next clock pulse causes both sections to overflow, which sets the counter to 0 and the sequence repeats.

The output of the first counter section is the 4-bit scan address (SCAN ADRS 01 (1) L, 02 (1) L, 04 (1) L, and 08 (1) L). The lines go to four Type 7404 Inverters (E5) and then to the select inputs of the data/address multiplexer. As the first section of the counter sequences through its 16 states, these lines cause the multiplexer to scan its 16 input lines and send the data serially to the shift register. Each of the four outputs of the second counter section goes to a Type 7416 inverter driver (E7). The open collectors of these inverters are connected together in a wired-OR configuration. The output is the SHIFT DISPLAY H signal, which is high only when all inverter inputs (E8 counter outputs) are low (0). The SHIFT DISPLAY H signal is a control signal input to the shift register. When it is low, the register is in the hold mode; when it is high, the register shifts serial data in to the right. The second counter section is 0 only during states 0 through 15. During this period, SHIFT DISPLAY H is high, and the shift register accepts serial data from the multiplexer and shifts it right. This data represents a complete scan of the 16 inputs to the multiplexer that are placed in the shift register. At state 16, a 1 is present in the second counter section. From this state up to and including state 255, one or more 1s are present in the second counter section. The counter states are shown in Table 3-2. During this period, SHIFT DISPLAY H is low, and the shift register is in the Hold mode. The data is static and is available for display.

A counter state change occurs in approximately 1  $\mu$ s; therefore, the Scan mode takes 16  $\mu$ s and the Hold mode lasts for 240  $\mu$ s. During manual console operation, data and addresses are generated by positioning switches. The information on the multiplexer input remains stable for a long time in comparison to the 256  $\mu$ s required for a counter scan/hold cycle. In effect, the multiplexer continually scans relatively stable information that can be displayed as static rather than transient information.

Table 3-2  
Counter States

Counter State (Decimal)	Counter States								Remarks	
	2nd Section				1st Section					
	D	C	B	A	D	C	B	A		
0	0	0	0	0	0	0	0	0	States 0-15 are Scan mode - Data is obtained and loaded into shift register.	
1	0	0	0	0	0	0	0	1		
2	0	0	0	0	0	0	1	0		
3	0	0	0	0	0	0	1	1		
4	0	0	0	0	0	1	0	0		
5	0	0	0	0	0	1	0	1		
:			:				:			
15	0	0	0	0	1	1	1	1		
16	0	0	0	1	0	0	0	0		States 16-255 are Hold mode - Data is held in shift register for display.
17	0	0	0	1	0	0	0	1		
18	0	0	0	1	0	0	1	0		
:			:				:			
31	0	0	0	1	1	1	1	1		
32	0	0	1	0	0	0	0	0		
:			:				:			
239	1	1	1	0	1	1	1	1		
240	1	1	1	1	0	0	0	0		
:			:				:			
255	1	1	1	1	1	1	1	1		
0	0	0	0	0	0	0	0	0	← Counter overflow	

### 3.3.4 Display Buffer and Driver

The display buffer and driver logic consists of a 16-bit buffer and 16 inverter drivers for the ADDRESS/DATA REGISTER lights (drawing C-1).

The 16-bit buffer is composed of four Type 8271 4-Bit Registers (E11, E10, E13, and E15). They are connected in a shift right configuration with a serial data input; the last bit output ( $D_O$ ) of the preceding section is connected to the series data input ( $D_S$ ) of the following section (Figure 3-4). The parallel data inputs are not used. The reset input ( $R_D$ ) is disabled by connecting it to +5V. The LOAD input (pin 10) is connected to ground (logic 0), and the SHIFT input (pin 13) is connected to the SHIFT DISPLAY H signal from the counter. With the LOAD input held low, the operating mode of the buffer is controlled by the SHIFT input. When the SHIFT DISPLAY H signal is high, the buffer accepts data and shifts right; this is the console scan mode. When the SHIFT DISPLAY signal is low, the buffer holds the data; this is the console display mode.

Each shift register output signal is sent to a Type 7416 Inverter Driver to illuminate an associated light-emitting diode (LED). The 16 LEDs are the indicators for the ADDRESS/DATA REGISTER display. A high output from the buffer causes the LED to illuminate, which indicates that the associated bit is a logical 1. The final stage of a 7416 inverter has an open collector that is connected to a LED, which in turn is connected to +5V via a current limiting resistor (Figure 3-5). When the inverter input is low (logical 0 = 0V), no current can flow through the LED because there is no conducting path to ground through the transistor; therefore, the LED is not illuminated. A high (logical 1) inverter input puts a positive voltage on the base of the transistor, which turns it on. Current flows from the +5V source through the resistor, LED, and transistor emitter to ground, which illuminates the LED.

### 3.3.5 Control Switches and Logic

The console contains six control switches (drawing C-3). The HALT/ENABLE (HLT/ENB) switch is a 2-position toggle type; HALT is the down position and ENABLE is the up position. The other five switches are momentary action type. They are: Load Address (LOAD



ADRS), Examine (EXAM), Continue (CONT), Deposit (DEP) and START. The DEP switch is activated when it is lifted: the others are activated when they are depressed.

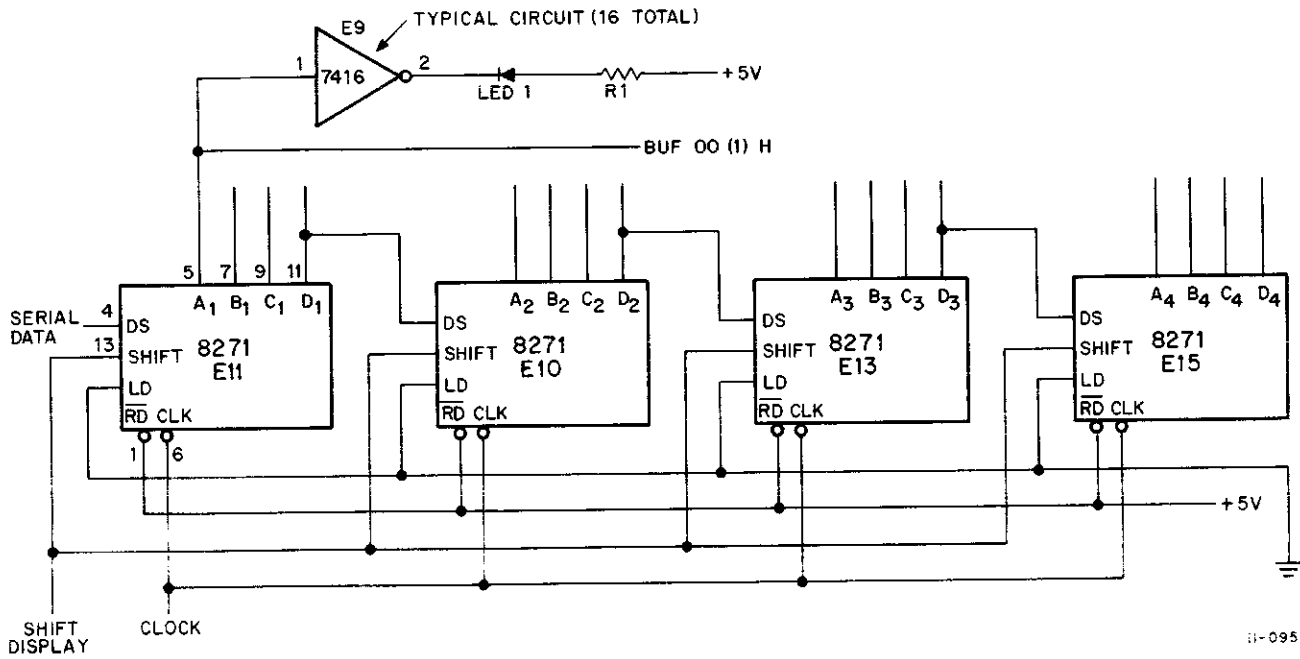


Figure 3-4 Display Buffer and Driver, Simplified Logic Diagram

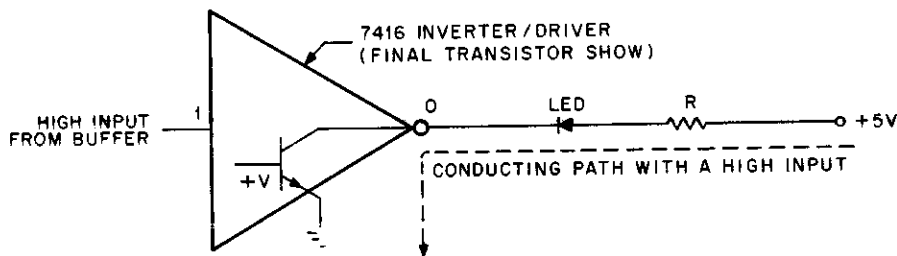


Figure 3-5 LED Driver Circuit

A bounce buffer is connected across the output contacts of each switch to eliminate interruptions of the output signal due to contact bounce when the switch is activated. The bounce buffer is a latch constructed of two cross-coupled 7416 inverter buffer/drivers.

When the switch is activated, the output signal is latched and any contact bounce, with accompanying signal loss, does not alter the output signal.

For the momentary action switches, the output is asserted low (logical 0) when the switch is activated. For the Halt/Enable switch, the output is asserted low when the switch is in the Halt position.

The input of each switch is connected to the output of a 7417 open-collector non-inverting buffer. The inputs of all the 7417 buffers are connected to the output of a very simple logic network that detects Power on/off and Panel Lock on/off. Power is sensed by monitoring the Power Up signal (PUP L) from the processor. Panel Lock is sensed by monitoring the PANEL LOCK signal from the OFF/POWER/PANEL LOCK switch on the console front panel. Panel Lock is a mode of operation that disables all console control switches. It prevents inadvertent switch operation from disturbing a running program.

3.3.5.1 Normal Operating Mode - Normal operation is performed with PANEL LOCK off. This discussion is referenced to engineering drawing C-3 and Figure 3-6, which is a simplified logic diagram.

The switch input logic network is composed of one 7404 inverter (E5) and one 7416 open-collector inverter (E7). The output of E7 is connected to PANEL LOCK, which is controlled by the key operated ON/OFF/PANEL LOCK switch on the console panel. When the switch is in the PANEL LOCK position, the Panel Lock mode is activated and the PANEL LOCK signal is high (logical 1). When the switch is in the ON position, the PANEL LOCK signal is low (logical 0). This is accomplished by grounding the PANEL LOCK signal in this switch position. The output of E7 (pin 6), which represents the state of input PUP L, is connected to the PANEL LOCK signal line. This point is the input to all switch 7417 buffers (E3). It is high only when PUP L and PANEL LOCK are both high.

With Panel Lock off, the PANEL LOCK signal is low and the input to each switch is low. Refer to momentary action switch S6 (LOAD ADDRESS), which is typical of the five switches of this type. The set input of the latch is the rest terminal, and the reset input

is the active terminal. With S6 in the rest position, a 0 is placed on the set input of the latch (E2 pin 13). The latch is set (S=0, R=1, Q=1) and the output (E2 pin 12) is high, which is the non-asserted state of the switch output (KEY LOAD ADRS (1) L = 1). With S6 in the active position, a 0 is placed on the reset input of the latch (E2 pin 11). The latch is reset (S=1, R=0, Q=0) and the output (E2 pin 12) is low, which is the asserted state of the switch output (KEY LOAD ADRS (1) L = 0). Note that the DEP switch (S1) is electrically identical to S6 but its active position is up rather than down.

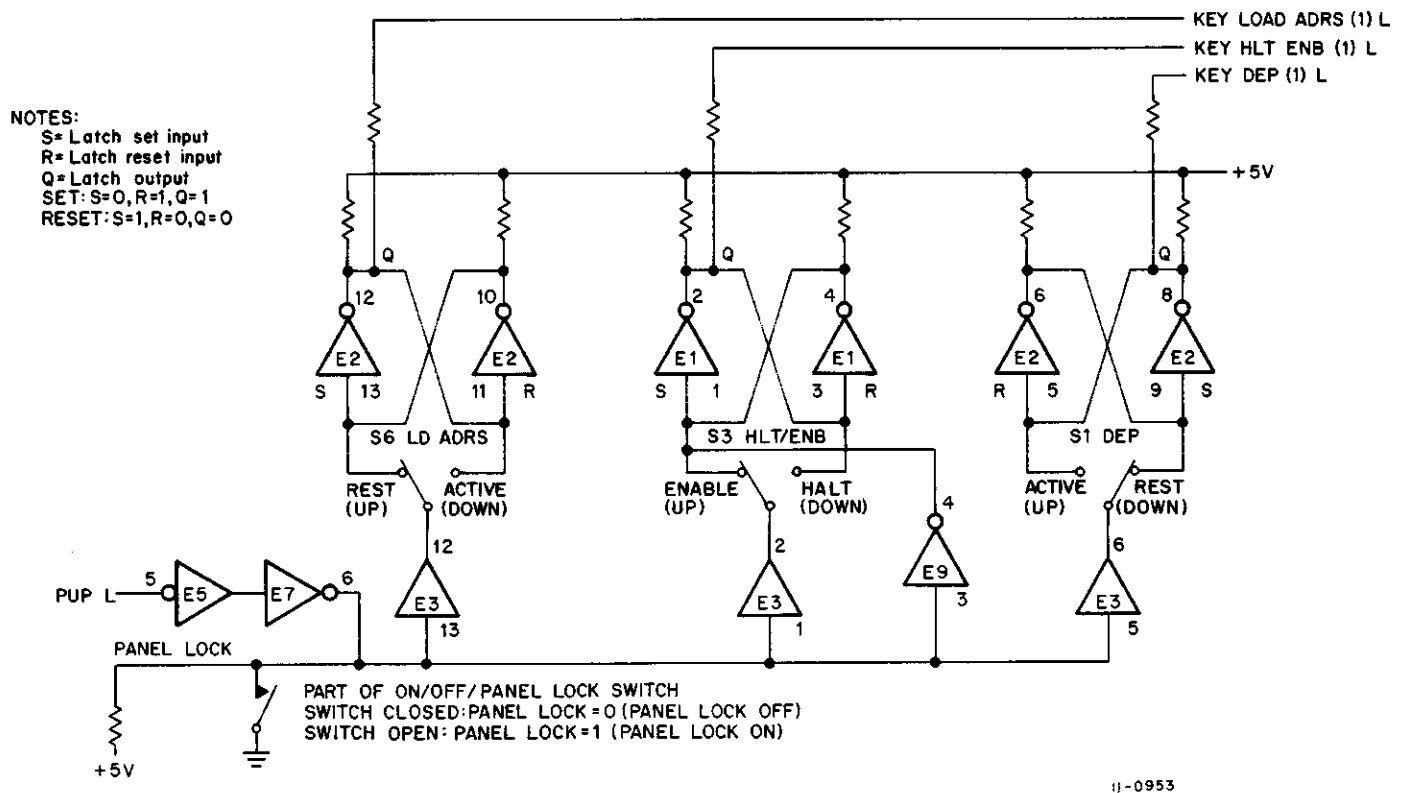


Figure 3-6 Control Switches and Bounce Buffers, Logic Diagram

With the HALT/ENABLE switch (S3) in the ENABLE (up) position, the latch is set and the switch output signal KEY HLT ENB (1) L = 1, which is its non-asserted state. This state allows a program to run. In the HALT (down) position, KEY HLT ENB (1) L = 0 is the asserted state and would halt an operating program. Type 7416 open-collector inverter E9 is used for power loss compensation and is described in a subsequent paragraph. In the normal operating mode, it has no effect on the switch operation.

3.3.5.2 Panel Lock Mode - In the Panel Lock mode, the PANEL LOCK signal is high (+5V via resistor R42). All switch inputs are now high. Panel Lock is applied after a program has started in the normal operating mode. All momentary action switches are in the rest position; switch outputs are high (not asserted) because the latches have been set previously (S=0, R=1, Q=1). The HALT/ENABLE switch is in the ENABLE position; the switch output is high (not asserted) because the latch has been set previously (S=0, R=1, Q=1). With respect to the momentary action switches, the high on the switch input has no effect if the switch is moved to the active position, because it puts a 1 on the reset input of the latch whose reset input is already a 1. With respect to the HALT/ENABLE switch, the high on the switch input has no effect if the switch is moved to the HALT position, because it puts a 1 on the reset input of the latch whose reset input is already a 1. Remember that the momentary action switch latches had been set (S=0, R=1, Q=1), and the HALT/ENABLE switch latch also had been set.

In this mode of operation, inadvertent switch operation cannot halt or otherwise alter a running program.

3.3.5.3 Power Loss During Operation - The processor contains a power fail circuit that allows the computer to tolerate an ac power loss without adverse effects. If a power loss occurs in the normal operating mode (Panel Lock off), the switches perform the functions determined by their current positions as soon as the +5V logic supply voltage is reestablished. PANEL LOCK = 0 is the signal that provides normal switch operation in this case.

If a power loss occurs in the Panel Lock mode, a forcing signal is required to ensure that the latches are driven to the states commensurate with the switch positions before the PANEL LOCK signal is applied again. Without the forcing signal, the latches could be set or reset in a random manner not related to switch position as the +5V logic supply voltage is reestablished.

As ac power is restored, PUP L is forced low for approximately 70 ms. This applies a 0 to the switch inputs to force the latches to the states commensurate with the switch positions: all momentary action switches are not asserted and KEY HLT ENB (1) L is not asserted

(HALT/ENABLE switch in ENABLE position). The processor resumes operation and when PUP L goes high again the Panel Lock mode is reestablished.

If the HALT/ENABLE switch is inadvertently placed in the HALT position during processor operation in the Panel Lock mode, the processor does not halt. However, if a power loss occurs with the switch in the HALT position, PUP L going low during the power-up sequence resets the latch and its output KEY HLT ENB (1) L is low, which halts the program. When PUP L goes high again and the Panel Lock mode is reestablished, the 1 on the switch input does not set the latch and eliminate the HALT signal.

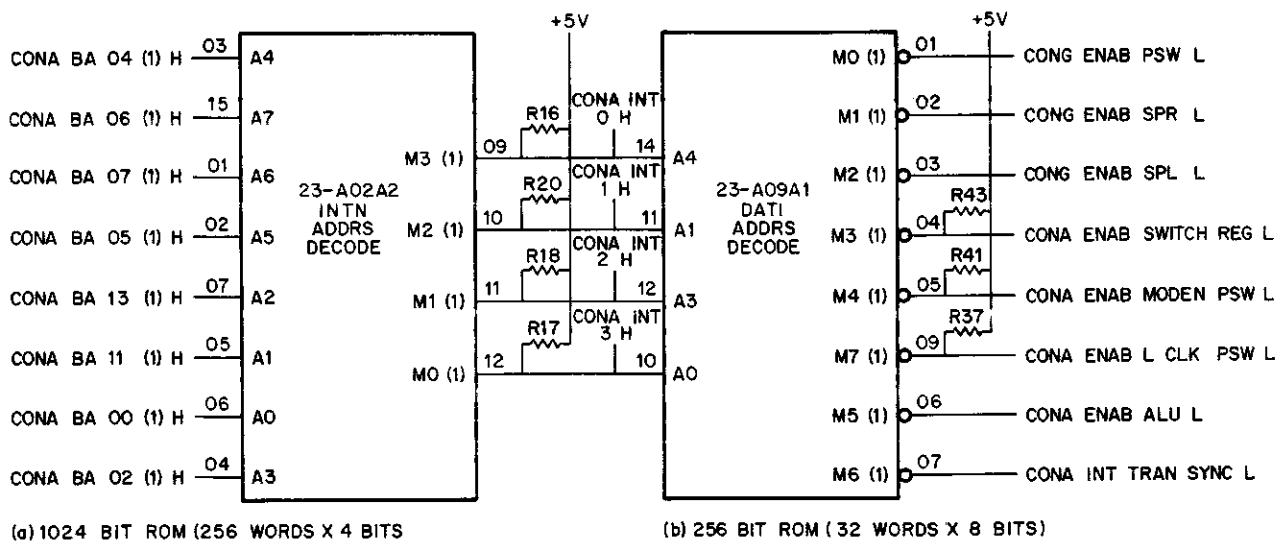
Open-collector inverter E9 solves this problem. When PUP L goes high during the power-up sequence, the 1 on the switch input is also inverted by E9 and a 0 is placed on the set input (E1 pin 1) of the latch. The latch is set and its output is not asserted (KEY HLT ENB (1) L = 1), which allows the processor to resume operation even though the switch is in the HALT position.

### 4.1 INTRODUCTION

This chapter describes the logic and physical implementation of the KD11-B Data Path (DP), Data Path Control (DPC), Unibus control, Serial Communications Line (SCL), and the line clock. Extensive use is made of bipolar, medium and large scale integrated circuits in the processor. There are a total of 28 Read-Only Memories (ROMs) used in the KD11-B. Details of the microprogram are described in Chapter 5.

### 4.2 ROMs AS GENERALIZED GATES

With the increasing availability of inexpensive bipolar ROMs, it is possible to replace rather complex combinational logic structures with one or two 16-pin dual in-line integrated circuits. In the processor, extensive use is made of two different ROM formats. As shown in Figure 4-1, one format states 256 bits (b), arranged in 32 words of 8 bits each.



II-1196

Figure 4-1 1024-Bit and 256-Bit ROMs

The other format stores 1024 bits, arranged in 256 words of 4 bits each. The 32-word ROM has 5 address lines, one output enable line, and 8 outputs. The 256-word ROM has 8 address lines, 2 output enable lines, and 4 outputs. Both devices have open-collector outputs.

Figure 4-2 illustrates the use of a 32 x 8 ROM as a generalized gate. In the example, a 32 x 8 ROM is used as a 5-input priority encoder. The output of the priority encoder follows the following equation:

$$\begin{aligned} \text{OUTPUT} = & V_0 \text{ if } I_0 = 1 \\ & V_1 \text{ if } I_0 = 0 \text{ and } I_1 = 1 \\ & V_2 \text{ if } I_0 = I_1 = 0 \text{ and } I_2 = 1 \\ & \vdots \\ & V_4 \text{ if } I_0 = I_1 = I_2 = I_3 = 0 \text{ and } I_4 = 1 \end{aligned}$$

A similar priority encoder is used in the KD11-B on print CONE where it is necessary to decide which switch function to perform if more than one console switch is depressed.

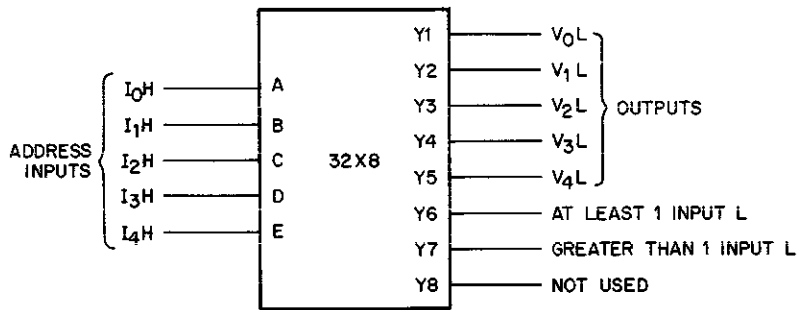
Many situations arise in which 5 or fewer input conditions result in combinations of 8 or fewer output conditions where a 32 x 8 ROM is used for implementing the function. Similar applications apply to 256 x 4 ROMs. For example, the KD11-B uses one 256 x 4 ROM to test all of the PDP-11 conditional branch instructions against the C, N, V, and Z condition code bits. The branch decode ROM may be found on print DPG in position E059.

### 4.3 KD11-B DATA PATH, SIMPLIFIED DESCRIPTION

Figure 4-3 contains a simplified diagram of the KD11-B Data Path. The heart of the DP is an arithmetic-logic unit (ALU), which is capable of performing all 16 Boolean operations and 16 different arithmetic operations on two 16-bit binary variables. The inputs to the ALU are storage registers on the A-leg input and the B-leg input. The output of the ALU feeds into a switch that is capable of introducing external data into the DP from the Unibus.

#### 4.3.1 Data Path (DP) Detailed Description

Figure 4-4 contains a detailed presentation of the KD11-B DP. The logic for all elements of the DP shown in Figure 4-4, with exception of the Bus Address Register and



11-1183

### 1 of 5 Priority Encoder

#### Truth Table

E	D	C	B	A	Address Octal	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>	Y <sub>6</sub>	Y <sub>7</sub>	Y <sub>8</sub>
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	1	0	0
0	0	0	1	0	2	0	1	0	0	0	1	0	0
0	0	0	1	1	3	0	1	0	0	0	1	1	0
					⋮								
0	0	1	1	1	7	0	0	1	0	0	1	1	0
					⋮								
0	1	1	1	1	17	0	0	0	1	0	1	1	0
					⋮								
1	0	1	1	1	27	0	0	0	0	1	1	1	0
					⋮								
1	1	1	1	1	37	0	0	0	0	1	1	1	0

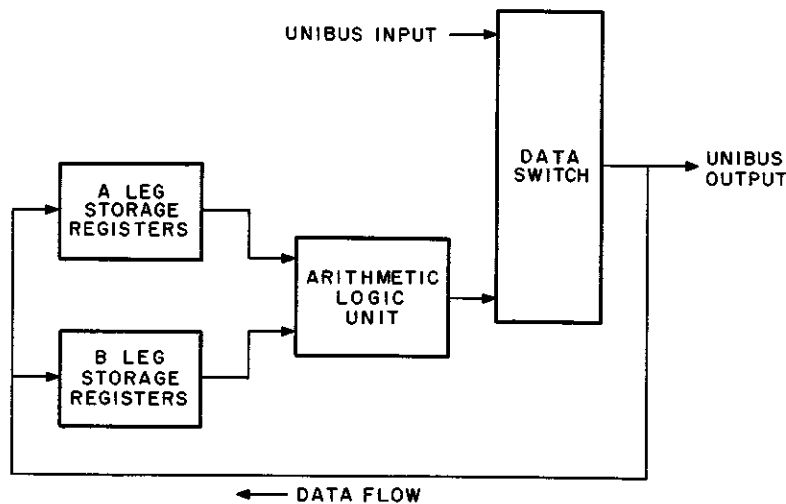
Figure 4-2 32 x 8 ROM used as Generalized Gate

associated Unibus drivers, is found in prints DPA through DPH1. It is important to recognize that this DP consists of a number of interconnected registers that are capable, when properly controlled, of executing the PDP-11 instruction set defined in Chapter 2.

#### 4.3.2 DP Data Polarities

It is useful to note the data polarity at various places in the processor. There are two signal levels used in the KD11-B. A high signal is represented by a voltage of +3V to +5V.





11-1195

Figure 4-3 KD11-B Simplified Data Path Block Diagram

A low signal is represented by a voltage between 0V and 0.4V. Positive and negative data polarities are defined as follows:

Negative Data Polarity:	Logic 1 = Low Signal = 0-0.4V
	Logic 0 = High Signal = 3-5V
Positive Data Polarity:	Logic 1 = High Signal = 3-5V
	Logic 0 = Low Signal = 0-0.4V

Data polarity is negative on the Unibus and within the dotted lines surrounding the ALU as shown in Figure 4-4. Throughout the remainder of the processor the data polarity is generally positive. Care has been exercised in the KD11-B logic to minimize the number of inverters used for correcting data polarity. This sometimes makes the circuit difficult to follow. However, note that in the KD11-B print set, the polarity of the asserted logic signal is given. For example, the signal DPF LOAD IR L is asserted, true or logic one, when it is at 0V (Low Signal).

#### 4.3.3 Data Path Control (DPC)

The DPC is shown on Figure 4-4 at the left side of the drawing. All functions performed by the processor, including instruction interpretation, trap handling, and Switch Register (SR) function execution, depend upon the contents of the Control Store (CS). For each PDP-11 action performed by the KD11-B, the DPC executes a sequence of microsteps stored in the CS.

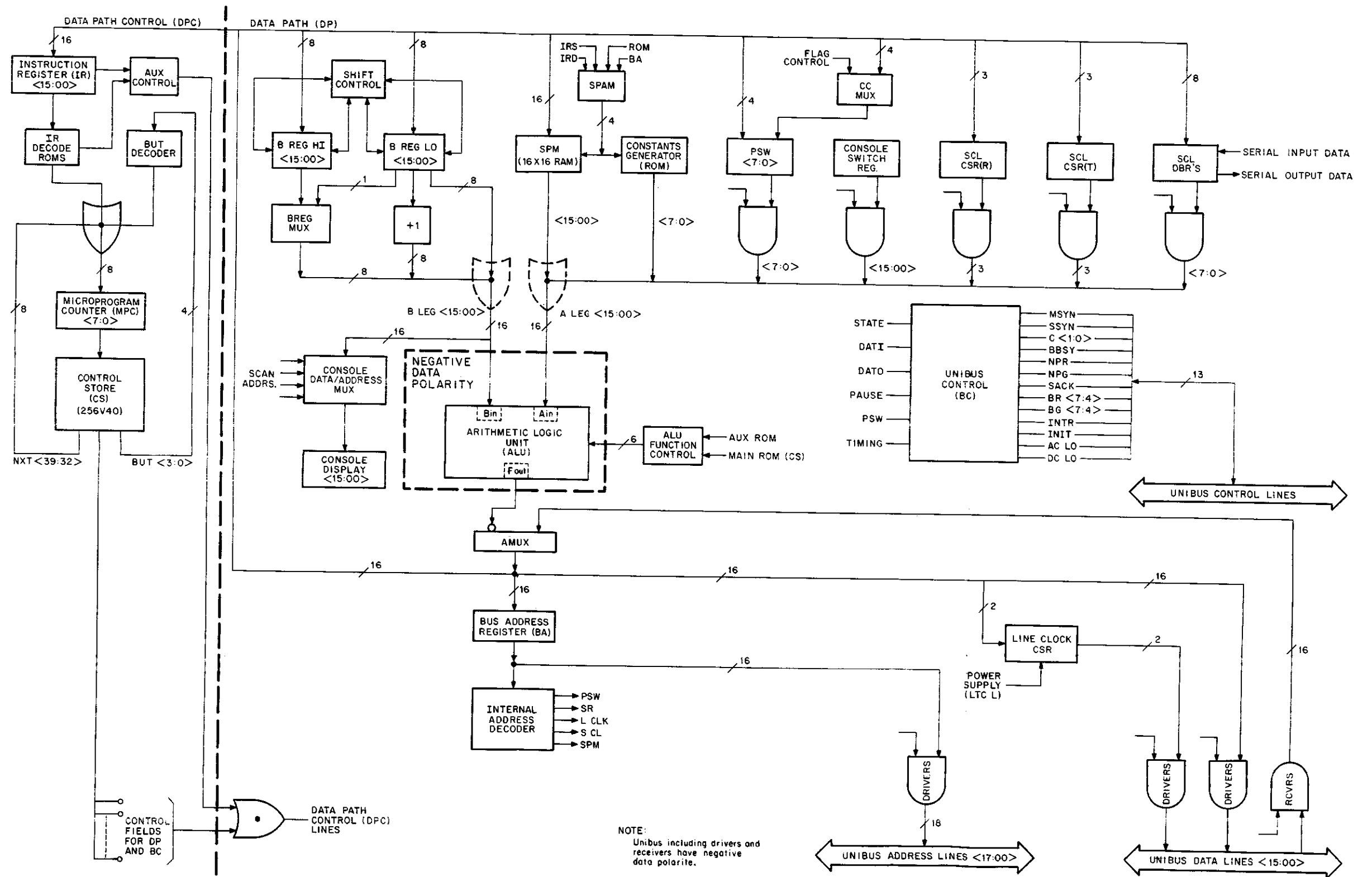


Figure 4-4 KD11-B Detailed Block Diagram

The microprogram contained in the CS consists of a series of microroutines that, when executed in the proper sequence, enable the KD11-B to perform as a PDP-11 processor. Details of the microprogram are described in Chapter 5.

The CS consists of ten 256 x 4 bipolar Read-Only Memories (ROM), shown on prints CONF and CONG. The outputs of the ROMs are used to control the registers and arithmetic elements in the DP. The current control step (microstep) is stored in a Microprogram Counter (MPC). The MPC is an 8-bit latch that is loaded at intervals of approximately 300 ns with a number generated by the output of the NXT field of the CS wire-ORed with the outputs of the microbranch network.

The functioning of the IR decode logic and the AUX control logic, shown in Figure 4-4, is described in Appendix B at the end of the manual.

#### 4.3.4 The A MUX1

As previously mentioned, the arithmetic logic unit (ALU) feeds a switch that can, on command, introduce external data from the Unibus into the DP. This switch is called the A MUX (A-multiplexer). The A MUX physically consists of four 8266 ICs that appear on prints DPA, DPB, DPC, and DPD. The outputs of the ALU feed the inverting inputs of the A MUX, while the outputs of the Unibus receivers feed the non-inverting inputs of the A MUX.

#### 4.3.5 The ALU

As previously mentioned, the ALU is the heart of the data path. It physically consists of four 74181 ICs with carry-look-ahead provided by one 74182 dual in-line circuit. The ALU is used in the processor to perform all arithmetic and logic operations with the exception of rotates and shifts. There is one 74181 on each of print pages DPA, DPB, DPC, and DPD. The five control lines for the ALU, S0, S1, S2, S3, and mode are driven by signals that originate on print CONF and are wire-ORed with signals on prints DPF and DPG.

In the execution of PDP-11 instructions, the source operand appears on the A-leg of ALU and the destination operand appears on the B-leg of the ALU. As discussed in Appendix B, this presents special problems with the handling of the Subtract instruction.

The ALU contains an output that detects the presence of all 0s on its output. This output, divided into two 8-bit bytes, is used to set the Z condition code.

#### 4.4 SCRATCH PAD STORAGE REGISTER

There are approximately 23 storage registers in the DP, of which 21 are attached to the ALU A-leg and one is attached to the ALU B-leg. Sixteen 16-bit storage registers are contained in a random access bipolar memory attached to the ALU A-leg called the Scratch Pad. The utilization of registers in the DP is listed in Table 4-1. The SP is implemented by four 16-word by 4-bit 3101A ICs shown in prints DPA through DPD.

Table 4-1  
Utilization of SP

Register No. (octal)	Designation
R0 - R5	General Purpose PDP-11 Registers
R6	Processor Stack Pointer
R7	PDP-11 Program Counter
R10	Source Operand Storage Register
R11	Destination Operand Storage
R12	Interrupt Vector (for diagnostic purposes)
R13 - R16	Unused
R17	Load Address Storage Register

##### 4.4.1 Scratch Pad Address Multiplexer (SPAM)

The SP register to be read from or written into can be selected from four different sources, depending on the state of the processor. These four sources (Table 4-2) are switched through two dual 4-to-1 multiplexers shown on print CONB.

##### 4.4.2 Processor Status Word Register

The Processor Status Word (PSW) is contained in an 8-bit register attached to the A-leg of the ALU. The PSW is loaded as a result of instruction execution, program traps, and program interrupts. The PSW contains the processor priority, the T-bit, and the four

Table 4-2  
Scratch Pad Address Sources through SPAM

Function	Source Signal	Print Location
Source Register of PDP-11 Instruction	IR (8:6)	DPF
Destination Register of PDP-11 Instruction	IR (2:0)	DPF
Access of General Register from the Console	BA (3:0)	CONA
Selection of Register by Microprogram	CONG ROM SPA (00-03)	CONG

condition codes. In the case of a program trap or interrupt, the PSW is loaded with the second word of the vector from the Unibus data lines via the A MUX. Otherwise, the PSW is loaded through a rather complex series of multiplexers and combinational logic dictated by the particular PDP-11 instruction being executed.

The C- and V-bit ROM - In the upper right-hand corner of print DPF there is a rectangle labeled "C- and V-bit ROM" along with six associated gates. This ROM determines the disposition of the C- and V-bits for normal arithmetic instructions. The exclusive OR gate attached to output pin 1 of the ROM is set to a non-inverting state, if the current PDP-11 instruction is a subtract and is set to an inverting state otherwise. The C- and V-bit ROM is not used to determine the disposition of the C- and V-bits for rotate or shift instructions.

#### 4.4.3 The Constants Generator

The constants generator is a single 32-word by 8-bit ROM attached to the A-leg of the ALU and shown on print DPB. The constants generator is used to generate the addresses of trap vectors and special Unibus addresses as shown in Table 4-3. Note that the inputs to the constants generator are the same signals from print CONB that are used to select SP registers. This is possible because the constants generator and the SP are never used simultaneously.

Table 4-3  
Contents of the Constants Generator (E025) ROM

OCTAL ADDRESS	DECIMAL ADDRESS	EDCBA	OCTAL DATA		
000	0	00000	11111111	377	
001	1	00001	01001110	116	K=207 SWR ADDRESS I.E. 177570=000207 .BAR
002	2	00010	01110011	163	K=64 RECVR. VECTOR
003	3	00011	00001111	017	K=360 CONDITION CODE MASK (CCM-1)
004	4	00100	10111011	273	K=30 EMT VECTOR
005	5	00101	00111111	077	K=14 T BIT VECTOR
006	6	00110	11111111	377	
007	7	00111	11111111	377	
010	8	01000	11111011	373	K=20 IOT VECTOR
011	9	01001	00111011	073	K=34 TRAP VECTOR
012	10	01010	11111111	377	
013	11	01011	11111111	377	
014	12	01100	10111111	277	K=10 RESERVED (ILLEGAL) INSTRUCTION VECTOR
015	13	01101	01111111	177	K=4 BUS ERROR OR STACK OVERFLOW ERROR
016	14	01110	11111111	377	
017	15	01111	11111111	377	
020	16	10000	01111011	173	K=24 PWR FAIL VECTOR
021	17	10001	11111111	377	
022	18	10010	11111101	375	K=100 LCLK INT VECTOR
023	19	10011	11111111	377	
024	20	10100	11111111	377	
025	21	10101	11111111	377	
026	22	10110	11111111	377	
027	23	10111	11111111	377	

Table 4-3 (Cont)

<u>OCTAL ADDRESS</u>	<u>DECIMAL ADDRESS</u>	<u>EDCBA</u>	↑↑↑↑↑↑↑↑ ↑↑↑↑↑↑↑↑	<u>OCTAL DATA</u>	
030	24	11000	11111111	377	
031	25	11001	11111111	377	
032	26	11010	11111111	377	
033	27	11011	11111111	377	
034	28	11100	11111111	377	
035	29	11101	11110011	363	K=60 TRANSMIT VECTOR
036	30	11110	11111111	377	
037	31	11111	11111111	377	
		↑↑↑↑↑			
		↑↑↑↑/( A(PIN #10) IS CONG SP WRITE H			
		↑↑↑/( B(PIN #11) IS CONG ROM SPA 00 H			
		↑↑/( C(PIN #12) IS CONG ROM SPA 01 H			
		↑/( D(PIN #13) IS CONG ROM SPA 02 H			
		/( E(PIN #14) IS CONG ROM SPA 03 H			

#### 4.4.4 The Console Switch Register

The settings contained in the 16 console switches are transmitted from the console over 16 parallel wires to a Berg connector on the M7260 module. The Switch Register (SR) contents are entered into the DP via 16 74H01 gates that appear on prints DPA through DPD. These gates are enabled by the ENAB SWITCH REG signal from print CONA, whenever Switch Register address  $177570_8$  appears in the Bus Address Register from a DATIP operation. The mechanism for the detection of internal bus addresses is described in Paragraph 4.8.

#### 4.4.5 Serial Communications Line

The KD11-B Serial Communications Line (SCL) interface consists of an oscillator, the interrupt circuitry, and the Universal Asynchronous Receiver-Transmitter (UART). The operation of the UART and its associated KD11-B circuitry is described in detail in Paragraph 4.11. As far as the DP is concerned, the SCL appears as four buffer registers. The appropriate register is enabled by the signals generated on print CONA.

### 4.5 B-LEG STORAGE REGISTER

The B-Register (B-Reg) is the only storage register on the B-leg of the ALU. The output of the B-Register, as shown in Figure 4-4, is attached to logic that permits its lower byte to be sign extended. This sign extension facility is used in the execution of PDP-11 byte instructions and PDP-11 branch instructions. The B-Reg is also located on prints DPA through DPD with other elements of the DP such as the ALU.

The B-Reg is used as a general purpose register as well as a left-right shift register. Whenever it is necessary to perform an operation that involves reading data from the SP, the result is stored in the B-Reg.

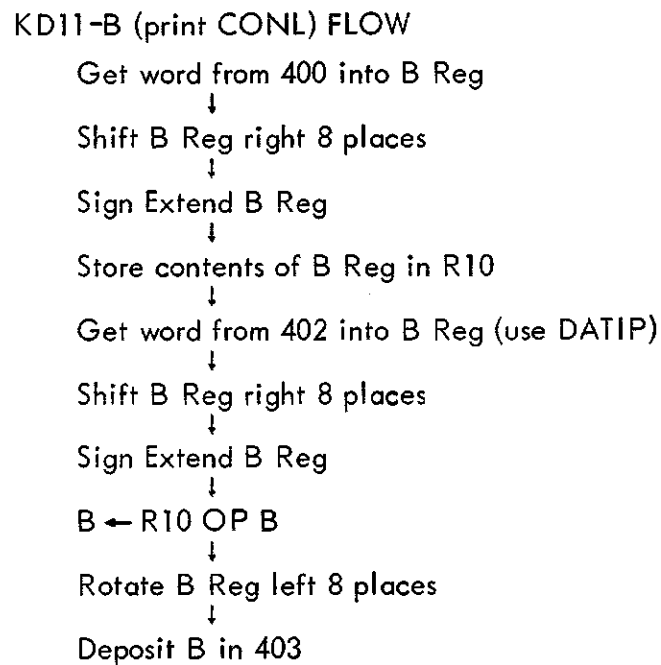
The B-Reg is also used as a left-right shift register to perform rotate and shift instructions as well as byte instructions. Byte instructions are handled by the processor as described in the next paragraph.

#### 4.5.1 Byte Instructions

For the correct execution of all instructions that operate on data, the least significant bit of both the source and destination must line up with bit 0 of the A-leg and B-leg,



respectively. This same rule applies even if the instruction being executed is a byte operator. For even bytes this is no problem, since the data received from the Unibus has the least significant bit of the low order byte lined up properly. For odd bytes, it is necessary to shift the data word right eight bit positions to properly line up the data. Then if the destination is an odd byte, the data must be shifted eight bits left before it is restored to its proper memory location. This operation is illustrated in the example in Figure 4-5 with the associated processor flow listed as follows:



Special multiplexing is required to determine the C, V, N, and Z condition codes and to complete rotates for byte instructions. This multiplexing is shown on print DPE.

#### 4.5.2 Instruction Register (IR) and IR Decode

The Instruction Register (IR), shown on print DPF, is used to store the 16-bit PDP-11 instruction loaded off the Unibus during interpretation by the processor. The output of the IR drives the decode logic found on prints DPF and DPG. The functions of the IR decode logic are described in Appendix B.

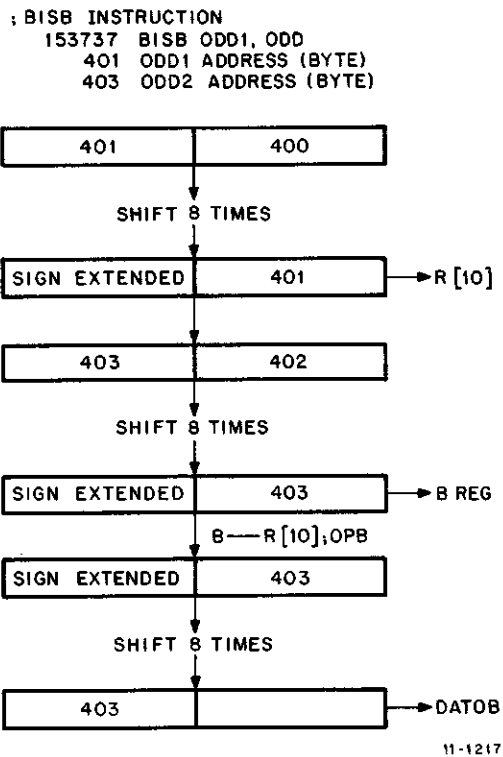


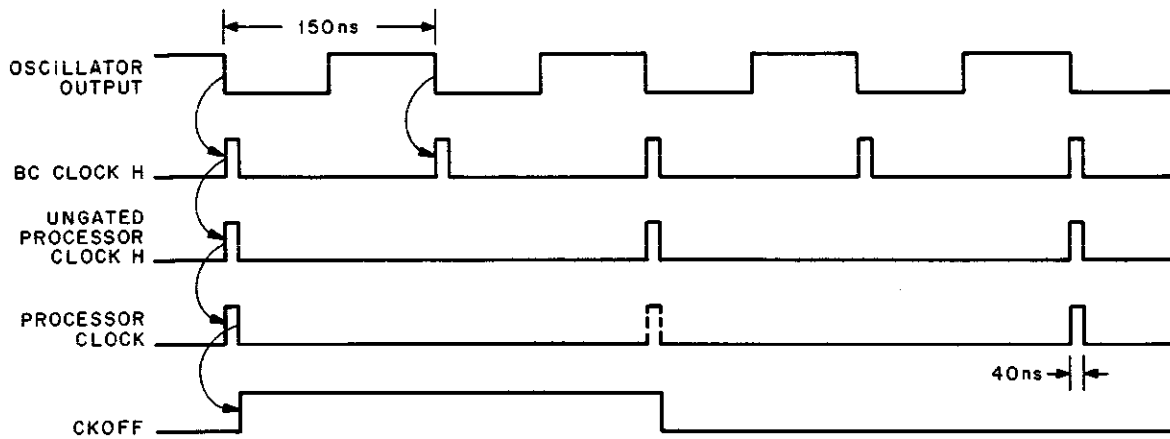
Figure 4-5 Byte Format for Shifting Instructions

#### 4.6 DATA PATH CONTROL AND CLOCKING

With the exception of provisions for the PSW, the KD11-B Data Path is rather general purpose. It is the Data Path Control (DPC) that specializes the KD11-B to execute the PDP-11 instruction set. The Control Store (CS) hardware is shown on prints CONF and CONG.

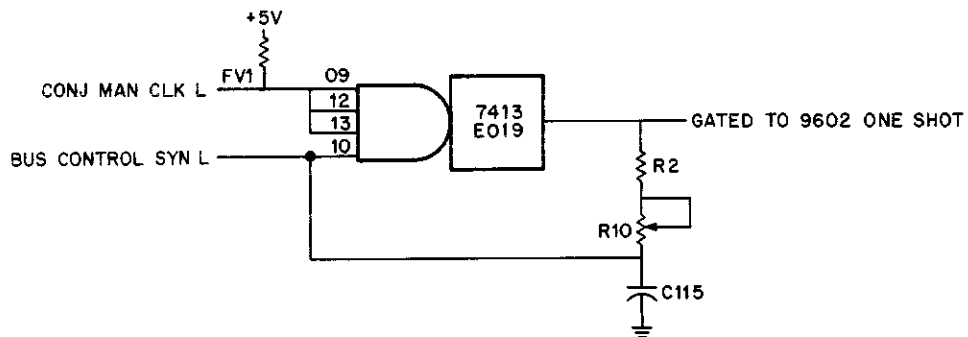
The KD11-B is a fully-clocked processor, in that events resulting in the alteration of storage registers occur only on a defined edge of the system clock pulse. For the most part, it is a single-clocked machine, in that there is a single astable oscillator that generates a pulse train to which the entire processor is synchronized.

A timing diagram of the system clocks is provided in Figure 4-6. The fundamental clock is provided by a free running astable oscillator that is similar to the circuit shown in Figure 4-7 (refer to print CONJ). Note that this particular astable oscillator circuit contains only one switch and therefore cannot become latched in a non-oscillating state, if R10 is adjusted to a sufficiently small value. R10 should be adjusted to provide a basic



11-1192

Figure 4-6 KD11-B Processor Clock Phasing



11-1193

Figure 4-7 KD11-B Basic Oscillator

clock period of 150 ns. The 40 to 60 ns pulse width of the systems clock is determined by a 9602 monostable IC (Appendix A), which is triggered by the falling edge of the basic oscillator output signal.

The output signal of the basic oscillator shaped by the 9602 is used as the Unibus control (BC) clock. The BC Clock is then divided by two to form the Processor Clock and the Ungated Processor Clock.

The three clock signals used in the KD11-B are as follows:

1. CONJ BC CLOCK  $\left(\begin{smallmatrix} H \\ L \end{smallmatrix}\right)$  PERIOD = 150 ns
2. CONJ PROC CLOCK  $\left(\begin{smallmatrix} H \\ L \end{smallmatrix}\right)$  PERIOD = 300 ns
3. CONJ UNG PROC CLOCK  $\left(\begin{smallmatrix} H \\ L \end{smallmatrix}\right)$  PERIOD = 300 ns

Clock signals 1 and 3 above are free running except when being synchronized with the Bus Control or when the maintenance mode clock disable pin on the M7261 module is grounded. Clock signal 2 may be halted to await the completion of a Unibus interrupt or the completion of a RESET instruction. Halting of the PROC CLOCK pulse train is accomplished by setting flip-flop CKOFF (print CONC). The equation for setting CKOFF is as follows:

$$\text{CONG CKOFF L} = \text{SET CKOFF} + [ (\text{BUT SER}) (\text{BR GRANT}) ]$$

Note that the signals SET CKOFF and BUT SER on print CONC are generated by the microprogram contained in the Control Store. The CS and microprogram will be discussed in greater detail in Chapter 5.

All edge-triggered storage elements in the KD11-B are triggered on the trailing edge of one of the three clock signals discussed above. Clocks 1, 2, and 3 are all in-phase with the exception of skew caused by propagation delays. All clock pulses are of the same 40-60 ns width. Chapter 6 discusses the use of the manual clock input, which enables the technician to troubleshoot the KD11-B using the KM11 maintenance panel.

#### 4.7 UNIBUS CONTROL

The Unibus control (BC) is found on prints CONC and CONC1, while the majority of the Unibus drivers are found on print COND. The BC can perform on request of the microprogram bus operations DATI, DATO, DATIP, DATOB, and retrieval of interrupt vectors. At the request of peripherals attached to the Unibus, the BC arbitrates BRs and NPRs. The BC is also responsible for detecting and causing a trap (Chapter 2), whenever there is an attempt by the processor to address non-existent memory or to access odd addresses illegally.

The BC operates in parallel with the DP. The microprogram may request a DATI and then perform other tasks, such as incrementing R7, as long as the Bus Address Register is unchanged. The Unibus Control proceeds with the DATI until the signal, slave sync (SSYN), is returned from the slave device. At this point, the BC waits for the microprogram to set the CKOFF flip-flop shown on print CONC. This signal indicates that the microprogram is ready to accept Unibus data. If the microprogram sets CKOFF before SSYN is received, the BC inhibits the oscillator until SSYN is received or a Unibus time-out occurs.

#### 4.7.1 DATI Timing

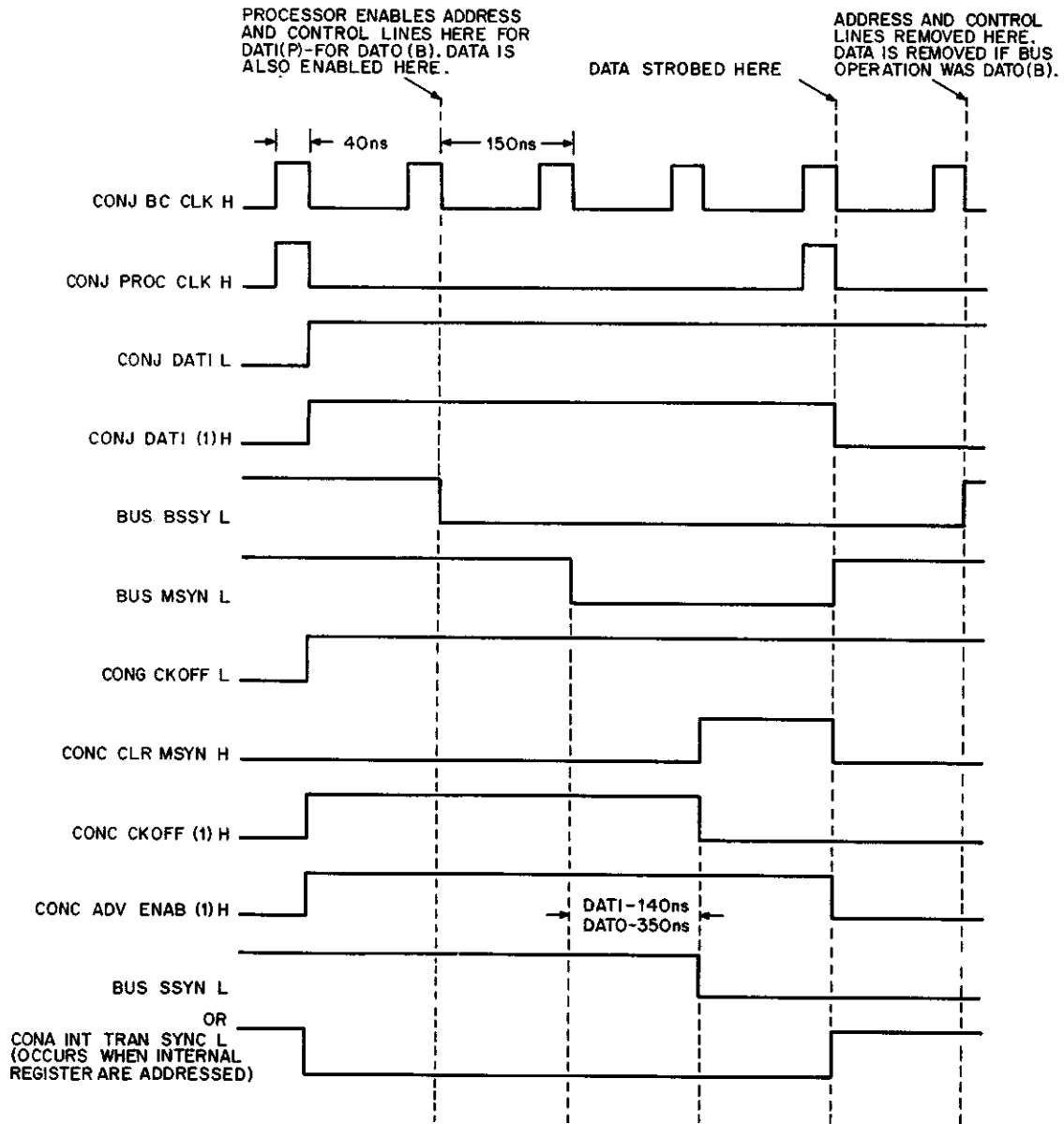
A DATI is used by the processor to retrieve data from devices attached to the Unibus. Figure 4-8 contains a timing diagram of the Unibus control signals for a DATI bus operation. Note that the signals BBSY, C0, C1, and the address lines may be set by the processor or bus master, whenever it is determined that the Unibus is free for use. The Unibus is free for use by the processor when the following equation is true:

$$\text{BUS FREE} = (\sim\text{BBSY}) (\sim\text{NPR}) (\sim\text{SACK})$$

Once BBSY, C0, C1, and the address lines are asserted, the master device must wait at least 150 ns before issuing MSYN. During this time the address and control lines of the Unibus are settling, so that when MSYN is issued, there will be no confusion as to the device addressed or to the direction of the data transfer. After MSYN is asserted, the BC must wait until SSYN returns from the Unibus and CKOFF is asserted. This indicates that data is available on the Unibus and the microprogram is ready to accept that data. Once the processor has strobed the data from the Unibus into a storage element, normally the B-Register, the signal MSYN is unasserted by the processor. BBSY, C1, C0, and the address are maintained for 150 ns after MSYN is unasserted.

#### 4.7.2 DATI Operation

The microprogram requests a DATI by asserting the signal CONG DATI L, which is the input to E05309 on print CONC. On the next processor clock following the assertion of CONG DATI L, the flip-flop DATI E017 on CONC is set. If the Unibus is free, BBSY is set.



11-1191

Figure 4-8 DAT1 and DAT0 Timing

Simultaneously with the assertion of CONC BBSY (1) L, the bus address drivers (print COND) enable the contents of the Bus Address (BA) Register onto the Unibus address lines. Note that the bus drivers for BUS A16 and BUS A17 are automatically enabled by the following equation:

$$\text{BUS A16 and BUS A17} = (\text{A15}) (\text{A14}) (\text{A13}) (\text{BBSY})$$

This allows PDP-11 processors, such as the KD11-B, that do not have extensive memory management facilities, to address peripherals registers that are located between 124K and 128K in the address space.

The MSYN flip-flop, E060 on print CONC, is normally set 150 ns after the issuance of BBSY. The setting of MSYN triggers a 9602 one-shot E025, shown at the lower left side of print CONC. This one-shot, which has a pulse width of 25 ns, is used to detect attempts at addressing non-existent memory by the processor. If SSYN does not appear on the bus before the signal CONC DAT TO (1) L is asserted by E034, then the microprogram is forced to execute an error trap sequence. For details on error trap execution refer to Paragraph 5.4.2.

SSYN is strobed into the holding register E005, shown on print CONC1, and generates the signal CONC SSYN (1) H. CONC SSYN (1) H enables an OR gate (E06208 shown in the center of print CONC). At this point, the following conditions exist:

1. BBSY, C0, C1, and MSYN are being applied to the Unibus by the KD11-B.
2. An address is enabled on the bus address lines by the processor.
3. Data is being driven onto the Unibus data lines by the addressed device or memory location.
4. SSYN is being generated by the addressed device.

The addressed peripheral device must maintain both its data and SSYN on the bus as long as MSYN is asserted. The Unibus control removes MSYN from the bus within 300 ns after SSYN and CKOFF are both set. The gating structure for removing MSYN can be traced back from the K input to the MSYN flip-flop (E060 on print CONC).

If MSYN, CKOFF, and the oscillator divider flip-flop are all set and the BC is waiting for SSYN, the oscillator input is inhibited and the oscillator stops. When SSYN is asserted, the input is released and MSYN is cleared. This method of synchronization causes no extra delay or flip-flop set-up problem.

4.7.2.1 DATIP Operation - Note that the sequence for DATI and DATIP are almost identical. DATIP is used by the processor to prevent the modification of a memory location by a device other than the processor, while the processor is operating on that memory location. To further understand the need for DATIP consider the operation of the DM11, a 16-line Asynchronous Serial Line Multiplexer (DEC-11-HOMA-D).. The Buffer Active

Register in the DM11 indicates status information and initiates message transmission. To begin the transmission of a message, the processor sets a 1 in the DM11 Buffer Active Register. When the message has been transmitted, the DM11 performs an NPR transfer to its own status register and clears the appropriate channel status bit.

Typically, the program to set an appropriate bit in the DM11 status register will use a BIS instruction. To execute this instruction, the processor must first execute a DATIP to the address of the DM11 status register and obtain a copy of the current contents of the status register. The specified bit is then set in the copy of the DM11 status register that is held by the processor. Finally, the processor performs a DATO to the status register and returns the altered copy of the status register to the DM11.

If, for instance, at the time of the DATIP, channels 0, 1 and 2 were active, the processor would retrieve a status word of  $000007_8$ . Suppose the program desired to activate channel 4; the return status word would equal  $000027_8$ . If channels 0, 1, or 2 completed their transmission between the time the processor issued the DATIP and the DATO and the processor permitted the DM11 to clear its status register before the DATO cycle of the BIS instruction, it is obvious that the copy of the DM11 status register held by the processor would be invalid.

Memories manufactured by DEC inhibit the normal restore cycle when a DATIP is issued. Therefore, when the following DATO is issued, the memory does not have to wait for the completion of the previous restore cycle before continuing with the DATO operation. However, the processor must inhibit NPRs from the issuance of the DATIP to the completion of the following DATO. Therefore DATIP operations lengthen the worst case NPR latency of the processor.

4.7.2.2 DATIP Logic — The BC executes a DATIP whenever the flip-flops DATI and DATIP (E017 and E008 on print CONC) are simultaneously set by the microprogram. The equation for setting DATIP, E017, is as follows:

$$\begin{aligned}
 (\text{SET DATIP E063, pin 12}) &= (\text{CONG ENAB IN PAUSE L})^{(1)} \sim (\text{DPG ENAB NON MOD H})^{(2)} \\
 &\quad (\text{CONI ALLOW PC L})^{(3)}
 \end{aligned}$$



Signal number 1 is an indication that the microprogram anticipates the need for a DATIP. Signal number 2 confirms that the current instruction in the IR is one that requires the destination to be restored. The instructions TST, CMP, BIT, JMP, and JSR can never result in the modification of the destination by the processor. Therefore, it is not necessary to use the DATIP operation during the execution of these instruction. Signal number 3 ensures that DATIP is set on a processor clock rather than a BC clock. DATIP remains set following the transfer and inhibits the setting of NPG flip-flop E00712. It is directly cleared when the processor enables the destination in data during the next DATO, and NPR's are again allowed to be granted.

#### 4.7.3 DATO

DATO differs from DATI in that for a DATO the Unibus data lines are driven by the processor. Figure 4-8 shows that data is maintained on the bus for the duration of BBSY. In the KD11-B, a DATO operation requires cooperation between the BC and the microprogram. The steps executed by the microprogram for a DATO operation are illustrated in flow chart example listed below. Note that CKOFF and DATO must be set simultaneously, and that the microprogram control step that follows the DATO specification must enable the data from the appropriate storage register through the ALU and A MUX.

DATO FLOW

	<u>LOC</u>	<u>NXT</u>	
DATO STARTS	334	065	D1-5 DATO; ALBYT; CKOFF /GET TO D1-6 FROM D0-18 VIA GOTO
DATA PUT ON UNIBUS	065	305	D1-6 DRIVERS B; GOTO B2-2 (BUT SERVICE)

The microprogram initiates a DATO operation by setting the DATO flip-flop (E017 on print CONC). The 7400 gate, E007, generates the signal CONC DAT ENAB L, which enables the data drivers shown on prints DPA, DPB, DPC, and DPD and also clears DATIP.

#### 4.7.4 Byte Operations

Byte operations have the following significance to the KD11-B Unibus control (BC):

- a. An odd address may be places on the Unibus.
- b. For a DATOB, both C0 and C1 are enabled.

Byte operations have the following significance on the Unibus slave:

- a. No significance for DATIP operations.
- b. For DATOP operations, only the upper or lower eight bits of the addressed location should be altered.

#### NOTE

The master must properly position the data during a DATOB operation. For instance, if the operation is a DATOB to the odd byte of a location, the data must appear on Unibus data lines <15:8>.

In the processor, the ALLOW BYTE flip-flop (E043 on print CONC) serves to both permit odd addresses and to generate the appropriate C0 and C1 signals. The microprogram attempts to set the ALLOW BYTE flip-flop, whenever the possibility of a legal odd address or DATOB is anticipated by asserting the signal, CONG ALLOW BYTE L. The signal DPG BYTE L (shown as an input to E06303 on print CONC) confirms that the current instruction (IR) is a byte operation.

#### 4.7.5 Bus Errors

The following situations cause the bus error trap sequence to be executed:

- a. An attempt to illegally address an odd location in the memory space. For instance if the contents of R7 is odd at the beginning of an instruction fetch, a bus error trap will be executed because instructions must start at even addresses.
- b. An attempt to access non-existent locations in the memory space. A non-existent location is recognized when SSYN does not appear on the bus within 25  $\mu$ s of the setting of MSYN by the processor.

Either type of bus error causes the BE flip-flop, E050 on print CONC to be set. The BE flip-flop inhibits the signal CONC MSYN OUT H which removes MSYN from the Unibus whenever a bus error is detected. The signal CONC BUS ERROR (1) H causes the 256 x 4 ROMs (E092 and E102 on print CONF) that generates the next address for the microprogram to be disabled. This forces the microprogram to execute its next control step from microaddress 010<sub>8</sub>.

A double bus error is defined by two successive unsuccessful attempts at addressing the memory. On the second successive bus error, the microprogram is forced to location  $110_8$  by the simultaneous setting of the BE and DBE flip-flops (E050 and E060 on print CONC). The microprogram in the KD11-B is designed to cause a processor halt after two successive bus errors.

#### 4.8 INTERNAL UNIBUS ADDRESSES

All presently implemented PDP-11 processors, including the KD11-B, contain internal registers that have associated addresses in the Unibus address space. To the program executed by the processor, the internal registers are indistinguishable from peripheral or memory registers. However, access to the internal registers is not available to devices attached to the Unibus other than the processor.

In the KD11-B, the concept of internal registers has been expanded to include the serial communications line control and the line clock. Table 4-4 lists the internal Unibus addresses.

Attempts to address internal Unibus addresses are detected by the logic detailed on print CONA and illustrated in Figure 4-9. A characteristic of all addresses listed in Table 4-4 is that the odd byte of the address is equal to  $377_8$ . The signal CONA INT BUS ADDR L, generated by E039, indicates that the odd byte of the currently addressed register is  $377_8$  and that the bus address may be that of an internal register.

Table 4-4  
Unibus Addresses

Octal Address	Function
177702	General registers R0 through R7
177701	
:	
177707	Hidden registers used by the microprogram
177710	
177717	Program status register
177776	

Table 4-4 (Cont)  
Unibus Addresses

Octal Address	Function
177570	Console switch register
177571	Odd byte of console switch register
177560	Receiver or keyboard status register
177562	Receiver or keyboard buffer
177564	Transmitter or printer status register
177566	Transmitter or printer buffer
177546	Line clock status register

The read-only memories of ICs E030, E069, and E068 decode the least significant eight bits of the Unibus address to determine which, if any, of the internal registers are currently being accessed.

The timing diagram contained in Figure 4-8 shows that the signal CONA INT TRAN SYNC L replaces SSYN for internal registers. Note that bus addresses, C1, C0, and MSYN are driven onto the Unibus during attempts to address internal registers. However, the signals output by ROMs E068 and E069 reconfigure the data path (DP) such that during a DATI from 177776, for example, the PSW is enabled onto the DP.

During transfers to and from the processor, to registers, and to memory, data is normally constrained to be written from and into the B-Reg. The reason is that most of the elements contained on the A-leg may be addressed with their corresponding Unibus address. Therefore, almost any data transfer may be from or to the DP. Since it is not possible to both read and write into the DP on the same clock pulse, it is necessary for the microprogram to receive and transmit Unibus data from the B-Reg.

In order to understand the decoding sequence for the ROMs F030, F068, and E069, it is necessary to refer to the ROM maps (K-RL-M7260-8 and K-RL-M7261-8).

#### 4.9 BUS REQUESTS (BR)

The KD11-B responds to BRs in a manner similar to that of the other PDP-11 processors. Peripherals may request the use of the Unibus in order to make data transfers or to interrupt the current processor program by asserting a signal on one of four BR lines, numbered

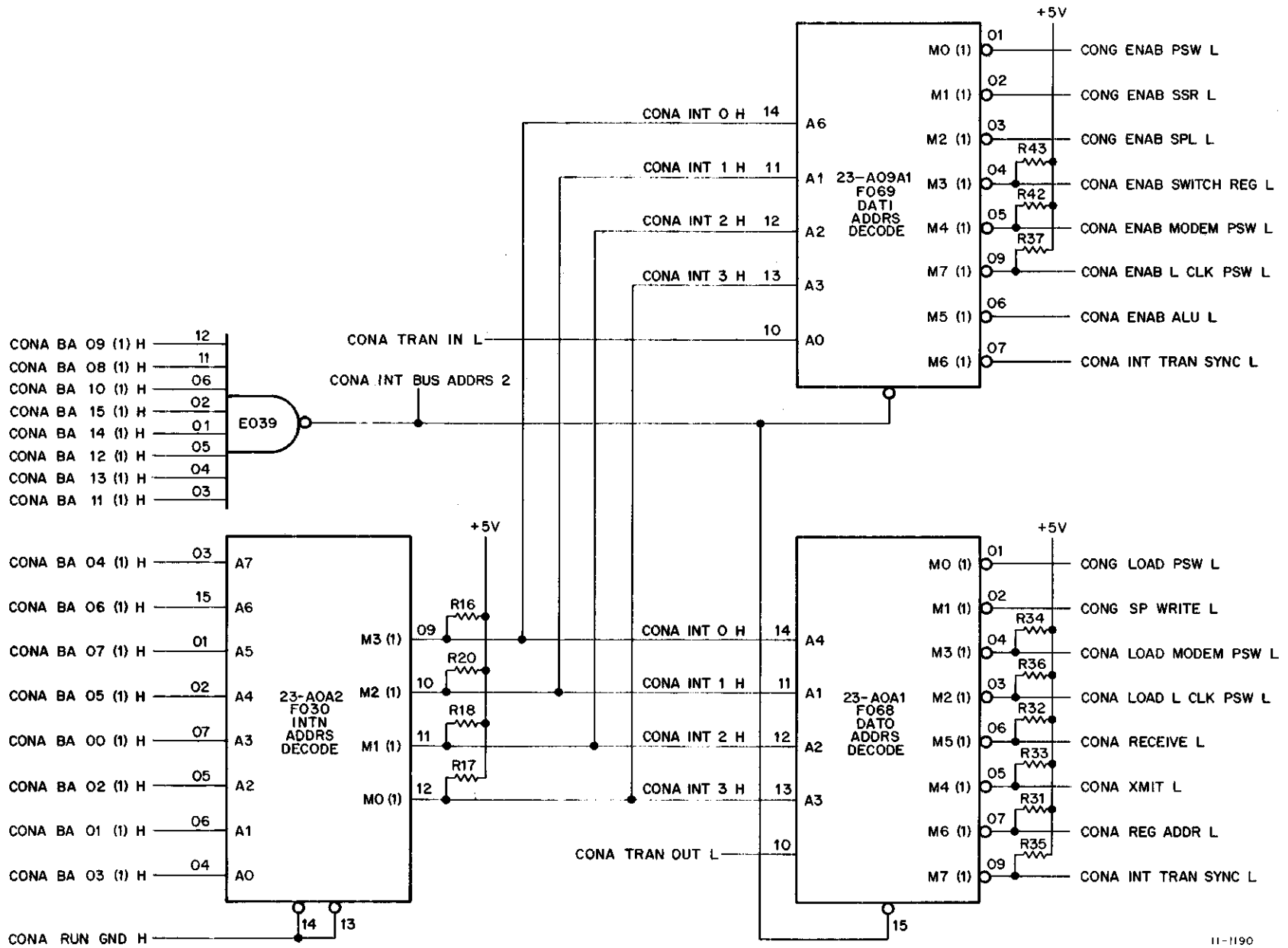


Figure 4-9 Unibus Address Decoding

4, 5, 6, and 7 in order of increasing priority. For instance, if two devices, one at priority 5 and the other at priority 7, assert BRs simultaneously, the device at priority 7 is serviced first. Furthermore if the processor priority, determined by <7:5> of the PSW, is at level 4, only devices that request BRs at levels higher than 4 such as BR 7, BR 6, or BR 5 are serviced. Table 4-5 contains the order of priorities for all BRs and other traps.

Table 4-5 Trap Priorities

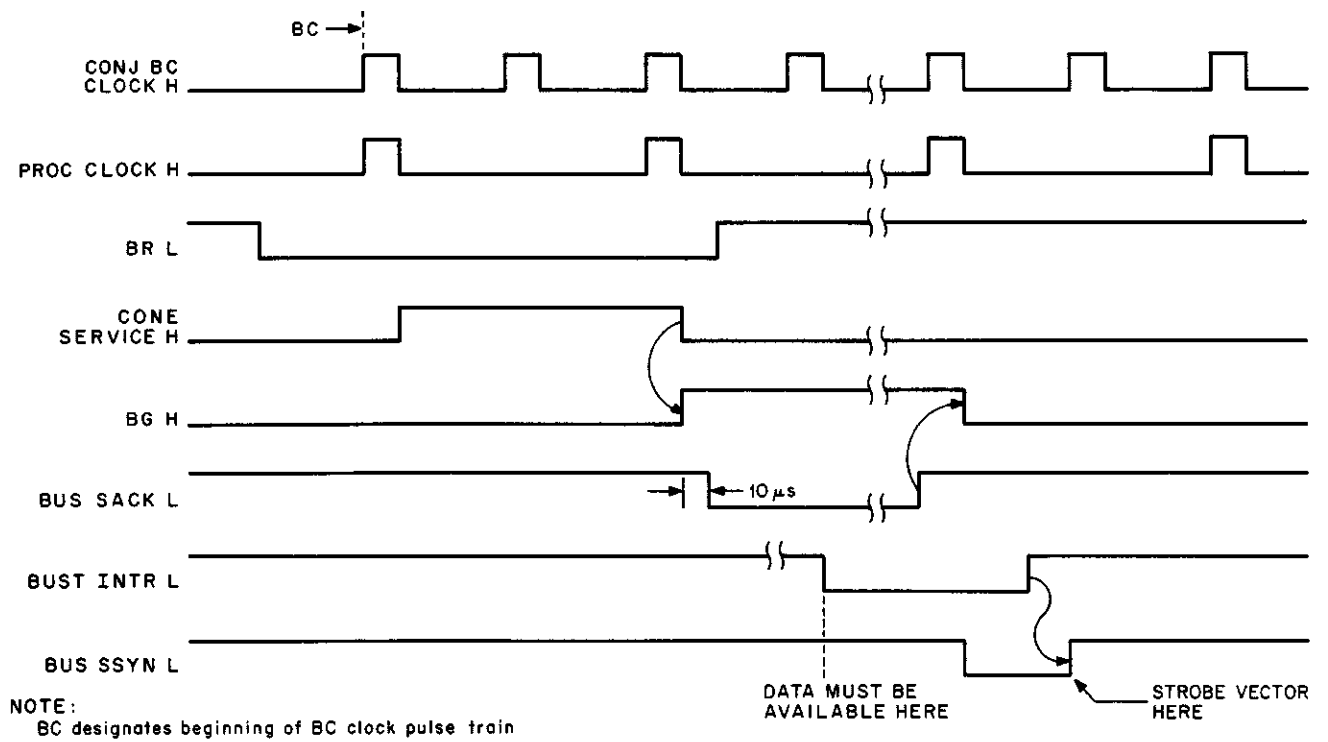
Priority	Service Priorities
<p>HIGHEST</p> <p>LOWEST</p>	<ol style="list-style-type: none"> <li>1. T-bit trap</li> <li>2. Stack overflow</li> <li>3. Power fail</li> <li>4. BR7</li> <li>5. BR6</li> <li>6. Internal line clock</li> <li>7. BR5</li> <li>8. BR4</li> <li>9. UART receive</li> <li>10. UART transmit</li> <li>11. Console stop</li> <li>12. Next instruction fetch</li> </ol> <p style="text-align: right;">Acknowledged by WAIT instruction.</p>

Since a BR can cause a program interrupt, it may be serviced only after the completion of the current instruction in the IR. A device that requests a program interrupt must at the appropriate time (Figure 4-10) place a vector address on the Unibus data lines. The processor first stacks away the current contents of PSW and R7; then a new R7 is loaded from the contents of the vector address, and a new PSW is loaded from the contents of the vector address plus two. An example of the flow that handles a BR is listed as follows:

LOC	NXT	*BUS GRANT SERVICE
		/GET TO BG-1 FROM BUT SERVICE
040	305	BG-1 BUT INTERRUPT; GO TO B2-2 (BUT SERVICE)
		↓ /IF INTERRUPT GO TO INT-1
		/IF NO INTERRUPT FALL THROUGH
		TO B2-2
LOC	NXT	*INTERRUPT SERVICING
		/GET TO INT-1 FROM BG-2 VIA BUT INT (TRUE)
325	246	INT-1 R(12) ← UNIBUS DATA; SET SLAVE SYNC; GO TO ET-3
		↓
LOC	NXT	
246	247	ET-3 B, BA ← R(6) - 2; ENAB OVER
247	226	ET-5 R(6) ← B; CKOFF; DATO
226	251	ET-6 DRIVERS ← PS
251	252	ET-7 B, BA ← R(6) - 2; ENAB OVER
252	253	ET-8 R(6) ← B; CKOFF; DATO
253	254	ET-9 DRIVERS ← PC
254	255	ET-10 BA ← R(12); DATI; CKOFF
255	256	ET-11 PC ← UNIBUS DATA
256	257	ET-12 BA ← R(12) + 2; DATI; CKOFF
257	305	ET-13 PS ← UNIBUS DATA; GO TO B2-2 (SERVICE)

The microprogram indicates the end of instruction execution by asserting the signal CONE BUT SERVICE L. BRs are arbitrated by the ROM E012 (shown on print CONC1). If there is an impending BR, the signal CONC BR GRANT H is asserted by E02208 (print CONC1). When CONE BUT SERVICE L is issued, the appropriate BG is clocked into the storage register (E021). Simultaneously, the microprogram address is forced to the bus grant sequence by the logic shown on print CONE.

In the KD11-B, interrupts for the SCL and the line clock are not entered the same way as interrupts from other devices attached to the Unibus. Interrupts from the SCL and line clock are handled in the same manner as power fail (Paragraph 4.13) and stack overflow traps. For all of these events, the microprogram address is altered when CONC BUT SERVICE L is issued to force the microprogram into the appropriate routine, which simulates the appropriate interrupt or trap.



11-1188

Figure 4-10 Bus Request (BR) Timing



Paragraph 5.4.2 describes the microprogram trap handling routine in detail. Note that the appropriate vector address for SCL and line clock interrupts are generated by the constants generator, which is the E025 ROM shown on print DPB.

#### 4.10 NON-PROCESSOR REQUESTS (NPR)

NPRs are a facility of the Unibus that permit devices on the Unibus to communicate with each other with minimal participation of the processor. The processor's function in servicing an NPR is simply to give up control of the bus in a manner that does not disturb the execution of an instruction by the processor. For instance, the processor may not relinquish the bus following a DATIP (Paragraph 4.7.2.1).

An NPR is received through a bus receiver (print COND) and clocked into storage register E005 (print CONCI). If conditions are appropriate to permit an NPG to be issued by the KD11-B, the signal CONC SET NPG H is issued by E01406. CONC SET NPG H is generated according to the following equation:

$$\text{CONC SET NPG H} = (\sim\text{DATIP}) \cdot (\sim\text{SACK DELAYED}) \cdot \text{RUN}$$

The signal CONC SET NPG H causes flip-flop E033 to be set, which in turn causes NPG to be placed on the Unibus. Note that both NPGs and BGs will be issued by the KD11-B for a period of 10 $\mu$ s. If the requesting device does not respond with SACK within this period, the 9602 timer IC (shown in the upper right hand corner of CONCI) trips, causing flip-flop E034 to be set. This in turn causes the pending BG or NPG to be cancelled, and the processor to continue operation.

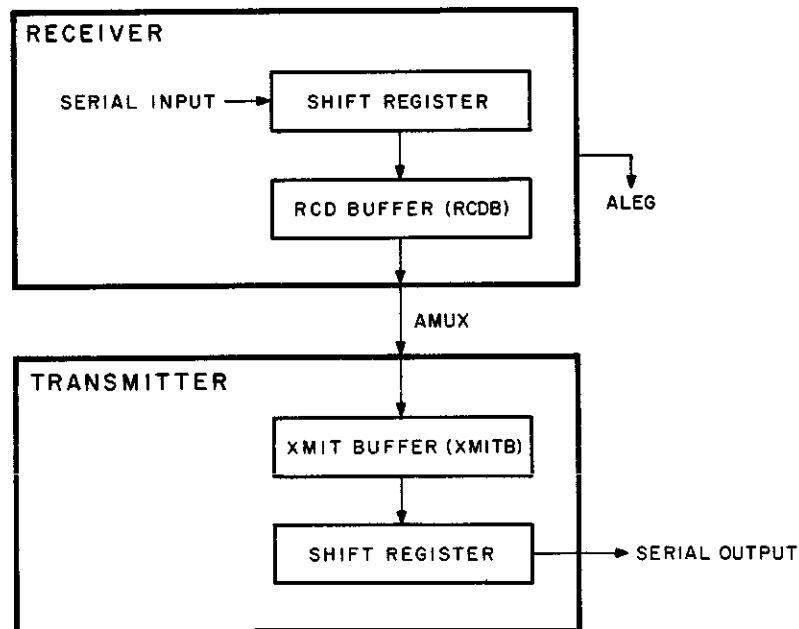
#### 4.11 SERIAL COMMUNICATIONS LINE DESCRIPTION (SCL)

The SCL of the KD11-B is essentially program compatible with the KL11 Teletype control. The heart of the serial communications line logic (prints DPH and DPH1) is the Universal Asynchronous Receiver Transmitter (UART), an MOS-LSI IC. The UART is easily recognized on the M7260 module because it is the only 40-pin dual in-line package used in the KD11-B. The UART is the only IC in the processor that requires two supply voltages, +5V and -12V. The -12V supply (print DPH) for the UART is generated by placing four diodes in series with the -15V supplied by the power supply (see Part 4).

The additional circuitry other than the UART (prints DPH and DPH1) serves the following purposes:

- a. Generation of the reader RUN signal that is used to control the low-speed paper-tape reader found on model 33 Teletypes.
- b. Generation of status bits and interrupts to make the KD11-B SCL program compatible with the KL11.
- c. Generation of the 20 mA current loop necessary to operate model 33 Teletypes, VTO5s and LA30s.

An important feature of the KD11-B SCL is double-buffering. An understanding of double-buffering (Figure 4-11) may be gained by studying the programming example provided. In order to receive or to transmit data at the maximum rate, it is necessary only to empty or to fill the appropriate UART buffer once every character time. Conversely, on single-buffered devices such as the KL11, it is necessary to empty or fill the appropriate buffer in one bit time.



11-1189

Figure 4-11 Double-Buffering Data Flow

### UART Sample Programs

```
LOOP:  TSTB   RCDSTA           ;Test for a received character
       BPL                    ;Go to Loop if no character
       TSTB   XMITST         ;Test the XMIT condition
       BPL
       MOVB   RCDB, XMIT     ;echo character
       ;If at this point it is desirable to ;If at this point it is desirable to issue a
       ;RESET it is necessary to send a mild
       ;character to insure that the desired
       ;character has been completely transmitted

       TSTB   XMITST
       BPL
       MOVB   NULL, XMIT
       TSTB   XMITST         ;When null character is
       BPL                    ;clear of the
       RESET                ;XMITB
```

The above programs are sample programs that utilize the UART. The first program simply echoes a character received from the UART into the transmitter of the UART. The second program illustrates the proper use of the RESET instruction, following an instruction that caused the SCL to transmit a character. RESET should not be issued until the last desired character has cleared the UART transmitter shift register.

On print DPH, the transmitter DONE flag is seen only as an indication that the transmitter buffer is empty (TBMT). It is guaranteed that the TBMT flag will set at least one character time before the UART has finished transmitting the last character transferred to it. A RESET instruction that occurs while a character is in the process of being transmitted aborts that character transfer. Therefore, the only safe way to issue RESET instructions, following an instruction that has transmitted a character through the UART, is to transmit a null character prior to the issuing the RESET instruction. Some care must be used in selecting the null character as it may be garbled by the RESET instruction. When the null character clears the UART's transmitter buffer, it is safe to issue the RESET instruction. When the SCL maintenance mode is enabled by setting the transmitter status bit (2), the serial output is fed back into the serial input just as in a

standard KL11. The transmitter status register address is 177564<sub>8</sub>. The SCL always appears to the program as the last device at the BR 4 interrupt level.

There is a provision in the SCL control (print DPH) to disable the internal clock and to provide an external clock for the UART. External clocks consisting of TTL compatible signals must be square waves of up to 160 kHz. The clock frequency must always be 16 times the SCL baud rate.

#### 4.12 LINE CLOCK

The line clock of the KD11-B is program compatible with the KW11-L. The line clock circuitry consists of 4 flip-flops and approximately 12 gates (print CONH). The line clock is the last device at the BR 6 interrupt level. The line clock derives its input (LTC L) from the power supply.

#### 4.13 POWER FAIL

The KD11-B power fail/auto restart circuitry (print CONH) serves the following purposes:

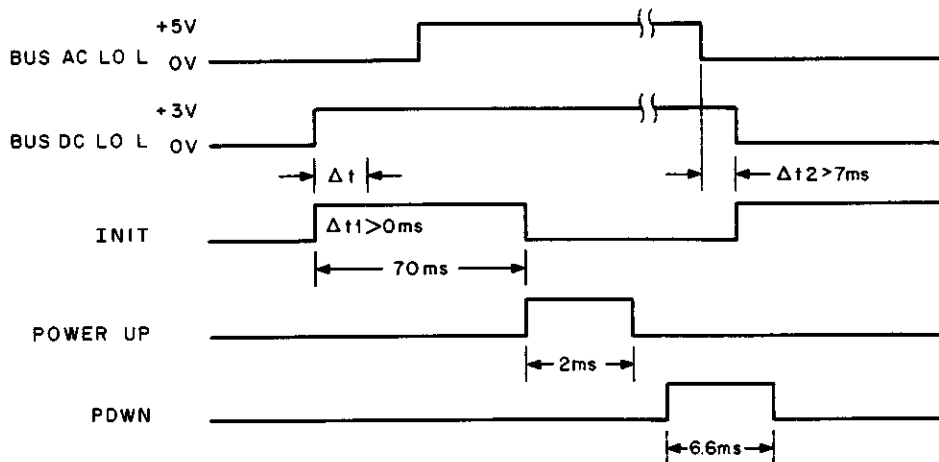
- a. Initializes the microprogram, the Unibus control (BC), and the Unibus to a known state immediately after power is applied to the computer.
- b. Notifies the microprogram of an impending power failure.
- c. Prevents the processor from responding to an impending power failure for 2 ms after initial start-up.

The actual power fail/auto restart sequences are microprogram routines. The operation of the power fail/auto restart circuitry depends on the proper sequencing of 2 bus signals AC LO and DC LO. Because of the electrical properties of the Unibus drivers and receivers, the entire computer system must be powered up for the machine to operate. Therefore, the processor is notified of a Power Fail in peripherals as well as its own ac source.

The notification of power status of any PDP-11 system component is transmitted from each device by the signals BUS AC LO L and BUS DC LO L (Figure 4-12). The power-up sequence shows that BUS DC LO L is unasserted before BUS AC LO L is unasserted. When BUS DC LO L is unasserted, it is assumed that the power in every component of the system is sufficient to operate. When BUS AC LO L is unasserted, there is sufficient stored energy in the regulator capacitors of the power supply to operate the computer for 5 ms, should power be shut down immediately.

As power is shut down, note that BUS AC LO L is asserted first. BUS AC LO L is an indicator that warns the processor of an impending power failure. When BUS DC LO L is asserted, it must be assumed that the computer system can no longer operate predictably. In fact, memories manufactured by DEC use BUS DC LO L as a switch signal. When BUS DC LO L is asserted, these memories turn themselves off even if power is available.

Time  $\Delta t_2$  (Figure 4-12) is the time delay between the assertion of BUS AC LO L and the assertion of BUS DC LO L; it must be greater than 7 ms. This allows for power to be rapidly cycled on and off. According to PDP-11 specifications, upon system start-up, a minimum of 2 ms of run time is guaranteed before a power fail trap occurs, even if the line power is removed simultaneously with the beginning of the power-up sequence. After the power fail trap occurs, a minimum of 2 ms of run time is guaranteed before the system shuts down. Given the tolerances permitted in the timing circuitry used in most equipment,  $\Delta t_2$  must be greater than 7 ms.



11-1187

Figure 4-12 BUS ACLO and BUS DC LO Timing Diagram

Whenever an impending power fail is sensed, a program trap occurs that causes the present contents of R7 and the PSW to be pushed onto the memory stack, which is determined by the contents of R6. R7 is then loaded with the contents of memory location  $24_8$ , and the PSW is loaded with the contents of location  $26_8$ . Processing is continued with the new R7 and PSW. The program must prepare for the impending power failure by storing away volatile registers and reloading location  $24_8$  and  $26_8$  with a power-up vector. This vector points to the beginning of a restart routine.

When power is restored, the processor loads R7 with the contents of location  $24_8$  and the PSW with the contents of location  $26_8$ . Note that no stacking is performed on an auto restart. Also the HALT switch is ignored if the console lock is set. After loading R7 and the PSW, processing continues if the HALT switch is not depressed. Presumably, the program will prepare locations  $24_8$  and  $26_8$  for another power failure. If the HALT switch is depressed and the console lock is not enabled, the processor powers up in the halt state.

Schematics of the power fail, auto restart, and bus reset logic are found on print CONH. As shown on Figure 4-12, E07106 generates a 70 ms processor INIT pulse as soon as BUS DC LO L is unasserted after power is applied to the computer. At the end of 70 ms the PUP one-shot, IC E08209, is fired if BUS AC LO L is unasserted. At this point, the processor begins to load R7 and the PSW if the HALT switch is not depressed. The PUP one-shot generates a 2 ms pulse during which time the assertion of BUS AC LO L is not recognized.

After PUP has been reset, the assertion of BUS AC LO L fires the one-shot E08206. Flip-flop E09708 is set by the leading edge of the one-shot's pulse. Note that E09708 is not synchronized to the processor clock. Flip-flop E09706 generates the signal CONH PDWN SYNC (1) L, which is synchronized to the processor clock. A power fail trap can be recognized by the microprogram whenever CONE BUT SERVICE L is issued. The various traps are arbitrated by the ROM F101 (print CONE).

If a momentary power failure occurs that causes the assertion of BUS AC LO L but does not cause the assertion of BUS DC LO L, the processor will restart when the PDWN (0) L one-shot times out retriggering the INIT one-shot simultaneously with DC LO H becoming unasserted.

## CHAPTER 5 MICROPROGRAM CONTROL

### 5.1 INTRODUCTION

This chapter describes the microprogram control implemented in the KD11-B processor. The flow notation used in the microprogram flow section of the prints is described in Paragraph 5.5.1. The difference between microprogram control and conventional control in a computer processor is described in Paragraph 5.2. Paragraph 5.3 describes the KD11-B Control Store (CS) structure; Paragraph 5.4 describes the technique of branching within microroutines in the CS; and Paragraph 5.5 describes the microprogram flow, including instruction interpretation, Unibus control coordination, interrupts, traps, and console functions.

### 5.2 MICROPROGRAMMED CONTROL VS CONVENTIONAL CONTROL

The control section of a conventional computer is a complex collection of specialized logic circuits. These circuits generate the timing signals that constitute the major and minor time states of a machine cycle. During each time state, these control signals configure the Data Path (DP), determine function performed within the Arithmetic/Logic Units (ALU), influence the Unibus control (BC), etc. Major disadvantages associated with this conventional approach are its complexity, the large amount of logic required, its inflexibility, and difficulty of making modifications.

A microprogrammed processor such as the KD11-B results in a reduction in the amount and complexity of the control logic, while facilitating a systematically implemented and easily modified control section. Basically, a microprogram involves the "execution" of a

sequence of microsteps from the Control Store (ROMs). Execution of a microstep causes the assertion of a set of control signals specified in the control store word associated with that microstep. By executing appropriate sequences of microsteps (known as a microroutines), the KD11-B can be made to interrupt PDP-11 instructions. Other functions such as console functions, interrupts, and traps are also accomplished by specialized microroutines.

### 5.3 CONTROL STORE

Figure 5-1 shows the format of the KD11-B Control Store word. There are 256 such words, each of which has the same fields. The fields, the possible values they may contain, and the significance of each value are described in Table 5-1. The Control Store is shown on prints CONF and CONG.

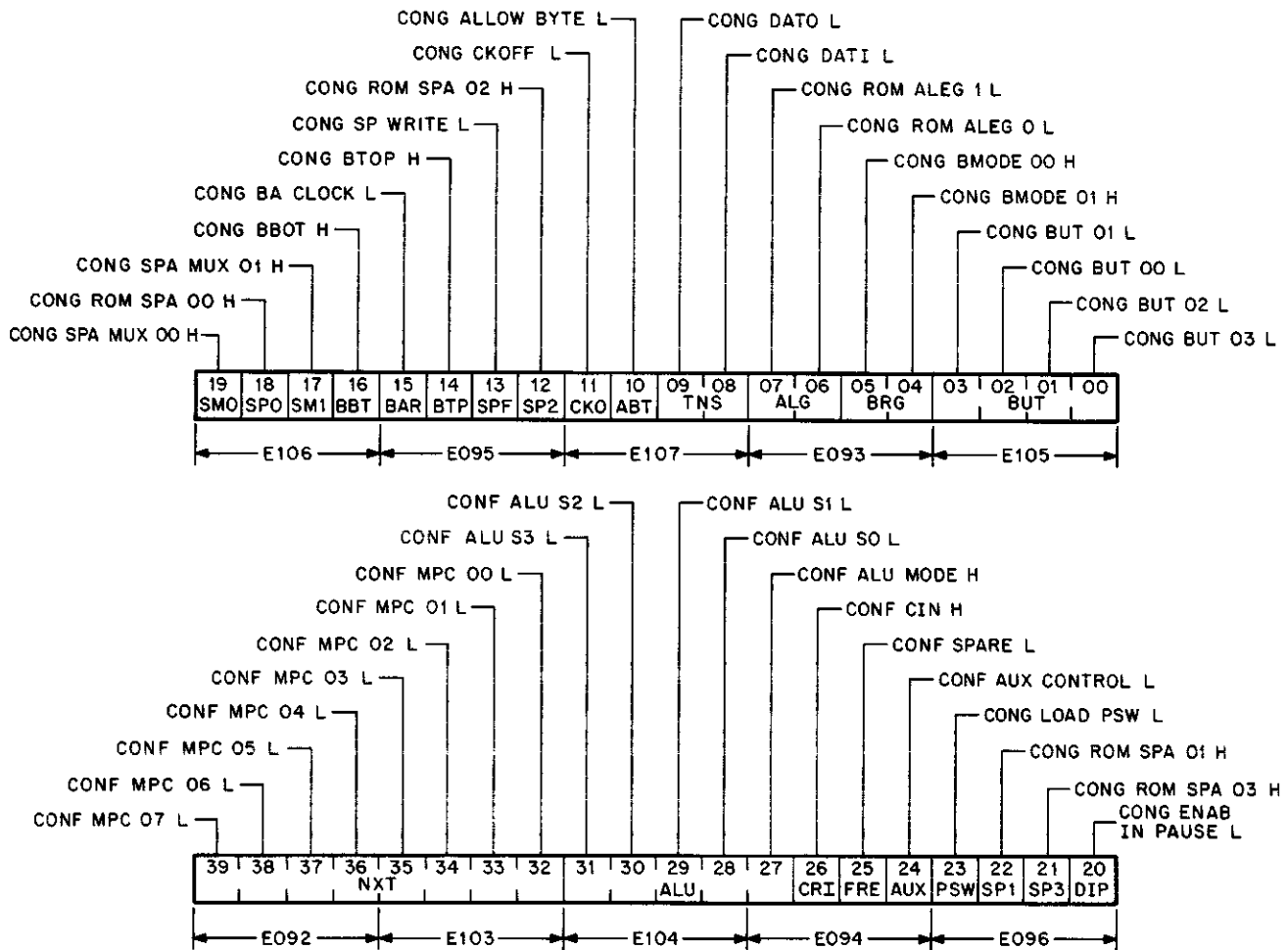
An explanation of the notation will aid in relating the Control Store word to the reset of the print set. Each field within the Control Store has been given a name (e.g., BUT, BRG, ALG.....ALU, NXT). These field names are used throughout documentation of the microprogram.

The signal coming from each bit is named according to the convention used throughout the print set. Note that several signals may be associated with a single field (e.g., the BUT field controls four signals, CONG BUT 01 L, CONG BUT 00 L, CONG BUT 02 L, and CONG BUT 03 L).

A field may contain any one of a number of different alternative bit patterns. To facilitate microprogramming, these alternatives have been given symbolic names, making it possible to work with the microprogram at a symbolic level rather than in binary. For example (Table 5-1), one of the alternative values that can be assigned to the ALU field is OR (A or B). This value corresponds to a bit pattern of 01001 (CS 37:33 = 01001).

The data word output from the CS is determined by the contents of the MPC registers (E091 and E102 shown on print CONF).





11-1216

Figure 5-1 Control Store Word Bit and Field Format

Table 5-1  
KD11-B Control Store Fields

Field	Description
BUT	<p>Branch on microtest. The BUT field has two uses; a) specify microprogram conditional branches, and b) as an encoded miscellaneous field. The values this field can assume are grouped by these two uses.</p> <p>Branching within the microprogram is accomplished by wiring conditional signals with the open-collector outputs of the NXT field of the Control Store. Each BUT condition has the minimum number of control bits</p>

Table 5-1 (Cont)  
KD11-B Control Store Fields

Field	Description
BUT (cont)	<p>required. This makes the range of branching restrictive, but it minimizes logic (print CONE). Table 5-2 lists the microstep in which each BUT is performed, the possible conditions and resulting destination of the microprogram branch.</p> <p>Microprogram conditional branches:</p> <p>NON No effect.</p> <p>JSRMP Microprogram branch on JMP or JSR instruction.</p> <p>IRD Microprogram branch on results of Instruction Register Decode.</p> <p>BYT Microprogram branch to distinguish a) byte and non-byte instructions and b) odd/even byte references.</p> <p>DST Microprogram branches on destination mode <math>IR &lt; 5:3 &gt;</math></p> <p>MOV Microprogram branch to distinguish both MOV and MOV B from other instructions.</p> <p>INT Microprogram branch on interrupt to be processed.</p> <p>UNY Microprogram branch to distinguish Unary instructions.</p> <p>SW Microprogram branch dependent on console switch action.</p> <p>NMD Microprogram branches to distinguish non-modifying instructions (e.g., CMP, TST, etc).</p> <p>SRV Microprogram branch at end of instruction sequence to determine if any condition requires service before going off to fetch next instruction.</p> <p>Miscellaneous encoded field:</p> <p>CON Enable the constants ship on the A-leg.</p> <p>INI Trigger BUS INIT L during the RESET instruction.</p> <p>SVS Set slave sync on Unibus during the interrupt sequence.</p> <p>ENO Enable the stack overflow detection logic.</p> <p>IRC Clock data into the Instruction Register.</p>

Table 5-1 (Cont)  
KD11-B Control Store Fields

Field	Description
BRG H L SR SL	Control the B-Register. Hold, do not modify. Load. Shift right once. Shift left once.
ALG SP NUL SPR PSW	A-leg control. Determines what is enabled onto the A-inputs of the ALU. Scratch pad. Nothing. Low orders eight bits (right half) of the scratch pad. Program Status Word.
TNS NON I O IP	Initiation of Unibus transfer. No effect. Initiate DATI. Initiate DATO. Initiate DATIP
ABT NO YES	Allow byte reference on current Unibus transfer.  No effect.
CKO OFF ON	Inhibit the processor clock until pending Unibus transfer is complete.  No effect.
SPA	Scratch pad address. This field is physically split in the control store word. It is made up of:  SPA = SP0 = CS < 18 > SP1 = CS < 22 > SP2 = CS < 12 > SP3 = CS < 21 >

Table 5-1 (Cont)  
KD11-B Control Store Fields

Field	Description
	R0 through R17 - Scratch pad address.
SPF	Scratch pad control function.
REA	Scratch pad contents not modified.
WRI	Write into scratch pad.
BLG	<p>B-leg control. Determines what is enabled onto the B input of the ALU. This field is physically split in control store word.</p> <p>BLG = BTP (B Top - Upper Byte) = CS &lt;14&gt;  BBT (B Bottom - Lower Byte) = CS &lt;16&gt;</p>
BRG	B-Register
SEX	B-Register sign extended. Bit 7 of the B-Register is propagated from bit 7 to bit 15.
+1	The constant.
BAR	Bus Address Register Control.
H	Hold, do not modify.
L	Load.
SAM	<p>Scratch pad address multiplexer control. This field is physically split in the control store word.</p> <p>SAM = SM0 &lt;19&gt;  SM1 &lt;17&gt;</p>
ROM	Scratch pad address taken from control store word (see SPA field).
IRS	Scratch pad address taken from source register bits of Instruction Register, IR <8:6>.
IRD	Scratch pad address taken from destination register bits of Instruction Register, IR <2:0>.
BAR	Scratch pad address taken from Bus Address Register low order 3 bits, BA <2:0>.
PSW	Program Status Word control.

Table 5-1 (Cont)  
KD11-B Control Store Fields

Field	Description
H	Hold
L	Load
AUX	Auxilliary ALU control enabled.
OFF	
ON	
CRI	Enable carry in to ALU.
OFF	
ON	
ALU	ALU function.
AL	A logical
AA	A arithmetic
AB	A and B
ABBAR	A and ones complement of B
ZERO	Output zero
A OR B	A or B
BL	B logical
A + B	A plus B
AXORB	A exclusive or B
A-B-1	A minus B minus 1
BBAR	1's complement of B
-1	Output the constant minus one
A-1	A minus one
ABAR	1's complement of A
ASL	Arithmetic shift B left
ROL	Rotate B left
ASR	Arithmetic shift B right
ROR	Rotate B right

These are used during shift and rotate instructions to control the serial shift inputs to the B register.

## 5.4 BRANCHING WITHIN MICROROUTINES

A microroutine is composed of a sequence of microsteps. Every microstep specifies the location of the next microstep in a sequence viz. the NXT field. During the execution of a microstep, the signals resulting from the NXT field are loaded into the MPC (Microprogram Counter). The MPC specifies the location from which the next microstep will be executed (print CONF). Conditional branching within a microroutine is accomplished by wire-ORing signals into those signals coming from the NXT field, while they are being loaded into the MPC. Each branch condition controls the minimum number of bits required. This restricts the range of branching, but it minimizes the logic (print CONE). This provides control for all the bits in the MPC. Table 5-2 shows the location of each microcode branch, the destinations, and associated conditions:

In general, microsteps are not executed from numerically sequential locations. This extra degree of complexity (and an extra eight bits in each CS word to specify the NXT location) enables the minimization of logic. Microprogram branching is illustrated in the example discussed in Paragraph 5.5.2.

Table 5-2  
Microprogram Branches (BUT)

BUT	Source	Destination	Comment
IRD (IR DECODE)	F-5	S0-1 THRU S7-1	ALL DOUBLE OPERAND INST.
		D0-1 THRU D7-1	SINGLE OPERAND INST.
		B-1	BRANCH, CHANGE PC
		B2-2D	BRANCH, PC UNCHANGED
		MCC-1	SET OR O CLEAR COND, CODES
		R1-1	RTS
		R2-1	RTI
		W-1	WAIT

Table 5-2 (Cont)  
Microprogram Branches (BUT)

BUT	Source	Destination	Comment
IRD (IR DECODE) (Cont)		H-1	HALT
		ET-1	EMT
		BT-1	BREAK POINT TRAP
		IT-1	IOT
		T-1	TRAP
		RT-1	RESERVED INST.
		RST-1	RESET
DST (DESTINATION)*	S0-2, SBE-2	D0-1 THRU D7-1	
	CCM-2	CC-1 SC-1	CLEAR COND. CODES SET COND. CODES
BYT (BYTE)	S0-1	SBE-1	BYTE SOURCE DATA (MODE 0)
	S1-2	SBE-1	EVEN BYTE SOURCE DATA
		SBO-1	ODD BYTE SOURCE DATA
MOVE	D0-1	DBO-1	BYTE INST. OTHER THAN MOVE
		MB-0	MOVB INST. (BYTE)
		D0-3A	MOV INST. (NOT BYTE)
NMD (NON-MODIFYING)	D0-3, D0-3A	B2-2A	NON-MODIFYING INST. TST, CMP, BIT
		D1-4	B2-2B
		DBO-2	B2-2
		DO-10	B2-2C
SRV (SERVICE)	B-3, B2-2, B2-2A, B2-2B,		IN ORDER OF PRIORITY HIGHEST TO LOWEST

\*Always have a branching destination (i.e., NXT field always modified).

Table 5-2 (Cont)  
Microprogram Branches (BUT)

BUT	Source	Destination	Comment
SRV (SERVICE) (Cont)	B2-2C, B2-2D, CC-1, CS-3 D0-4, DB0-3, J1-2, J2-8, MB-2, SC-1	BT-1	T BIT TRAP
		ERT-IA	STACK OVERFLOW TRAP
		PF-1	POWER FAIL
		BG-1	BR 7 (BUS REQUEST LEVEL)
		BG-1	BR 6
		LC-1	INTERNAL LINE CLOCK
		BG-1	BR 5
		BG-1	BR 4
		URTR	UART RECEIVE
		URTX	UART TRANSMIT
		H-1	CONSOLE STOP
		F-1	NONE OF THE ABOVE
		W-1	WHEN EXECUTING WAIT INST. LOOP ON W-1 INSTEAD OF GOING TO F-1.
		SW (SWITCH)	H-2
CCS-1	CONTINUE		
CE1-1	EXAMINE 1st.		
CE2-1	EXAMINE		
CD1-1	DEPOSIT 1st		
CD2-1	DEPOSIT		
CL-1	LOAD		



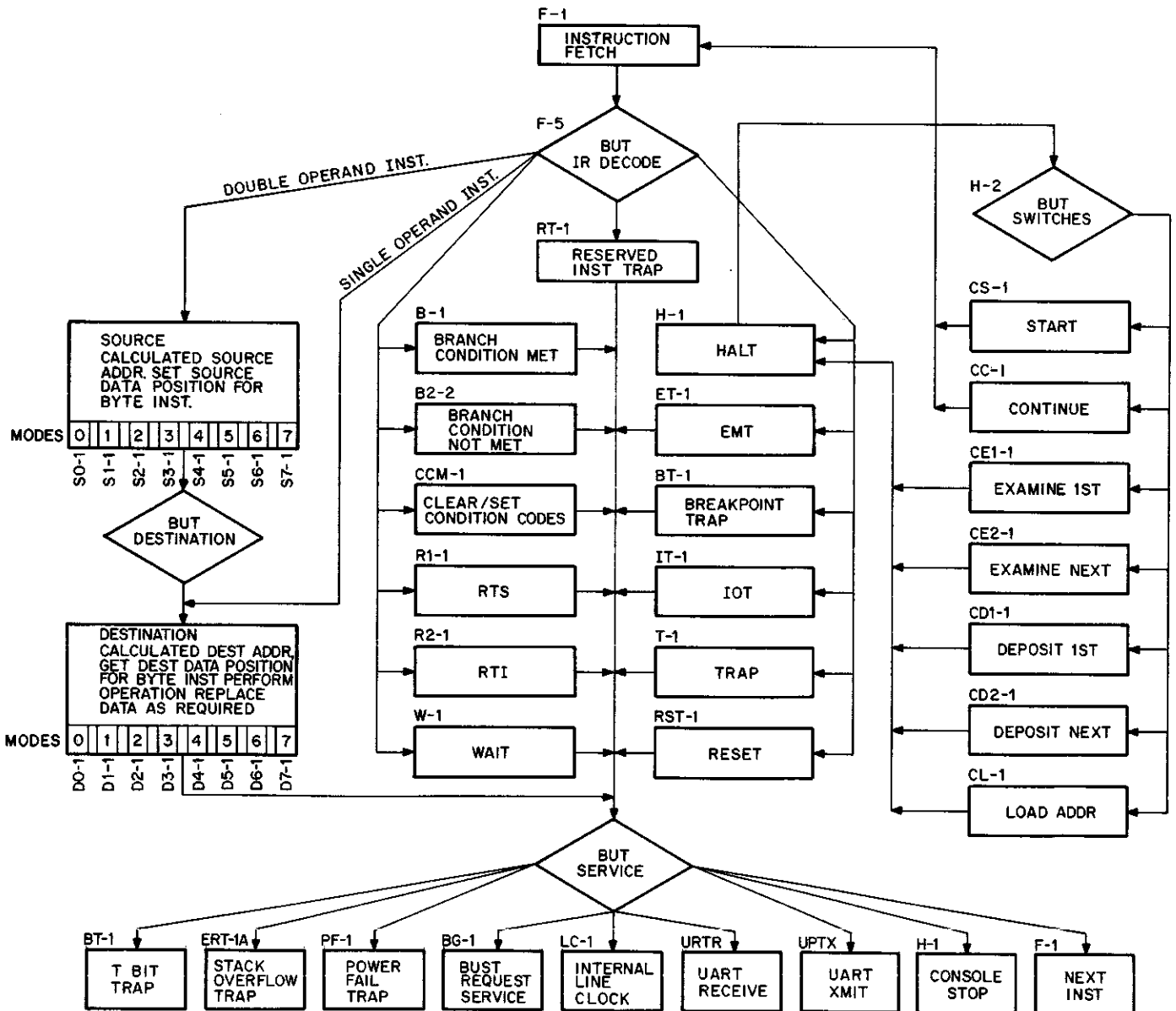
Table 5-2 (Cont)  
Microprogram Branches (BUT)

BUT	Source	Destination	Comment
SW (SWITCH) (Cont)		H-2	NONE
		CE1-1	LOOP UNTIL EXAMINE IS RELEASED.
INT (INTERRUPT)	BG-1	INT-1	INTERRUPT SERVICE
JSRMP	D-1-, D2-3, D3-5, D6-5	J1-1	JMP INSTRUCTION MODE OF OPERATION TO CHANGE PC.
		J2-1	JSR INST.
INITIALIZE	D6-5		
	RST-1		INITIALIZE COMPUTER RESET INSTRUCTION.
UNY (UNARY)	D0-2	ERT-1	JMP or JSR MODE 0 - ILLEGAL INST.
		SB1-1	SWAB
		U1-1	OTHER UNARY
	D1-3	SB2-1	SWAB
		U2-1	OTHER UNARY
	DB0-1	U3-1	UNARY OTHER THAN JMP, JSR, or SWAB
	DE-1	U5-1	UNARY OTHER THAN JMP, JSR, or SWAB
	DO-9	U4-1	UNARY OTHER THAN JMP, JSR, or SWAB
NON (NONE)			NO BRANCH TEST.

### 5.5 MICROPROGRAM FLOW

The microprogram flow chart is shown in full detail in engineering drawing K-MP-KD11-B-1. Figure 5-2 is a simplified flow that provides an overview and aids in using the detailed

flow. No attempt is made in this manual to trace through each path of the microcode. An explanation of the detailed flow notation (Paragraph 5.5.1) is provided along with examples to illustrate instruction interpretation (Paragraph 5.5.2), interrupts and traps (Paragraph 5.5.3), and console functions (Paragraph 5.5.4).



11-1212

Figure 5-2 KD11-B Simplified Flow Diagram

### 5.5.1 Flow Chart Notation

Figure 5-3 illustrates an excerpt from the microprogram flow section of the prints. Notice that the listing is grouped into microroutines (source mode 0 through mode 3 are shown in Figure 5-3); these microroutines start with an identifying comment, the first character of which (disregarding the LOC and NXT columns) is an asterisk. Other comment lines begin with a slash.

All microsteps have mnemonic names such as S0-1, (source mode 0, step 1), S2-2 (source mode 2, step 2), etc. Very often a microroutine will weave back and reuse part of another. For example, the source mode 1 routine weaves back into the source mode 0 routine by the "GOTO S0-2" in S1-2 (Figure 5-3).

To the left of every microstep is the location of that step in the Control Store (in octal) and the contents of the NXT field. Observe the Microprogram Counter (MPC) while single stepping through the microprogram, the LOC and NXT columns provide useful information relating to the path taken by the microprogram.

The flow is well commented and should be self-explanatory. Table 5-3 is a useful glossary of flow notation.

```

LOC  NXT  * SOURCE MODE 0 (REGISTER), GET SOURCE DATA
      / GET TO S0-1 FROM F-5 VIA BUT IR DECODE IR 11:9 = 0
201  007  S0-1 B R[S]; BUT BYTE
      / IF BYTE INST GOTO SBE-1 (MUST BE EVEN BYTE)
007  001  S0-2 R[10] B; BUT DESTINATION
      / IF IR 5:3 = 0 GOTO D0-1
      /           = 1      D1-1
      /           = 2      D2-1
      /           = 3      D3-1
      /           = 4      D4-1
      /           = 5      D5-1
      /           = 6      D6-1
      /           = 7      D7-1

LOC  NXT  * SOURCE MODE 1 (REG. DEFERRED) GET SOURCE DATA
      / GET TO S1-1 FROM F-5 VIA BUT IR DECODE IR 11:9
203  244  S1-1 BA R[S]; DATI; CKOFF; ALBYT
      / GET TO S1-2 FROM S2-3 VIA GOTO
      /           "      S3-5  "
      /           "      S6-5  "
244  007  S1-2 B UNIBUS DATA; BUT BYTE; GOTO S0-2
      / IF ODD BYTE GOTO SBO-1
      / IF EVEN BYTE GOTO SBE-1
      / IF NOT BYTE FALL THROUGH TO S0-2

LOC  NXT  * SOURCE MODE 2 (AUTO-INC.) GET SOURCE DATA
      / GET TO S2-1 FROM F-5 VIA BUT IR DECODE IR 11:9 =2
205  301  S2-1 BA R[S]; DATI; ALBYT
301  014  S2-2 B R[S]+1+BYTE. BAR
      / GET TO S2-3 FROM S4-1 VIA GOTO
214  244  S2-3 R[S] 8; CKOFF; GOTO S1-2

LOC  NXT  * SOURCE MODE 3 (AUTO-INC DEFERRED) GET SOURCE DATA
      / GET TO S3-1 FROM F-5 VIA BUT IR DECODE IR 11:9 =3
207  016  S3-1 BA R[S]; DATI (MUST BE AN EVEN ADDRESS HERE)
216  017  S3-2 B R[S]+2

```

Figure 5-3 Excerpt from Microprogram Flow (K-NL-KD11-B-1)

Table 5-3  
Flow Notation Glossary

Designation	Definition
BA	Bus Address Register.
←	Assignment operator.
;	Separator.
DATI	Initiate DATI operation on Unibus.
+	Plus, the arithmetic operator.
PC	Program Counter = R [7].
CKOFF	Set the Clock Off bit of the Control Store.
B	B-leg Register.
IR	Instruction Register.
B Sex	B-leg Register sign extended (bit 7 repeated in bits 8 through 15).
R [S]	Scratch Pad Register specified by the source portion of the current inst. (IR 8:6 ).
R [D]	Scratch Pad Register specified by the destination portion of the current inst. (IR 2:0 ).
R [n]	Scratch Pad Register n specified by the control ROM.
BUT	Branch on microtest.
ALBYT	Allow byte Unibus reference.
BYTE.BAR	A signal indicating the absence of a byte in instruction.
ENABOVER	Enable the stack overflow detection logic (working BUT).
DATO	Initiate DATO operation on Unibus.
DATIP	Initiate DATIP operation on Unibus.
INIT	Initialize the logic (working BUT).
SVS	Set slave sync (working BUT).
IRC	Clock the Instruction Register (working BUT).

Table 5-3 (Cont)  
Flow Notation Glossary

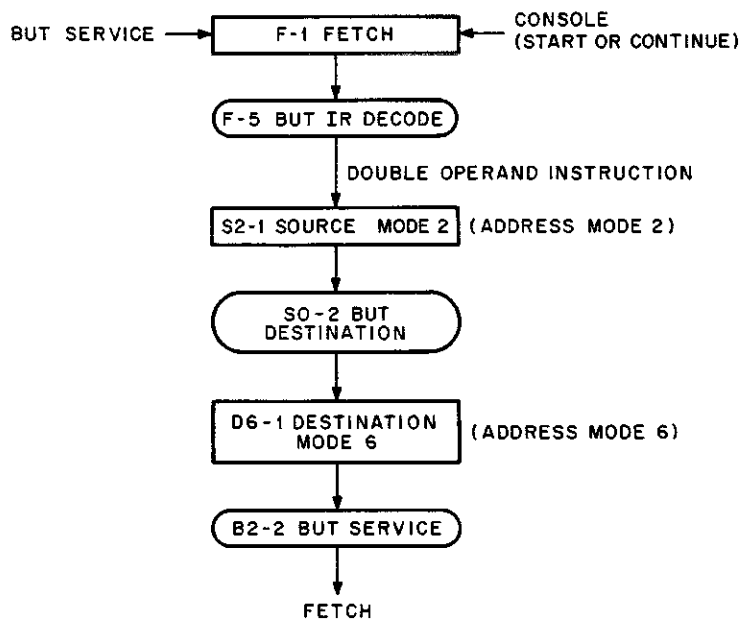
Designation	Definition
K [n]	That location of the constants chip (on the data path A-leg) containing the value n.
R [10] OP B	ALU function determined by the auxillary ALU control logic as a function of the instruction currently in the Instruction Register.
GOTO X	NXT field is to contain the address of X. Unconditional GOTO.

To illustrate the interpretation of PDP-11 instructions, the execution of a CMP instruction is traced through the microcode. The machine is in the RUN state (i.e., the machine is executing instructions), and the instruction is located in memory location 1000.

Location	Assembler Symbolic	Octal
1000	CMP # 15, CHAR	022767
1002		000015
1004		000100
⋮		
1106	CHAR: WORD 0	

This instruction compares the literal 15 to the contents of CHAR and sets the condition code accordingly. Source mode is immediate (mode 2, register 7 = PC) and destination mode is relative (mode 6, register 7 = PC) (refer to Chapter 3). Figure 5-4 shows the simplified flow for the CMP example.

First the instruction is fetched from memory (microsteps F-1 through F-5). This is the same fetch microroutine used to get each instruction from memory and update the PC.



11-1215

Figure 5-4 CMP # 15, CHAR (022767) Simplified Flow Diagram

Location	NXT	Microstep Name	Action	Comment
062	053	F-1	BA ← PC: DATI	/Load the Bus Address Register (BA) with the contents of the Program Counter (PC) (R7) and initiate a DATI by the Unibus Control (BC).
053	365	F-2	B ← PC+2	/Load the B Register with the contents of the PC+2.
365	364	F-3	PC ← B; CKOFF	/Update the PC. CKOFF inhibits execution of the next microstep until the pending Unibus transfer (DATI, initiated in F-1) is complete.
364	061	F-4	B, IR ← UNIBUS DATA	/Load the data from the Unibus (instruction fetched from memory) into the B Register and Instruction Register (IR).

Location	NXT	Microstep Name	Action	Comment
061	001	F-5	B B SEX; BUT IR DECODE	/Sign extend the low order 8 bits of the copy of the instruction in the B Register (used in Branch instruction interpretation) and Branch on microtest (BUT) determined by the IR decode logic. Note that NXT (F-5) = 1 which is the CS location of the RESERVED instruction microroutine. If the IR decode logic does not recognize the instruction, no signals are wire-ORed into the MPC and the RESERVED instruction microroutine (RT-1) is executed by the microprogram. In this example, CMP is recognized (by the IR decode logic), and 204 is wire-ORed with NXT (F-5 = 1) to cause the MPC to be loaded with 205, the location of the microroutine which operates on source mode 2 (S2-1).

Since the instruction is of the double operand group, the next step is to get the source data. Source mode 2 is autoincrement. (Autoincrement implies one level of deferred addressing, Chapter 3.) When used with R7 (the PC), it becomes an immediate mode.

Location	NXT	Microstep Name	Action	Comment
205	301	S2-1	BA R[S]; DATI; ALBYT	/Load the BA with the contents of the register specified by IR 8:6. The register will contain the location of the source data (1002) in this example. Initiate a Unibus DATI to actually get the data. ALBYT will allow an odd Unibus transfer, if the IR contains a byte instruction and the BA contains an odd address. Without the ALBYT,



Location	NXT	Microstep Name	Action	Comment
205 (Cont)				a Unibus transfer that an odd BA results in a bus error (Paragraph 5.3).
301	014	S2-2	B R [ S ] + 1 + BYTE . BAR	/For byte instructions, the auto-increment is by one, for non-byte instructions, autoincrement is by two. BYTE BAR indicates that BLG (S2-2) = +1, and this signal is conditioned by the logic, such that it is true (+1) only when the IR does not contain a byte instruction. So actually, R [ S ] is on the A-leg of the ALU, CARRY IN is enabled, and the +1 constant (enabled only if the IR does not contain a byte instruction) is on the B-leg. The ALU function is A + B.
014	244	S2-3	R [ S ] B;	/Update the register which is to be autoincremented. Inhibit the processor clock until the DATI initiated in S2-1 is complete. From here, the microroutine is woven back into S1-2 (i.e., NXT (S2-3) = S1-2).
244	007	S1-2	B UNIBUS DATA;	/Load the source data which has come in from memory into the B-Register. The microcode at this point joins the microroutine associated with source mode 0 (S0-2). Not a byte instruction, so go to S0-2.
007	001	S0-2	R [ 10 ] B; BUT DESTINATION	/Source data is stored in the scratch pad register, R [ 10 ] while the destination data is retrieved. BUT DESTINATION will cause a microcode branch dependent on IR 3:5 . In this case, the destination mode of 6 will cause 114 to be wire-ORed into the NXT (S0-2) = 1, such that the MPC will be loaded with 115 = LOC (D6-1).

The microroutine starting in D6-1 will get the destination data and perform the operation indicated by the op code of the instruction. Mode 6, when used with the PC, requires that the index contained in the word currently pointed to by the PC be added to the updated PC (address of the index word plus two) to get the location of the source data.

Location	NXT	Microstep Name	Action	Comment
115	075	D6-1	BA ← PC; DATI	/Initiate the Unibus transfer to get the index word from memory.
075	077	D6-2	B ← PC + 2	/Prepare to update PC to next word.
077	057	D6-3	PC ← B; CKOFF	/Update the PC and inhibit the processor clock until the Unibus DATI initiated in D6-1 is complete.
057	300	D6-4	B ← UNIBUS DATA	/Receive the index word into the B-Register.
300	200	D6-5	B, BA ← B+R (D); DATI; BUT JSRMP; ALBYT; CKOFF; GOTO D1-2	/The actual location of the destination data is formed by adding the index (in the B-Register) to the destination register (IR <2:0> ), which is the PC in the example. This address is loaded into the BA, and a DATI is issued to retrieve the data from memory. As in S2-1, ALBYT makes odd byte Unibus transfers legal. BUT JSRMP involves a collection of logic which examines the contents of the IR to see if the instruction is a JMP or JSR. If either of these instructions are present, the appropriate bit is wire-ORed with NXT (D6-5) = D1-2 into the MPC, such that the MPC is loaded with J1-1 or J2-1, respectively for JMP or JSR instruction. In the example, neither of these instructions are present and the MPC is loaded with NXT (D6-5) = D1-2. CKOFF inhibits the processor clock until the DATI initiated in this microstep is complete.

Location	NXT	Microstep Name	Action	Comment
300 (Cont)				Note that this is the first time in this example that memory reference has not been overlapped with micrograms.
200	210	D1-2	$D \leftarrow \text{UNIBUS DATA};$ BUT BYTE	/Receive the destination data from memory. If the instruction had been a byte instruction (e.g., CMPB), the microprogram would be diverted to D0-1 (for odd byte address) to get the byte operand into the right half of the B-Register. This is not the case in this example.
210	143	D1-3	$R [ 11 ] \leftarrow B;$ BUT UNARY	/It is at this point in the micro-routine that a branch occurs for Unary instructions (e.g., SWAB, CLR, COM, etc.). Unary instructions would have caused the BUT IR DECODE done in F-5 to take the appropriate destination microroutine (there is no source field in a Unary instruction). R [ 11 ] is used in Unary instruction interpretation.
163	334	D1-4	$B \leftarrow R [ 10 ] \text{ OP } B;$ BUT NON MOD	/This is the microstep which involves the AUX ALU control (Appendix B) ROM (print DPF) to: 1) cause the ALU to perform the appropriate function, and 2) set or clear condition codes in accordance with the instruction in the IR and the results of the ALU operation. In the case of the example, it is the setting of condition codes which count. Since CMP is an instruction that does not modify memory, (NONMOD), the microprogram is ready to branch to the microstep in which a BUT SERVICE is done. If the instruction requires a memory modification (e.g. MOV, ADD, INC., etc.), D1-5 and D1-6 are executed before going to BUT SERVICE.

Location	NXT	Microstep Name	Action	Comment
335	040	B2-2B	BUT SERVICE	<p>/At the end of each instruction, various situations that attempt to intervene before the next instruction are tested. Their priorities are arbitrated in the F101 ROM shown on the CONE print. These conditions and their relative priorities are as follows:</p> <p>High priority</p> <ol style="list-style-type: none"> <li>1. T-bit trap</li> <li>2. Stack overflow</li> <li>3. Power fail</li> <li>4. Bus request level 7</li> <li>5. Bus request level 6</li> <li>6. Internal line clock</li> <li>7. Bus request level 5</li> <li>8. Bus request level 4</li> <li>9. UART receive</li> <li>10. UART transmit</li> <li>11. Console stop</li> </ol> <p>Low priority</p> <ol style="list-style-type: none"> <li>12. Next instruction</li> </ol> <p>If no condition with a higher priority exists, the microprogram proceeds to F-1 and commences with the fetch of the next instruction.</p>

This completes the example of the microprogram interpretation of CMP #5, CHAR. It may be useful to trace this or some other instruction through the detailed flow (K-WL-KD11-B-1).

### 5.5.2 Interrupts and Traps

Interrupts and traps are also accomplished by the microprogram. Interrupts are sent from Unibus devices. Bus requests (BR) are received by the BC. At the end of each instruction (not microstep), if a BR is present, and if it has the highest priority (see microstep B2-2 in previous example), the microprogram goes to BG-1. In BG-1, a BUT INTERRUPT is done to distinguish BRs that are associated with interrupts from those that are not. If an interrupt is required, the microcode is diverted to INT-1 where the interrupt vector location is loaded

into R (12) from the Unibus data lines. At this point, the microprogram joins the ET-2 microroutine, which stacks the PSW and PC and retrieves a new PSW and PC from the interrupt vector words. At the end of microroutine ET-13, another BUT SERVICE is done to determine if anything (e.g., another higher priority interrupt or the occurrence of stack overflow) is asserted. If none are, the microprogram proceeds to F-1 where it commences to fetch the next instruction.

Power fail trap, stack overflow trap, and T-bit traps are also recognized during BUT SERVICE. Each of these routines has a microroutine associated with it that loads the B-Register with the appropriate trap vector location (from the constants ship, F025 on the DPB print). In each case, the microprogram then joins the ET-2 microroutine, which stacks the PSW and PC and loads the new PSW and PC just as with external interrupts. The main difference is that the vector location comes from the constants chip rather than from the UNIBUS DATA.

Bus error traps are treated in a different manner, because they may prevent an instruction from being completed. When a bus error is detected (Paragraph 4.7.5), the NXT field of the CS (E092 and E103 on print CONG) is disabled, and the microprogram is forced to ERT-1. This microroutine picks up the respective trap vector location from the constants chip and from that point on operates as do all other traps. The difference is the method in which the microprogram gets to ERT-1.

### 5.5.3 Console Functions

When the processor is in the HALT state, the microprogram is looping on microstep H-2 doing BUT SWITCH. As a console switch is depressed (or lifted in the case of DEPOSIT), the microprogram branches to an associated microroutine. Additional logic comes into play here to distinguish the first of a sequence of EXAMINES or DEPOSITS. This is illustrated in the following examples.

Suppose the console operator wants to examine locations 1000 and 1002. The processor is the HALT state, with the microprogram looping of microstep H-2. First the operator must

set the switches to 1000 and depress LOAD ADDRESS. When this is done, the BUT SWITCH causes the microprogram to branch to CL-1.

Location	NXT	Microstep Name	Action	Comment
302	300	H-2	BUT SWITCH	/Loop on H-2 waiting for switch action. When LOAD ADDRESS is depressed branch to CL-1.
311	375	CL-1	BA ← K [207]. BAR *; DATI; CKOFF	/The Switch Register (SR) is logically on the Unibus at location 177570. This constant is obtained from the 8-bit wide constants chip (F25 on DPB print) by taking 207 and forming the complement through the ALU on the way to the BA. A request for the contents of the SR is initiated (DATI) and the processor clock is inhibited until the data is available (CKOFF).
375	367	CL-2	B ← UNIBUS DATA	/Since the SR is physically on the A-leg of the Data Path (DP) (prints DPA, DPB, DPC, and DPD), it cannot be written directly into register 17 of the scratch pad. Rather it is first loaded into the B-Register.
367	302	CL-3	R [ 17] ← B; GOTO H-2	/Load SR into the Load Address Register, R [ 17]. Microprogram goes to H-2.
		H-2	BUT SWITCH; GOTO, H-2	/Loop here looking for switch activity. The microprogram loops on CL-1, CL-2, CL-3, and H-2 as long as LOAD ADDRESS is depressed.

\*(207). BAR = 1's complement of 207.

Now the operator has loaded 1000 from the SR into the Load Address Register R [ 17 ]. The lights are attached to the B-Register and will display the loaded address.

To examine location 1000, the operator depresses EXAMine. So long as the EXAMine switch is depressed, the location to be examined is displayed in the lights. When it is released, the contents of that location are displayed.

Location	NXT	Microstep Name	Action	Comment
317	307	CE1-1	BA, B ← R [ 17 ]; BUT SWITCH	/The lights are connected to the B-Leg. By loading the B-Register with the contents of the Load Address Register, R [ 17 ], the address of the location is displayed. The BA is also loaded for subsequent retrieving of the data itself. BUT SWITCH causes the microprogram to loop on CE1-1 until EXAMine is released.
307	326	CE1-2	DATI; CKOFF	/When the switch is released, the data is requested from the Unibus, and the processor clock is inhibited until it is available.
326	302	CE1-3	B ← UNIBUS DATA; GOTO H-2	/Display the data by loading it into the B-Register and return to the H-2 microprogram loop to await the next switch action.

While the microprogram loops in H-2, the B-Register remains unchanged and the contents of location 1000 are displayed. When EXAMine is depressed a second time, the logic associated with F100 (print CONE) causes BUT SWITCH in H-2 to branch the microcode to CE2-1. In this case, the Load Address Register must be incremented before using its contents.

Location	NXT	Microstep Name	Action	Comment
302	300	H-2	BUT SWITCH	/Loop waiting for switch action.
315	371	CE2-1	$B \leftarrow R [ 17 ] + 2$	/Increment the Load Address Register so that sequential words can be examined.
371	317	CE2-2	$R [ 17 ] \leftarrow B$ ; GOTO CE1-1	/Update R [ 17 ]. The rest of this microroutine merges with CE1-1.
317	307	CE1-1	BA, $B \leftarrow R [ 17 ]$ ; BUT SWITCH	
307	326	CE1-2	DATI; CKOFF	
326	302	CE1-3	$B \leftarrow \text{UNIBUS DATA}$ ; GOTO H-2	

This completes the example of console function microroutines. The remaining console functions are quite similar.

## 5.6 MICROPROGRAM SYMBOLIC LISTING

The microprogram section of the prints (K-MP-KD11-B-1 through 4) contains four useful tools. Paragraph 5.5 describes the microprogram flow. Flow is probably the most useful level to work with the microprogram when tracing through processor action on any specific operation. Flow tells what happens in each microstep and why. To determine how a microstep accomplishes its task, refer to the Microprogram Symbolic Listing (K-MP-KD11-B-2), an excerpt of which is shown in Figure 5-5. In this listing, microsteps are listed alphabetically by their name (e.g., F-1, F-2. . .). Each of the Control Store fields described in Table 5-1 is listed along with its symbolic values. For example, in F-2 of the example in Paragraph 5.5.2, flow indicates:

F-2  $B \leftarrow PC + 2$

The Symbolic Listing is useful for determining how this is to be accomplished in terms of Control Store fields (such as ALU function). Refer to the excerpt in Figure 5-5. Scan the alphabetically ordered list of names for F-2.



A-leg	(ALG)	=	SP (scratch pad)
ALU function	(ALU)	=	A + B
B-leg	(BLG)	=	+ 1 (the constant)
B Register	(BRG)	=	L (load)
Carry In	(CRI)	=	ON
Scratch Pad Address	(SPA)	=	R7 (that is the PC)
Scratch Pad Function	(SPF)	=	REA (read)
Next MPC	(NXT)	=	F-3 (go to F-3 next)

$B \leftarrow PC + 2$  is accomplished by gating Register 7 (the PC) onto the A-leg of the ALU, gating + 1 onto the B-leg, and causing the ALU to perform on A + B operation ( $=R7 + 1$ ) with Carry In enabled ( $=R7 + 1 + \text{carry in}$ ). The B-Register is loaded with the results, and the MPC is loaded with the address of F-3, which is the next microstep.

Only eight of a total of eighteen fields are described in the above example. The rest of the fields have values but they are not of immediate interest. If another step is closely examined, other fields may not be of immediate interest. In all cases, every field value is given.

## 5.7 MICROPROGRAM BINARY LISTING

In addition to the Flow and Symbolic Listing, a Binary Listing of Control Store is included in the microprogram section of the prints (K-MP-KD11-B-3). An excerpt is shown in Figure 5-6. As with the Symbolic Listing, the Binary Listing is alphabetically ordered by microstep name. The fields are located across the top of the listing; however, they relate closely to the actual signals that are controlled (Figure 5-1). A high is represented by a 1 in this listing.

From the previous example, flow indicates F-1  $B \leftarrow PC + 2$ . The Symbolic Listing shows that the ALU function to accomplish this is A + B; the Binary Listing shows the actual logic level value of CONF ALU S3 L, CONF ALU S2 L, CONF ALU S1 L, CONF ALU S0 L, and CONF ALU MODE H (Figure 5-1 and 5-6). Notice that these five signals are grouped

together under the heading ALU. They physically come from chips E104 and E094. The Binary Listing is spaced to show signals grouped both by field (ALU) and also chip (E104 and E094).

If the PC is not being properly incremented during program execution, the flow is a useful tool in determining what is supposed to happen during the fetch microroutine. To see the problem at a deeper level, the Symbolic Listing is used to determine how it is to be accomplished. If the Symbolic Listing does not identify the problem, use the Binary Listing and a scope probe to locate the bad signal and/or malfunctioning chip.

## 5.8 MICROPROGRAM CROSS REFERENCE LISTING

The microstep name (e.g., F-2) is the key that ties the Flow, Symbolic, and Binary Listings together. When working with the processor, it is often useful to determine the name of a microstep from a location or visa versa. This information is provided in the Cross Reference Listings (K-MP-KD11-B-4 in the microprogram section of the prints).

KD11-B MICROPROGRAM SYMBOLIC LISTING

NAME	LOC	ABT	ALG	ALU	AUX	BAR	BLG	BRG	BUT	CON	CKO	CRI	PSW	SAM	SPA	SPF	TNS	NXT
DO-9	132	NO	SP	BL	OFF	H	SEX	L	UNY	NON	OFF	OFF	H	ROM	R11	WRI	NON	A145
ERT-1	010	NO	NUL	AL	OFF	H	BRG	L	CON	4	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
ERT1A	046	NO	NUL	AL	OFF	H	BRG	L	CON	4	OFF	OFF	H	ROM	R0	WRI	NON	ET2-2
ERT1B	153	NO	NUL	AL	OFF	H	BRG	L	CON	4	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
ET-1	011	NO	NUL	AL	OFF	H	BRG	L	CON	30	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
ET-10	254	NO	SP	AL	OFF	L	BRG	H	IRC	NON	ON	OFF	H	ROM	R12	REA	I	ET-11
ET-11	255	NO	SP	AL	OFF	L	BRG	L	NON	NON	OFF	OFF	H	ROM	R7	WRI	NON	ET-12
ET-12	256	NO	SP	A+B	OFF	L	+1	L	NON	NON	ON	ON	H	ROM	R12	REA	I	ET-13
ET-13	257	NO	SP	AL	OFF	H	BRG	L	NON	NON	OFF	OFF	L	ROM	R0	REA	NON	B2-2
ET-2	245	NO	SP	BL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R12	WRI	NON	ET-3
ET-3	246	NO	SP	A-B-1	OFF	L	+1	L	END	NON	OFF	OFF	H	ROM	R6	REA	NON	ET-5
ET-5	247	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R6	WRI	O	ET-6
ET-6	226	NO	PSW	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R0	REA	NON	ET-7
ET-7	251	NO	SP	A-B-1	OFF	L	+1	L	END	NON	OFF	OFF	H	ROM	R6	REA	NON	ET-8
ET-8	252	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R6	WRI	O	ET-9
ET-9	253	NO	SP	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R7	REA	NON	ET-10
ET2-2	003	NO	SP	BL	OFF	H	BRG	L	NON	NON	OFF	OFF	H	ROM	R12	WRI	NON	ET2-3
ET2-3	004	NO	SP	A-B-1	OFF	L	+1	L	NON	NON	OFF	OFF	H	ROM	R6	REA	NON	ET2-5
ET2-5	036	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R6	WRI	O	ET2-6
ET2-6	037	NO	PSW	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R0	REA	NON	ET2-7
ET2-7	051	NO	SP	A-B-1	OFF	L	+1	L	NON	NON	OFF	OFF	H	ROM	R6	REA	NON	ET-8
F-1	062	NO	SP	AL	OFF	L	BRG	H	NON	NON	OFF	OFF	H	ROM	R7	REA	I	F-2
F-2	053	NO	SP	A+B	OFF	H	+1	L	NON	NON	OFF	ON	H	ROM	R7	REA	NON	F-3
F-3	366	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R7	WRI	NON	F-4
F-4	364	NO	NUL	AL	OFF	H	BRG	L	IRC	NON	OFF	OFF	H	BAR	R0	REA	NON	F-5
F-5	061	NO	SP	BL	OFF	H	SEX	L	IRD	NON	OFF	OFF	H	ROM	R0	REA	NON	RT-1
H-1	041	NO	SP	AL	OFF	H	BRG	L	NON	NON	OFF	OFF	H	ROM	R7	REA	NON	H-2
H-2	302	NO	SP	AL	OFF	L	BRG	H	SW	NON	OFF	OFF	H	ROM	R17	REA	NON	D6-5
INT-1	325	NO	SP	AL	OFF	H	BRG	H	SVS	NON	OFF	OFF	H	ROM	R12	WRI	NON	ET-3
IT-1	273	NO	NUL	AL	OFF	H	BRG	L	CON	20	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
J1-1	204	NO	SP	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R0	REA	NON	J1-2
J1-2	260	NO	SP	BL	OFF	H	BRG	H	SRV	NON	OFF	OFF	H	ROM	R7	WRI	NON	BG-1
J2-1	212	NO	SP	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R0	REA	NON	J2-1A
J2-1A	261	NO	SP	BL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R11	WRI	NON	J2-2
J2-2	262	NO	SP	A-B-1	OFF	L	+1	L	NON	NON	OFF	OFF	H	ROM	R6	REA	NON	J2-3
J2-3	214	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R6	WRI	O	J2-4
J2-4	206	NO	SP	AL	OFF	H	BRG	H	ENO	NON	OFF	OFF	H	IRS	R0	REA	NON	J2-5
J2-5	216	NO	SP	AL	OFF	H	BRG	L	NON	NON	OFF	OFF	H	ROM	R7	REA	HON	J2-6
J2-6	263	NO	SP	BL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	IRS	R0	WRI	NON	J2-7
J2-7	264	NO	SP	AL	OFF	H	BRG	L	NON	NON	OFF	OFF	H	ROM	R11	REA	NON	J2-8
J2-8	265	NO	SP	BL	OFF	H	BRG	H	SRV	NON	OFF	OFF	H	ROM	R7	WRI	NON	BG-1
LO-1	042	NO	NUL	AL	OFF	H	BRG	L	CON	100	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
MB-0	154	NO	SP	AL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R0	REA	NON	MB-1
MB-1	242	NO	SP	ABAR	ON	H	BRG	L	NON	NON	OFF	ON	H	ROM	R10	REA	NON	MB-2

Figure 5-5 Excerpt of (K-WL-KD11-B-2)  
Microprogram Symbolic Listing

## KD11-B MICROPROGRAM BINARY LISTING

N A M	L O C	N X T	A L U	C F A R R U I E X	P S S D S P P I W 1 3 P	S S S B M P M B 0 0 1 T	B B S S A T P P R P F 2	C A T K B N O T S	A B L R G G	B U T	
DO-1B	143	1100	1010	0000	1001	1001	1011	1110	0001	11 01	1111
DO-2	123	1010	1011	0000	1001	1001	1011	1110	1111	11 10	1111
DO-3	124	1010	1010	0000	1001	1001	1011	1110	11 11	11 00	1111
DO-4	125	1010	1001	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-5	126	1010	1000	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-6	127	1010	0111	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-7	132	1010	0110	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-8	131	1010	0101	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-9	132	1001	1010	0101	1001	1011	1111	1000	11 11	11 11	1010
ERT-1	012	0101	1010	0000	1001	1101	1011	1101	11 11	10 11	1101
ERT1A	046	1111	1100	0000	1001	1101	1011	1101	11 11	10 11	1101
ERT1B	153	0101	1010	0000	1001	1101	1011	1101	11 11	10 11	1101
ET-1	011	0101	1010	0000	1001	1101	1011	1110	11 11	10 11	1101
ET-10	254	0101	0010	0000	1001	1111	1011	0110	01 10	11 00	0000
ET-11	255	0101	0001	0000	1001	1101	1111	0101	11 11	11 11	1111
ET-12	256	0101	0000	0110	0101	1111	1010	0110	01 10	11 11	1111
ET-13	257	0011	1010	0000	1001	0001	1011	1110	11 11	11 11	1111
ET-2	245	0101	1001	0101	1001	1111	1011	1100	11 11	11 00	1111
ET-3	247	0110	1001	0101	1001	1101	1011	1101	01 01	11 00	1111
ET-5	247	0110	1001	0101	1001	1101	1011	1101	01 01	11 00	1111
ET-6	226	0101	0110	0000	1001	1001	1011	1110	11 11	00 00	1111
ET-7	251	0101	0101	1001	0001	1101	1010	0111	11 11	11 11	0100
ET-8	252	0101	0100	0101	1001	1101	1011	1101	01 01	11 00	1111
ET-9	253	0101	0011	0000	1001	1101	1111	1111	11 11	11 00	1111
ET2-2	003	1111	1011	0101	1001	1111	1011	1100	11 11	11 11	1111
ET2-3	004	1110	0001	1001	0001	1101	1010	0111	11 11	11 11	1111
ET2-5	036	1110	0000	0101	1001	1101	1011	1101	01 01	11 00	1111
ET2-6	037	1101	0110	0000	1001	1001	1011	1110	11 11	00 00	1111
ET2-7	051	0101	0131	1001	0001	1101	1010	0111	11 11	11 11	1111
F-1	062	1101	0100	0000	1001	1101	1111	0111	11 10	11 00	1111
F-2	053	0000	1010	0110	0101	1101	1110	1111	11 11	11 11	1111
F-3	365	0000	1011	0101	1001	1101	1111	1101	01 11	11 001	1111
F-4	364	1100	1110	0000	1001	1001	0001	1110	11 11	10 11	0000
F-5	061	1111	1110	0101	1001	1001	1011	1010	11 11	11 11	0111
H-1	041	0011	1101	0000	1001	1101	1111	0111	11 11	11 11	1111
H-2	302	0011	1111	0000	1001	1111	1111	0111	11 11	11 00	0110
INT-1	325	0101	1001	0000	1001	1111	1011	1100	11 11	11 00	1000
IT-1	273	0101	1010	0000	1001	1001	1011	1111	11 11	10 11	1101
J1-1	204	0100	1111	0000	1001	1001	1011	1110	11 11	11 00	1111
J1-2	260	1101	1111	0101	1001	1101	1111	1101	11 11	11 00	1100

Figure 5-6 Excerpt of Microprogram Binary Listing (K-W-KD11-B-3)

## CHAPTER 6 MAINTENANCE

### 6.1 INTRODUCTION

This chapter describes techniques for isolating and repairing failures in the KD11-B and the console. The basic procedures are aimed at differentiating between failures in the processor and the remainder of the computer. If it is established that the processor is at fault, then it is necessary to determine which of the two KD-11B modules is defective. Finally, the KM11 maintenance panel can be used in conjunction with the KD11-B documentation to isolate failure to specific integrated circuits.

The easiest method of isolating failures and determining if a system will function under worst-case conditions is to use diagnostic programs that have been designed by DEC to test the processor and memory. For most DEC computers it is possible to assemble a hierarchy of diagnostics that progressively tests more and more of the computer. In fact, with large systems it is possible to test the KD11-B beyond its performance specifications. Diagnostic programs are written and commented in a manner that guides the use in determining computer malfunctions.

This chapter also describes special techniques for troubleshooting the KD11-B. The exact determination of failures and their repair requires careful application of the tools described in this chapter, in addition to a general knowledge of PDP-11 systems. A section on console maintenance is also included and provides a console troubleshooting procedure.

### 6.2 DIAGNOSTICS

The diagnostic programs are useful preventive maintenance tools. The diagnostic programs supplied by DEC provide a rigorous test of the computer, and they can indicate the need for service even before a failure occurs. Preventive maintenance is especially important on machines that include mechanical components, such as line printers or tape drives.

### 6.3 TYPES OF FAILURES

Failures can be broken down into three classes: basic, complex, and peripheral. A basic failure of the processor, memory, or program read-in device does not permit diagnostic software to be loaded; thus, fault isolation and repair in a computer with a basic failure requires an elementary approach. A complex failure typically occurs only with programs that generate interaction on the Unibus between several peripherals and the processor. DEC provides a number of systems diagnostics, such as the General Test Program (GTP) and the Communications Test Program (CTP), that are useful in isolating complex failures.

Often the failure is caused by a peripheral problem that is unrelated to the processor or memory. In this case, the processor itself may be used as a troubleshooting tool. For instance, a diagnostic program is available that tests the alignment of TU10 magnetic tape drive and reports significant parameters via the serial communications line (SCL).

### 6.4 SUGGESTED EQUIPMENT

Table 6-1 provides a list of test equipment, maintenance devices, and tools used to perform the processor maintenance procedures and adjustments.

Table 6-1  
Test Equipment and Tools

Equipment		Description	
Test Equipment	}	Oscilloscope	Tektronix Model 453 (or equivalent)
		Volt-ohmmeter	Triplett Model 630 (or equivalent)
Devices	}	Extender Board	Three W984A Double Extender Boards
		Maintenance Module Set	One W130 (two are desirable)
		Maintenance Module Overlays	One W131 (two are desirable)
		IC Test Clip	KM1-DEC Part No. 55-09081-9 KM2-DEC Part No. 55-09081-10
Tools		Small Flatblade Screwdriver	

## 6.5 PROCEDURES

It is most useful to know the precise condition of the computer at the time of the failure. The user is advised to record the state of the computer, in as much detail as needed, to reproduce the problem whenever a failure occurs. At least the following information should be noted:

1. Any peripherals attached to the Unibus not usually present.
2. The name of the program running when the failure occurred.
3. The state of the processor indicators (console) when the failure occurred.
4. If possible the sequence of events preceding the failure should be noted.

When running a program on the KD11-B for the first time, it should be noted that certain subtle differences exist among the several PDP-11 processors that can cause problems, when non-standard programming practices are used. A list of differences between the KD11-B and other PDP-11 processors is contained in Chapter 2.

Once it is established that a hardware failure exists, the following checks are advised before dismantling the computer:

1. Verify that the power supply is attached to a live ac source and is functioning normally.
2. Verify that the Unibus is properly routed.
3. Be certain that grant continuity cards are properly placed whenever missing peripherals would break the BUS GRANT lines.
4. Be certain that no Unibus address conflicts exist.

Programs can be executed from the Scratch Pad Memory (SP) (register) locations, and if processor problems are suspected, this procedure should be tried to isolate the problem. Communications between the console and processor must be functioning properly in order to use this procedure, and should be the first thing that is checked when a processor problem is suspected. Executing programs from the SP is advantageous for troubleshooting or checking the processor. When executing a program from the SP, the PC (R7) is incremented by one; however, BR instructions always modify the PC by multiples of two. Consequently, a BR instruction must be carefully used in a program to prevent the PC from being modified to an incorrect address. An example of a simple program that loops on two SP register locations is as follows:

Address	Instruction	Octal
1177700 (R0) 177701 (R1)	NOP BR.-1*	000240 000777

To load the above program from the console, perform the following steps:

1. Enter 177700 in the Switch Register and depress LOAD ADDRESS (this is the address of register 0).
2. Enter 000240 in the Switch Register and lift DEPOSIT. (This places a NOP instruction in R0.)
3. Enter 000777 in the Switch Register and lift DEPOSIT.
4. Enter 177700 in the Switch Register and depress LOAD ADDRESS (this specifies the starting address).
5. Lift ENABLE/HALT to the ENABLE position.
6. Depress START. The RUN light should come on. The program is now being executed.
7. If ENABLE/HALT is pressed, the ADDRESS/DATA display should contain either 177700 or 177701.

When executing programs from the SP (registers), do not use the registers that the processor uses (R6, R7, R10, R11, R12, and R17).

## 6.6 ADJUSTMENTS

Adjustments to the processor are listed as follows:

1. The processor clock should have a 310 ns period. Adjust, if necessary, performing the following procedures:
  - a. Extend the M7261 module.
  - b. Observe with an oscilloscope the processor clock at E045 pin 6.
  - c. Adjust the potentiometer on the M7261 until the processor clock period is 310 ns.
  - d. Remove oscilloscope probe and insert M7261 module back in the back panel.

\*  $777_8$  is normally a BR self-instruction. However, when executed from the SP, it is a BR.-1, because the SP registers are located on BYTE ADDRESSES.



2. The SCL clock frequency should be 16 times the desired baud rate. Adjust, if necessary, using the following procedure:
  - a. Extend the M7260 module.
  - b. Observe with an oscilloscope the SCL clock at E084 pin 6.
  - c. Adjust potentiometer R74 for 16 times the desired baud rate, according to Table 6-2.
  - d. Remove oscilloscope probe and insert M7260 module back into back panel.

Table 6-2  
Baud Rate Adjustment

Baud Rate	Period ( $\mu$ s)	Frequently (Hz)
110	568	1760
150	416	2400
300	208	4800

An alternate method of adjusting the SCL clock that does not require extending the module is to run any program, such as T-17, that causes a continuous stream of characters to be printed on the console. The potentiometer on the M7260 should then be adjusted to the center of the range for which satisfactory characters are printed.

## 6.7 KD11-B PRINT FUNCTION TABLE

The principles of operation of the KD11-B logic are described in Chapter 4. The microprogram is described in Chapter 5. The KD11-B print set is described in Chapter 1. Table 6-3 lists each engineering drawing for the KD11-B processor and describes the functions of the items shown on that drawing.

## 6.8 EXTERNAL CLOCK INPUTS

External clock inputs and corresponding internal clock disables are provided for the serial communications line (SCL) clock and the processor clock. The external input for the SCL clock permits the reception and transmission of serial asynchronous data at rates up to 10,000 baud. High baud rate signals should be input on pin FM2 of the M7261, rather than the low frequency input on pin FN1. The SCL clock, its external disable, and external clock input are shown on print DPH.

The external clock input for the processor clock permits the synchronization of two processors or the use of a manual clock. The manual clock input and the internal processor clock disable are shown on print CONJ.

Table 6-3  
Engineering Drawing Print List and Functions

Print Designation	Print Title	Function of Logic on Print
D-CS-M7260-0-01		
DPA	Data Path <3:0>	This print contains the least significant four bits of the DP components, including the ALU, the Scratch Pad, the B-Register, the A MUX, Unibus data drivers and receivers, and additional A-leg gating for the PSW, and console switches. Prints DPB, DPC, and DPD contain the three other 4-bit chunks of the Data Path.
DPB	Data Path <7:4>	In addition to the items mentioned above, DPB contains the constants generator described in Chapter 4.
DPC	Data Path <11:8>	Same as DPA.
DPD	Data Path <15:12>	Same as DPA.
DPE	PSW	DPE contains the 8-bit Program Status Word and the multiplexers required to load the PSW. Rotate multiplexers are also shown on DPE. The console (MUX shown in the lower right-hand corner of DPE) converts the data presented on the B-leg into a serial bit stream for the console display.
DPF	AUX ALU CONTROL	In addition to the auxiliary ALU control, the Instruction Register (IR) and the C and V-bit encoder are shown on DPF.
DPG	IR DECODE	The major elements of the IR decoder are shown on DPG.
DPH and DPH1	SCL CONTROL	The UART and other elements of the SCL control are shown on DPH and DPH1.

Table 6-3 (cont)  
Engineering Drawing Print List and Functions

Print Designation	Print Title	Function of Logic on Print
CONA	INT ADDR	The Bus Address Register (BR) is shown on the left side of CONA. On the right half of the print, the logic required to detect reference to internal registers is diagrammed.
CONB	STACK FLOW AND SPAM	The left half of CONB contains the Scratch Pad Address Multiplexer (SPAM) while the right side contains the Stack Overflow and Run flip-flops.
CONC	UNIBUS CONTROL (BC)	Data Requests flip-flops are shown towards the left edge of CONC. The lower right-hand quarter of the print contains Bus Error and CKOFF flip-flops. The 9602 that detects nonexistent memory is shown in the lower left-hand corner of CONC.
D-CS-M7261-0-01 CONC1	PRIORITY ARBITRATION	The priority arbitration logic for bus requests is shown along the bottom edge of CONC1. Towards the left and top of CONC1 there are three 4-bit latches used to hold signals received from the Unibus. The 9602 shown on the upper right of CONC1 is used to clear the bus if SACK is not received with 22 $\mu$ s after NPG or BG.
COND	DRIVE AND RECEIVERS	COND contains all of the Unibus drivers and receivers except those used for the data lines and two drivers used for the Line Clock circuit.
CONE	MICRO BRANCH LOGIC	A 4-to-16 line decoder associated with the BUT field of the microprogram is located in the upper left of CONE. The function, switch buffers, and decoders are shown in the upper middle and upper right. Two flip-flops associated with the console example and DEPOSIT keys are shown in the lower left and lower center in the trap arbitration logic.

Table 6-3 (cont)  
Engineering Drawing Print List and Functions

Print Designation	Print Title	Function of Logic on Print
CONF	MPC	Sixteen bits of the Control Store are shown located on the left side of CONF. The MPC is located along the right edge of the print.
CONG	CONTROL STORE (CS)	The remaining 23 bits of the CS are shown on CONG.
CONH	POWER FAIL	Initialize and Power Fail circuitry is shown on CONH. The 9602 contained in the lower left-hand corner of CONH is used to generate bus instruction during the RESET instruction.
CONI	LINE CLOCK	The circuit equivalent to the KW11-L is contained on CONI.
CONJ	PROCESSOR CLOCK	The circuit consisting of E19, R2, R10, and C115 comprises the oscillator that generates the processor clock. The input to E02713 is used by the KM11 to generate clock signals from an external source.

### 6.9 KM11 MAINTENANCE PANEL

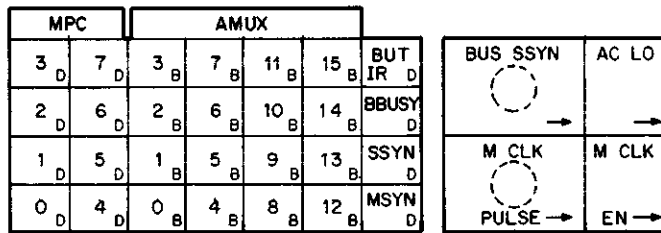
The discussion to this point has not considered the backplane or configuration. Every KD11-B contains the necessary logic to permit single step operation; however, the use of these facilities depends on the specific configuration (see Chapter 1 or Part 1). Two module slots are provided in the computer for the maintenance panel. Figure 6-1 contains a diagram of the KM11 overlays for slots KM-1 and KM-2 in the computer backplane (Figures 1-2 and 1-3). Table 6-4 provides description of the overlay designations. Note the following:

- a. The KM-1 switches have the same function in slots KM-1 and KM-2.
- b. When the manual clock is enabled, bus error timeouts are disabled. Non-existent memory trap cannot occur in manual mode.
- c. Each actuation of the manual clock with line EV1 of the M7261 grounded produces Bus Control (BC) clock. It normally requires two BC clock pulses to advance the Microprogram Counter (MPC) to the next address.

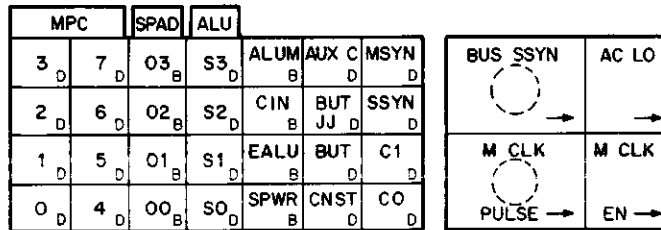
- d. The MPC is duplicated on both KM11 slots. This permits the user who has only one KM11 to plug the unit into either KM-1 or KM-2.
- e. The MPC displayed on the KM11 is the address of the next microstep to be selected and not the present one.
- f. Some lights on the maintenance panel indicate the assertion of a signal when illuminated and others indicate unassertion when illuminated. This fact is indicated on the KM11 overlay drawing by the letter B for bright or D for dim appearing under each indicator light.

B = bright for assertion or logic 1  
D = dim for assertion of logic 1

- g. The wiring for KM-1 appears in slot A2, and the wiring for KM-2 appears in slot B2 for a Configuration 2 backplane. KM-1 and KM-2 are wired to slots A1 and B1, respectively, for a Configuration 1 backplane.



(a) KM-1 OVERLAY



(b) KM-2 OVERLAY

NOTE:  
D = Dim when asserted.  
B = Bright when asserted.

11-1271

Figure 6-1 KM11 Maintenance Module, KD11-B Overlays

Table 6-4  
KM-1 and KM-2 Overlay Designations

Display	Definition
	<b>KM-1 OVERLAY</b>
MPC <7:0>	The address of the next microinstruction to be executed.
AMUX <15:0>	The 16-bit output of the A MUX.
BUT IR	BUT IR DECODE signal. When asserted, the microprogram is at F-5 and does a branch on the contents of the IR.
BBUSY	BUS BUSY. When asserted, BBSY indicates that a device has control of the Unibus.
SSYN	BUS SLAVE SYNC. When asserted, SSYN indicates that the Unibus slave device has responded to the master.
MSYN	BUS MASTER SYNC. When asserted, MSYN indicates that the master device on the Unibus is informing the selected slave that address and control information are present.
BUS SSYN	When actuated in the direction of the arrow (ON), SWITCH BUS SSYN asserts BUS SLAVE SYN as long as the switch is ON.
AC LO	When actuated in the direction of the arrow (ON), AC LO asserts BUS AC LO as long as the switch is ON.
M CLK PULSE (MANUAL CLOCK PULSE)	Each actuation in the direction of the arrow (ON), processor generates one bus control clock, provided that CLK EN switch has been actuated. Two actuations will generate a processor clock.
M CLK EN	When actuated in the direction of the arrow (ON), it disables the processor clock logic and allows the M CLK PULSE switch to generate processor clocks.
	<b>KM-2 OVERLAY</b>
MPC 7 through 0	The address of the next microinstruction to be executed.
SPAD (SCRATCH PAD ADDRESS)	The address of the register (location) in the scratch pad memory.
ALU S3 through S0	These five signals together indicate the function that the ALU is performing.
ALU M	

Table 6-4 (cont)  
KM-1 and KM-2 Overlay Designations

Display	Definition															
CIN	Carry in signal to bit 0 of the ALU.															
E ALU	Enable ALU is the signal that switches the A MUX from inputting the Unibus data lines to inputting at the output of the ALU.															
SP WR	Scratch Pad Write indicates that the SPM is doing a write function as opposed to a read.															
AUX CNTRL	Auxiliary Control enables the AUX ALU ROMs on print DPF.															
BUT J J	Signifies that a branch test for a JMP or JSR instruction is occurring.															
BUT UN	Signifies that a branch test for a Unary instruction is occurring.															
CNST	Signifies that the constants ROM, F025 on M7260, is enabled.															
MSYN	Same as MSYN on KM-1.															
SSYN	Same as SSYN on KM-1.															
C1 and C0	BUS C1 and BUS C0 together signify the type of Unibus cycle that is occurring: <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>C1</th> <th>C0</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>DATI</td> </tr> <tr> <td>0</td> <td>0</td> <td>DATIP</td> </tr> <tr> <td>1</td> <td>0</td> <td>DATO</td> </tr> <tr> <td>1</td> <td>1</td> <td>DATOB</td> </tr> </tbody> </table>	C1	C0		0	0	DATI	0	0	DATIP	1	0	DATO	1	1	DATOB
C1	C0															
0	0	DATI														
0	0	DATIP														
1	0	DATO														
1	1	DATOB														

KM-1 is the more useful configuration and should be used to begin any repair attempts requiring the use of the maintenance panel. The console indicators display the B-leg input to the ALU, and the KM-1 configuration maintenance panel displays the output of the A MUX. If the ALU and A MUX are functioning, it is possible to deduce the contents of the A-leg by observing the console and the Maintenance Panel.

#### 6.10 USING KM11 MAINTENANCE PANEL

Assume that the maintenance panel is plugged into slot A2 for the KM-1 overlay configuration. The M CLK EN switch must be activated in the direction of the arrow, which disables the processor M CLK PULSE. The following example uses the sequence of micro-

steps described in Paragraph 5.4.3.

With the HALT switch depressed, hold down the START switch. Toggle the M CLK PULSE switch advance two times, then release the START switch and toggle two more times. The processor should now be in microstep CS-2. The MPC should read  $321_8$ , which is the contents of the NXT field of LOC  $100_8$  of the CS. Repeated actuation of the M CLK PULSE switch should cause the microprogram to proceed as follows:

LOC	NXT (MICRO PC)	STEP NAME
100	322	CS-1
322	321	CS-2
321	$40 + 1^*$	CS-3
41	302	H-1
302	$300 + 2$	H-2

## 6.11 CONSOLE MAINTENANCE

If any malfunctions are suspected in the console display logic, the console can be put into Service mode. This mode of operation induces known data into the serial data line from the computer in order to verify that the counters, clock, and shift registers of the console logic on the console board are functioning properly. If the data on the console display does not match the known data, then the closed loop can be probed with an oscilloscope to determine the faulty area.

Refer to Figure 2-1 in the Console Description (Chapter 3); the procedure takes the four Scan Address lines and feeds them one at a time into the serial data output line. The Address/Data Multiplexer is bypassed. Since the clock is free-running, each scan address line displays a known data pattern in the console lights. The troubleshooting procedure for the console is as follows:

1. Make certain the computer power is off.
2. Disconnect the console cable connector from the M7260 module and then turn on the computer power.
3. After step 2 is completed, the data pattern  $177777_8$  should be displayed on the console lights.

\* In step CS-3 the NXT field contains 40. However, if the HALT switch is depressed, a 1 is ORed into the NXT field to cause a branch to H-1.



4. At the Berg cable connector that plugs into the M7260 module, use a piece of small gauge wire and jumper pin F (signal DAK, serial output line) to pin B2 (ground). All the console lights should be off. Remove the jumper before proceeding to the next step.
5. At the cable connector, jumper pin F (signal DAK) to pin N (SCAN ADDRESS 01). The pattern displayed on the lights, should be  $052525_8$ . Remove the jumper before proceeding to the next step.
6. At the cable connector, jumper pin F to pin L (SCAN ADDRESS 02). The pattern displayed on the lights should be  $031463_8$ . Remove the jumper before proceeding to the next step.
7. At the cable connector, jumper pin F to pin J (SCAN ADDRESS 04). The pattern displayed on the lights should be  $007417_8$ . Remove jumper before proceeding to the next step.
8. At the cable connector, jumper pin F to pin D (SCAN ADDRESS 08). The pattern displayed on the lights should be  $000377_8$ . Remove jumper after completing the step.

**PART 3**  
MM11-K, MM11-L Memories

## CHAPTER 1 GENERAL DESCRIPTION

### 1.1 INTRODUCTION

This chapter provides the user with theory of operation and logic diagrams necessary to understand and maintain the MM11-K and L Read/Write Core Memories. The level of discussion assumes that the reader is familiar with basic digital computer theory. Both general and detailed descriptions of the core memories are included.

Although memory control signals and data pass through the Unibus, it is beyond the scope of this discussion to describe the operation of the Unibus. A detailed description of the Unibus is presented in the PDP-11 Unibus Interface Manual, DEC-11-HIAB-D.

A complete set of engineering logic drawings is shipped with each core memory. These drawings are bound in a separate volume entitled MM11-K and L Core Memories, Engineering Drawings. The drawings reflect the latest print revisions and correspond to the specific memory shipped to the user.

This section of the manual is divided into three chapters.

1. General Description
2. Detailed Description
3. Maintenance

### 1.2 GENERAL DESCRIPTION

This paragraph provides a physical description and specifications for the memory. The

major functional units of each memory are briefly described, and the basic memory operations are discussed.

### 1.2.1 Physical Description

The MM11-K provides 4096 (4K) 16-bit words; the MM11-L provides 8192 (8K) 16-bit words. Both configurations require three standard 8-1/2 inch wide modules: two are hex-height and one is quad-height.

The quad-height module contains the memory stack: module H213 for 4K; and module H214 for 8K. One hex-height module (G110) contains the control logic, inhibit drivers, sense amplifiers, and 16-bit data register; the other hex-height module (G231) contains the address selection logic, current generator, and switches and drivers. Pin to pin compatibility exists between the C, D, E and F connectors of both these modules are also compatible with the standard Unibus pin assignments.

It is recommended that the G231 Driver Module be installed between the G110 Control Module and the H213 or H214 Stack Module. Photographs of the component side of the modules are shown in Figures 1-1, 1-2, and 1-3.

### 1.2.2 Specifications

The general memory specifications are listed in Table 1-1.

Table 1-1  
MM11-K and L Memory Specifications

Type	Magnetic core, read/write, coincident current, random access.
Organization	Planar, 3D, 3-wire
Capacity	4096 (4K) words for MM11-K 8192 (8K) words for MM11-L

Table 1-1 (Cont)  
MM11-K and L Memory Specifications

Access Time and Cycle Time

Bus Mode	Cycle Time	Access Time
DATI	900 ns	400 ns
DATIP	450 ns	400 ns
DATO DATOB (PAUSE L)	900 ns	200 ns
DATO-DATOB (PAUSE H)	450 ns	200 ns

X-Y Current Margins

$\pm 6\%$  @  $0^\circ\text{C}$ ,  $\pm 7\%$  @  $25^\circ\text{C}$ ,  $\pm 6\%$  @  $50^\circ\text{C}$

Strobe Pulse Margins

$\pm 30\text{ ns}$  @  $0^\circ\text{C}$ ,  $\pm 40\text{ ns}$  @  $25^\circ\text{C}$ ,  $\pm 30\text{ ns}$  @  $50^\circ\text{C}$

Voltage Requirements

$+5\text{V} \pm 5\%$  with less than 0.05V ripple

$-15\text{V} \pm 5\%$  with less than 0.05V ripple

Average Current Requirements

Stand by

$+5\text{V}$ : 1.7A

$-15\text{V}$ : 0.5A

Memory Active

$+5\text{V}$ : 3.4A

$-15\text{V}$ : 6.0A

Power Dissipation (Worst Case)

Control Module (G110):  $\cong 60\text{W}$

Drive Module (G231):  $\cong 40\text{W}$

Stack Module (H213 or H214):  $\cong 20\text{W}$

Total at maximum repetition rate: 120W

Environment

Ambient temperature:  $0^\circ\text{C}$  to  $50^\circ\text{C}$  ( $32^\circ\text{F}$  to  $122^\circ\text{F}$ )

Relative Humidity: 0-90% (non-condensing)

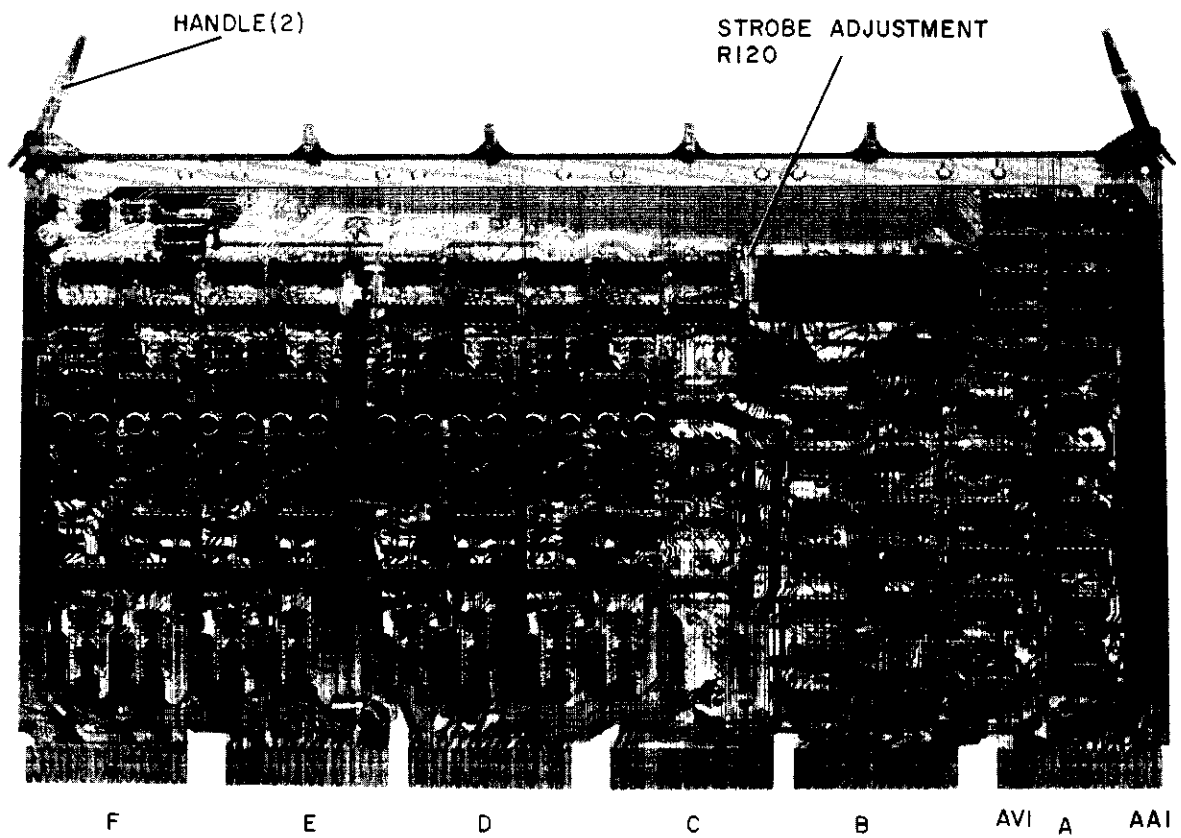


Figure 1-1 Component Side of G110 Control Module

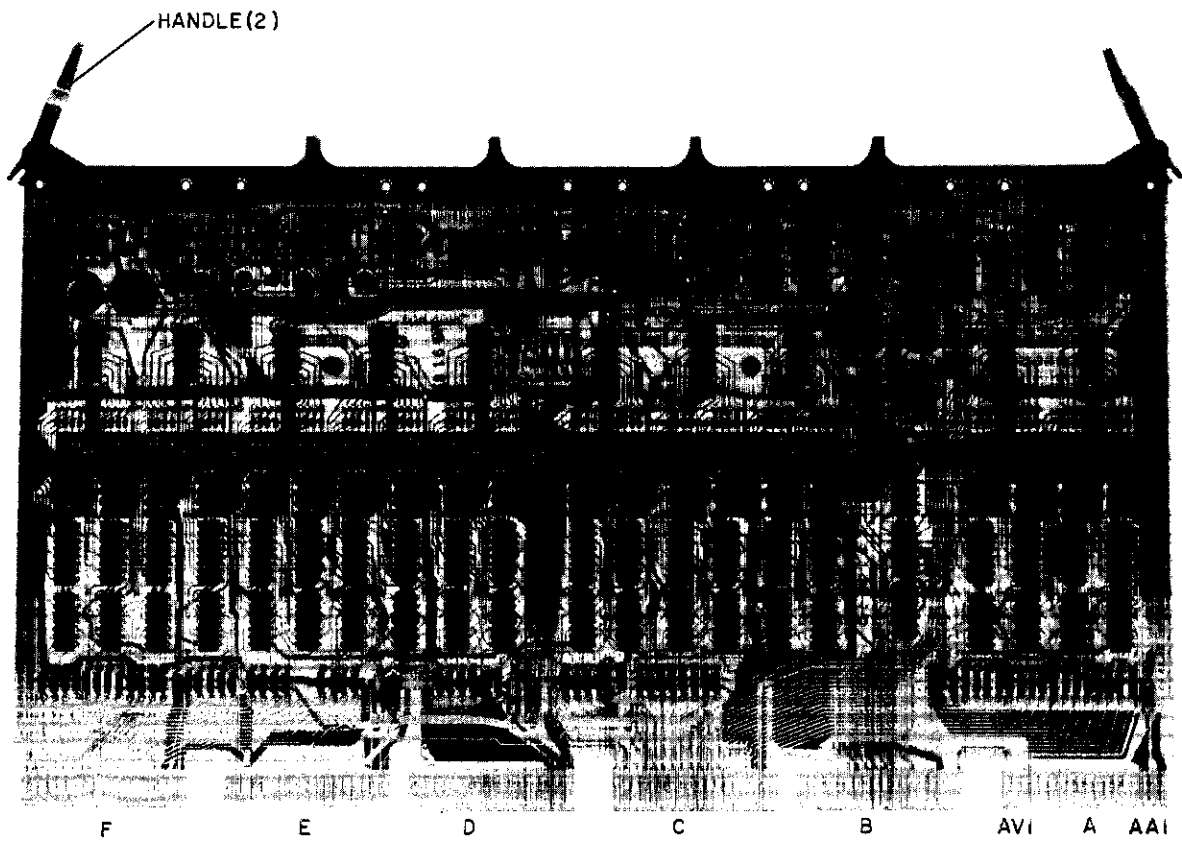


Figure 1-2 Component Side of G231 Drive Module

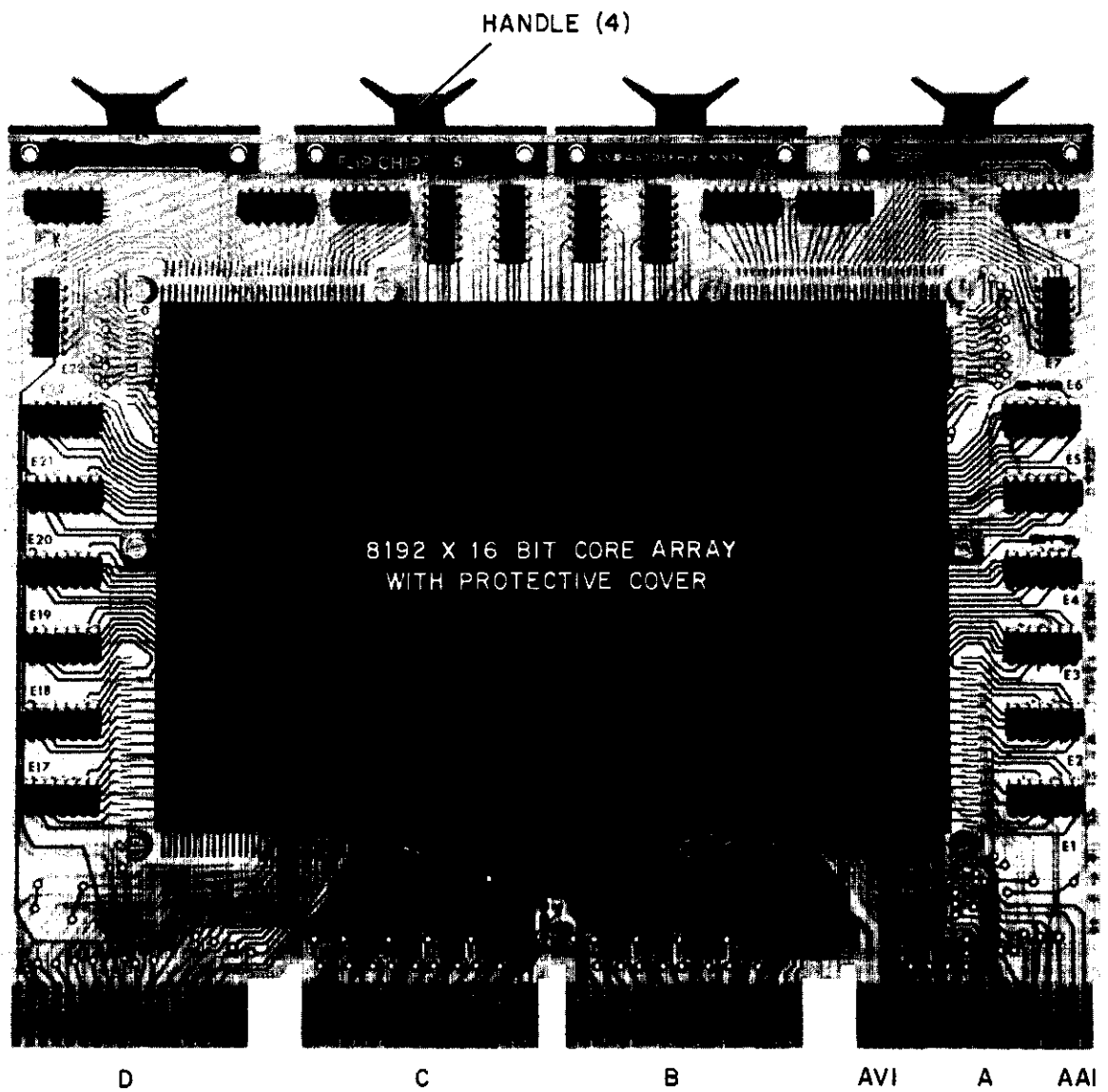


Figure 1-3 Component Side of 8K H214 Stack Module



### 1.2.3 Functional Description

The memory is a read/write, random access, coincident current, magnetic core type with a cycle time of 900 ns and an access time of 400 ns. It is organized in a 3D, 3-wire planar configuration. Word length is 16 bits, and the memory is offered in two word capacities: MM11-K contains 4096 (4K) words; and MM11-L contains 8192 (8K) words. The major functional units of the memory (Figure 1-4) are briefly described in the following paragraphs.

1.2.3.1 G110 Control Module - The G110 Control Module contains the memory control circuits, inhibit drivers, sense amplifiers, data register, threshold circuit, -5V supply, and device selector.

#### 1. Memory Control Circuits

Control circuits are provided to acknowledge the request of the master device; determine which of the four basic operations (DATI, DATIP, DATO or DATOB) is to be performed; and set up the appropriate timing and control logic to perform the desired read or write operation. If a byte operation has been selected, address line A00 L determines the byte to be selected. The actual read or write operation is selected by control lines (C00 and C01). The memory control logic also transfers data to and from the Unibus.

#### 2. Inhibit Driver

Each bit mat contains a single inhibit/sense line that passes through all cores on the mat. To write a 0 into a selected bit, an inhibit current is passed through the inhibit/sense line that cancels the write current in the Y-line. The core does not switch so it remains in the 0 state. With no inhibit current, the currents in the X- and Y-lines switch the core to the 1 state.

#### 3. Sense Amplifiers

During a read operation, the sense amplifier picks up a voltage induced in the sense/inhibit winding when a core is switched from a 1 to a 0. This signal is detected and amplified by the sense amplifier whose output sets a data register flip-flop to store a 1. In effect, a 1 is read but the core is switched to the 0 state. Cores which were previously set to 0 are not affected.

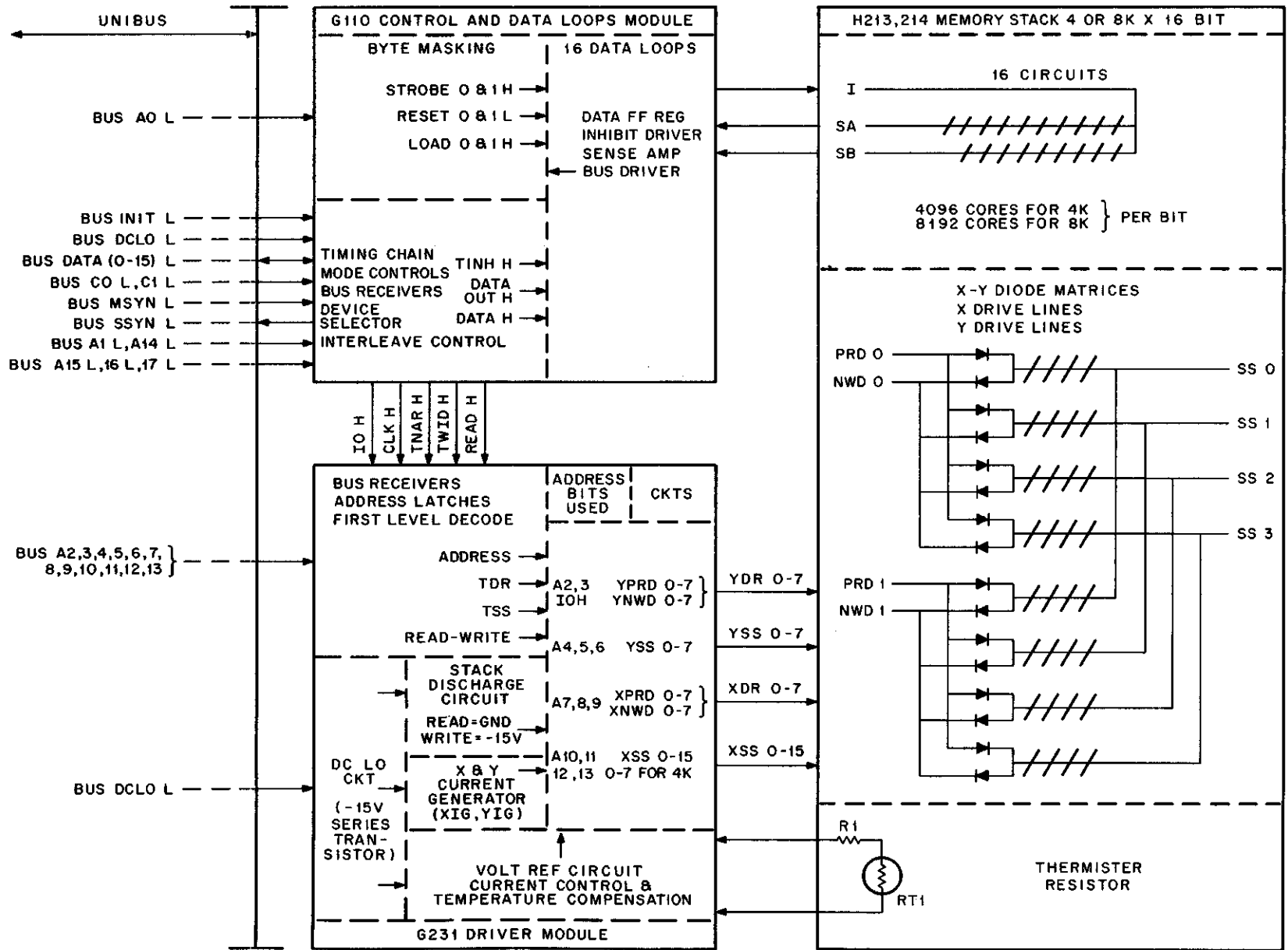


Figure 1-4 MM11-K, L Memory Block Diagram

#### 4. Data Register

The data register is a 16-bit flip-flop register used to store the contents of a word after it is read out of the destructive memory; the same word can then be written back into memory (restored) when in the DATI mode. The register is also used to accept data from the Unibus lines to accommodate the loading of incoming data into the core memory during the DATO or DATOB cycles.

#### 5. Device Selector

The device address is decoded in the device selector to determine if the memory bank has been addressed.

#### 6. Threshold Circuit and -5V Supply

The threshold circuit provides a reference threshold voltage to the sense amplifiers. During a read operation, if the threshold voltage (-5.2V) is exceeded, the sense amplifier produces an output. The -5V supply provide a negative voltage for the sense amplifiers.

1.2.3.2 G231 Drive Module - The G231 Drive Module contains the address selection logic, switches and drivers, current generator, stack discharge circuit, and DC LO protection circuit.

##### 1. Address Selection Logic

The core memory receives an 18-bit address from the master device. The address is latched and decoded to determine if the memory is the selected device and to determine the core location specifically addressed. If the operation is a byte operation, bus line A00 L indicates the byte to be used. The X- and Y-portion of the address is decoded through selection switches and a diode matrix to enable passage of read/write current through the selected X- and Y-drive lines of the memory. The coincidence of these currents selects the specific 16-bit core memory location desired.

##### 2. Switches and Drivers

The switches and drivers direct the flow of current through the magnetic cores to ensure the proper polarity for the desired function. This action is necessary because a single read/write line is used, and the current for a write operation is opposite in polarity to the current required for a read operation. There are separate switches and drivers for the read and write circuits in the selection matrix.

##### 3. Current Generators

X- and Y-current generators provides the current necessary to change the state of the magnetic cores. The linear rise time and amplitude of the output-current waveform have been selected to provide optimum switching of the core states and maximum signal-to-noise ratio for a wide range of temperatures.

#### 4. Stack Discharge Circuit

The stack discharge circuit maintains the proper stack charge voltage during operation: approximately 0V during a read operation and approximately -14V during a write operation.

#### 5. DC LO Protection Circuit

If any dc voltage is out of tolerance, DC LO is asserted on the Unibus. It is sensed by the DC LO protection circuit, which inhibits the memory operation by opening the -15V line to the current source. This prevents spurious memory operation.

1.2.3.3 H213 or H214 Stack Module - The stack module contains the ferrite core array and the X-Y diode matrices. For the 4K memory (H213), 16 core mats are used, each wired in a 64 x 64 matrix; 16 core mats, each wired in a 128 x 64 matrix are used for the 8K memory (H214). The stack also contains the resistor-thermistor combination to control the X-Y current generator temperature compensation.

#### 1.2.4 Basic Memory Operations

The core memory has four basic modes of operation. The main function of the memory is simply to read and write data. Additional modes are provided, however, to allow for byte operation and to eliminate the restore cycle when it is not needed, thereby increasing overall system efficiency. The four basic memory operations are:

- a. Read/restore (DATI)
- b. Read only (DATIP)
- c. Write (DATO)
- d. Write byte (DATOB).

These four modes are discussed briefly in the following paragraphs.

#### NOTE

In the following discussions, all operations refer to the master (controlling) device. For example, the term data out indicates data flowing out of the master and into the memory.

1.2.4.1 Data In (DATI) Cycle - The DATI cycle is a read/restore memory cycle. During this operation, the memory reads the information from the selected core location, transfers it to the Unibus, and then writes the information back into the memory location. This last step is necessary because the core memory is a destructive readout device. During the first part of the cycle, the memory loads the data into a register; at the same time, the memory applies the data to the Unibus. Then, during the second part of the cycle, the memory takes the data from the register and writes it back into the addressed memory location.

1.2.4.2 Data In, Pause (DATIP) Cycle - Normally in reading from memory, the information is destroyed in the particular location accessed, and the data must be restored. However, sometimes it is not actually necessary to restore the information after reading, because the location is to have a new data written into it. In this instance, eliminating the restore operation decreases the memory cycle time by approximately 50 percent. The DATIP operation is used for this purpose. The data is read from memory and the restore cycle is inhibited. Because no restore cycle is used, a DATIP must always be followed by a write cycle (either DATO or DATOB) on the same address or data in both addresses will be destroyed. The location having the DATIP will go to all 0s and the new address will have the OR function, bit by bit, of what was in it and what was put on the bus for the DATO.

1.2.4.3 Data Out (DATO) Cycle - The DATO cycle is a write memory cycle used by the master device to transfer data into core memory. To ensure that proper data are stored, the memory unit must first be cleared by reading the cores (thereby setting them all to 0) before writing in the new data. During a normal DATO, the memory first performs the read operation to clear the cores and then performs a write cycle to transfer data from the bus into the selected core location. If a DATO follows a DATIP (rather than a DATI), the sequence is not the same. The DATIP clears core and generates a Pause flag; the DATO skips the read cycle and immediately begins the write cycle. This process reduces DATO cycle time by approximately 50 percent.

1.2.4.4 Data Out, Byte (DATOB) Cycle - The DATOB cycle is similar in function to the DATO cycle, except that during DATOB data is transferred into the core memory from the bus in byte form rather than as a full word. Actually an entire word is loaded into the selected memory location: the selected byte, which is new data from the bus; and the non-selected byte, which is restored data from the word previously stored in that memory location. During the read cycle, the non-selected byte is saved by storing it in the data register while the selected byte is cleared. During the write cycle, only the selected byte portion of the word is loaded into the memory location from the bus. At the same time, the non-selected byte is restored from the data register into the memory location. In effect, the memory is first cleared and then simultaneously performs a restore cycle for the non-selected byte and a write cycle for the selected byte. This mode can follow a DATIP as described above.

## CHAPTER 2 DETAILED DESCRIPTION

### 2.1 INTRODUCTION

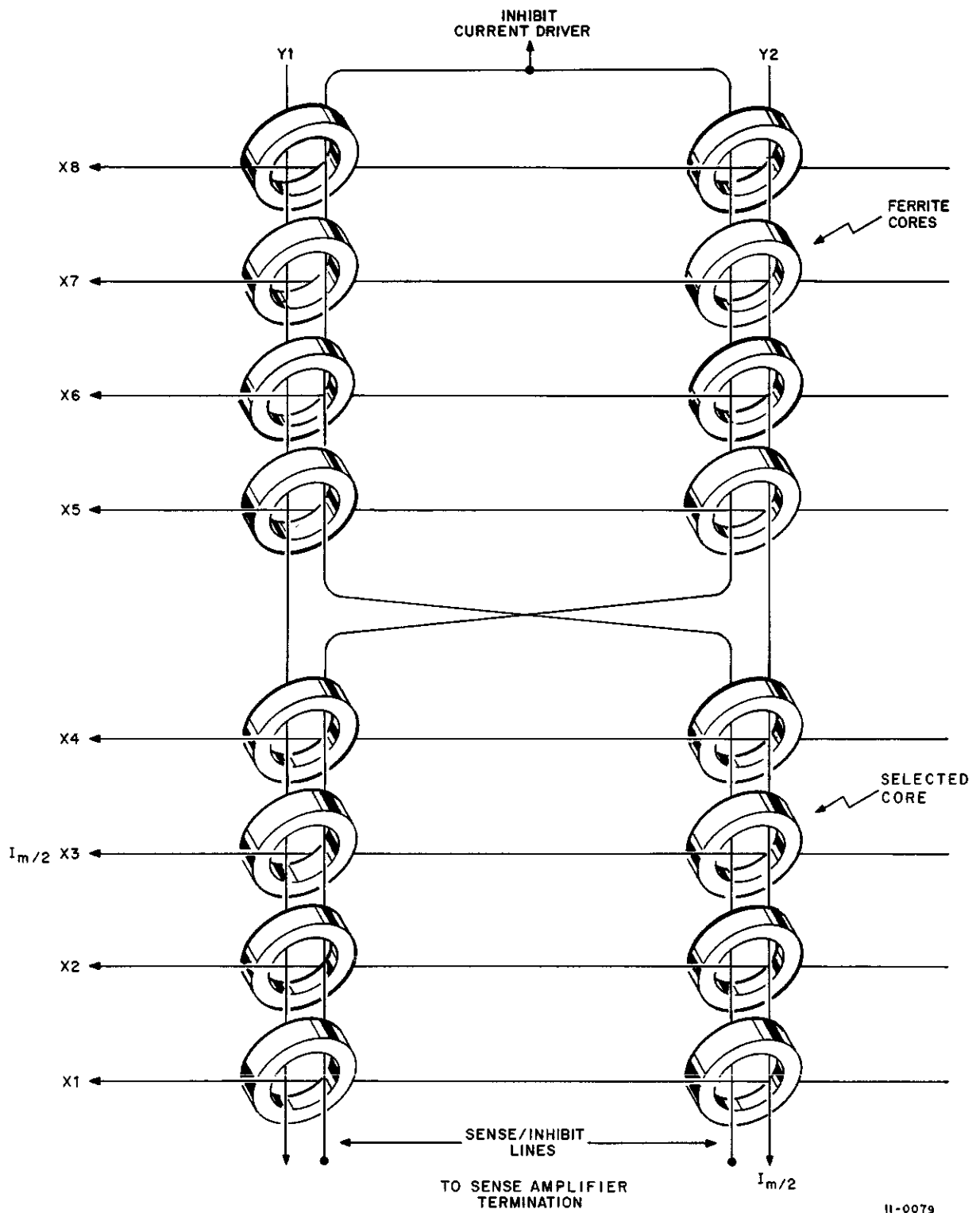
This chapter provides a detailed description of the MM11-K and L memories. The discussion is related to the 8K memory (MM11-L). The description of the 4K memory (MM11-K) is basically the same: only the differences are discussed.

The detailed description covers the core array, device and word selection, switches and drivers, current generation, stack discharge circuit, DC LO circuit, sense/inhibit circuitry, control and timing logic, and memory operating cycles.

### 2.2 CORE ARRAY

The ferrite core array for the 8K memory consists of 16 mats arranged in a planar configuration. Each mat contains 8192 ferrite cores arranged in a 128 x 64 array. Each mat represents a single bit position of a word. This planar configuration provides a total of 8192 16-bit word locations. The 4K memory core array consists of 16 mats each arranged in a 64 x 64 planar configuration to provide a total of 4096 16-bit word locations. Each ferrite core can assume a stable magnetic state corresponding to either a binary 1 or a binary 0. Even if power is removed from the core, the core retains its state until changed by appropriate control signals. The outside diameter of each core is 18 mil; the inside diameter is approximately 11 mil. Each core is 4.5 mil thick.

Selection and switching of the cores is provided by three wires traversing each core in a special selection technique. An X-axis read/write winding passes through all cores in each horizontal row for all 16 mats. AY-axis read/write winding passes through all cores in each vertical row for all 16 mats. Through the use of selection circuits which control the current applied to specific X-Y windings, any one of the 8192 or 4096 word locations can be addressed for writing data into memory or reading data out of memory.



11-0079

Figure 2-1 Three-Wire Memory Configuration



A third line passes through each core on a mat to provide the sense/inhibit functions. There is one sense/inhibit line per mat. This single sense/inhibit line, as well as the selection circuits, are discussed in subsequent paragraphs.

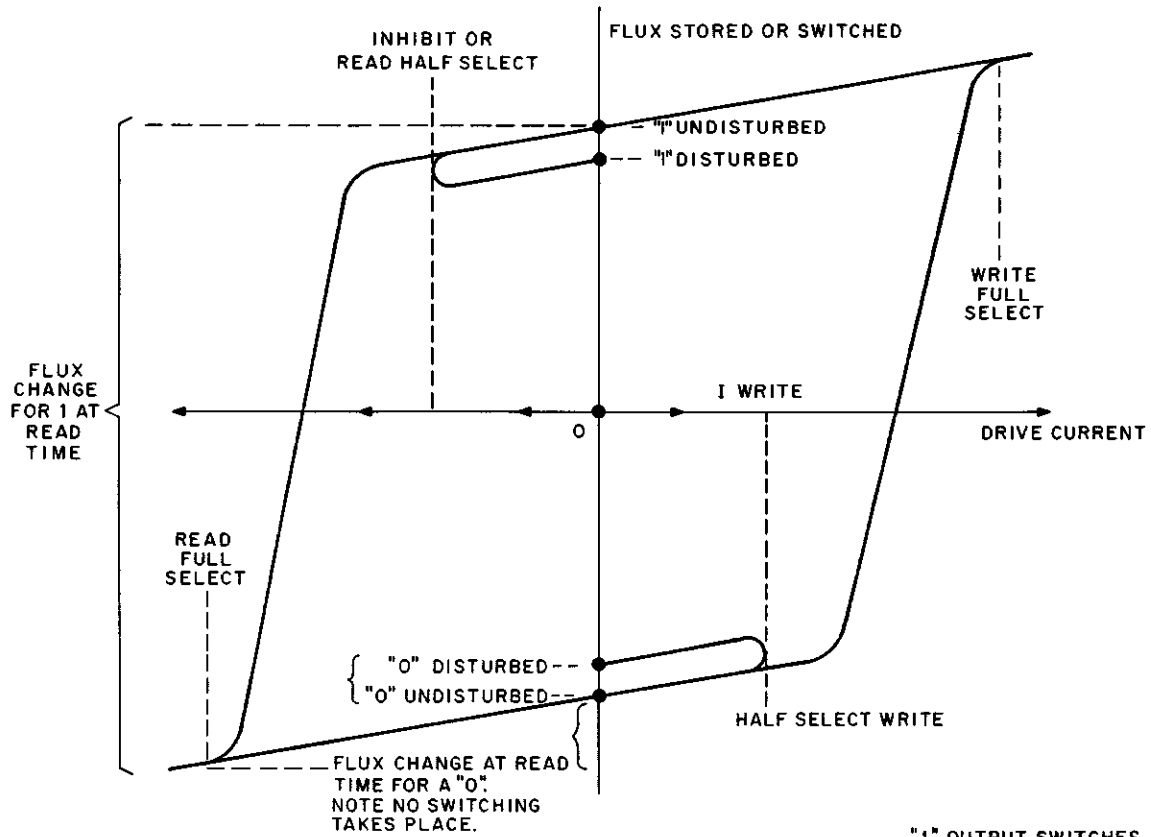
### 2.3 MEMORY OPERATION

Figure 2-1 illustrates a typical portion of the core memory. An X- and Y- winding pass through each core in the mat. The current passing through any one winding is such that no single winding produces a magnetic field strong enough to cause a core to change its magnetic state. Only the reinforcing magnetic field caused by the coincident current of both an X- and a Y- winding can cause the core located at the point of intersection to change states. It is this principle that allows the relatively simple winding arrangement to select one and only one memory core out of the total contained on each mat. The current passing through either an X- or Y- winding is referred to as the half-select current.

A half-select current passing through the X3 winding (Figure 2-1) from left to right produces a magnetic field that tends to change all cores in that horizontal row from the 0 to 1 state. The flux produced by the current is, however, insufficient to complete the state transition in any core. Simultaneously passing a half-select current through the Y- winding from top to bottom produces the same effect on all cores in that particular vertical row. Note, however, that both currents pass through only one core which is located at the intersection of the X3 and Y- windings. This is the selected core and the combined current values are sufficient to change the state of the core. The arrows in Figure 2-1 show current direction for the write cycle. All X- and Y- windings are arranged in such a manner that whenever a half-select current is passed through each, the resultant magnetic fields combine in the core at the point of intersection. This combined, full-select current ensures that the selected core is left in the binary 1 state. The currents used to select the core are referred to as write currents. A typical hysteresis loop for a core is shown in Figure 2-2.

In the MM11-K and L Core Memories, the X3 windings in all 16 mats are connected in series as are the Y1 windings. Therefore, whenever a full-select current flows through a selected core on one mat, it also flows through an identical core on the other 15 mats. The X3-Y1 cores on all mats switch to a binary 1, causing each of the 16 cores to become one bit of a 16-bit storage cell.

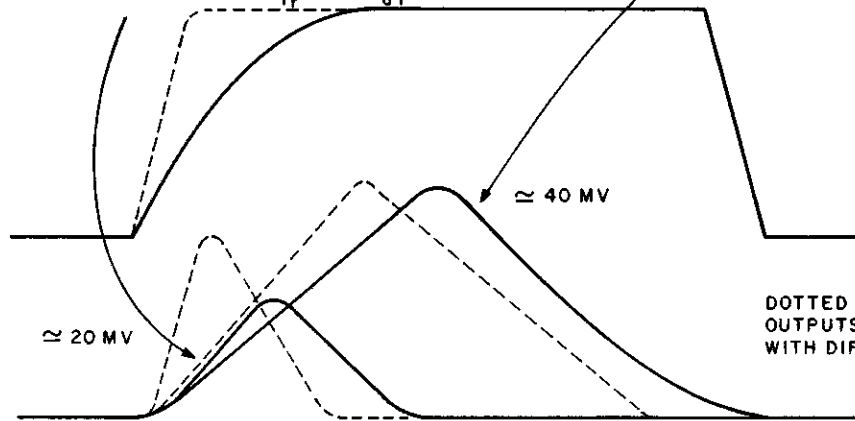
HYSTERESIS LOOP FOR CORE



"0" OUTPUT COMES DURING I RISE TIME AND IS A FUNCTION OF IT AND CURRENT AMPLITUDE.

"0" OUTPUT  $\propto \frac{1}{T}$  OR  $\propto \frac{di}{dt}$

"1" OUTPUT SWITCHES AT THE CORE TIME CONSTANT AND IS PRIMARILY DEPENDENT ON CURRENT AMPLITUDE. IT WILL SWITCH FASTER AND GROW AS RISE TIME IS DECREASED.



DOTTED LINES SHOW HOW OUTPUTS WOULD BEHAVE WITH DIFFERENT CURRENTS

11-0088B

Figure 2-2 Hysteresis Loop for Core

Because of the serial nature of the X-Y windings, a method is used that allows cores to remain in the 0 state during a write operation; otherwise, every 16-bit word selected would be all 1s. The method used in the MM11-K and L Core Memories is to first clear all cores to the 0 state by reading and then, by using an inhibit winding during the write operation, to inhibit cores on particular mats. The inhibited cores remain 0s even when identical cores on other mats are set to 1s.

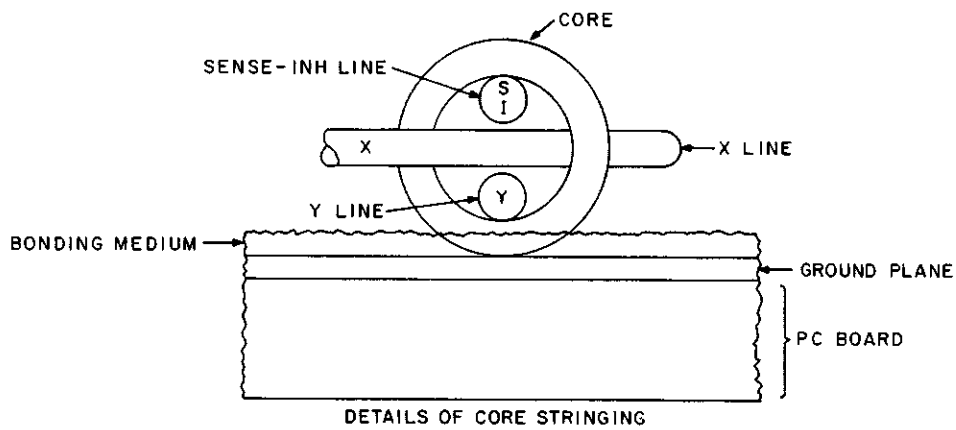
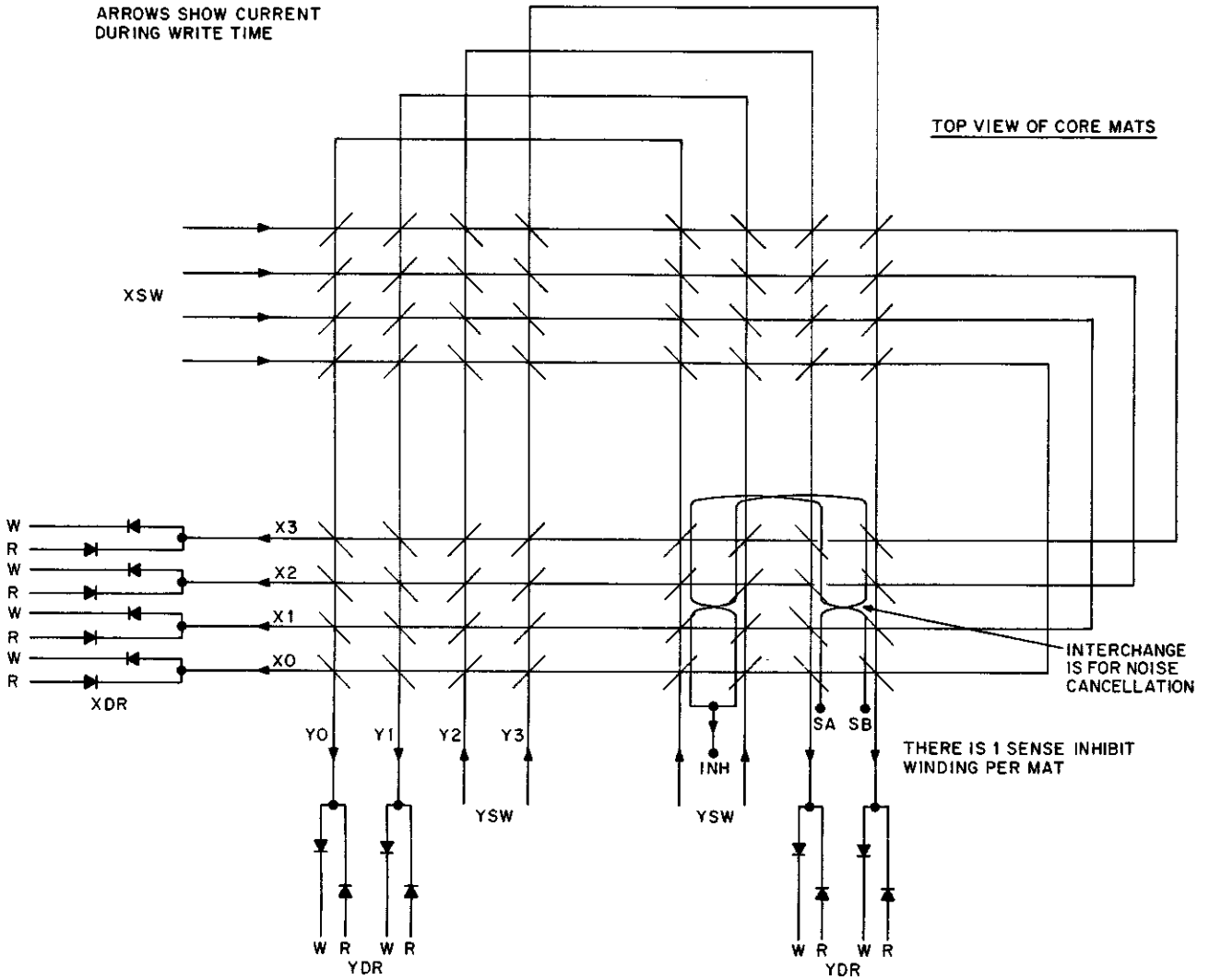
The half-select current for the inhibit lines is applied from an inhibit current driver, which is a switch and a resistor between the inhibit line and -15V. The current in the inhibit line flows in the opposite direction from the write current in all Y- lines and cancels out the write current in any Y- line. There is a separate inhibit driver for each memory mat, and each mat represents one bit position of a word; thus, selected bits can be inhibited to produce any combination of binary 1s and 0s desired in the 16-bit word. It must be remembered that the inhibit function is active only during write time.

The sense/inhibit lines are also used to read out information in a selected 16-bit memory cell. The specific core is selected at read time in the same manner as during the write cycle with one notable exception: the X- and Y- currents are in the opposite direction. These opposite half-select currents cause all cores previously set to 1 to change to 0; cores previously set to 0 are not affected. Whenever the core changes from 1 to 0, the flux change induces a current in the sense winding of that mat. This current is detected and amplified by a sense amplifier. The amplifier output is strobed into the data register for eventual transfer to the Unibus. Figure 2-3 shows a 16-word by 4-bit planar memory. The MM11-L Core Memory (8K) functions in the same manner, except that it has 128 X-lines, 64 Y-lines, and 16 core mats. The core stringing is identical, and the sense windings are strung through all 8192 cores with the interchange between X63 and X64 instead of between X1 and X2. For the 4K memory, the interchange is between X31 and X32 and it has 64 X-lines and 64 Y-lines.

## 2.4 DEVICE AND WORD SELECTION

When the processor or a peripheral device attempts to perform a transaction with the memory, the processor asserts an 18-bit address on Unibus address lines A <17:00>. Six of these 18 bits (A01 and A <17:13>) indicate the address of the memory as a device.

Depending on the memory configuration, only four or five bit combinations of these bits are used as shown in Table 2-1. Eleven of the 12 remaining bits (A <12:02>) plus A01



11-0088A

Figure 2-3 Three-Wire 3D Memory, Four Mats Shown for a 16-Word - 4-Bit Memory

and A13 indicate the address of a specific word within the memory. Address bit A00 is used to select the byte (8 bits) transaction when in DATOB mode.

The memory address is decoded by the device selection circuit on the G110 Control Module. The word address is stored in a register on the G231 Driver Module whose output is decoded to activate the X-Y line switches and drivers which select the addressed word. These circuits contain jumpers which are included or excluded to establish a specific device address and select 4K or 8K word capacity. Jumpers are provided to select interleaved or non-interleaved operation for the 8K model; however, the memory is to be operated in the non-interleaved mode only.

Table 2-1  
Addressing Functions

Bus Address	Function	
	4K Mode	8K Mode
A00	Controls byte mode	Controls byte mode
A01	Becomes A01H to G231	Becomes A01H to G231
A02, A03, A01H*	Decode Y-Drivers	Decode Y-Drivers
A04, A05, A06	Decode Y-Switches	Decode Y-Switches
A07, A08, A09	Decode X-Drivers	Decode X-Drivers
A10, A11, A12	Decode X-Switches	Decode X-Switches
A13	Goes to device selector	Decode X-Switches
A14	Goes to device selector	Goes to device selector
A15, A16, A17	Goes to device selector	Goes to device selector

\*A01H is not a Unibus signal.

Table 2-1 lists the function of each address bit. Figure 2-4 is a simplified block diagram of the device and word address selection circuits.

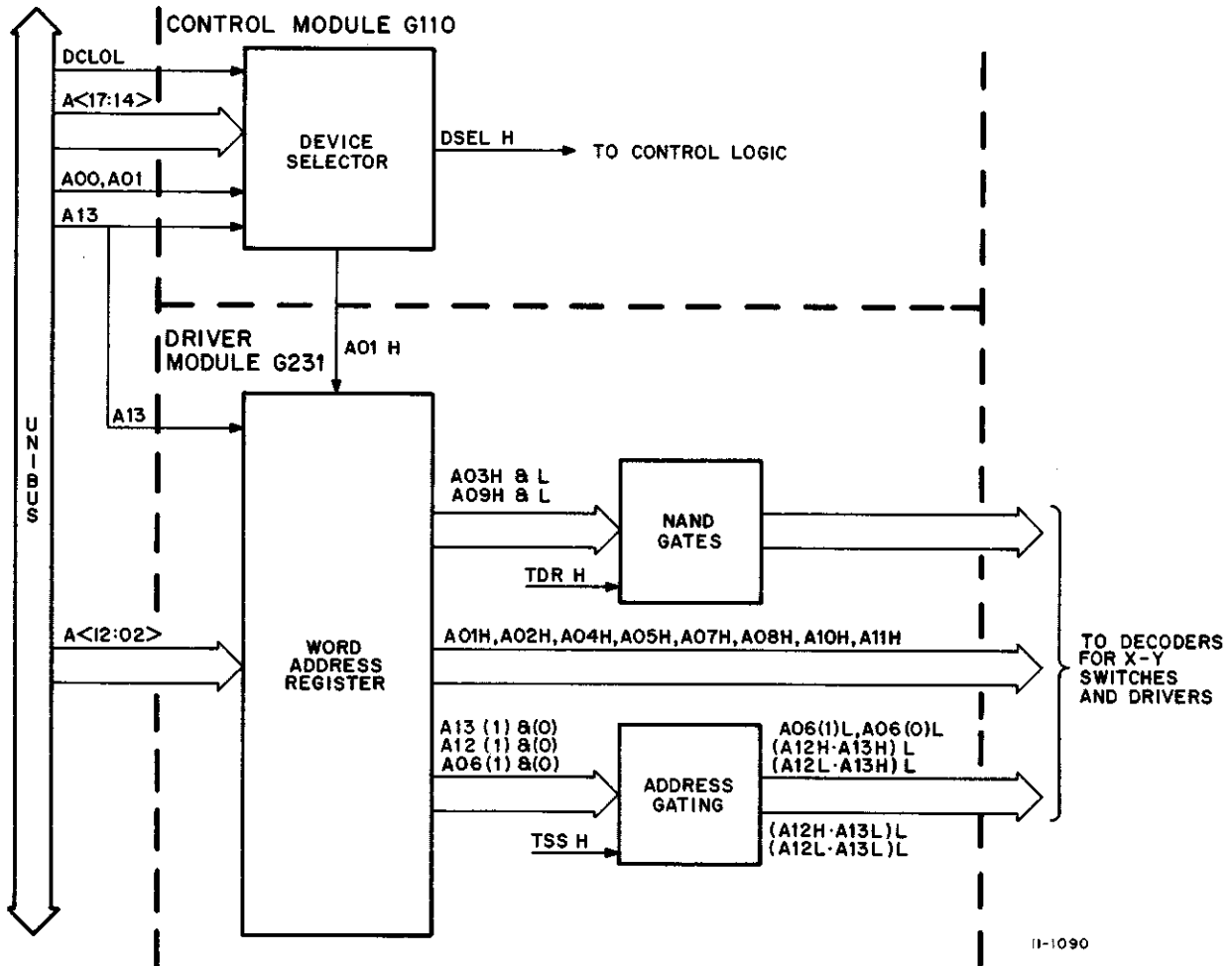
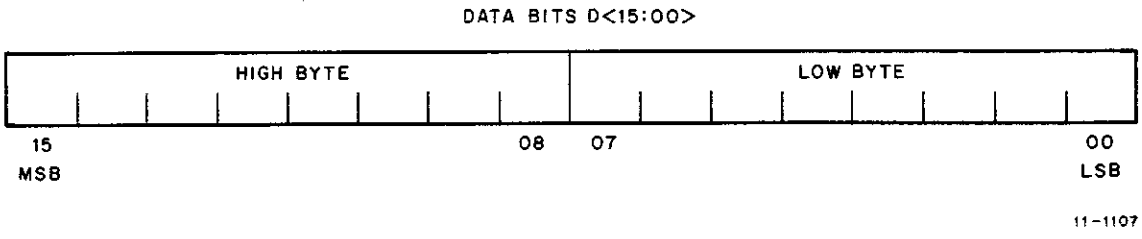


Figure 2-4 Device and Word Address Selection Logic, Block Diagram

#### 2.4.1 Memory Organization and Addressing Conventions

Prior to a detailed discussion of the address selection logic, it is important to understand memory organization and addressing conventions.

The memory is organized in 16-bit words each consisting of two 8-bit bytes. The bytes are identified as low and high as shown below.



Each byte is addressable and has its own address location: low bytes are even numbered and high bytes are odd numbered. Words are addressed at even numbered locations only: the high (odd) byte is automatically included.

For example, an 8K word memory has 8192 words or 16,384 bytes; therefore, 16,384 locations are assigned. The address locations are specified as six digit octal numbers. The 16,384 locations for the 8K memory are designated 000000 through 037777. Figure 2-5 shows the organization for an 8K memory.

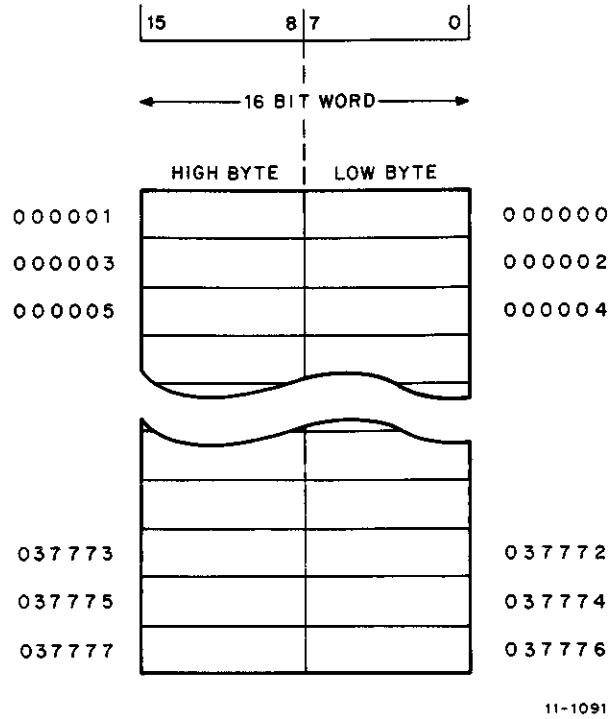


Figure 2-5 Memory Organization for 8K Words

The address selection logic responds to the binary equivalent of the octal address. The binary equivalent of 017772 is shown below as an example.

ADDRESS BITS A<17:00>

17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	BIT POSITION
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	BINARY
0			1			7			7			7			2		OCTAL	

11-1106

Each memory bank (4K or 8K words) requires its own unique device address. For example, assume that a system contains three 8K memory banks (Figure 2-6). The device selector for the 8K non-interleaved memory decodes four address lines (A <17:14>). Examination of the binary states of these bits for the three memory banks shows that the changes in the states of bits A14 and A15 allow the selection of a unique combination for each bank. The combination, which is the device address, is hardware-selected by jumpers in the device selector.

During system operation, the processor generates the binary equivalent of the octal address on Unibus address lines A <17:00>. The processor uses positive logic and the Unibus uses negative logic. With this in mind, the following note is included to remind the reader of the negative logic convention of the Unibus.

#### NOTE

##### Processor (Positive Logic)

Signal Asserted: High = Logical 1 = +3V

Signal at Rest: Low = Logical 0 = 0V

##### Unibus (negative Logic)

Signal Asserted: Low = Logical 1 = 0V

Signal at Rest: High = Logical 0 = +3V

### 2.4.2 Device Selector

The device selector located on the G110 Control Module (drawing G110-0-1, sheet 2) shows a logic diagram of the device selector in the 4K configuration.

Address bits A01 and A <17:13> are decoded in the device selector to provide the device selection signal D SEL H that is used in the control logic. Two combinations of these bits are decoded, depending on the memory configuration as shown below.



Memory Configuration

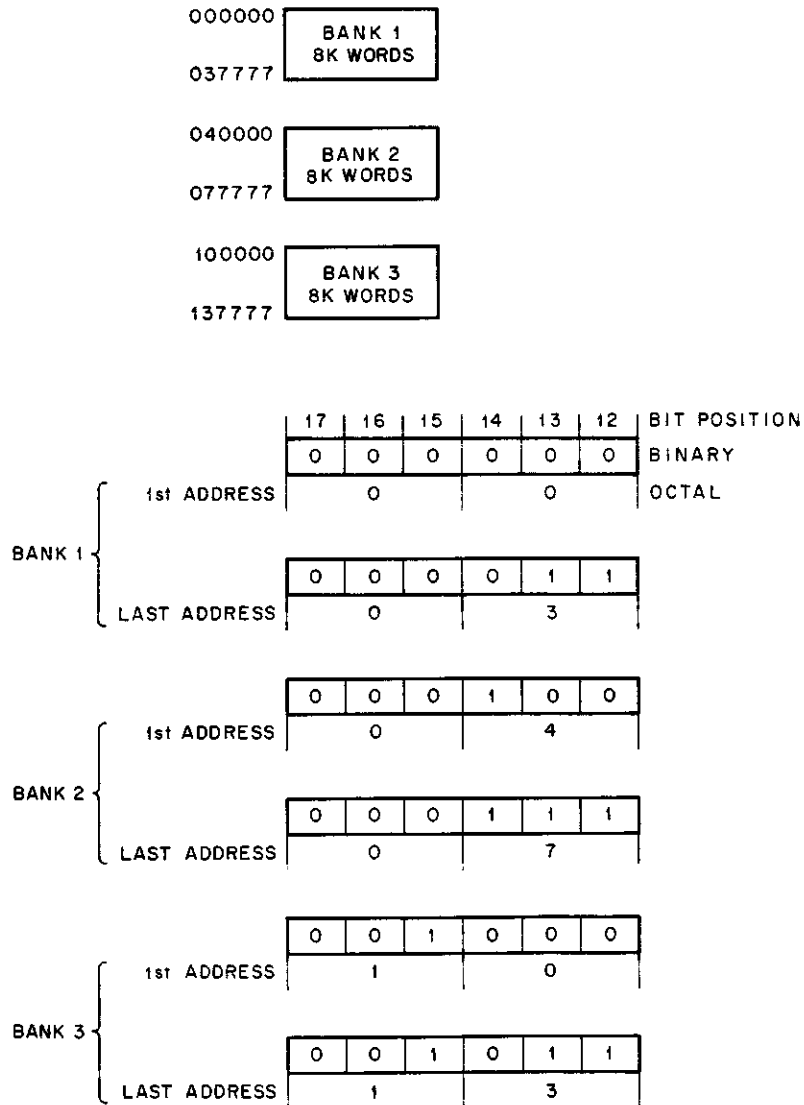
4K Words

8K Words

Address Bits

A <17:13>

A <17:14>



11-1092

Figure 2-6 Address Assignments For Three Banks of 8K Words Each

Obviously, the memory capacity is determined by the stack module: H213 for 4K words and H214 for 8K words. The same control module is used for both 4K and 8K memories; therefore, two jumpers (W9 and W10) are provided to include or exclude address bit A13 commensurate with the memory word size. Two jumpers (J3 and J4) on the G231 Driver Module (drawing G231-0-1, sheet 2) are provided for A13 inclusion or exclusion in the word addressing logic. The same driver module is used for both memory capacities. In the 4K word size, the components associated with the additional X-line read and write switches needed for 8K words may be removed. Two jumpers (W7 and W8) in the device selection logic on the control module are used to select interleaved or non-interleaved operation of the 8K memory. They are configured to provide non-interleaved operation only.

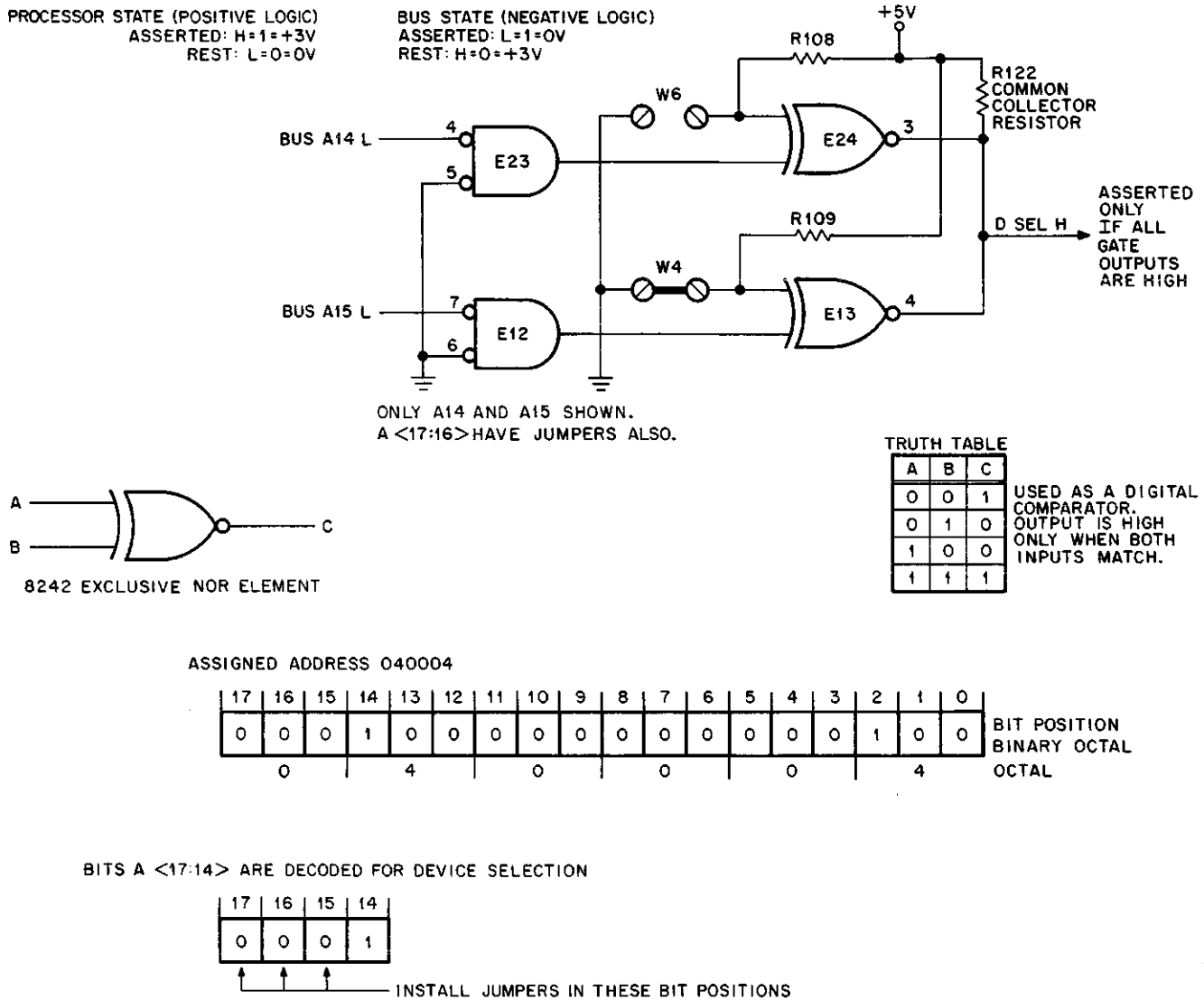
Each memory bank (4K or 8K) must have its own unique device address. Five jumpers (W2-W6) in the device selector provide this capability. On drawing G110-0-1, sheet 2, all the jumpers are shown in place and the device selector responds only when high signals appear on the Unibus address lines A <17:13>. Some jumpers can be removed to allow the device selector to respond to a particular combination of high and low signals on these address lines.

All highs at the inputs of the 7380 Unibus receivers (E12 and E23) give lows at their outputs. Each receiver output goes to one input of a type 8242 Exclusive NOR gate. Because of jumpers W7 and W8, bit A14 is decoded for 4K and 8K configurations. An additional receiver is used to sense BUS DC LO L and its output (E23 pin 14) is sent to an 8242 gate (E24 pin 5). BUS DC LO L is asserted only when the dc voltages from the power supply drop below specified limits.

The other input of the 8242 gates associated with bits A14, A13, A15, A16 and A17 can be connected to +5V or ground, depending on whether or not jumpers W2-W6 are installed. The input is low (ground) with the jumper in; with the jumper removed, the input is high (+5V). Each 8242 gate is used as a digital comparator: its output is high only when both inputs are identical. The 8242 gates have open-collectors and they are connected in common; therefore, the comparator output D SEL H is high only when all gates detect matched inputs (both lows or both highs).

An installed jumper requires a low signal at the output of the 7380 Unibus receiver. The 7380 is connected as an inverter so this signal is reflected as a high on the Unibus (logi-

cal or asserted state for the Unibus). To configure the jumpers for a specific device address, find the binary equivalent of the assigned octal address and insert a jumper in each bit position that contains a 0. A specific jumper configuration is shown in Figure 2-7.



11-1093

Figure 2-7 Jumper Configuration For A Specific Memory Address

The previous discussion dealt with the 4K memory configuration of the device selector as shown in drawing G110-0-1, sheet 2. Address bits A <17:13> are decoded and the output of bit A01's Unibus receiver (E23 pin 2) is sent via jumper W8 to the word address register as A01H.

In the 8K memory configuration, jumper W9 is removed and W10 is installed. This removes bit A13 from the input of Unibus receiver E12 on G110 and replaces it with +5V via resistor R107. This receiver output (pin 14) always remains low so that jumper W5 must remain installed to ensure a match on pins 12 and 13 of gate E13. The jumper configurations for memory systems up to 128K words are shown in Figure 2-8.

### 2.4.3 Word Selection

Word selection requires two levels of decoding. The word address bits are placed in the 13-bit word address register: 12 bits are used for a 4K memory, and 13 bits are used for an 8K memory. Some bits from the register output are combined in a gating network. The outputs from the gating network and some outputs directly from the register are used as inputs to a group of decoders (Figure 2-4). The outputs of the decoders select the proper X- and Y- read/write switches and drivers.

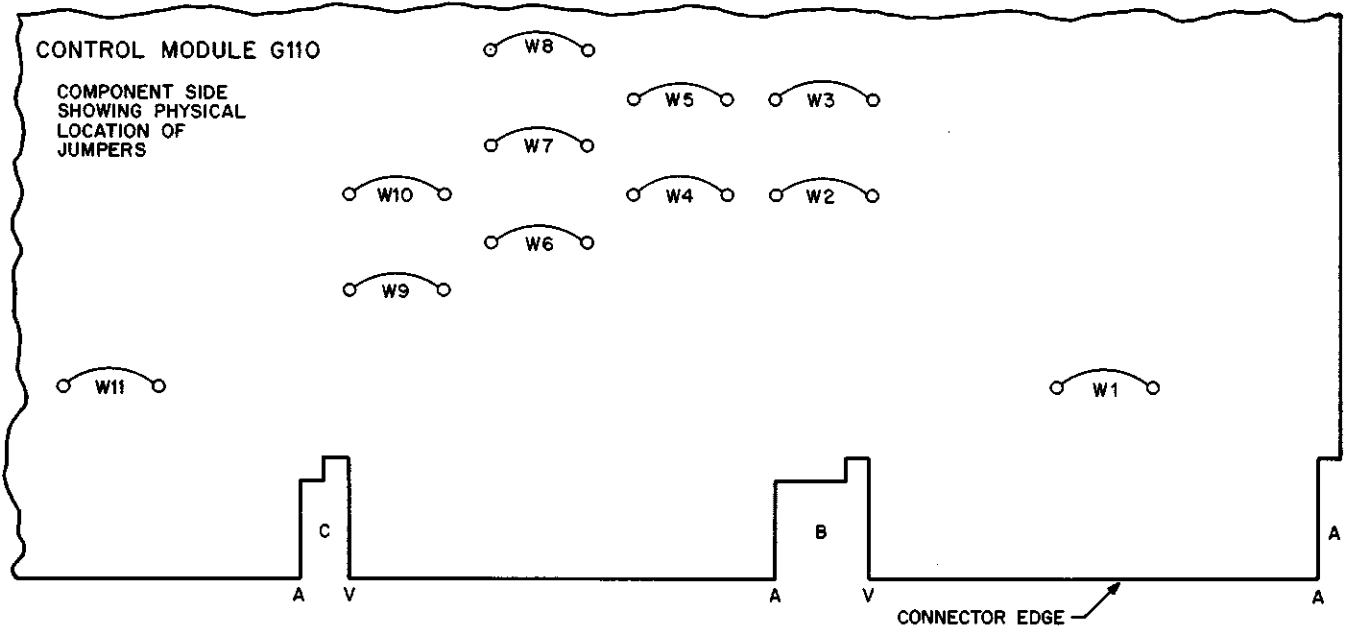
**2.4.3.1 Word Address Register and Gating Logic** - The word address register and gating logic are contained on the G231 Driver Module. The circuit schematic is shown in drawing G231-0-1, sheet 2. The register is composed of 13 74H74 dual D-type edge-triggered flip-flops. They are identified as E11, E12, E13, E14, E18, E19, and E20. The output (pin 3) of gate E9 provides a high signal on the preset input (pin 4 or pin 10) of each flip-flop, which prevents direct presetting of the flip-flop. Direct clearing of each flip-flop is prevented by a high signal on the clear input (pin 1 or pin 13) via the output (pin 2) of gate E9. The register cannot be directly cleared or preset: its output responds only to the signal at its data (D) input.

Address bits A <13:02> are picked off the Unibus via type 7380 receivers (E15, E16, and E17). The receiver outputs are sent to the corresponding flip-flop D-inputs. The input to the receiver associated with bit A13 has two sources: Unibus signal BUS A13L via jumper J4, or +5V via jumper J3. These jumpers are associated with the memory word size. A 4K memory requires J3 in and J4 out: an 8K memory requires J4 in and J3 out. Because BUS A13L is used on the G110 module as part of the device selector, this arrangement prevents loading BUS A13L twice per memory bank.

The E11 flip-flop associated with bit A01 receives its input from the device selector (drawing G110-0-1, sheet 2). The input signal is A01 H, which is obtained from bit A01 Unibus receiver for both 4K and 8K memories.

Memory Bank (Words)	Machine Address (Words)	Device Address Jumpers				
		W5 A13	W6 A14 or A01	W4 A15	W3 A16	W2 A17L
0-4K	000000-017776	IN	IN	IN	IN	IN
4-8K	020000-037776	OUT	IN	IN	IN	IN
8-12K	040000-057776	IN	OUT	IN	IN	IN
12-16K	060000-077776	OUT	OUT	IN	IN	IN
16-20K	100000-117776	IN	IN	OUT	IN	IN
20-24K	120000-137776	OUT	IN	OUT	IN	IN
24-28K	140000-157776	IN	OUT	OUT	IN	IN
28-32K	160000-177776	OUT	OUT	OUT	IN	IN
32-36K	200000-217776	IN	IN	IN	OUT	IN
36-40K	220000-237776	OUT	IN	IN	OUT	IN
40-44K	240000-257776	IN	OUT	IN	OUT	IN
44-48K	260000-277776	OUT	OUT	IN	OUT	IN
48-52K	300000-317776	IN	IN	OUT	OUT	IN
52-56K	320000-337776	OUT	IN	OUT	OUT	IN
56-60K	340000-357776	IN	OUT	OUT	OUT	IN
60-64K	360000-377776	OUT	OUT	OUT	OUT	IN
64-68K	400000-417776	IN	IN	IN	IN	OUT
68-72K	420000-437776	OUT	IN	IN	IN	OUT
72-76K	440000-457776	IN	OUT	IN	IN	OUT
76-80K	460000-477776	OUT	OUT	IN	IN	OUT
80-84K	500000-517776	IN	IN	OUT	IN	OUT
84-88K	520000-537776	OUT	IN	OUT	IN	OUT
88-92K	540000-557776	IN	OUT	OUT	IN	OUT
92-96K	560000-577776	OUT	OUT	OUT	IN	OUT
96-100K	600000-617776	IN	IN	IN	OUT	OUT
100-104K	620000-637776	OUT	IN	IN	OUT	OUT
104-108K	640000-657776	IN	OUT	IN	OUT	OUT
108-112K	660000-677776	OUT	OUT	IN	OUT	OUT
112-116K	700000-717776	IN	IN	OUT	OUT	OUT
116-120K	720000-737776	OUT	IN	OUT	OUT	OUT
120-124K	740000-757776	IN	OUT	OUT	OUT	OUT
124-128K	760000-767776	OUT	OUT	OUT	OUT	OUT

Figure 2-8 Device Decoding Guide



- NOTES:
1. Jumper W1 is for test purposes only. It must be installed for normal operation.
  2. Jumper W11 should be removed for normal operation. When installed the memory responds to DATI only, regardless of state of control lines CO0 and CO1.
  3. Jumpers W7 and W8 must remain in the factory installed positions.
  4. When used as an 8k bank, jumpers W5 and W10 must be installed and jumper W9 must be removed.
  5. When used as a 4k bank, jumper W10 must be removed and jumper W9 must be installed. Jumper W5 determines the location of the bank on the bus.

11-1149

Figure 2-8 Device Decoding Guide (continued)

The register flip-flops are clocked synchronously by CLK 1 H from the control logic (drawing G110-0-1, sheet 2). Clocking occurs on the positive-going edge of CLK 1 H. The generation and timing of this clock signal is discussed in Paragraph 2.8. When the register is clocked, the outputs of flip-flops A01, A02, A04, A05, A07, A08, A10 and A11 are sent to the type 8251 X-Y decoders on the G231 Driver Module (drawing G231-0-1, sheets 3 and 4). The outputs of flip-flops A06, A12, and A13 are combined in a group of six type 74H10 NAND gates (three E22s, and three E25s), which are enabled by signal TSS H. Table 2-2 lists the states of flip-flops A06, A12, and A13 that are required to enable these gates. The outputs of flip-flops A03 and A09 are gated with TDR H in high-speed 2-input NAND gates and then applied to the decoders for the drivers only.

The six signals listed in Table 2-2 are sent only to the X-Y line read/write switch decoders on the driver module.

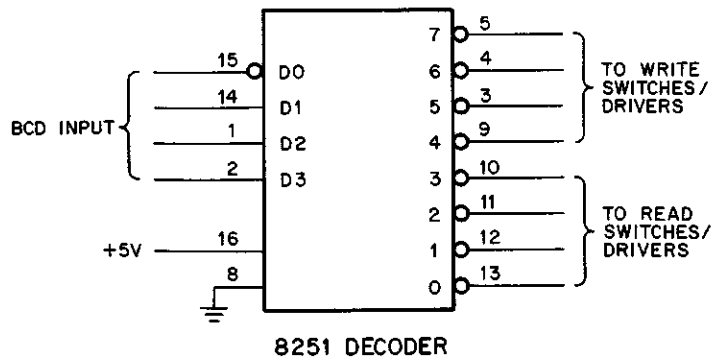
Table 2-2  
Enabling Signals for Word Register Gating

Output Signals		Enabling Signals		
Gate	Asserted Signal	FF A06	FF A12	FF A13
E22 pin 12	(A06H) L	set	X	X
E22 pin 8	A06L	reset	X	X
E22 pin 6	(A12H · A13H) L	X	set	set
E25 pin 12	(A12L · A13H) L	X	reset	set
E25 pin 8	(A12H · A13L) L	X	set	reset
E25 pin 6	(A12L · A13L) L	X	reset	reset

Signal ISS H is generated at the output (pin 3) of negative input OR gate E4 during a read or write operation. During a read operation, the enabling signal is produced at NAND gate E4, pin 8 by ANDing READ H and TNAR H. During a write operation, the enabling signal is produced at NAND gate E4, pin 6 by ANDing WRITE H and TWID H. Signals READ, TNAR and TWID are generated by the control logic on the G110 Control Module. WRITE is the complement of READ (produced by inverter E6). Signal READ H comes from the 1 output of R/W flip-flop E13 (drawing G110-0-1, sheet 2): the READ H signal is produced when the flip-flop is set. When the R/W flip-flop is cleared, READ H is low and is inverted by E6 to produce WRITE H.

2.4.3.2 X- and Y- Line Decoding - The basic decoding unit is a type 8251 BCD-to-decimal decoder that converts a 4-bit BCD input code to a one-of-ten output; however, only eight outputs are used. Figure 2-9 shows an 8251 and associated truth table. The inputs are D0, D1, D2, and D3: they are weighted 1, 2, 4, and 8 with D0 being the least significant bit. The outputs are 0-7 and are mutually exclusive. The selected output is low and all others are high.

For the 8K memory, ten decoders are used: six for the X-axis and four for the Y-axis. Each decoder controls four read/write switch pairs. Each pair is associated with a specific switch or driver. This switch matrix is combined with the stack X-Y diode matrix to allow



INPUTS				OUTPUTS							
D3	D2	D1	D0	0	1	2	3	4	5	6	7
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0

TRUTH TABLE

11-1095

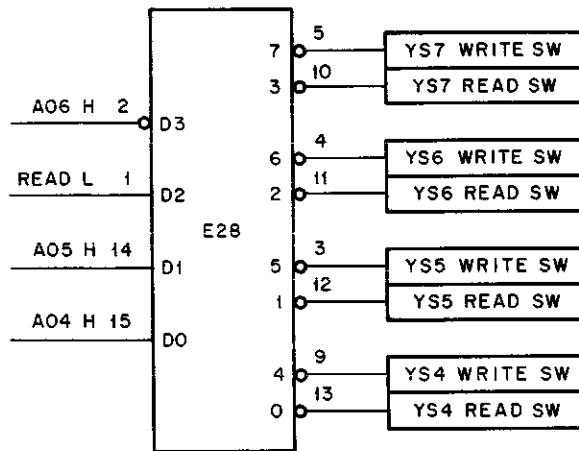
Figure 2-9 Type 8251 Decoder, Pin Designation and Truth Table

selection of any location out of the total 8192 Locations (stack drawing DCS-H214-0-1 for interconnections).

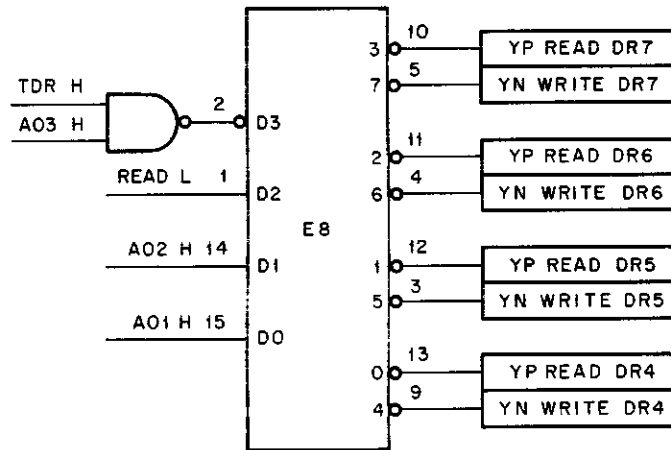
For the 4K memory, eight decoders are used: four for each axis. The stack X-diode matrix is halved to allow selection of any location out of the total 4096 locations (stack drawing DCS-H213-0-1 for interconnections). A discussion of the configuration and operation of the switches and diode matrices is given in Paragraph 2.4.3.3.

The X- and Y- line switches are first differentiated as switches and drivers. The drivers are those switches that are connected to the diode end of the stack. Drivers and switches are further differentiated by function: either read or write. Another differentiation is made by polarity: negative or positive depending on the physical connection. Read drivers and write switches are connected to the current generator outputs and are considered positive; write drivers and read switches are connected to -15V and are considered negative.





DECODER FOR READ AND WRITE SWITCHES YS4 - YS7



DECODER FOR READ AND WRITE DRIVERS Y4-Y7

11-1096

Figure 2-10 Decoding of Read/Write Switches and Drivers Y4-Y7

Figure 2-10 shows the decoders associated with Y-line read and write switches 4-7 and Y-line read and write drivers 4-7. Refer also to the truth table in Figure 2-9. In both decoders (E28 for switches and E8 for drivers), the signal to input D3 selects the block of switch pairs. This signal must be low for any output to be selected. The signal to input D2, which is READ L for all decoders, controls the selection of read or write switches/drivers. When READ L is low, outputs 0-3 are selected: these are read switches and read drivers. When READ L is high, outputs 4-7 are selected: these are write switches and write drivers. The four combinations of the states of inputs D0 and D1 select the particular switch/driver.

The four driver decoders (E3, E8, E43, and E46 on drawing G231-0-1, sheets 3 and 4) have a NAND gate connected to input D3. Signal TDR H is an input to each gate; therefore, the driver decoders cannot be enabled unless TDR H is high. This signal is generated on the G231 Driver Module (drawing G231-0-1, sheet 2, coordinates A-8) by ANDing TWID H and READ H or TNAR H and WRITE H.

Each switch/driver is connected to the decoder output through a transformer-coupled base drive circuit. When the decoder output is at ground (low), the switch/driver is turned on; it is turned off when the decoder output is at +3.5V (high). The base drive circuit for write switch YS7 shown in Figure 2-11 is typical.

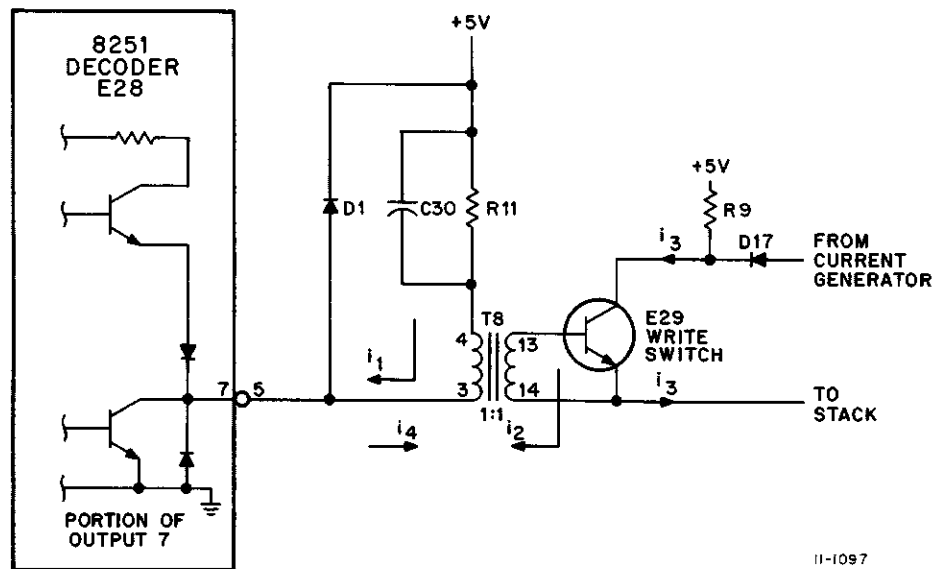


Figure 2-11 Switch or Driver Base Drive Circuit

In this example, the decoder inputs have selected output 7, which is at ground. Current  $i_1$  flows into this decoder output circuit from the +5V supply via resistor R11 and the primary winding (terminals 4 and 3) of transformer T8. The value of  $i_1$  is determined by the value of R11 and the voltage reflected into the transformer primary (approximately 1.0V). An equal current  $i_2$  is induced in the base-emitter circuit of write switch E29, which is connected to the transformer secondary winding (terminals 13 and 14). This current turns on E29. All the base current for E29 is provided by this circuit:  $i_3$  is the collector current. When the decoder is turned off, its output pull-up transistor tries to drive the turn-off current  $i_4$  in the opposite direction. This reverse current removes the forward bias from the

base of E29 and turns it off. Capacitor C30 allows the decoder to pump reverse current  $i_4$  into the transformer primary; it also speeds up turn-on current  $i_7$ . Diode D1 prevents reverse breakdown of the base-emitter junction of E29; it also protects the decoder output.

2.4.3.3 Drivers and Switches - Drivers and switches direct the current through the X- and Y- lines in the proper direction as selected by the read and write operations.

For an 8K memory, 16 pairs of read/write switches and 8 pairs of read/write drivers are provided in the X-axis; eight pairs of read/write switches and eight pairs of read/write drivers are provided in the Y-axis. In conjunction with the stack diode matrix (drawing H214-0-1, sheet 2), one driver and any one of 16 switches select 16 lines in the X-axis; one driver and any one of eight switches select eight lines in the Y-axis. This allows selection of 128 lines in the X-axis and 64 lines in the Y-axis. This provides a 128 x 64 matrix that selects any location out of 8192 locations.

For a 4K memory, eight pairs of read/write switches and eight pairs of read/write drivers are provided for each axis (X and Y). One driver and any one of eight switches select eight lines in both axes, which allows selection of 64 lines in each axis and provides a 64 x 64 matrix that selects any location out of 4096 locations. The size of the X-diode matrix for the 4K memory is one half the size of the corresponding matrix for the 8K memory (drawing H213-0-1, sheet 2). In both memories, the diodes prevent sneak currents in the stack and steer all switched current into the selected stack line.

Figure 2-12 is one fourth of a Y-selection matrix showing the interconnection of the diodes and the lines from the switches and drivers. It also shows how four pairs of switches and drivers are connected to select 16 locations. Refer to drawing H213-0-1, sheet 2 for an extension of this method that uses eight pairs of switches and drivers to select 64 locations.

Figure 2-12 shows four pairs of drivers and four pairs of switches for the Y-axis only; polarities are shown for convenience. The diodes are identified to assist in associating them with the drivers and switches. Each line from a twin diode interconnection to a read/write switch pair passes through 64 cores and represents one line on each bit mat. Assume that a write operation is to be performed and the word address decoders have selected write switch WYS00 and write driver YNWD1. The Y-current generator sends current through write switch WYS00 (conventional flow), which puts a positive voltage on the anodes of diodes 03W, 02W, 01W and 00W. The non-selected write drivers (YNWD3,

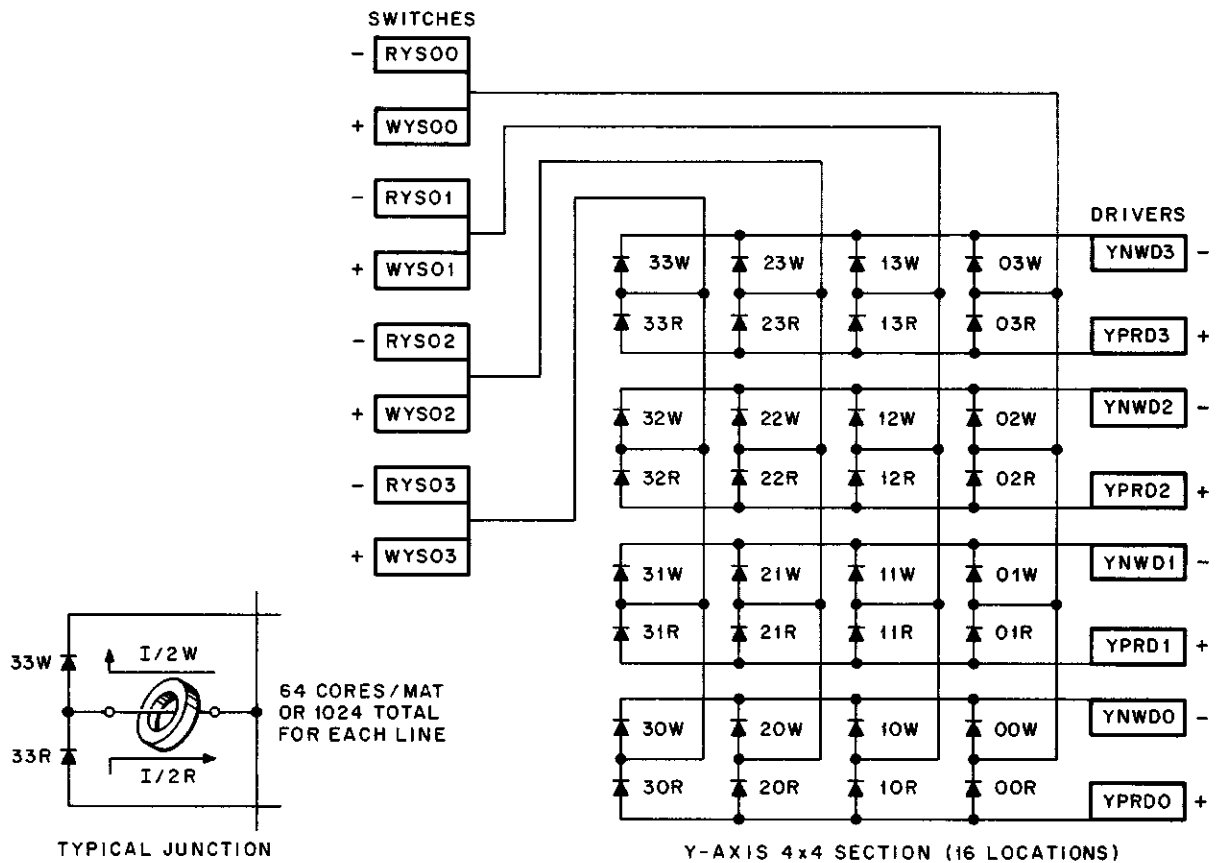
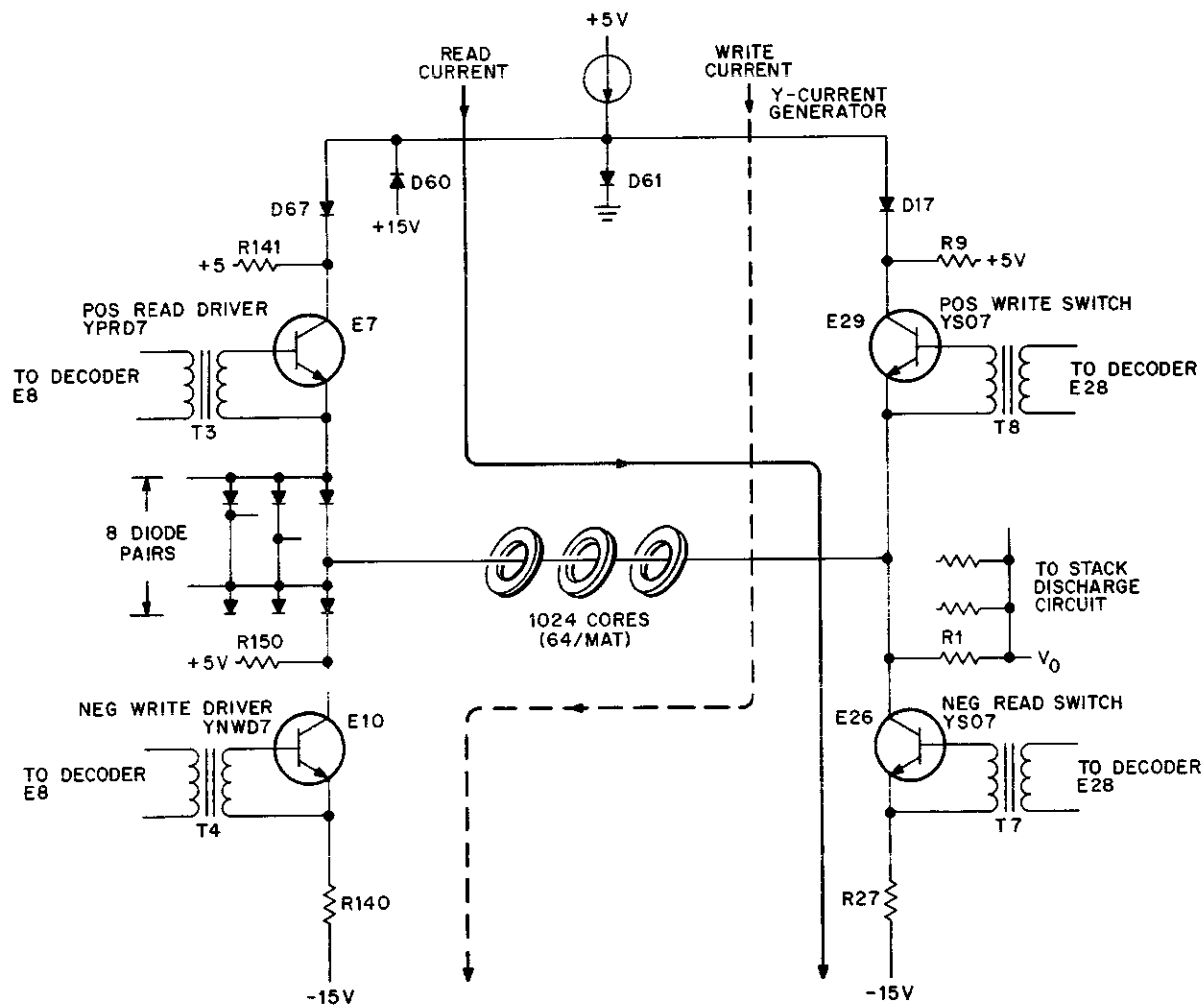


Figure 2-12 Y-Line Selection Stack Diode Matrix

YNWD2, and YNWD0) provide a positive voltage on the cathodes of their associated diodes (03W, 02W and 00W, respectively), which reverse biases them and prevents conduction. Write driver YNWD1, which has been selected, turns on and makes the cathode of diode 01W negative with respect to the anode that forward biases it. The diode conducts and allows current to flow to write driver YNWD1. A half-select current now flows through this line that links 64 cores per bit mat (1024 total for 16 mats).

Figure 2-13 is a simplified schematic of two pairs of switches and drivers interconnected with the core stack and current generator. Read/write switches YS07 and read/write drivers YD7 are used as examples. These switches and drivers are chosen for convenience. For a read or write operation, there are 64 switch/driver combinations available on each axis. For a read operation, decoder E8 selects positive read driver E7 via transformer T3; and decoder E28 selects negative read switch E26 via transformer T7. Both E7 and E28 are



11-1089

Figure 2-13 Typical Y-Line Read/Write Switches and Drivers

turned on when they are selected. E7 conducts and removes the reverse bias on diode D67, which allows current from the Y-current generator to flow through D67, E7, the associated matrix diode, and the cores on the selected line. After passing through the cores, the current flows through E26 and R27 to the -15V line. For a write operation, decoder E28 selects positive write switch E29 via transformer T8; and decoder E8 selects negative write drivers E10 via transformer T4. Both E29 and E10 are turned on. E29 conducts and removes the reverse bias on diode D17, which allows current from the Y-current to flow through D17, E29, and the cores in the opposite direction. After passing through the cores, the current flows through the associated matrix diode, E10, and R140 to the -15V

line. Read current flow is shown as a solid line: a broken line shows write current flow.

2.4.3.4 Word Address Decoding and Selection Sequence - This paragraph takes a specific word address through the decoding and X- and Y- line selection sequence.

The word address is 017772, and it is assumed that a specific memory bank has been selected. The binary equivalent of the address is shown below. A read operation is to be performed.

ADDRESS BITS A <17:0>																		
17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	Bit Position
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	Binary
	0			1			7			7			7			2		Octal

Bits A <13:01> are used to decode the word address. Bit A01 is sent to the device selector (drawing G110-0-1, sheet 2) and appears at word address flip-flop E11, pin 2 as A01 H (drawing G231-0-1, sheet 2). Bits A <12:02> are sent to the Unibus receivers, which are inputs to the associated word address flip-flops. Bit A13 is not used. The input to the Unibus receiver associated with this bit is connected directly to +5V through jumper J3 (for a 4K memory, J3 is in and J4 is out). Table 2-3 shows the state of bits A <13:01> and the decoding signals generated by the word address flip-flops after they are clocked.

The output signals from flip-flops A06, A12, and A13 are not used directly from the flip-flops; they are sent to gating logic (E22 and E25) and are ANDed with signal TSS H. In this case, only two out of a possible six signals are generated: A06H is low from E22, pin 12 and (A12H . A13L) L is low from E25, pin 8. These signals and the outputs from the other word address flip-flops are sent to the inputs of the type 8251 decoders to select the appropriate switches and drivers. READ L is an input to each 8251 decoder. A read operation is to be performed; therefore, READ L is low.

The decoders, switches, and drivers are shown in drawing G231-0-1 sheet 3 and 4. Using the decoding signals in Table 2-3 and the operating characteristics of the decoders, it is possible to determine which decoders have been selected for word address 017772. A decoder is selected only when its D3 input is low. In this case, the selected decoders are E34 and E46 for the X-line (drawing G231-0-1, sheet 3), and E23 and E8 for the Y-line

Table 2-3  
Word Address Decoding Signals

Address Bit	Unibus Receiver Input	Receiver Output	Flip-Flop State	Flip-Flop Output Signals
A01	L	H	set	A01H = H
A02	H	L	reset	A02H = L
A03	L	H	set	A03H = H, A03L = L
A04	L	H	set	A04H = H
A05	L	H	set	A05H = H
A06	L	H	set	A06H = H, A06L = L
A07	L	H	set	A07H = H
A08	L	H	set	A08H = H
A09	L	H	set	A09H = H, A09L = L
A10	L	H	set	A10H = H
A11	L	H	set	A11H = H
A12	L	H	set	A12H = H, A12L = L
A13	-	-	reset	A13H = L, A13L = H

(drawing G231-0-1, sheet 4). READ L is low and is sent to input D2 of each decoder; it selects read drivers and switches in this case. To verify this point, refer to the truth table and diagram in Figure 2-9. Decoder inputs D0 and D1 select the particular switch or driver as shown below.

- a. Decoder E34  
D1 is high, D0 is high: selects output 3 (pin 10), which is read switch XS07.
- b. Decoder E46  
D1 is high, D0 is high: selects output 3 (pin 10), which is read driver XPRD7.
- c. Decoder E23  
D1 is high, D0 is high: selects output 3 (pin 10), which is read switch YS03.
- d. Decoder E8  
D1 is low, D0 is high: selects output 1 (pin 12), which is read driver XPRD5.

The last step is to follow the outputs of the drivers and switches to the stack diode matrix (drawing H213-0-1, sheet 2). For the X-line, the circuit is from driver XPRD7 to diode junction E7-11, across termination 35 to switch XS07. For the Y-line, the circuit is from driver YPRD5 to diode junction E4-9, across termination 15 to switch YS03. The termination indicates the point on the stack printed circuit board where the X- or Y-line is soldered. Physically, the wire that is connected across the termination is strung through 64 cores per bit mat (total of 1024 cores in series for 16-bit memory).

## 2.5 READ/WRITE CURRENT GENERATION AND SENSING

In addition to the addressing and control logic, four functional units are involved in generating current to switch the cores and detect their state. The X- and Y-line current generators supply the drive current (via switches and drivers); the inhibit drivers allow 0s to be written during a write operation; the sense amplifiers detect 1s during a read operation; and the memory data register (MDR) temporarily stores data to be written or data that has been read from the memory. The following paragraphs describe each functional unit and their interrelationship.

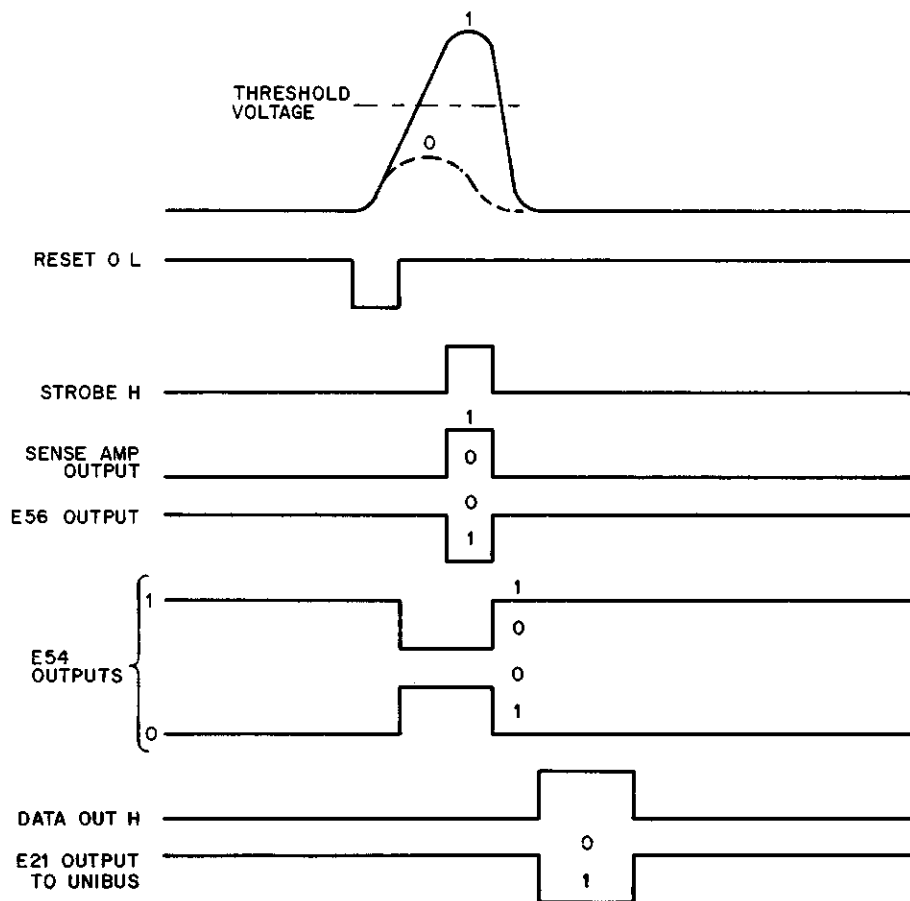
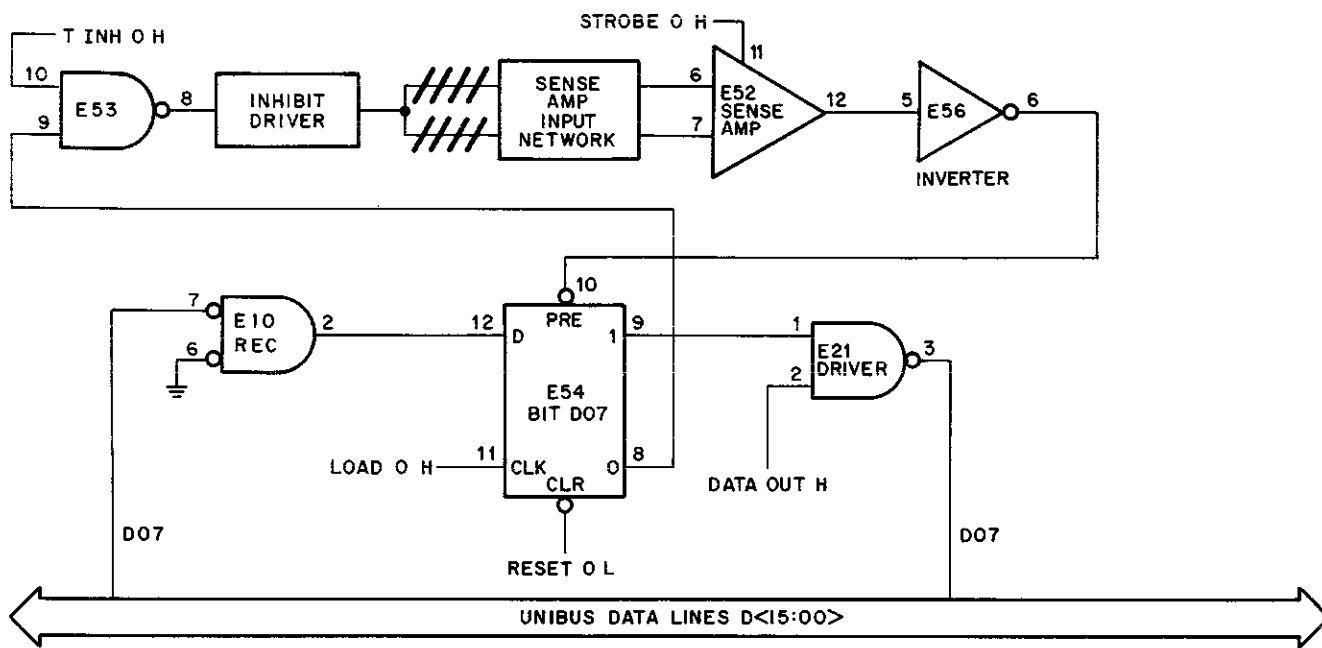
### 2.5.1 Read/Write Operations

The read/write operations are discussed in terms of the interrelation of the current generator, inhibit drivers, sense amplifiers, and memory data register. Details of operation of each functional unit are discussed in subsequent paragraphs. Several control signals are mentioned; however, details of their generation and timing are described in Paragraph 2.8.

For clarity, one data bit (D07) of the selected word is discussed and the text is referenced to Figure 2-14, which is a simplified block diagram. Detailed logic for the Memory Data Register (MDR), Unibus receivers and drivers, sense amplifiers, and inhibit drivers for all 16 data bits is shown on drawings G110-0-1, sheets 3 and 4.

During a read operation, half-select currents flow in the X- and Y-lines for the selected word in each bit mat. These currents flow opposite to the write currents; therefore, cores in the 1 state are switched to the 0 state and cores in the 0 state are unchanged. Switching the core from the 1 state to the 0 state induces a voltage pulse in the sense winding. This pulse is detected by sense amplifier E52 as a differential voltage on input pins 6 and 7 that exceeds the threshold reference voltage. This pulse is amplified and when STROBE





11-1100

Figure 2-14 Interconnection of Unibus, Data Register, Sense Amplifier, and Inhibit Driver

○ H is generated at pin 11, the output of sense amplifier E52 goes high. Just prior to the strobe signal, the control logic generates RESET ○ L, which clears (resets) flip-flop E54. The sense amplifier output is inverted by E56 and sent to the preset input (pin 10) of MDR flip-flop E54. A low on the preset input sets the flip-flop: its 1-output (pin 9) is a high and its 0-output (pin 8) is a low. The high from pin 9 of the flip-flop is sent to input pin 1 of the Unibus driver E21. The other input to this gate is the data out signal. When the control logic generates DATA OUT H, the output of E21 is low (logical 0 for memory logic and logical 1 for Unibus logic). This is the read-out of bit D07 and is sent to the requesting device via the Unibus. Timing diagrams for the sense operation are also shown in Figure 2-14.

The read operation is destructive: all cores at the specified location are now 0. The data that was read must be restored by a write operation, which immediately follows the read operation. Flip-flop E54 is still in the set state; therefore, its 0-output (pin 8), which is low, is sent to input pin 9 of NAND gate E53. The control logic generates the inhibit driver control signal TINHO H, which is the other input to gate E53. The gate is not asserted (pin 8 is low), and the inhibit driver is not turned on. With no inhibit current in the inhibit line to oppose the half-select Y line current, a 1 is written back into the appropriate cores.

In this example, if bit D07 is a 0 in core, it does not switch during the read operation and the output of sense amplifier E52 does not go high. Flip-flop E54 remains cleared (reset): its 1-output (pin 9) is low and its 0-output (pin 8) is high. When the control logic generates DATA OUT H, the output of Unibus driver E21 is high (logical 1 for memory logic and logical 0 for Unibus logic). The 0-output of flip-flop E54, which is high, is sent to NAND gate E53. During the subsequent write operation, TINHO H is generated, producing a low output signal at E53, pin 8 to activate the inhibit driver that in turn produces a current that opposes the Y line current and prevents a 1 from being written into this bit of the selected word.

The read/write operation that has been discussed is a read/restore operation (DATI). The requesting device wants to read a word from memory, and as an internal requirement, the memory must restore the word by writing it back into core. In this case, the MDR flip-flop are preset by the sense amplifier outputs when 1s are read from the core. The MDR flip-flop outputs are used in the subsequent write (restore) operation to control the inhibit drivers. If the requesting device wants to write a word into memory (DATO), it must load

the data into the MDR flip-flops. The requesting device then asserts the data on the Unibus, from which it is picked off via Unibus receivers. In this example, bit D07 is sent to pin 7 of Unibus receiver E10. Bit D07 is inverted by the receiver and sent to the D input (pin 12) of flip-flop E54. During the write operation, the control logic generates LOAD O H, which clocks the flip-flop. If the D input is high, E54 is set and its O-output is low. Control gate E53 is not asserted by TINHO H, and the inhibit driver is not turned on. A 1 is written into the selected core. If the D input is low, E54 is reset and its O-output is high. Control gate E53 is asserted by TINHO H, and the inhibit driver is turned on. A 0 is written into the selected core.

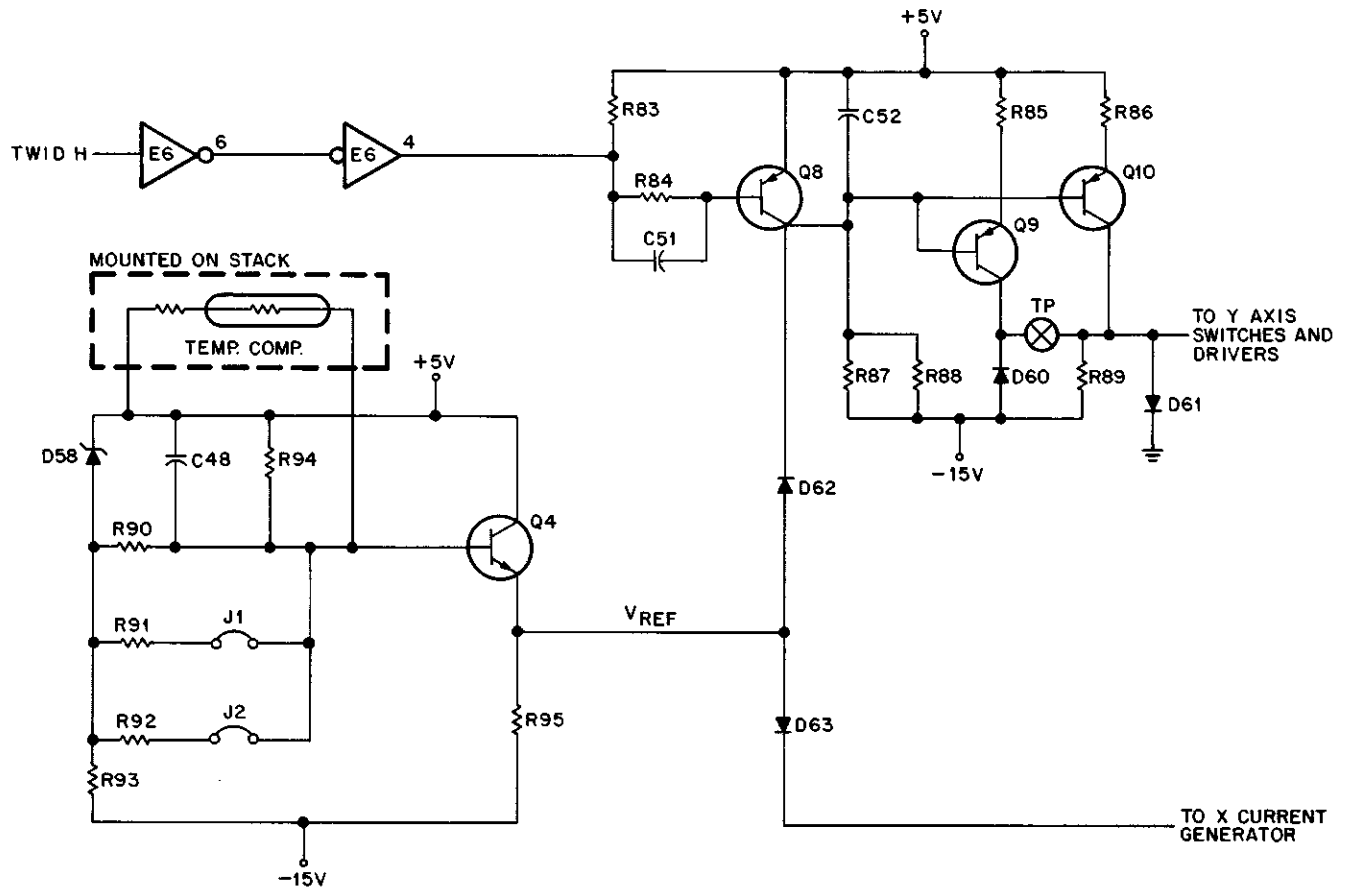
### 2.5.2 X- and Y-Current Generators

Two identical current generators are provided: one each for the X- and Y-drive lines. They generate the current pulses that are used during read and write operations to switch the cores. The current generators and associated reference voltage supply are shown in drawing G231-0-1, sheet 2. Refer to Figure 2-15, which shows the Y-current generator and reference voltage supply.

Optimum core switching requires repeatable current pulses of constant amplitude with a linear rise time. The current generator and reference voltage circuit provide current pulses that meet these requirements. The amplitude of the output current pulse is determined by the reference voltage circuit; the rise time is determined by an RC circuit in the current generator, and pulse duration is determined by the length of the triggering pulse TWID H.

During the quiescent state of the current generator, input transistor Q8 is on; its collector voltage is 4.7V; and it is connected to the cathode of diode D62, which reverse biases it. The anode of D62 is connected to the emitter of transistor Q4, which is the output of the reference voltage circuit. In this state, D62 blocks the output from the reference voltage circuit to the current generator. With Q8 on, both output transistors Q9 and Q10 are turned off; the current generator is off.

Operation of the current generator is triggered by a high TWID H signal from the control logic. TWID H is double inverted by two E6 inverters and sent to the base of Q8, which turns it off. When Q8 is cut off, capacitor C52 starts charging, which provides base drive to output transistors Q9 and Q10 and they begin to conduct. With Q8 off, its collector



11-1101

Figure 2-15 Y-Current Generator and Reference Voltage Supply

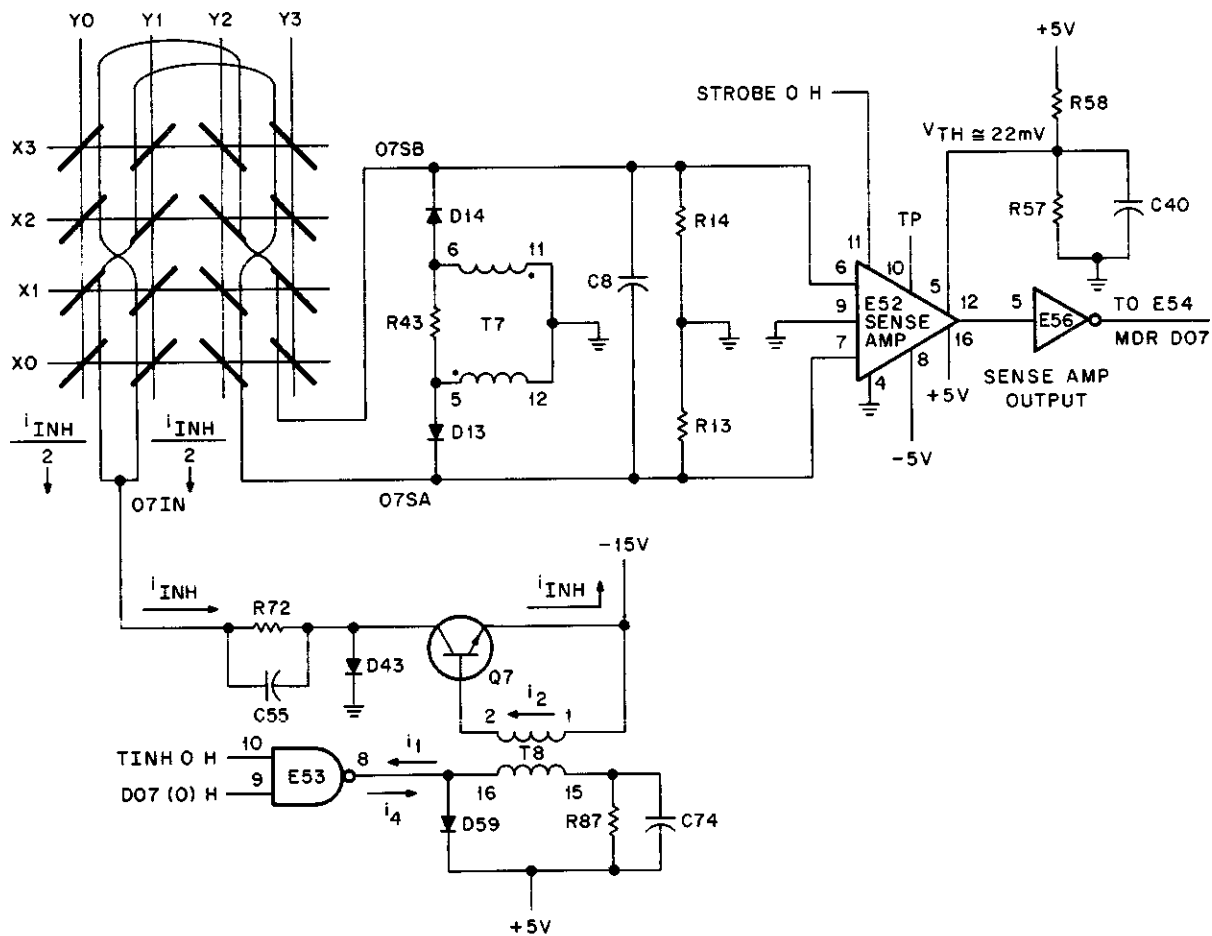
goes negative until it reaches the forward bias level of D62, which is the value of the reference voltage minus the voltage drop across D62. The rise time of the current pulse is determined by the time constant of C52, R87, and R88. The amplitude of the pulse is determined by the value of the reference voltage. When TWID H goes low again, the current generator is turned off and the output pulse is terminated.

A resistor network in the base circuit of Q4 (in the reference supply) is used to set the amplitude of the current generator to approximately 410 mA. The total resistance of parallel network R90, R91, and R92 is changed by the configuration of jumpers J1 and J2. The amplitude of the current generator output pulse is factory set as close as possible to 410 mA at 25°C. It should not be changed in the field.

The base circuit of Q4 is temperature compensated by a resistor and thermistor that are mounted on the stack. This ensures that the amplitude of the current generator output pulse remains within specified tolerances over a temperature range of 0°C to 50°C. This temperature compensation is approximately  $-0.8 \text{ mA}/^\circ\text{C}$ .

### 2.5.3 Inhibit Driver

A detailed schematic of the inhibit driver for bit D07 is shown in Figure 2-16: it is typical of all 16 inhibit drivers (drawing G110-0-1, sheets 3 and 4).



11-1102

Figure 2-16 Sense Amplifier and Inhibit Driver

When the inhibit driver is off, none of the currents shown in the schematic are flowing. Transistor Q7 is held off by the negative voltage on its base. The output of NAND gate E53 goes low (ground) when this inhibit driver is selected. Current  $i_1$  flows into the output circuit of E53 from the +5V supply via resistor R87 and the primary winding (terminals 15 and 16) of transformer T8. An equal current is induced in the base-emitter circuit of Q7, which is connected to the transformer secondary winding (terminals 1 and 2). This base current overcomes the reverse bias voltage and turns on Q7. Current  $i_1$  and therefore induced-current  $i_2$  are determined by resistor R87 and the reflected base-emitter voltage  $V_{be}$  of Q7. When Q7 is turned on, current flows from ground through Balun transformer T7, isolation diodes D13 and D14, and the sense/inhibit winding to the common inhibit terminal (07IN). The Balun transformer balances the two inhibit half-currents. At terminal 07IN, the full inhibit current flows through resistor R7 and Q7 to -15V. The value for inhibit current is calculated as follows:

$$i_{inh} \approx \frac{15V - V_{ce\ sat\ Q7} - V_{be\ diodes}}{R_{72} + R_{core\ mat}} \\ \approx \frac{15 - 0.8 - 1.2}{17.5} = 740\text{ mA}$$

Each leg of sense/inhibit sees half the inhibit current: approximately 370 mA. Capacitor C55 decreases the rise time of the current.

The inhibit driver is turned off when the output (pin 8) of gate E53 goes from low to high. At turn-off time, the back caused by the stack inductive reactance tries to drive the collector of Q7 highly positive; however, diode D43 clamps this voltage to ground. When the output of E53 goes high (approximately +3.2V), its output pull-up transistor (an integral part of the gate circuit) tries to drive the turn-off current  $i_4$  in the opposite direction through the transformer primary winding. An equal current induced in the secondary winding removes the forward bias from the base of Q7 and turns it off. With Q7 off, all dynamic current flow ceases in the circuit and the negative voltage on the base of Q7 keeps the circuit turned off until the output of gate E53 goes low again.

Capacitor C74 allows the gate to pump reverse current  $i_4$  into the transformer primary; it also helps to decrease the turn-on time of Q7. Diode D59 prevents reverse breakdown of the emitter junction of Q7.

#### 2.5.4 Sense Amplifier

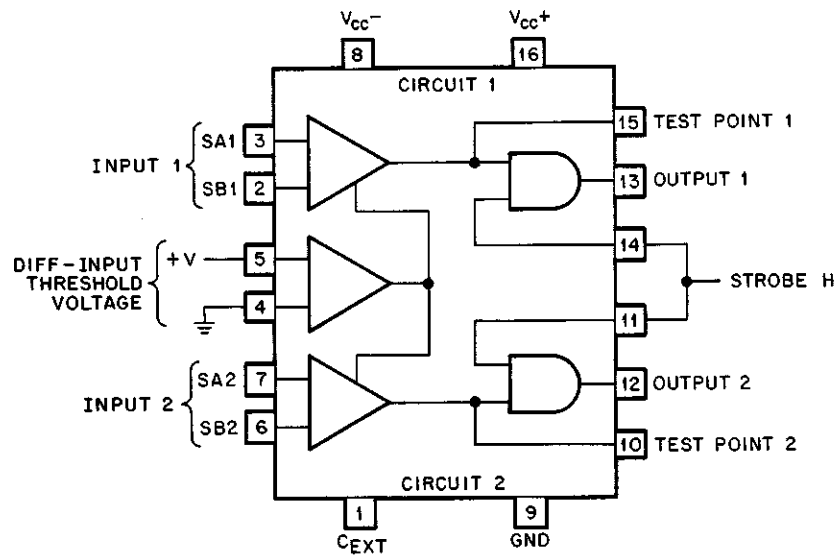
A detailed schematic of the sense amplifier circuit for bit D07 is shown in Figure 2-16; thus circuit is typical of all 16 sense amplifier circuits (drawing G110-0-1, sheets 3 and 4). The circuit consists of the sense amplifier, terminating network for the sense/inhibit winding, and threshold voltage network.

The sense amplifier input (E52, pin 6 and 7) is across the sense/inhibit winding (points 07SB and 07SA). Resistors R13 and R14 are matched to terminate the sense/inhibit line in the desired impedance. Practically speaking, during the sense operation, the inhibit driver connection is a short circuit between the two sides of the sense/inhibit line and an open circuit through the driver transistor Q7. The effect of the inhibit driver circuit, Balun transformer T7, and isolation diodes D13 and D14 can be ignored during the sense operation, because the diodes are reverse biased.

Sense amplifier E52 is one half of a dual IC package (type 7528). A simplified block diagram of the package is shown in Figure 2-17. The two identical circuits are marked 1 and 2. Each one consists of a preamplifier and sense amplifier. The output of the preamplifier is available as a test point to observe the amplified core signal and to facilitate accurate strobe timing. Both circuits share a reference voltage (or threshold voltage) amplifier (pins 4 and 5). In this application, pin 4 is grounded and a positive threshold voltage of approximately 20 mV is supplied to pin 5. This voltage is obtained from the +5V supply through resistor voltage divider R57 and R58; C40 is a bypass capacitor. Operation of the sense amplifier is discussed in Paragraph 2.5.1.

#### 2.5.5 Memory Data Register

The Memory Data Register (MDR) is a 16-bit flip-flop register that is used to store a word after it is read out of the memory; or to store a word from the Unibus prior to its being written into the memory. The MDR is composed of eight 74H74 dual high-speed D-type flip-flops: bits D00-D07 are shown in drawing G110-0-1, sheet 3 and are identified as E54, E57, E60, and E63; bits D08-D15 are shown in drawing G110-0-1, sheet 4 and are identified as E42, E45, E48, and E51.



11-1103

Figure 2-17 Type 7528 Dual Sense Amplifiers With Pre-amplifier Test Points

At the start of a memory operation, the MDR is cleared directly via the Clear input (pin 1 or pin 13) of each flip-flop: the clear signal is RESET 0 L for bits D00-D07 and RESET 1 L for bits D08-D15.

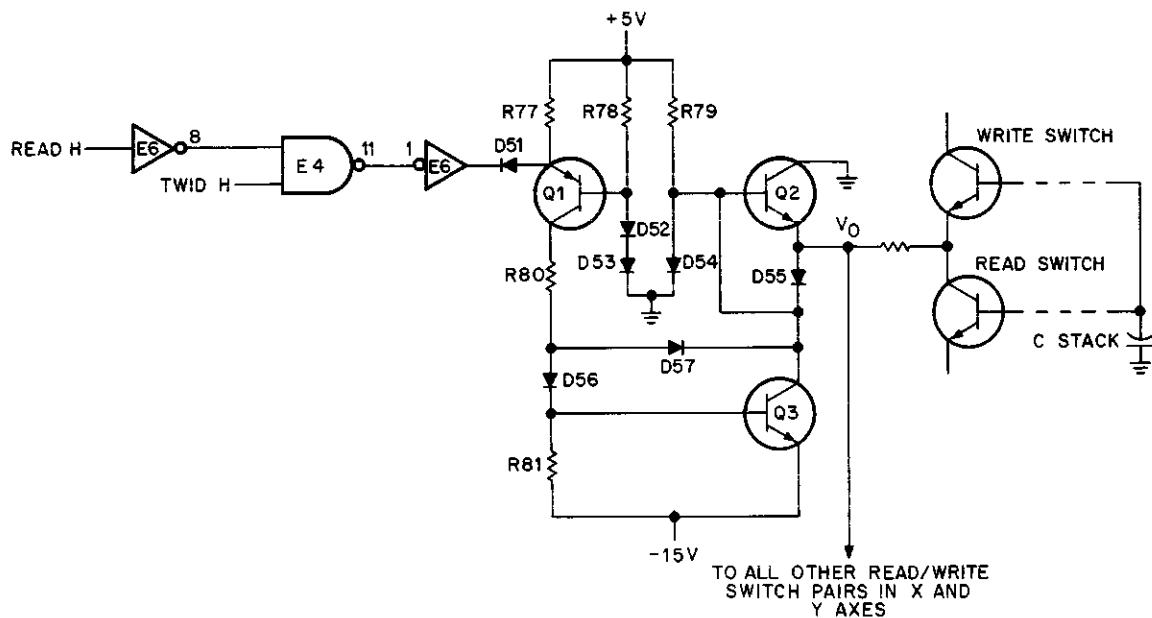
The operation of the MDR during a read/restore operation (DAT1) and a write operation (DAT0) is discussed in Paragraph 2.5.1.

## 2.6 STACK DISCHARGE CIRCUIT

The stack discharge circuit assists the stack capacitance in recovering and shortens the rise time of the stack current. It also reduces unwanted currents in the seven unselected lines associated with the selected driver.

Figure 2-18 shows the stack discharge circuit. Its output is taken from the emitter of transistor Q2 and goes to the junction of each X- and Y- read/write switch pair via a resistor. This common interconnection is labeled  $V_0$ . It is desired that  $V_0 \cong 0V$  (ground) during a read operation; and  $V_0 \cong -15V$  during a write operation. The effective stack capacitance associated with each line is shown as  $C_{stack}$ .





11-1104

Figure 2-18 Stack Discharge Circuit

During a write operation, READ H is low; it is inverted and ANDed with TWID H at NAND gate E4. The low output (pin 11) of E4 is inverted by E6 and sent to the cathode of diode D51, which reverse biases it. The emitter becomes more positive, overcomes the constant positive base bias, and turns on transistor Q1. When Q1 conducts, it provides base drive for Q3, which also turns on. When Q3 conducts, it reduces the base drive on Q2 and it turns off. The emitter voltage of Q2 goes to approximately  $-14V$ , which is  $V_0$  on the switch node for the stack. Diode D57 prevents hard saturation of Q3; diode D55 holds Q2 off. During a write operation,  $V_0 = -14V$  and the stack discharge circuit is considered to be turned on (input transistor Q1 is on).

During a read operation, READ H is high: it is inverted and ANDed with TWID H at NAND gate E4. The gate is not asserted and its output (pin 11) is high. This signal is inverted by E6 and sent to the cathode of diode D51, which forward biases it. The voltage on the emitter of Q1 produced by the current through R77 and D51 is not enough to overcome the constant positive bias and Q1 is turned off. With Q1 off, Q3 loses its base drive and turns off. Now, D55 cannot hold Q2 off. As long as the stack capacitance is charged negatively, base current exists for Q2 and it remains on. The stack capacitance now charges in the positive direction until it reaches ground potential. During a read

operation,  $V_0 \cong 0V$  and the stack discharge circuit is considered to be off (input transistor Q1 is off).

Figure 2-13 shows how the stack discharge circuit reduces unwanted currents on the seven unselected lines associated with the selected driver.

During a read operation, the stack discharge circuit is on the  $V_0 = 0V$ . The current generator drives the read driver node of the stack towards ground; the current generator output is clamped to ground by diode D61. The anodes of the eight read diodes are at ground. The stack discharge circuit is on and the cathodes of the seven unselected diodes are also at ground, which back biases them off. The read switch pulls the cathode of the selected line towards  $-14V$ , which forward biases it and allows conduction through the diode. Current flows only through the selected line. Reverse biasing of the diodes in the unselected lines prevents current from flowing between the unselected nodes and the selected read driver. The stack discharge circuit performs the same task during the write operation by back biasing the anodes of the diodes in the unselected lines with  $-14V$ .

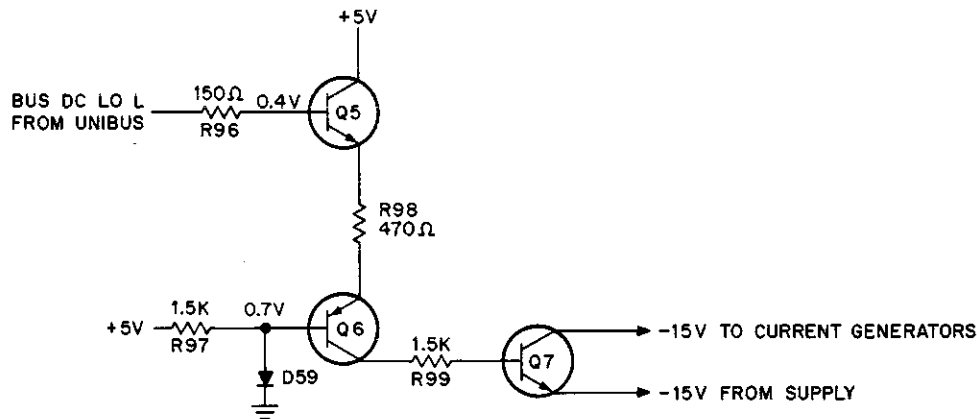
## 2.7 DC LO CIRCUIT

A circuit on the G231 Driver Module (drawing G231-0-1, sheet 1) opens the 15V supply line to the current generators when power is interrupted to the power supply. When power is interrupted, the +5V supply is lost and the operation of all logic is indeterminate. In this state, it is necessary to cut off the -15V supply to the X- and Y-line current generators to prevent them from destroying stored data. The circuit that performs the -15V cut off is called the DC LO circuit (Figure 2-19).

The -15V supply for the X- and Y-line current generators passes through transistor Q7 in the DC LO circuit. Q7 must be turned on for the -15V to reach the current generators. The circuit monitors BUS DC LO L from the power supply via the Unibus. This signal is sent to the base of transistor Q5. When power is on, BUS DC LO L is high (not asserted).

The voltage across R96 forward biases Q5 and it turns on which turns on Q6. The conduction through Q5 and Q6 forward biases Q7 which turns it on. The -15V flows through Q7 to the X- and Y-line current generators.

When power is interrupted, BUS DC LO L goes low (asserted). Q5 is now reverse biased and it turns off which turns off Q6. With Q5 and Q6 off, Q7 is also turned off which



11-1105

Figure 2-19 DC LO Circuit, Schematic Diagram

opens the -15V line to the current generators. This circuit still functions when BUS DC LO L is asserted even if the +5 supply drops to zero.

## 2.8 OPERATING MODE SELECTION LOGIC

When the memory is addressed by the master device, one of four bus transactions is selected. The transaction (or operation) selected is determined by the states of control bits C01 and C00 and address bit A00 as placed on the Unibus by the master device. Table 2-4 shows the states of these bits for each transaction.

The logic that decodes the mode and byte control bits is shown in drawing G110-0-1, sheet 2; it appears at the bottom of the sheet and is identified as the byte masking logic. Bits BUS C01, BUS C00, and BUS A00 are taken from the Unibus to three E29 receivers. One input of each gate associated with C01 and C00 is connected to the output of the PROTECT LOW gate (E29 pin 3). Both inputs to this gate are tied to +5V so that its output is always low. For troubleshooting purposes, a jumper (W11) can be installed that makes the gate output high which allows only DAT1 operations to be performed regardless of the states of bits C01 and C00. This jumper hardwires the memory as a read only device.

The outputs of the three E29 receivers (C01, C00, and A00) are sent to the byte masking logic to generate LOAD 0 H and LOAD 1 H and to qualify a group of gates, which are enabled by control signals to generate RESET 0 L, RESET 1 L, STROBE 0 H, STROBE 1 H,

Table 2-4  
Selection of Bus Transactions

Transaction	Mnemonic	Mode Control		Byte Control A00	Function
		C <01:00 >	Octal		
Data In	DATI	00	0	X	Data from memory to master. Memory performs operations.
Data In, Pause	DATIP	01	1	X	Data from memory to master. Restore operation is inhibited. Must be followed by DATO or DATOB: Read operation is inhibited.
Data Out	DATO	10	2	X	Data from master to memory (words).
Data Out, High Byte	DATOB	11	3	1	Data from master to memory. High byte on data lines D <15:08 >.
Data Out, Low Byte	DATOB	11	3	0	Data from master to memory. Low byte on data lines D <07:00 >.

and DATA OUT H. The logic also conditions the D-input of the PAUSE flip-flop (E4, pin 12) to allow it to be set or reset. It also applies conditioning signals to the wired-AND that provides the clocking signal to the Slave Sync (SSYN) flip-flop. The PAUSE flip-flop and the SSYN flip-flop are part of the control logic.

The signals generated for each bus transaction are shown in Table 2-5. The memory operational sequences are discussed in subsequent paragraphs. To avoid confusion in interpreting the transactions listed in Table 2-5, the purpose of the PAUSE flip-flop is discussed briefly. During DATIP, the PAUSE flip-flop is set during the read operation, which inhibits the restore (write) operation. The DATIP must be followed by a DATO or DATOB on the same address. The DATO or DATOB that follows a DATIP is shorter than a standard DATO or DATOB because the initial read operation is eliminated. In Table 2-5, the suffix PAUSE L identifies the standard transactions; the suffix PAUSE H identifies the DATO and DATOB transactions that must follow a DATIP.

Table 2-5

## Generation of Memory Operating Signals

Mode	Byte Control A00	Mode Control		State of PAUSE Flip-Flop	Signals Generated							Operation Sequence
		C01	C00		STROBE 0	STROBE 1	RESET 0	RESET 1	LOAD 0	LOAD 1	DATA OUT H	
DATI	X	0	0	Reset	X	X	X	X			X	Read-Restore.
DATIP	X	0	1	Reset-Set	X	X	X	X			X	Read-Pause. Restore inhibited by PAUSE flip-flop.
DATO PAUSE L	X	1	0	Reset					X	X		Clear-Write.
DATO PAUSE H	X	1	0	Set					X	X		Write. Must follow DATIP.
DATOB PAUSE L	0	1	1	Reset		X		X	X			Clear-Write selected byte 0. Clear-Restore non-selected byte 1.
DATOB PAUSE H	0	1	1	Set		X		X	X			Write selected byte 0. Restore non-selected byte 1. Must follow DATIP.
DATOB PAUSE L	1	1	1	Reset	X		X			X		Clear-write selected byte 1. Clear-restore non-selected byte 0.
DATOB PAUSE H	1	1	1	Set	X		X			X		Write selected byte 1. Restore non-selected byte 0. Must follow DATIP.

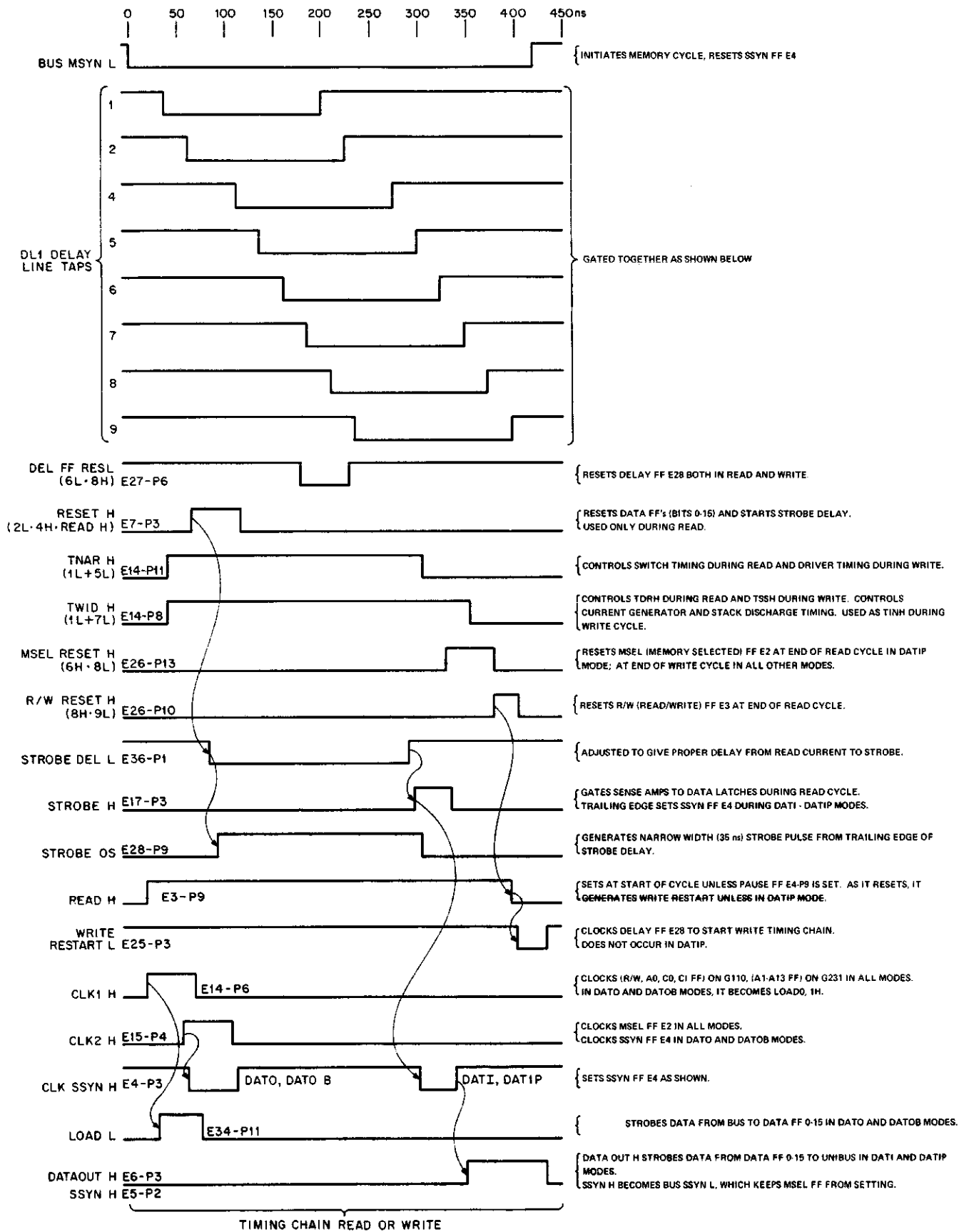
## 2.9 CONTROL LOGIC

The control logic generates the precisely timed signals that initiate and stop the memory operations that are requested as a result of the decoding of the bus transaction. The heart of the control logic is the delay line timing circuit. For better understanding, the timing circuit, slave sync circuit, pause/write restart circuit, and strobe generating circuit are described separately. Each bus transaction is also discussed in detail. The discussion is to the detailed logic level but the signals are not traced through each component. The text is referenced to logic drawing G110-0-1, sheet 2 and the timing diagrams in drawing MM11-L-3.

### 2.9.1 Timing Circuit

The heart of the memory control logic is the timing circuit. When activated, it generates a series of precisely timed signals that control memory operation. The major component of the timing circuit is a delay line (DL1) with multiple 25-ns taps (drawing C110-0-1, sheet 2). The delay line outputs are gated to produce the control signals. Figure 2-20 shows the timing of the delay line outputs and the timing of the control signals obtained by gating these outputs. A brief statement of the function of each control signals is included. Absolute timing is obtained from the engineering timing diagram (drawing MM11-L-3). The discussion is referenced to Figure 2-20 and the control logic drawing G110-0-1.

When the system is turned on, the processor asserts BUS INIT L on the Unibus. This initializing signal is sent to pins 6 and 7 of bus receiver E7. It is inverted by E7 to produce a high, which is sent to pins 9 and 10 of the memory select reset (MSEL RESET) gate E16. The output (pin 8) of E16 is low and is used to clear (reset) MSEL flip-flop E2 via the 100-ns delay DL3. The output of E7 is also inverted by E18 to provide a low that clears read/write (R/W) flip-flop E3. The output of E7 is also inverted by E15 to provide a low that clears PAUSE flip-flop E4. The low output of E15 is double inverted by two E38 gates to clear the DEL flip-flop E28. The master places the address, mode control state, and data (if required) on the Unibus. The device address is decoded and DSEL H is generated and sent to pin 13 of E1, which is one of four input signals (pins 10, 11, 12, and 13). Pin 11 is high via the 0-output of MSEL flip-flop E2. Ssyn flip-flop E4 is preset, making pin 10 of E1 high via its 1-output (pin 5). When the master asserts BUS MSYN L to bus receiver E23, pin 12 of E1 is high also. The output of E1 (pin 8) is low and is sent to pin 13 of E5, pins 4 and 5 of E14, and pin 1 of delay line DL2. E14 inverts the low from E1 to start the positive CLK 1 H pulse. DL2 provides a 30-ns delay for the low signal



11-1108

Figure 2-20 Basic Timing and Control Signal Functions

from E1, which is inverted by E15 to start the positive CLK 2 H pulse. The output (pin 3) of DL2 is also sent to the preset input (pin 4) of MSEL flip-flop E2, and pin 6 goes low which in turn is fed back to pin 10 of E1 to disable it. The output (pin 8) of E1 is now high, and this signal terminates both clock pulses (CLK 1 H and CLK 2 H) via gates E14 and E15. These pulses are approximately 50-ns wide.

Gate E5 also inverts the low from E1 because pin 12 (WRITE RESTART L) of E5 is high. The positive transition at the output (pin 11) of E5 clocks delay (DEL) flip-flop E28 which sets it. Pin 5 of E28 is high and is connected to pins 1 and 2 of DL1 driver gate E34. The low from the E34 output (pin 3) is the input to delay line DL1. This signal remains low for approximately 225-ns until DEL flip-flop E28 is cleared by DELAY FF RESET L. This provides a negative pulse that propagates through the delay line and can be picked off at 25-ns intervals.

DL1 taps 1, 2, 4, 5, 6, 7, 8, and 9 are used to generate control signals. Figure 2-20 depicts each control signal and relates it to logic drawing G110-0-1, sheet 2.

#### DELAY FF RESET

Tap 6L is inverted by E15 and sent to pins 3 and 5 of 3-input NAND gate E27; the third input (pin 4) is tap 8H. The output (pin 6) of E27 clears the DEL flip-flop E28; however, it is ORed with INIT L in gate E28 (pins 9 and 10) and inverted by E38, pin 11 so that either (6L \* 8H) or BUS INIT L can produce DELAY FF RESET L, which clears E28 via its clear input (pin 1). This signal is generated in both read and write operations.

#### RESET H

Tap 2L, tap 4H, and signal READ H are gated to generate RESET H, which triggers the strobe delay circuit and generates RESET 0 L and RESET 1 L during the read operation only. Tap 4H and READ H (high during read operation) are ANDed at pins 10 and 9 of E17. The low output of E17 is ANDed with tap 2L in gate E7. The high output (pin 3) is RESET H.

#### TWID H and TNAR H

The 0-output of DEL flip-flop E28 is ORed with tap 5L and tap 7L in separate gates (E14) to produce signals TWID H and TNAR H. Tap 5L is sent to pin 13 of E14; the other input



to this gate (pin 12) is from the 0-output of DEL flip-flop E28. Tap 7L is sent to pin 10 of another E14 gate; pin 9 of this gate is also connected to the 0-output of DEL flip-flop E28. These gates are 2-input NAND gates (type 7437); however, they are shown as logically equivalent negative input OR gates, because it is desired to have them asserted high (logical 1) when TWID H or TNAR H is asserted.

At the start of a read or write cycle, just before E28 is set, TNAR and TWID are low because both inputs to each gate are high. E28 is set and pins 12 and 9 of E14 go low; TNAR and TWID are both high, which starts the positive TNAR and TWID pulses simultaneously. When taps 5 and 7 go low (E28 is still set), TNAR and TWID remain high. At the end of the read or write cycle, E28 is cleared (taps 5 and 7 are still low) and TNAR and TWID still remain high. When tap 5 is high again, TNAR goes low because both inputs (pins 12 and 13) of E14 are high. This terminates the positive TNAR pulse. Approximately 50-ns later, tap 7 is high again and TWID goes low, terminating the positive TWID pulse. In summary, TNAR H and TWID H are started together by setting DEL flip-flop E28 before taps 5 and 7 are low; TNAR H and TWID H are not affected when taps 5 and 7 go low. Signals TNAR H and TWID H are terminated when taps 5 and 7 return high. The intervening clearing of E28 does not affect TNAR H or TWID H.

Signals TNAR H and TWID H provide various control functions related to the operation of the switches, drivers, current generators, inhibit drivers, and stack discharge circuit. At this point, the discussion digresses to follow TNAR H and TWID H through some additional logic in order to understand their functions. The logic is spread throughout several engineering drawings. To simplify the discussion, all the logic is shown in Figure 2-21.

Signal TWID H is ANDed with the 0-output (pin 8) of R/W flip-flop E3 at pins 9 and 10 of gate E25. With TWID H high, E25 is asserted only when E3, pin 8 is high; this occurs only during a write operation. The output (pin 8) of E25 is inverted by E14 to procure TINH 0 H and TINH 1 H. The output of E14 is physically divided into two paths: TINH 0 H activates the inhibit drivers for bits D <07:00>, and TINH 1 H activates the inhibit drivers for bits D <15:08>. These signals do not leave the control module because the inhibit drivers are on this module also.

Signals TWID H and TNAR H leave the control module (G110) and are sent to the drive module (G231). TWID H is sent to pin 4 of E2R, and TNAR H is sent to pin 2 of E2W. Gates E2 and E4 are marked W and R in Figure 2-21 to show their association with write or read operations. READ H is sent from the 1-output (pin 9) of R/W flip-flop E3 on the

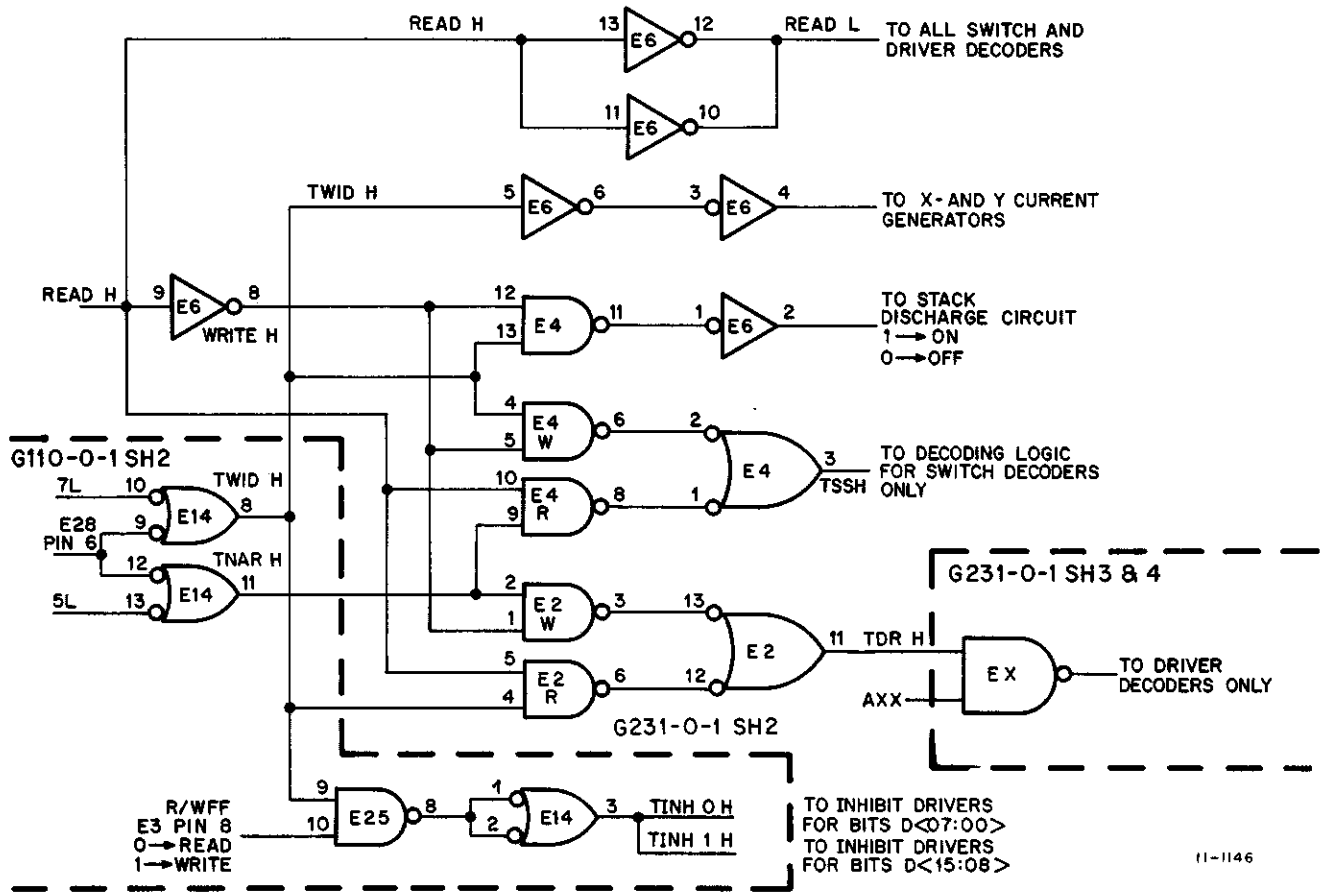


Figure 2-21 TWID H and TNAR H Control Logic

control module to pin 9 of inverter E6 on the driver module. READ H is high during a read operation and low during a write operation. Assume that a read operation is selected. READ H is high at pin 9 of E6 and is sent to pin 5 of E2R to be ANDed with TWID H. This gate is asserted and its low output is sent to pin 12 of negative-output NOR gate E2, which inverts it to produce TDR H. This signal is a decoding input for the memory read/write drivers only. Gate E2W is not asserted because WRITE H, which is the inversion of READ H, is low. Therefore, TWID H controls decoding signal TDR H during a read operation. During a write operation, READ H is low and WRITE H is high. Signal TDR H is asserted via the output of gate E2W, using the ANDing of WRITE H and TNAR H. Decoding signal TDR H is controlled by TNAR H during a write operation.

A similar logic network is used to control signal TSS H, which enables six decoding signals that are in turn used to control memory read/write switches only. When gates E4W, E4R, and E4 are used: TSS H is generated at the output (pin 3) of E4. During a read operation, TNAR H controls enabling signal TSS H; signal TWID H controls TSS H during a write operation.

TWID H controls the operation of the X- and Y-current generators. During read and write operations, when TWID H is high, the signal is double inverted by two E6 inverters to turn both current generators on.

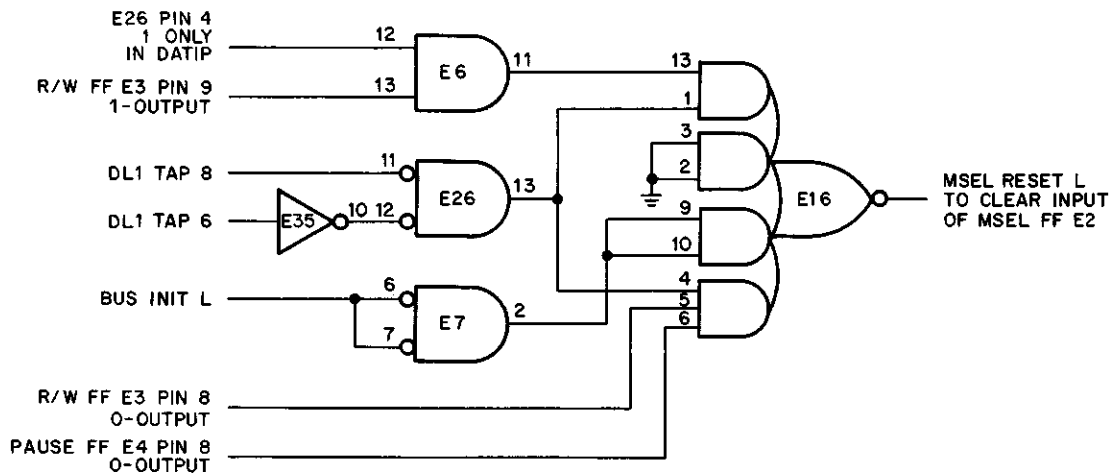
The TWID H signal also controls the operation of the stack discharge circuit. It is ANDed with WRITE H at pins 13 and 12 of NAND gate E4. The output (pin 11) of E4 is inverted by E6 to control the stack discharge circuit. This circuit is considered to be turned on when the output (pin 2) of E6 is high. This occurs during a write operation when TWID H and WRITE H both high.

Although not part of the timing circuit, Figure 2-21 shows READ H inverted by two E6 inverters to become READ L, which is a decoding input to all type 8251 decoders for the memory switches and drivers. During a read operation, READ H is high and READ L is low, which selects only read switches and drivers; conversely, during a write operation, READ L is high which selects only write switches and drivers (Paragraph 2.4.3.2).

## MSEL RESET

The memory select (MSEL) flip-flop E2 is cleared (reset) at the end of a read operation in DATIP mode and at the end of a write operation in all other modes (DATI, DATO, and DATOB) by signal MSEL RESET L. The MSEL RESET L signal is generated at the output (pin 8) of gate E16 (a type 74H53 2-2-2-3 input AND-OR-invert gate). Three of its four AND inputs are used to facilitate the various methods used in generating MSEL RESET L (Figure 2-22).

When the system is turned on, the processor asserts BUS INIT L on the Unibus. The output of bus receiver E7 is high; this high output is sent to pins 9 and 10 of E16 to generate MSEL RESET L at its output (pin 8). The MSEL RESET L signal is passed through a 100-ns delay line (DL3) to the clear input (pin 1) of MSEL flip-flop E2, which directly clears (resets) it. All memory operations start with E2 cleared; however, this flip-flop is set approximately 75-ns after the processor asserts BUS MSYN L. It remains set until it is cleared by one of



11-1145

Figure 2-22 Generation of MSEL RESET L

the following operations.

In the DATIP mode, pin 12 of AND gate E6 is high; in all other modes, it is low, disqualifying E6. A read operation is performed in DATIP, and R/W flip-flop E3 is set. The 1-output of E3 is sent to pin 13 of E6. At this time, pin 13 is high and a high is generated at the output (pin 11) of E6. This AND input is qualified when pin 1 is also high, which occurs when DL1 tap 6 is high and DL1 tap 8 is low. Tap 6H is inverted by E35 and sent to pin 12 of E26. Tap 8L is sent directly to the other input (pin 11) and the gate is asserted; this gate sends a high to pin 1 of E16, which generates MSEL RESET L at the output (pin 8) of E16. This low signal clears MSEL flip-flop E2 at the end of the read operation (timed by 6H and 8L).

In all other modes (DATI, DATO, and DATOB), signal MSEL RESET L is generated at the end of the write operation (except DATO or DATOB following a DATIP). The R/W flip-flop is set, making its 0-output (pin 8) low which disqualifies the 3-input (pin 4, 5, and 6) AND gate in E16.

Taps 6H and 8L cannot qualify this AND input or the other AND input (pins 1 and 13) because the memory is not in the DATIP mode. Therefore, the read operation is completed and MSEL RESET L is not generated. The write operation is now started and the R/W flip-flop is cleared, which puts a high on input 5 of E16. Input 6 is high because the PAUSE flip-flop is reset (pin 8 is a 1). Now, when tap 6 is high and tap 8 is low, input 4 of E16

is high. This generates signal MSEL RESET L to clear MSEL flip-flop E2 at the end of the write operation.

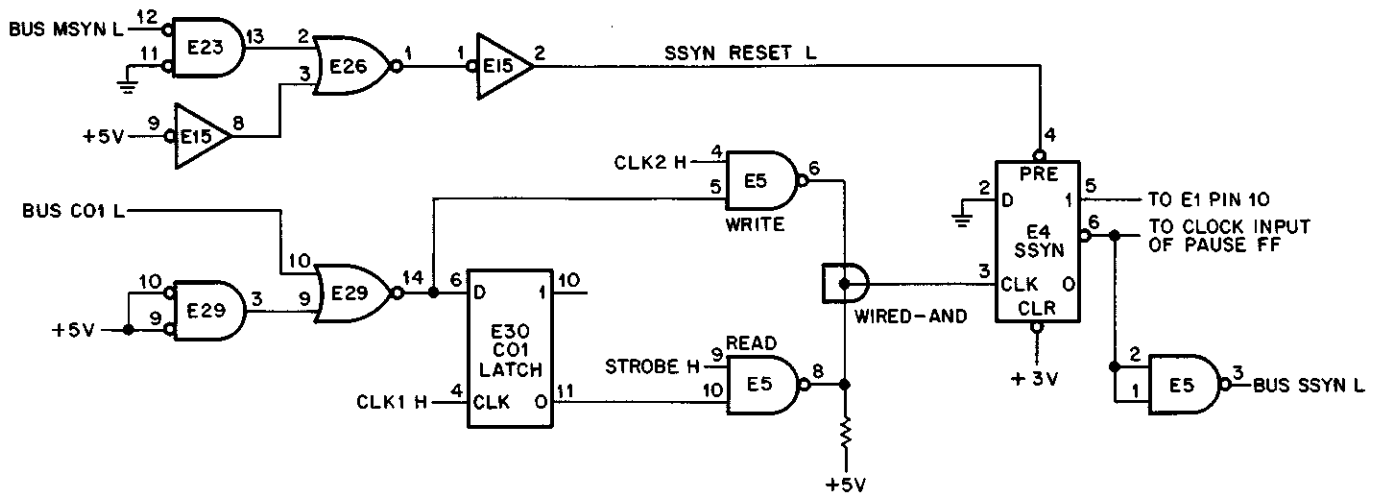
## R/W RESET

The timing for the generation of the signal to clear (reset) R/W flip-flop E3 is obtained from taps 8 and 9 of DL1. Tap 9 is sent directly to pin 8 of E26. Tap 8 is inverted by E35 and sent to pin 9 of E26. When tap 9 is low and tap 8 is high, E26 is asserted (output pin 10 is high). This signal is sometimes called R/W RESET H. It is ANDed with READ H at pins 2 and 1 of NAND gate E18 to generate R/W RESET L. When this signal is a low, it directly resets R/W flip-flop E3 via its clear input (pin 13). READ H is high when the R/W flip-flop is set because it comes from the 1-output (pin 9). The remainder of the control signals shown in Figure 2-20 are discussed in the circuit descriptions contained in Paragraph 2.9.2 Slave Sync Circuit, Paragraph 2.9.3 Pause/Write Restart Circuit, and Paragraph 2.9.4 Strobe Generating Circuit.

### 2.9.2 Slave Synchronization (SSYN) Circuit

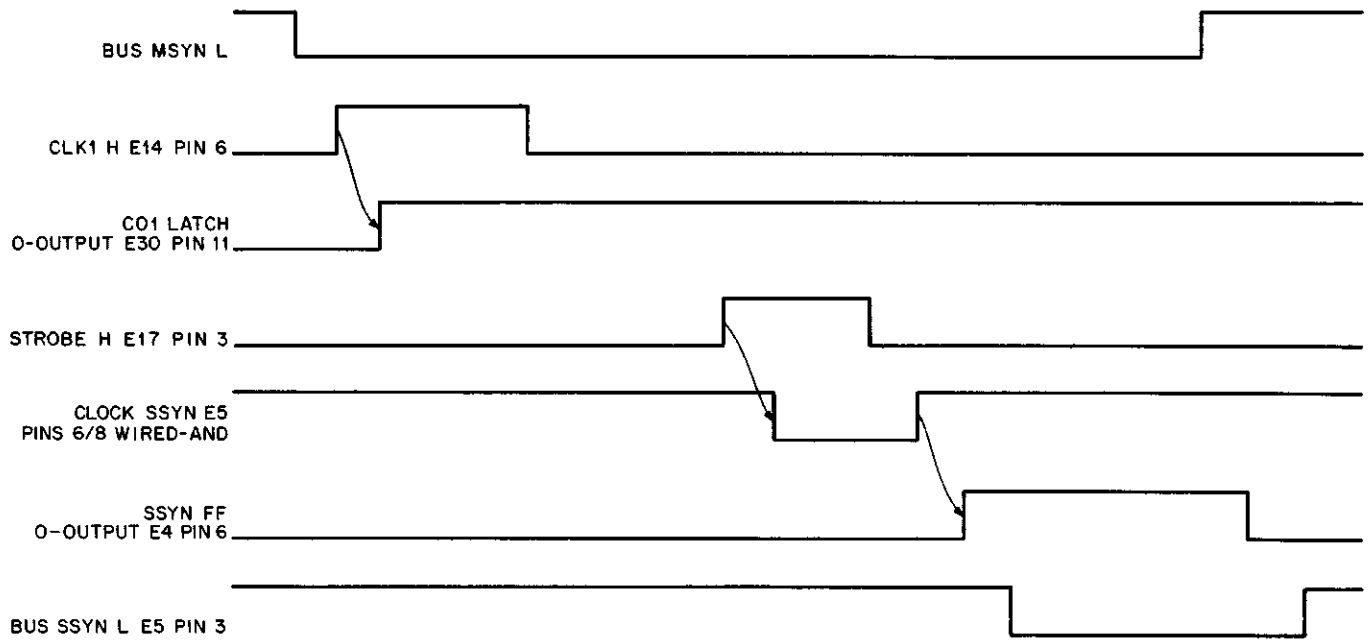
Slave synchronization (SSYN) is the slave device's response to the master: usually a response to master synchronization (MSYN). The master places address information, mode control information, and data (if a DATO or DATOB is selected) on the Unibus. It then asserts BUS MSYN L but only if BUS SSYN L from the slave is cleared, which indicates that the slave can participate in a bus transaction. The slave asserts BUS SSYN L when it has data to send (DATI or DATIP) or when it has received data (DATO or DATOB). The master receives BUS SSYN L in both cases and clears BUS MSYN L. When the slave receives the cleared BUS MSYN L, it clears BUS SSYN L which frees the bus. This brief statement of the SSYN/MSYN interaction is necessary to understand the operation of the memory SSYN circuit. Details of the SSYN/MSYN interaction during all bus transactions can be found in the PDP-11 Unibus Interface Manual, DEC-11-HIAB-D. The SSYN circuit is shown in drawing G110-0-1, sheet 2; however, for clarity, only the SSYN circuit is shown in Figure 2-23 along with appropriate timing diagrams.

During a DATI or DATIP transaction, signal BUS SSYN L is asserted by the memory when the data is placed on the Unibus by the Memory Data Register. During a DATO or DATOB transaction, BUS SSYN L is asserted by the memory when it receives data from the Unibus.



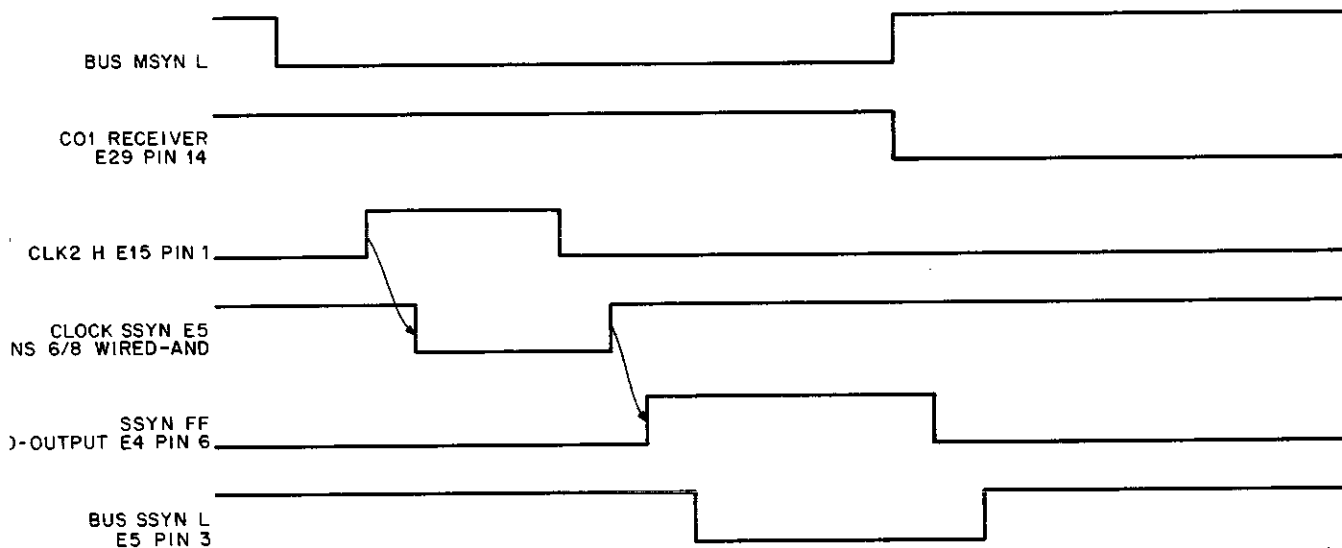
11-1139

Figure 2-23 Slave Sync (SSYN) Circuit



11-1140

Timing Diagram For SSYN Circuit During DAT1 and DATIP  
(Part of Fig. 2-23)



11-1141

Timing Diagram For SSYN Circuit During DATO and DATOB  
(Part of Fig. 2-23)

At the start of each transaction, the master first places the memory address (device and word) and mode control information on the Unibus. (Data is included if the transaction is DATO or DATOB.) For a DATI or DATIP transaction, BUS C01 L is high at pin 10 of bus receiver E29. The output (pin 14) of E29 is low and is sent to the D-input (pin 6) of C01 latch E30 and to pin 5 of the E5 WRITE gate. Signal BUS MSYN L has not yet been asserted; thus, the output (pin 13) of bus receiver E23 is low. This signal is sent to pin 2 of NOR gate E26: the other input (pin 3) of this gate is always low because MSYN A L is normally not connected. The output (pin 1) of E26 is inverted by E15 to produce SSYN RESET L; this signal sets SSYN flip-flop E4 via its preset input (pin 4). The low 0-output (pin 6) is sent to both inputs of bus driver E5. The output of this gate is the slave sync signal (BUS SSYN L) and, at this point, it is not asserted.

As long as BUS MSYN L is not asserted, the SSYN flip-flop is preset. The master now asserts BUS MSYN L, which in turn disables the preset signal to the SSYN flip-flop (SSYN RESET L is high). Clock signal CLK 1 H is generated and clocks C01 latch E30. Latch E30 is reset and its high 0-output (pin 11) is sent to pin 10 of the E5 READ gate in the wired-AND. The wired-AND output CLK SSYN is high, and it remains high as long as both E5 NAND gate outputs are high; this occurs when at least one input of each gate is low. The output of E5 WRITE remains high because input pin 5 is held low by the output of

C01 receiver E29. The output of this gate is not changed when the CLK 2 H pulse appears at pin 4. The output of E5 READ remains high until STROBE H goes low again; the wired-AND output is high again. This positive transition clocks the SSYN flip-flop, which now resets because its D-input is tied to ground (low). The high 0-output (pin 6) of the SSYN flip-flop asserts BUS SSYN L at the output (pin 3) of bus driver E5. The master receives the asserted BUS SSYN L signal and clears BUS MSYN L. The memory receives the cleared BUS MSYN L from the master at bus receiver E23 and generates signal SSYN RESET L via gates E26 and E15 to set the SSYN flip-flop. The memory is now ready for the next transaction.

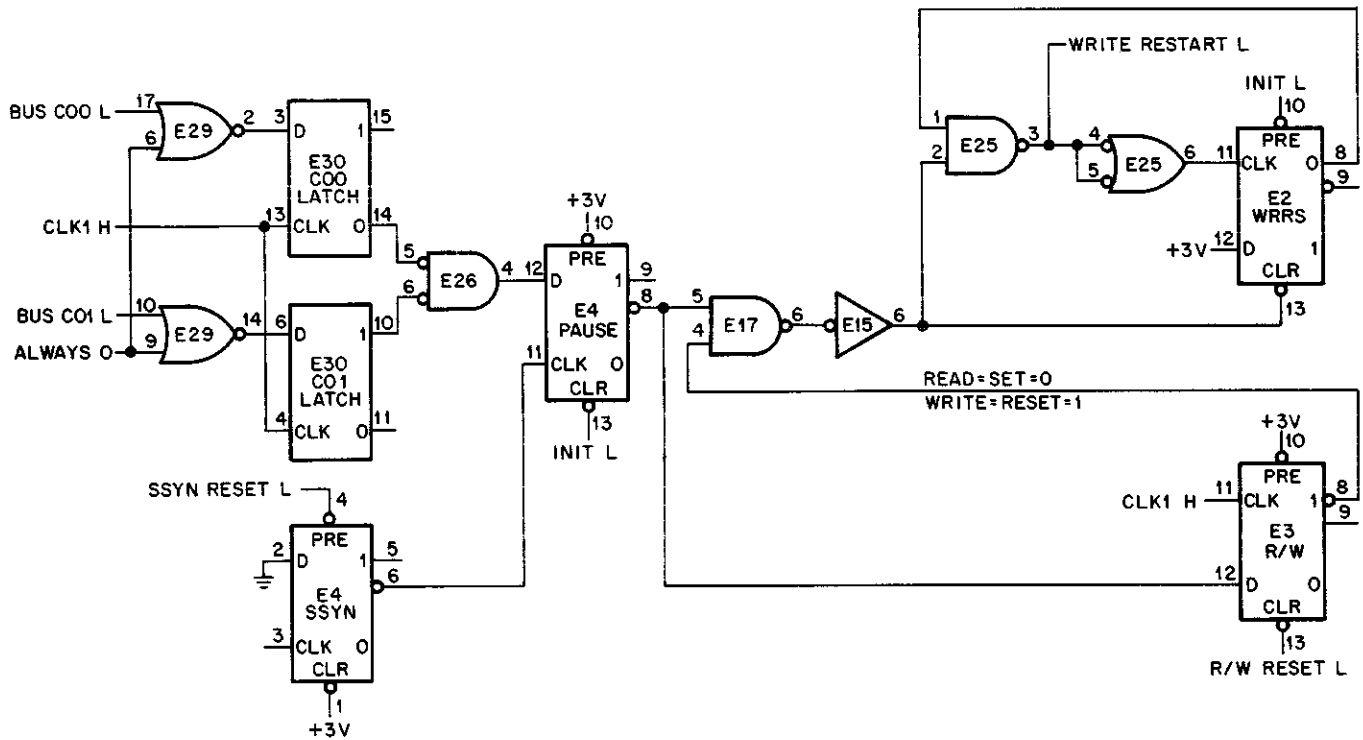
For a DATO or DATOB, the sequence is the same except that BUS C01 L is low at pin 10 of bus receiver E29. This conditions the wired-AND so that the output of E5 READ remains high. In this case, the CLK 2 H pulse generates the CLK SSYN pulse that clocks the SSYN flip-flop via E5 WRITE.

### 2.9.3 Pause/Write Restart Circuit

The PAUSE flip-flop is used to inhibit the restore (write) operation during a DATIP transaction. This transaction is useful when there is no need to restore data after reading because the location is to have new data written into it. By eliminating the restore operation, memory cycle time is decreased by approximately 50 percent. A DATIP must always be followed by a DATO or DATOB. In this case, the DATO or DATOB is shortened by eliminating the clear (read) operation that is normally performed prior to the restore (write) operation. The location has been cleared previously by the DATIP; consequently, the DATO or DATOB need only perform the restore (write) operation. The pause/write restart circuit is shown in drawing G110-0-1, sheet 2; however, for clarity, only the pause/write restart circuit is shown in Figure 2-24.

At the start of all bus transactions, the PAUSE flip-flop is reset; it remains reset throughout the bus transactions except during a DATIP, in which case it is set during the read operation. The PAUSE flip-flop is clocked by the reset 0-output (pin 6) of the SSYN flip-flop. The state (set or reset) of the PAUSE flip-flop is determined by its D-input (pin 12): the D-input is high to set and low to reset. The state of the D-input is controlled by Unibus mode control bits C01 and C00. (Only the mode control representing a DATIP provides a high to the D-input of the PAUSE flip-flop.) During a DATIP, C01 is high and C00 is low at bus receivers E29, pin 10 and E29, pin 7. These signals are in-





11-1144

Figure 2-24 Pause/Write Restart Circuit

verted by the bus receivers and applied to the D-input of the C01 and C00 latches: C01 latch E30, pin 2 is high, and C00 latch E30, pin 6 is low. When the latches are clocked by the CLK 1 H signal, latch C01 is reset and C00 is set. This action puts a low on each input of negative input AND gate E26, which generates a high output. This high output is the D-input to the PAUSE flip-flop. The PAUSE flip-flop is now conditioned to set when it is clocked.

Returning to the start of the DATIP operation, the PAUSE flip-flop is reset. Its D-input is conditioned (D is high) but the PAUSE flip-flop has not been clocked; thus, its 0-output (pin 8) is high. This high output goes to the D-input of the Read/Write flip-flop (R/W E3); this flip-flop is clocked early in the sequence by CLK 1 H. The R/W flip-flop is then set which permits a read operation. The low 0-output (pin 8) of the R/W flip-flop is sent to pin 4 of E17. The other input (pin 5) of E17 comes from the 0-output (pin 8) of the PAUSE flip-flop, and it is a high at this time. The output of E17 is a high and is inverted by E15, which puts a low on the clear input of the Write Restart flip-flop (WRRS E2). The

output of E15 also goes to input 2 of E25. The WRRS flip-flop is cleared (reset) and its high 0-output (pin 8) is sent to the other input (pin 1) of E25. The output of E25 is the WRITE RESTART L signal. This signal is produced to trigger the timing circuit and to initiate a write operation. Signal WRITE RESTART L is now high: its proper state when a read operation is being performed.

At the end of the read operation, the SSYN flip-flop is clocked which resets it. The positive transition at its 0-output (pin 6) clocks the PAUSE flip-flop, which sets the SSYN flip-flop and puts a low on pin 5 of E17. The timing circuit clears (resets) the R/W flip-flop, which in turn puts a high on pin 4 of E17. The output of E17 remains high, inhibiting the WRITE RESTART L signal and preventing the initiation of a write operation.

For any transaction other than DATIP (DATI, DATO, or DATOB), the PAUSE flip-flop is not set when it is clocked because its D-input is low. It remains reset which keeps a high on pin 5 of E17.

When the R/W flip-flop is cleared, it puts a high on pin 4 of E17. The low output of E17 is now inverted by E15 and sent to pin 2 of E25. The WRRS flip-flop is reset so that pin 1 of E25 is also high. The output (pin 3) of E25 goes low, which generates WRITE RESTART L. This starts the formation of a low WRITE RESTART L pulse. This output is inverted by E25, pin 6 which clocks the WRRS flip-flop and sets it, because its D-input is connected to +3V. Pin 8 of the WRRS flip-flop now goes low, which is in turn fed to pin 1 of E25. Thus, the output of E25 becomes high again, which terminates the low WRITE RESTART L pulse. This pulse triggers the timing circuit and initiates a write operation.

For a DATO or DATOB following a DATIP, the PAUSE flip-flop is reset by the SSYN flip-flop, because the DATO or DATOB transaction started with the PAUSE flip-flop set previously by the DATIP.

#### 2.9.4 Strobe Generating Circuit

The strobe generating circuit produces a narrow positive pulse (STROBE H) during the read operation to enable the STROBE 0 H and STROBE 1 H signals for the sense amplifiers. The strobe generating circuit is shown in drawing G110-0-1, sheet 2; however, for clarity, only the strobe generating circuit is shown in Figure 2-25 along with an appropriate timing diagram.



During the read operation, the timing circuit generates a positive RESET H pulse. The RESET H pulse is sent to pin 5 of the strobe delay one-shot (ST DEL E36); this 74121 one-shot provides complementary outputs but only the  $\overline{Q}$  (negative pulse) output (pin 1) is used. Pins 3 and 4 of the ST DEL one-shot are connected to ground so that it can be triggered by a positive going edge at pin 5. Prior to receiving the triggering signal (RESET H), the strobe generating circuit is in the quiescent state. The STROBE OS flip-flop E28 is in the reset state. (When the memory is powered up, E28 is driven to the reset state by E36 if it did not come up reset randomly.) The low 1-output (pin 9) of E28 is sent to pin 13 of E17. The ST DEL one-shot is inhibited so that its  $\overline{Q}$  output (pin 1) is high, which is sent to pin 12 of E17. The output (pin 11) of E17 is high and is inverted by the next E17 gate (pin 3). This is the STROBE H signal, and it is low at this time.

The timing circuit generates a positive RESET H pulse that is sent to pin 5 of E36. The positive edge of RESET H triggers E36, and its  $\overline{Q}$  output (pin 1) goes to low. This is the start of a single negative pulse whose duration is determined by an external RC circuit connected to pins 10 and 11 of E36. The output of E36 directly sets STROBE E OS flip-flop E28 via its preset input (pin 10). The 1-output (pin 9) of E28 goes high and is sent to pin 13 of E17. The other input to this gate (pin 12) is now low. E17, pin 11 is high and is inverted so that E17, pin 3 is still low (no strobe pulse yet). When E36 times out, its output (pin 1) is high again. Pins 12 and 13 of E17 are now both high, and the output (pin 11) of E17 is low. This signal is inverted and E17, pin 3 is high. This is the beginning of the STROBE H pulse. The positive transition of E17, pin 3 also clocks flip-flop E28. E28 is reset because its D-input is connected to ground (low); pin 9 of E28 is now low. It is fed back to pin 13 of E17, which makes E17, pin 3 low again. This terminates the positive STROBE H pulse. The circuit is back to its quiescent state where it remains until another RESET H pulse comes along to trigger ST DEL one-shot E36.

#### 2.9.5 Data In (DATI) Operation

In the discussion of the DATI operation (as well as the DATIP, DATO, and DATOB operation) signals are not traced through circuit components; rather, various events are integrated to describe a complete memory operating cycle. All the circuits involved have been discussed in detail in the preceding paragraphs of this chapter. Refer to engineering logic drawings G110-0-1, sheets 2, 3, and 4; G231-0-1, sheets 2, 3, and 4; MM11-L-3 (timing diagram); and Figure 2-26, which is a flow chart for memory operation.

In a DATI operation, the master requests that a selected memory location be read and the information transferred to the master via the Unibus. The readout is destructive because the read operation forces all cores in the selected location to 0. However, during readout, the information is temporarily stored in the Memory Data Register (MDR) and is automatically restored to the selected location by a write operation that immediately follows the read operation.

At the start of the DATI, MSEL flip-flop is reset, DEL flip-flop is reset, R/W flip-flop is reset, PAUSE flip-flop is reset, and SSYN flip-flop is set. The address lines and mode control lines (C01 and C00) are decoded. The master asserts the BUS MSYN L signal and the cycle begins. Signal CLK 1 H is generated, the DEL flip-flop is set, and the R/W flip-flop is set. Setting the DEL flip-flop initiates the timing chain via delay line DLT. The timing chain generates TWID H and TNAR H. At the same time, CLK 2 H is generated and it presets the MSEL flip-flop, which prevents the start of another cycle until it is reset. Signal READ H from the R/W flip-flop and signals TNAR H and TWID H go to the driver module to select the appropriate read drivers and switches; turn on the X- and Y-current generators; and control the stack discharge circuit. As a result of these signals, the X- and Y-half currents are directed to the selected memory location, and all 16 cores (one per bit plane) are set to 0. Just prior to this event, the timing chain generates RESET 0 L and RESET 1 L; these signals clear the Memory Data Register. The timing chain then generates STROBE H, which asserts STROBE 0 H and STROBE 1 H; these signals are sent to the sense amplifiers. The strobe pulses are timed to arrive at the same time as the pulses induced in the sense/inhibit line. If a selected core is a 1, a pulse is induced in the sense/inhibit line that exceeds the sense amplifier threshold and produces an amplified positive pulse. This output is inverted and presets its associated MDR flip-flop and a 1 is stored in the flip-flop. Signal STROBE H also clocks the SSYN flip-flop which resets. The SSYN flip-flop output asserts DATA OUT H and BUS SSYN L. Signal DATA OUT H gates the output of the Memory Data Register to the Unibus. BUS SSYN L is a Unibus signal that informs the master that the memory has read the selected location and placed the data on the Unibus. The master takes the data and clears BUS MSYN L, which in turn generates SSYN RESET L to set the SSYN flip-flop. BUS SSYN L is cleared to indicate that the Unibus is free; however, another bus transaction cannot be initiated even if the master asserts BUS MSYN L because the lockout feature of the MSEL flip-flop is still set. Prior to the assertion of BUS SSYN L, the timing chain generates DELAY FF RESET L, which resets the DEL flip-flop and allows the TNAR H and TWID H pulses to terminate as

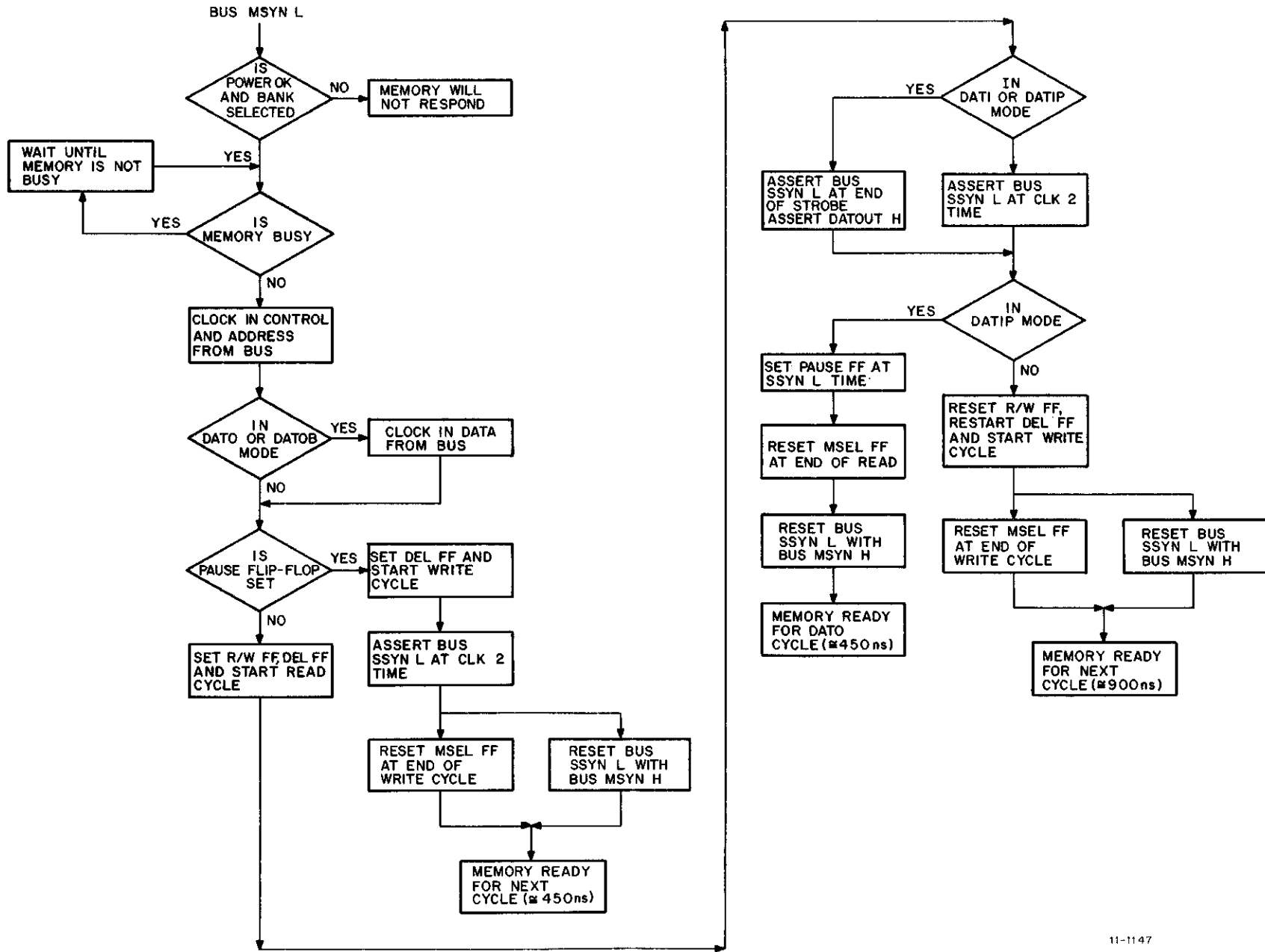


Figure 2-26 Flow Chart For Memory Operation

a function of taps 5 and 7 of the delay line. The timing chain also generates R/W RESET L, which resets the R/W flip-flop.

The memory now enters the write (or restore) cycle. With the R/W flip-flop and PAUSE flip-flop both reset, the pause/write restart circuit generates the WRITE RESTART L signal, which initiates another timing cycle by setting the DEL flip-flop.

The timing chain generates TWID H and TNAR H. These signals plus a low READ H signal from the R/W flip-flop go to the driver module to select the appropriate write drivers and switches; turn on the X- and Y-current generators; and control the stack discharge circuit. In addition, TWID H and an output from the R/W flip-flop are ANDed to generate TINH 0 H and TINH 1 H. These signals control the operation of the inhibit drivers. Signals TINH 0 H and TINH 1 H are ANDed with the outputs of the MDR flip-flops. If a 1 is stored in the MDR flip-flop, the associated inhibit driver is not turned on and a 1 is written into this bit of the selected memory location. If a 0 is stored in the MDR flip-flop, the associated inhibit driver is turned on and produces a current that opposes the Y-line current and prevents a 1 from being written into this bit. The timing chain generates DELAY FF RESET L, which resets the DEL flip-flop and allows TNAR H, TWID H, and the inhibit pulses (TINH 0 H and TINH 1 H) to terminate. The timing chain also generate MSEL RESET L, which resets the MSEL flip-flop.

#### 2.9.6 Data in Pause (DATIP) Operation

In a DATIP operation, the master requests that a selected memory location be read and the information transferred to the master via the Unibus. However, unlike the DATI this information is not to be restored after reading; this location is to have new information written into it. The DATIP performs only a read operation, the write operation is inhibited. A DATIP must always be followed by a write operation (either DATO or DATOB).

The read operation of a DATIP is identical to that of a DATI (Paragraph 2.9.5) until the time the SSYN flip-flop is reset (clocked by STROBE H). At this time, the SSYN flip-flop output clocks the PAUSE flip-flop, which sets it because its D-input is a 1 (only during DATIP due to the state of mode control bits C01 and C00). The timing chain generates R/W RESET L which resets the R/W flip-flop. The output of the PAUSE flip-flop and R/W flip-flop prevents the pause/write restart circuit from generating WRITE RESTART L. With this signal inhibited, the write operation is not produced. The timing chain gen-

erates DELAY FF RESET L, which resets the DEL flip-flop and terminates TNAR H and TWID H. The timing chain also generates MSEL RESET L, which resets the MSEL flip-flop.

The memory is now ready to accept another request from the master. The next operation must be a DATO or DATOB. Normally, a DATO or DATOB starts with a read operation to set all selected cores to zero (clear) before writing new information into them. A DATO or DATOB following a DATIP has this initial clear operation eliminated because the cores have been cleared by the previous DATIP operation.

The DATO or DATOB following a DATIP starts when the master asserts BUS MSYN L. Pulse CLK 1 is generated but it does not set the R/W flip-flop because the PAUSE flip-flop is set. The master places the information to be written on the Unibus where it is picked off by bus receivers and sent to the D-input of the MDR flip-flops. Decoding the mode control bits (C01 and C00) for a DATO or DATOB generates LOAD 0 H and LOAD 1 H, which clock the MDR flip-flops. The outputs of the MDR flip-flops are gated with TINH 0 H and TINH 1 H to control the associated inhibit drivers to write 1s or 0s into the selected memory location. As in the write operation of a DATI, the timing chain generates TWID H and TNAR H which select the appropriate write drivers and switches; turn on the X- and Y-current generators; and control the stack discharge circuit. They also generate inhibit driver control signals TINH 0 H and TINH 1 H. Signal CLK 2 H clocks the SSYN flip-flop (resets it), which asserts BUS SSYN L to tell the master that the data has been taken from the Unibus. When the master clears BUS MSYN L, the SSYN flip-flop is reset, which in turn resets the PAUSE flip-flop. At the end of the write operation, the timing chain generates DELAY FF RESET L and MSEL RESET L to restore the control signals to their original states.

#### 2.9.7 Data Out (DATO) Operation

In a DATO operation, the master sends a 16-bit word to be written into the selected memory location. The transaction starts with a read (clear) operation to set the selected cores to 0 before writing new data into them. The standard DATO consists of a read operation followed by a write operation. (As described in Paragraph 2.9.6, a DATO following a DATIP does not perform the read operation.)

The read operation of a DATO is similar to a read operation of a DATI except that no RESET 0 L, RESET 1 L, STROBE 0 H, and STROBE 1 H pulses are generated. The Mem-



ory Data Register is not cleared and the sense amplifiers are not strobed. This read operation is required only to clear the memory location by setting all the selected cores to 0; it is not necessary to readout and store the information in the MDR. The information on the Unibus data lines is sent to the inputs of the MDR flip-flops. Decoding the mode control bits (C01 and C00) generates LOAD 0 H and LOAD 1 H, which clock the MDR flip-flops. The MDR outputs (16 bits) are gated with TINH 0 H and TINH 1 H to control the associated inhibit drivers. The timing chain generates the other control signals that provide the selection of the appropriate write drivers and switches and a write operation is initiated. This write operation is the same as that described in Paragraph 2.9.6 for a DATO following a DATIP.

#### 2.9.8 Data Out Byte (DATOB) Operation

In a DATOB operation, the master sends a byte (8 bits) to be written into the selected memory location. A high byte (bits D <15:08>) or a low byte (bits D <07:00>) can be selected. Byte selection is made by the state of address bit A00. A DATOB is the same as a DATO except that the selected and non-selected bits are handled differently.

Assume that the low byte (bits D <07:00>) is selected (A00 = 0). Neither RESET 0 L or STROBE 0 H are generated for the selected byte because new data is to be written into bits D <07:00> (low byte). The LOAD 0 H signal is generated so that the data on Unibus Bits D <07:00> can be written into the selected byte location.

The non-selected byte (bits D <15:08>) is to be restored so that RESET 1 L and STROBE 1 H are generated. These signals strobe the byte into the MDR for restoration during the write operation. Restoration is necessary because this byte does not receive new data. The LOAD 1 H signal is not generated; therefore, any data on Unibus bits D <15:08> has no effect on the non-selected byte.

When the DATOB is complete, the selected byte contains new data and the non-selected byte remains unchanged. A DATOB operation following a DATIP is the same, except that the read portion is eliminated.

### 3.1 INTRODUCTION

This chapter provides the preventive and corrective maintenance procedures for the MM11-K and L memories. The user should have a thorough understanding of the normal operation of the memory (Chapter 2). This knowledge plus the maintenance information will aid the user in isolating and correcting malfunctions.

### 3.2 PREVENTIVE MAINTENANCE

Preventive maintenance consists of specific tasks performed at intervals to detect conditions that could lead to subsequent performance deterioration or malfunction. The following tasks are considered preventive maintenance items.

1. Visual inspection of modules for broken wires, connectors, or other obvious defects.
2. +5V and -15V checks: both must be within  $\pm 3\%$ .
3. X- and Y-current generator check (Paragraph 3.2.2).

Two pieces of test equipment are recommended for checking and troubleshooting the memory. Tektronix 453 dual trace oscilloscope or equivalent; and Honeywell 33R Digital Voltmeter or equivalent with 0.5% accuracy.

#### 3.2.1 Initial Procedures

Before attempting to check, adjust, or troubleshoot the memory, perform the following steps.

#### NOTE

All tests and adjustments must be performed in an ambient temperature range of 20°C to 30°C (68°F to 86°F).

1. Verify that all modules are properly and securely installed.

#### CAUTION

Ensure that all power is off before installing or removing modules.

2. Visually check modules and backplane for broken wires, connectors, or other obvious defects.
3. Verify that power buses are not shorted together.
4. Turn on primary power and check that both the -15V and +5V power is present and within tolerances ( $\pm 3\%$ ).
5. Start the system. The memory should operate without errors. If not, check the output of the current generator (Paragraph 3.2.2). If the memory still does not operate properly, a malfunction has occurred. Proceed with corrective maintenance (Paragraph 3.3).

#### 3.2.2 Checking Output of Current Generators

The amplitude of the current pulse from each current generator (X and Y) is factory set at  $410 \pm 5$  mA. It is not adjustable in the field.

The X- and Y-current generators are located on the G 231 Driver Module. Each output has a current loop on its output line for attaching a test probe. Loop J5 is for the Y-generators and loop J6 is for the X-generator (drawing G 231-0-1, sheet 2). The amplitude of each READ current pulse should be  $410 \pm 5$  mA. At the time of measurement, -15V and +5V power must be within the specified tolerance of  $\pm 3\%$ .

#### 3.3 CORRECTIVE MAINTENANCE

This paragraph describes the method of interchanging the positions of the memory modules to gain access to test points. It also describes the strobe delay adjustment, which is a specific corrective maintenance procedure. Further, three aids are included for performing corrective maintenance: a troubleshooting chart and waveforms for the drive and sense/inhibit circuits.

### 3.3.1 Strobe Delay Check and Adjustment

#### CAUTION

Strobe delay is factory adjusted and should be adjusted only when one of the three modules is replaced. It is a critical adjustment and must be done carefully.

The strobe must be set while cycling worst-case noise test patterns (MAINDEC-11-DIGA). The proper setting is mid-way between the two end points where the memory starts to error as strobe time is moved from earliest to latest. As the strobe time is varied, allow adequate time to cycle completely through the worst-case noise test at each strobe position. Figure 3-1 shows the strobe pulse waveform and the READ pulse waveform and the points at which they are picked off for display. Strobe-adjusting potentiometer R120 is on the G110 module next to the large delay line (DL1) and is accessible without putting the module on the extender.

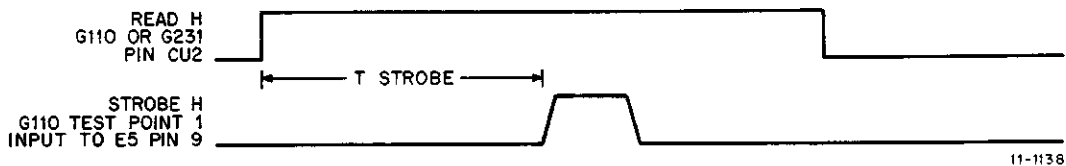


Figure 3-1 Strobe Pulse Waveform

### 3.3.2 Corrective Maintenance Aids

Figure 3-2 is a troubleshooting chart arranged as a two-axis grid that identifies faults versus cause location. Figure 3-3 illustrates the sense/inhibit waveforms, and Figure 3-4 illustrates the drive waveforms. Both figures include schematics to indicate the points in the circuit where the waveforms occur. In addition to nominal waveforms, dotted lines are used to indicate waveforms that appear if specific components are faulty.



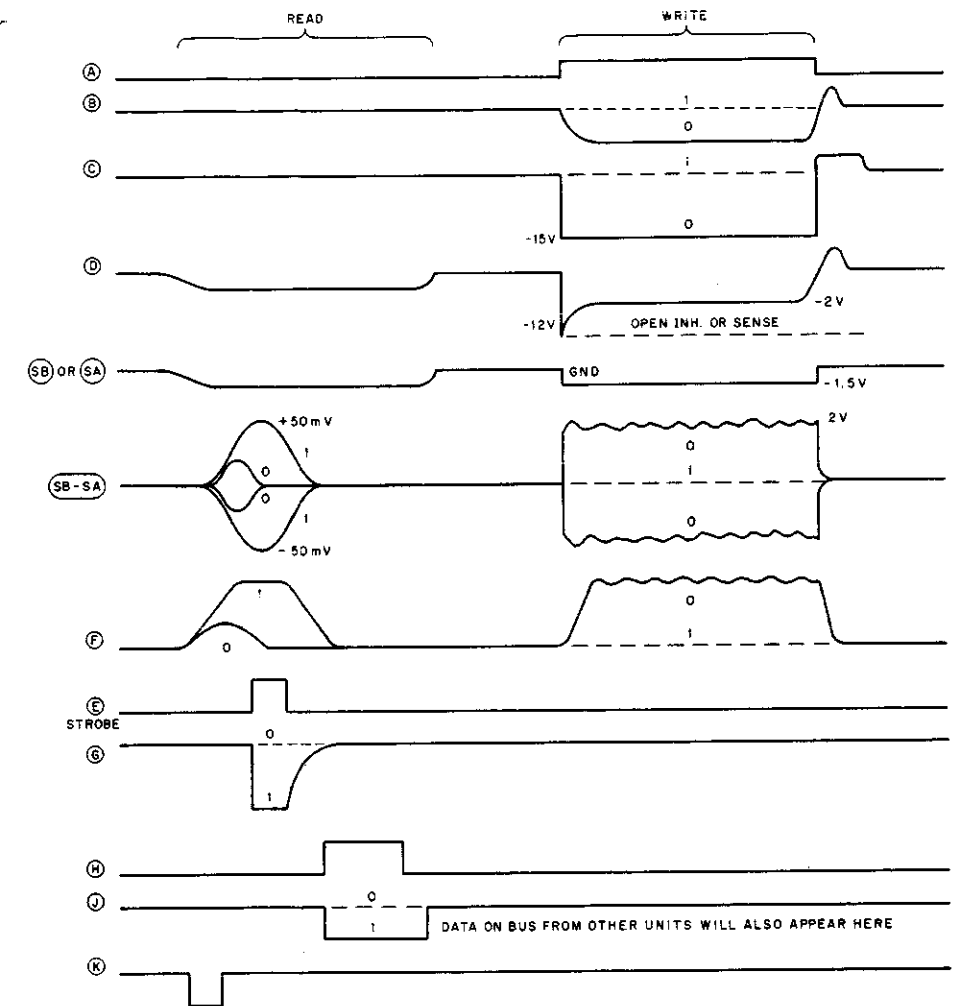
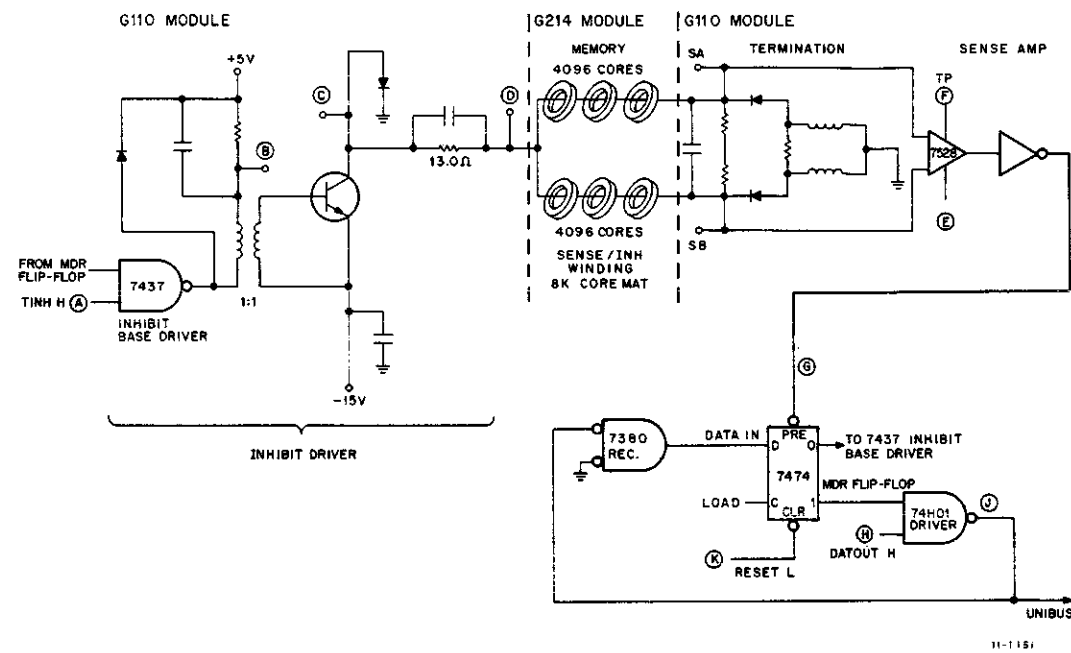
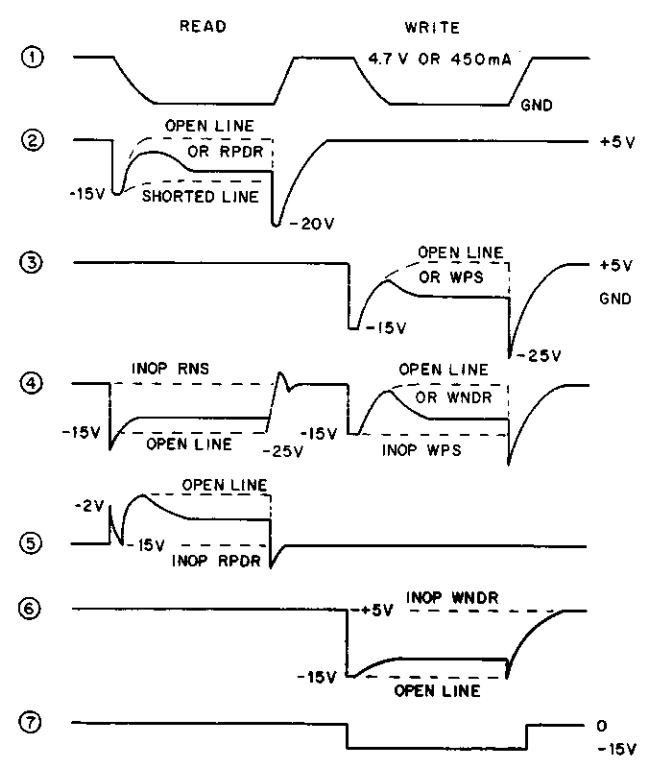
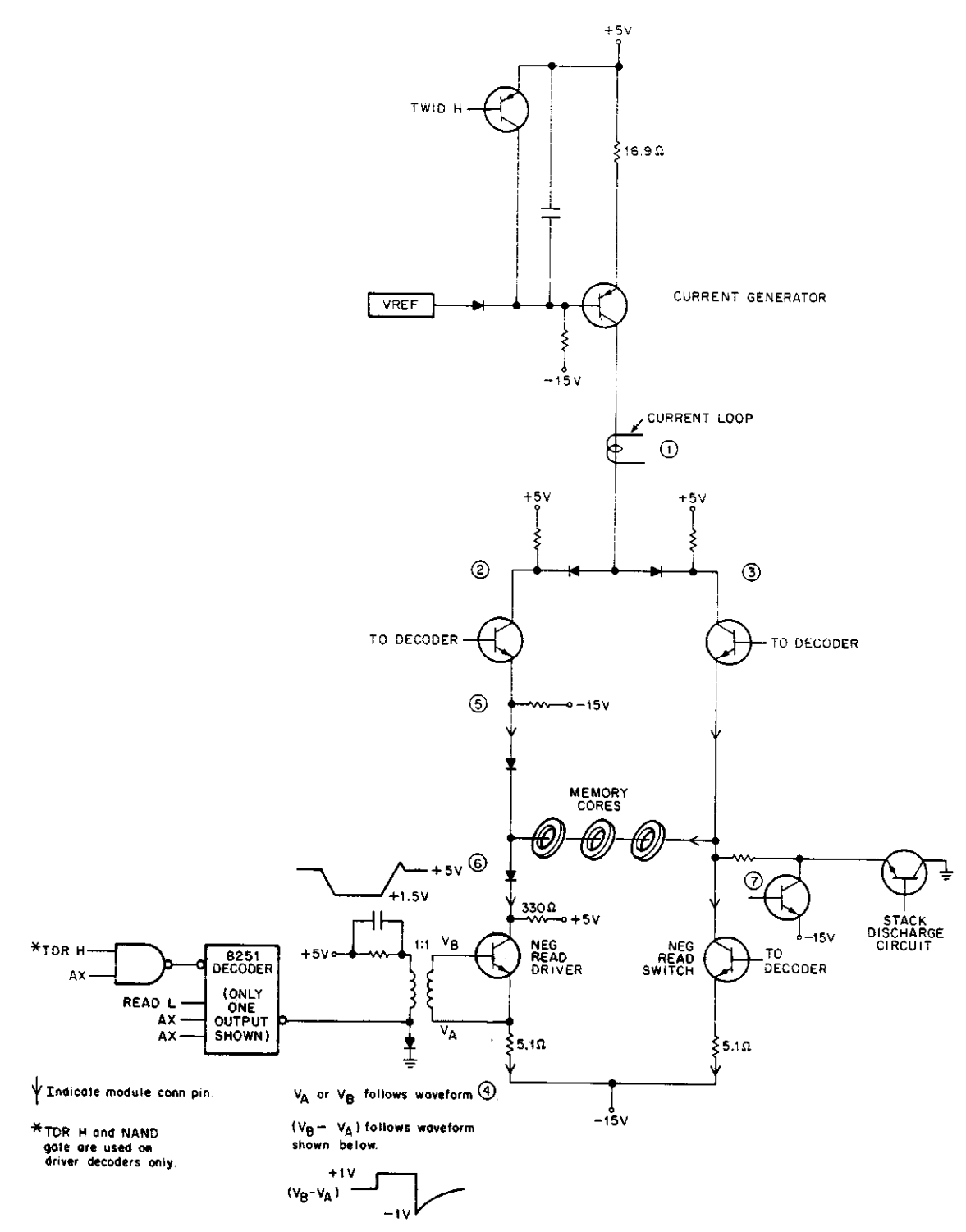


Figure 3-3 MM11-K Sense/Inhibit Waveforms



---- Dotted line show possible failure waveforms.

11-1154

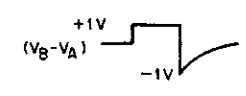


↓ Indicate module conn pin.

\*TDR H and NAND gate are used on driver decoders only.

$V_A$  or  $V_B$  follows waveform ④.

$(V_B - V_A)$  follows waveform shown below.



11-1153

Figure 3-4 Drive Waveforms

### 3.4 PROGRAMMING TESTS

Certain DEC programs can be used to test various memory operations as an aid to troubleshooting. The purpose of each of these memory-related test programs, as well as the program abstract, is given in the following paragraphs. Each program contains instructions for use.

#### 3.4.1 Address Test Up (MAINDEC-11-DIAA)

The purpose of the Address Test Up program is to demonstrate that the selected memory area is capable of basic read and write operations when address propagation is upward through memory. This test program writes the address of each memory location (within the test limits) into itself and then increments through memory until the address corresponding to the high limit is reached. After this location has been written, the memory enters the read cycle. The read cycle starts with the high limit location and reads and compares each word location, decrementing down to the low limit location. The program halts on an error.

The program ensures that all addresses are selectable and can also be used to isolate bad switches, wiring errors, or address selection errors. It will also find double selection errors when two bus addresses select the same core address.

#### 3.4.2 Address Test Down (MAINDEC-11-DIBA)

The purpose of the Address Test Down program is to demonstrate that the selected memory area is capable of basic read and write operations when address propagation is downward through memory. It is a companion test to the Address Test Up program (Paragraph 3.4.1).

This test program writes the address of each location into itself, downward through memory. After writing down, the program reads and checks back up through the memory test area. The program halts on an error.

The Address Test Down program resides in the high portion of core memory. It does not check memory below address 100, as these locations are reserved for trap and vector locations. The program verifies that all modules can perform their basic functions, checks that all addresses are selectable, and can also be used to isolate faulty switches, wiring errors, or address selection errors.



### 3.4.3 No Dual Address Test (MAINDEC-11-DICA)

The purpose of the No Dual Address Test program is to check the unique selection of each memory address tested. This test is divided into two parts. The first portion of the test fills the test field with 1s and writes 0s into the first test location. This is followed by a read check from this location. The program then checks each field location to ensure there are no variations from the 1s configuration. Upon completion of this test, the test location pointer is incremented. The next location is then write-read exercised with 0s, and the test field rechecked for any change in content. When the selected test field has been tested in this mode, the program sets a flag and the second portion of the test is begun. The program fills the test field with 0s and the field is then tested with a write-read exercised with 1s.

This program checks for faulty switches or wiring errors, checks the complete address selection scheme, and checks all 16 bits in the data field for 1s and 0s operation.

### 3.4.4 Basic Memory Patterns Test (MAINDEC-11-DIDA)

The Basic Memory Patterns Test program has two main purposes:

- a. Verify that the selected memory test field is capable of writing and reading fixed data patterns.
- b. Verify that the memory plane is properly strung.

This test program writes a specific pattern throughout a given memory zone, then reads the pattern back and compares it with the original for correctness. If the pattern read fails to compare correctly with the original, the program initiates a call to the error subroutine. After completely checking the pattern, the program continues on to the next pattern test.

### 3.4.5 Worst-Case Noise Test (MAINDEC-11-DIGA)

The purpose of the Worst Case Noise Test program is to generate the maximum possible amount of plane noise during execution of memory reference instructions to check system operation under worst case conditions.

This test program is designed to produce the greatest amount of plane noise possible during memory read and write cycles. The noise parameters are affected by a number of factors. The noise generated is distributed across the core plane algebraically and adds to the

normal dynamic noise present on the sense lines. This can cause misreading of data (within the plane) that is in the low (1) or high (0) category. The sense windings of most memories are such that worst-case patterns can be caused by alternately writing -1 and 0 data configurations throughout memory. Under these conditions, worst-case noise is generated by performing a read, write, complement operation at each location. The test is repeated after complementing all of the pattern data stored in the memory test zone; thus, all cores are tested for worst-case as both 1s and 0s. The pattern or its complement is written into the memory test zone as determined by the exclusive OR between address bits 3 and 9.

The Worst-Case Noise Test program is divided into two parts. Part 1 is run first and, during this part of the program, a -1 configuration is written into all locations having an address with an exclusive OR state between bits 3 and 9. All other locations are loaded with the 0 configuration. After the test zone has been loaded, the memory is rescanned. This time, each location is read, complemented, read, and complemented (RCRC). Any location detected as being disturbed by a previous RCRC operation is flagged as an error. Upon conclusion of the read scan loop, the program automatically switches to Part 2.

During Part 2 of the program, the data patterns stored in memory are complemented. In other words, 0 patterns are stored in locations having addresses with an exclusive OR between bits 3 and 9. All other locations are loaded with the -1 configuration.

The exclusive OR pattern distribution for Parts 1 and 2 is summarized for reference as follows:

Part 1  
Exclusive OR (3 and 9) = 1 pattern  
No Exclusive OR (3 and 9) = 0 pattern  
Part 2  
Exclusive OR (3 and 9) = 0 pattern  
No Exclusive OR (3 and 9) = -1 pattern

After memory is loaded, it is scanned again with a read, complement, read, complement (RCRC) loop as previously described. Any location detected as being disturbed by a previous RCRC operation is flagged as an error.

Before writing or reading any location (in either part of the program), the program issues a call to subroutine XORCK (exclusive OR check) that tests bits 3 and 9 and sets the XORFLG if the exclusive OR condition is present.

Subroutine ERRORA is called for any location disturbed from the -1 configuration; subroutine ERRORB is called for any location disturbed from the 0 configuration.

The program prints out errors and repeats when complete without interruption. Upon completion, the program rings the Teletype bell and then halts if switch 12 is present. A continue from the halt initiates another pass.

If the program indicates an error, use the troubleshooting chart as a guide to locate the fault.

**PART 4**  
**Power Supply**

CHAPTER 1  
GENERAL DESCRIPTION

1.1 INTRODUCTION

The power supply is a forced air-cooled unit that converts single phase 115V or 230V nominal, 47-63 Hz line voltage to the three regulated output voltages required by the computer. The output voltages and their principal uses and characteristics are:

Voltage	Use	Characteristics
+15V	Communication Circuits	Series regulated and overcurrent protected.
+5V	IC Logic	Switching regulated and overvoltage and over-current protected.
-15V	Core Memory	Switching regulated and overvoltage and over-current protected.

The power supply is used in conjunction with the BC05HXX (115V) or BC05JXX (230V) Power Control Assemblies, which contain a line cord, circuit breaker and RFI capacitors. Line cord length is specified in the part number, e.g., 115V, 6 feet is designated BC05H06.

The power circuitry also generates BUS ACLO L and DCLO L power fail early warning signals, and the LTC L real-time clock synchronizing signal.

A thermal control mounted on the heat sink will interrupt the ac input should the heat sink temperature become excessive due to fan failure or other cause.

## 1.2 PHYSICAL DESCRIPTION

The power supply comprises three major subassemblies and two cables: the Power Control, Power Chassis Assembly, DC Regulator Module, DC Cable and AC Cable.

### 1.2.1 Power Control

The Power Control (drawings C-IA-5409824-0-0 and C-IA-5409825-0-0) is mounted to the rear of the computer by two screws. It contains line cord, circuit breaker, RFI capacitors, 115V or 230V connections for the power supply transformer and an output 6-socket Mate-N-Lok connector. Physically it consists of a sheet metal bracket and a slide-on-cover that is locked in place by one screw. A single pole thermal breaker and a line cord strain relief grommet are mounted on the flange of the bracket, making the line cord and breaker reset button accessible on the rear of the computer.

A small printed circuit card is mounted directly to the breaker terminals. This card interconnects and mounts the RFI dual disc ceramic capacitor, the output Mate-N-Lok connector and three fast-tabs for ac input and ground connections. A dual fast-tab is connected directly to the bracket. The black and white line cord wires are connected via fast-tab to the PC card; the green (ground) line cord wire is connected to the dual fast-tab, which in turn is connected to the third fast-tab on the PC card.

The 115V and 230V models differ in only two respects: breaker current rating and (printed circuit) jumpers for parallel or series connection of the power supply transformer primaries. Power control part numbers are:

BC05HXX - 115V, 7A  
BC05JXX - 230V, 4A

where XX denotes line cord length, e.g., BC05H06 has a 6 foot line cord.

### 1.2.2 Power Chassis Assembly

The 700 8731-1 Power Chassis Assembly (Figures 1-1 and 1-2) consists of a long, inverted U-shaped chassis, 700-8726 power transformer, and a 5 inch fan. It is secured to the bottom of the computer by four 8-32 x 3/8 inch Phillips pan-head bolts.

The chassis is mounted to the right of the connector blocks, when viewed from the front, and airflow is from front to rear. The fan is held to one end of the chassis by two screws; the transformer is held to the other end by four mounting studs. The transformer may be removed by loosening four nuts, which are accessible through large holes on the bottom of the power chassis.

The DC Regulator Module is mounted to the chassis assembly by six screws and must be removed for cable access. The dc cable enters a slot on the connector block side of the chassis; the ac cable enters a slot on the other side.

Connections to the fan are made by small fast-tabs; connections to the transformer are made via Mate-N-Lok connectors, 6-pin for primary, 3-socket for secondary.

### 1.2.3 DC Regulator Module

The 5409728 DC Regulator Module (Figures 1-3 and 1-4) is a printed circuit assembly, mounted to the Power Chassis Assembly by four 6-32 x 9/16 inch and two 6-32 x 1/4 inch Phillips pan-head screws. This module contains all the circuitry between the transformer secondary winding and the power supply output cable. The transformer secondary 3-socket Mate-N-Lok connector is plugged into a mating connector that is soldered directly to the printed circuit board and is accessible underneath it. The 9-pin Mate-N-Lok connector on the dc output cable to the computer is similarly mated to a connector underneath the other end of the board.

The dc regulator module may be probed for troubleshooting purposes from the top; all points on the circuit are available. It may also be removed from the top for cable access and for parts replacement by removing the six mounting screws.

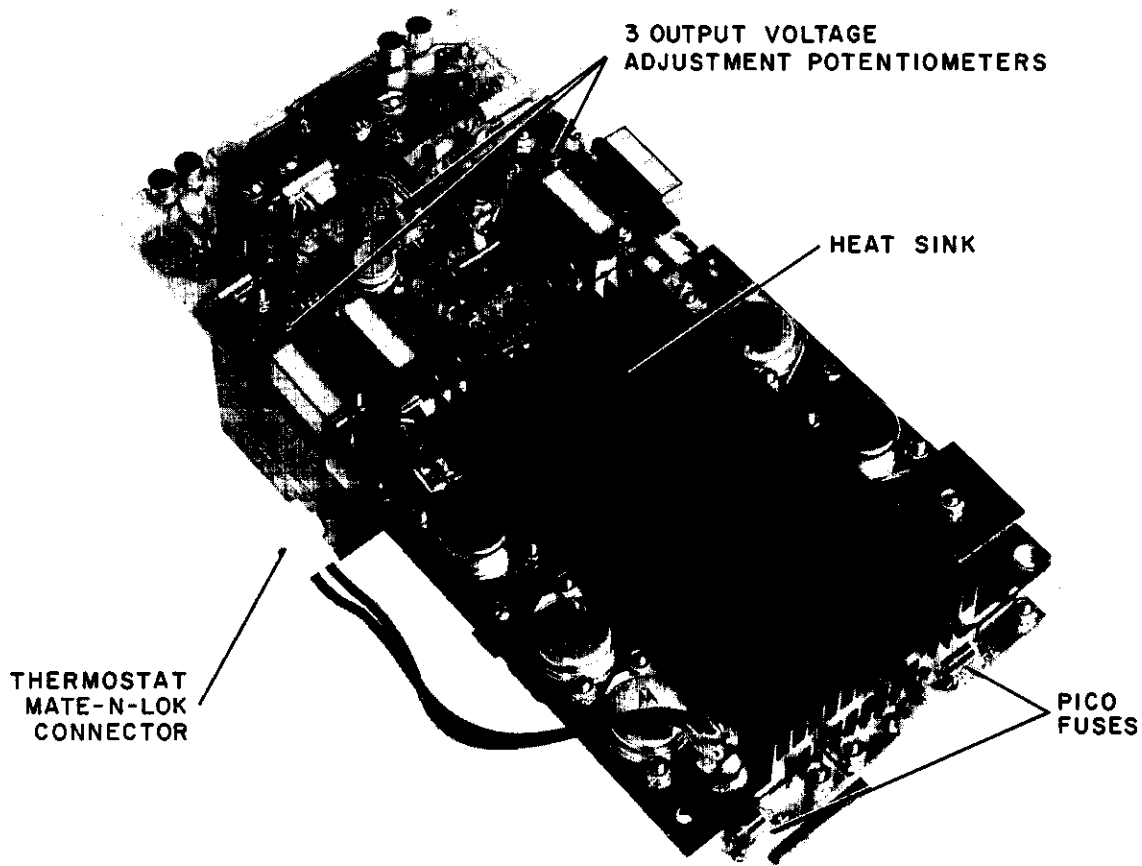


Figure 1-1 Power Chassis Assembly (with DC Regulator Module)



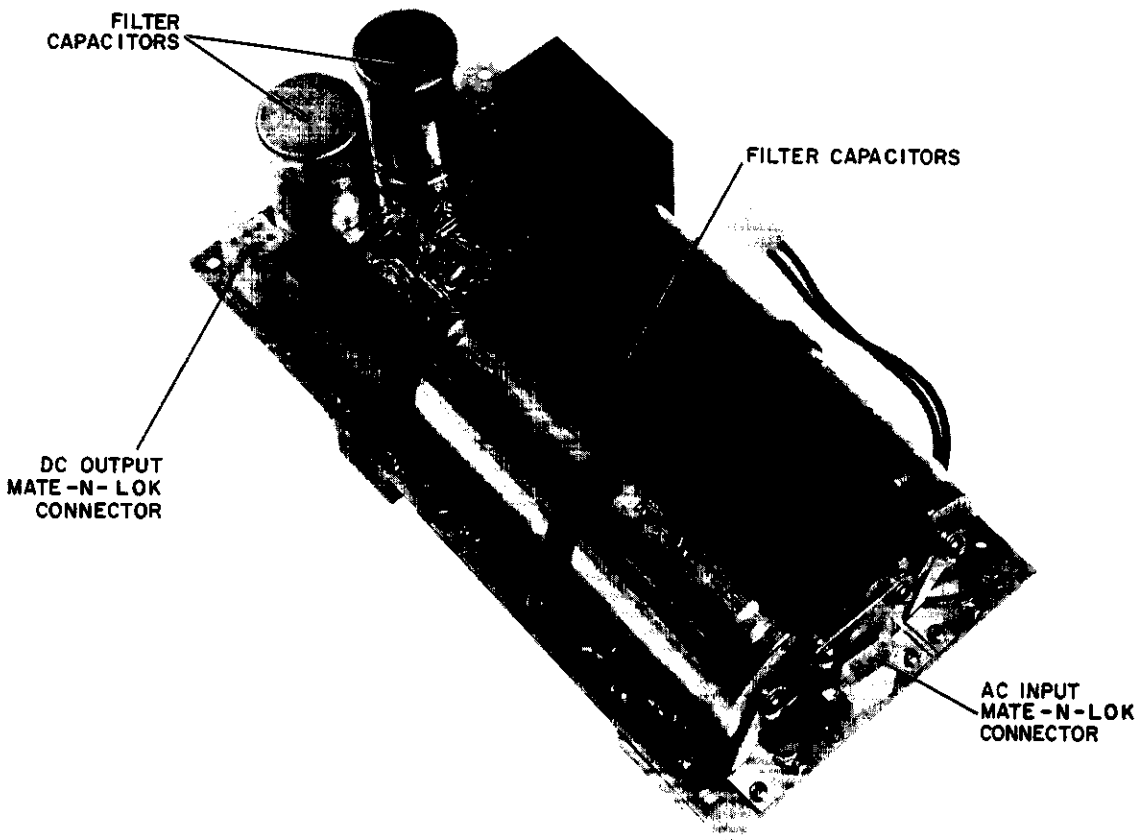


Figure 1-2 Power Supply Assembly (with DC Regulator Module Removed)

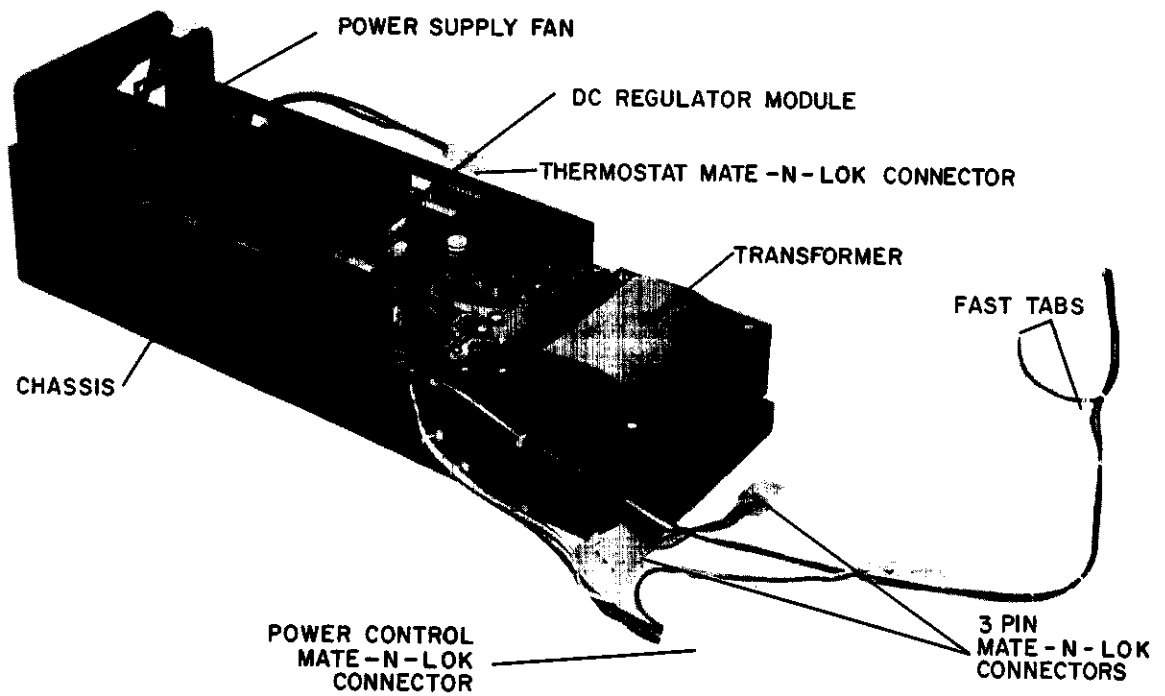


Figure 1-3 DC Regulator Module (Top View)

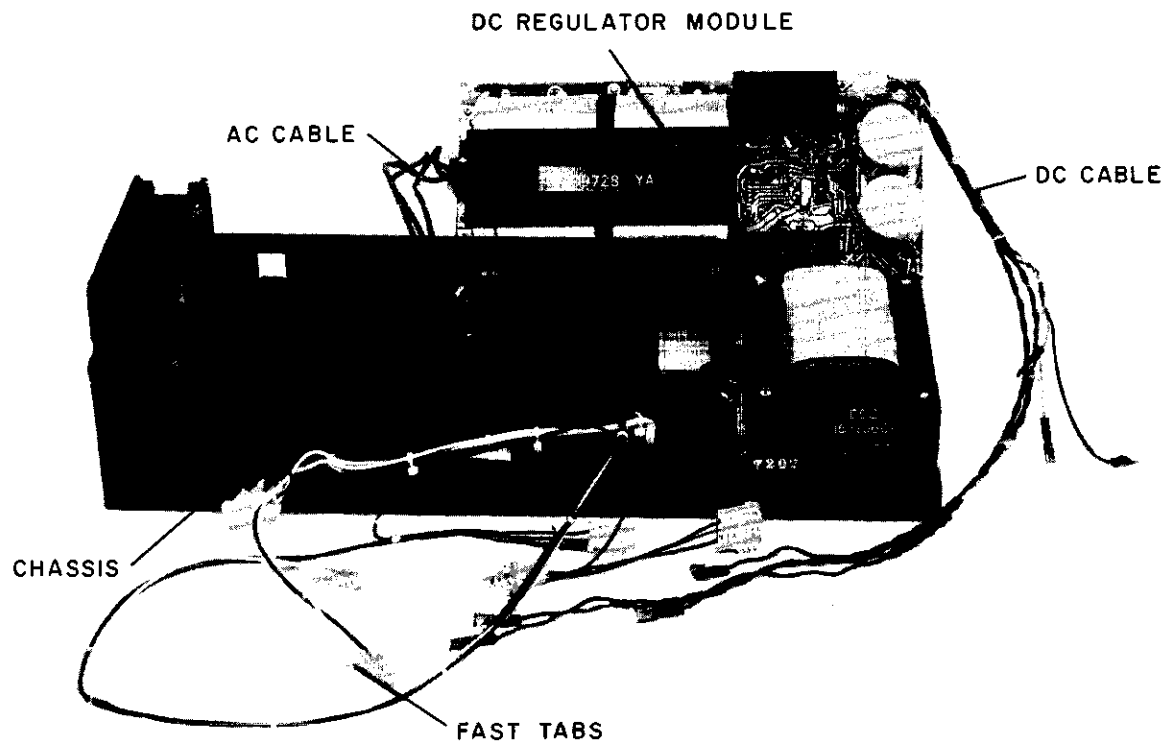


Figure 1-4 DC Regulator Module (Bottom View In Mounting Box)

The printed circuit is approximately 5 x 10 inch, with about half of the top surface devoted to the heat sink. The power transistors and power rectifiers are bolted to two shelves on the sides of the heat sink and make contact with the circuit board directly underneath via solder and screw connections. The heat sink is hard anodized for electrical insulation.

The other half of the top surface is devoted to interconnecting and mounting the balance of the circuit. Three small output voltage adjustment potentiometers are accessible on this top portion of the board.

Two small picofuses are mounted on the top of the PC board on the fan end. These fast-acting fuses will typically only blow when some component is defective or when the +5V or -15V is too high. The two input filter capacitors are held to the underside of the board by a bracket and are connected to the circuit via jumper tabs on the fan end.

The +5V and -15V output filter capacitors and inductors are also mounted under the board, the former by screws and the latter by nuts.

Care must be taken to ensure that all electrical and mechanical connections are secure. In manufacturing, the hardware is tightened with a torquing device set to 12 inch-pounds.

#### 1.2.4 DC Cable

This is a simple cable connecting the computer module to the dc power module via a 9-pin Mate-N-Lok. The latter is made accessible by loosening the six mounting screws and lifting out the dc module.

Cable access is through a slot on the computer module side of the power chassis.

#### 1.2.5 AC Cable

This cable interconnects all ac portions of the computer chassis (Figures 1-1 - 1-4). The ac portions of the computer chassis are listed as follows:

Power Supply Fan	2 fast-tabs
Power Supply Thermostat	2-pin Mate-N-Lok
Computer Fan	2 fast-tabs
Transformer Primary	6-socket Mate-N-Lok
Power Control	6-pin Mate-N-Lok
Key Switch AC Section	2 fast-tabs
Key Switch Remote Turn On Section	2 fast-tabs
Output for Remote Turn On	2 three-pin Mate-N-Lok on rear of computer

The ac cable generally runs on the right-hand side and rear of the computer and is inherently shielded by the power chassis and the computer chassis.

### 1.3 SPECIFICATIONS

Table 1-1 lists all the power supply specifications in three groups: Input Specifications, Output Specifications, and Mechanical and Environmental Specifications.

Table 1-1  
Power Supply Specifications

Specifications	Description
Input Specifications	
NOTE	
Input voltage selection, 115V or 230V, is made by specifying the appropriate Power Control Box, DEC Model BC05 (Paragraph 1.2.1). All specifications are with respect to the BC05 input.	
Input voltage (1 phase, 2 wires & ground)	95-135/190-270V
Input frequency	47-63 Hz
Input current	5/2.5A RMS
Input power	325W at full load

Table 1-1 (Cont)

Specifications	Description
Inrush	80/40A peak, 1 cycle
Rise time of output voltages	30 ms max. at full load, low line
Input overvoltage transient	180/360V, 1 sec
	360/720V, 1 ms
Storage after line failure	25 ms min., starting at low line, full load
Input breaker	7A/4A single pole
(Part of BC05 power control)	Manually reset, thermal
Thermostat mounted on heat sink	277V, 7.2A contacts
(Opens transformer and fan power)	Opens 98-105°C
	Automatically resets 56-69°C
Input connection	Line cord on BC05 power control
	length & plug type specified with BC05 (paragraph 1.2.1)
Turn-on/Turn-off by	Console keyswitch
Hipot (input to chassis & output)	2.1K Vdc, 60 seconds
<b>Output Specifications</b>	
+15V Output	
Load Range	
Static	0-1A
Dynamic	0-1A
Max. bypass capacitance in load	500 mF
for 30 ms turnon	
Overvoltage protection	None
Current limit at 25°C	1.3A to 1.7A (-6.2 mA/°C)
Backup fuse	15A (also used for +5V)

Table 1-1 (Cont)

Specifications	Description
Adjustment	±5% min.
Regulation (All causes including line, load, ripple, noise, drift ambient temperature)	±5%
+5V Output	
Load Range Static Dynamic #1 Dynamic #2	0-17A +5A (within 0-17A load range) No load ↔ Full load
Max. bypass capacitance in load for 30ms turnon	2000 mF
Overvoltage crowbar (blows fuse)	5.7 - 6.8V actuate (7V abs. max. output)
Current limit at 25°C	24-29.4A (-0.1A/°C)
Backup fuse (series with raw dc)	15A
Adjustment range	±5% min.
Regulation Line Static load Dynamic load #1 Dynamic load #2 Ripple and noise 1000 hour drift Temperature (0-60°C)	±0.5% 3% ±2% ±10% 4% p-p ±0.25% ±1%
-15V Output	
Load range Static Dynamic #1 Dynamic #2	0-5A 0.5 ↔ 5A (0.5A/ s) No load ↔ Full load (0.5A/ s)
Max. bypass capacitance in load for 30 ms turnon	1000 mF

Table 1-1 (Cont)

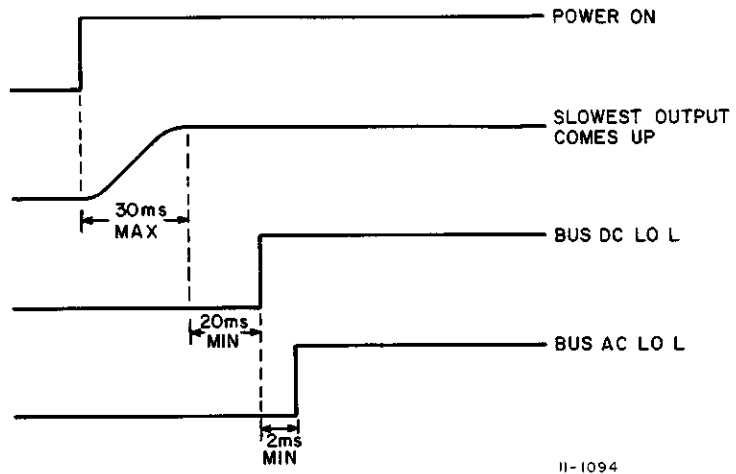
Specifications	Description
<p>Overvoltage crowbar (blows fuse)</p> <p>Current limit at 25°C</p> <p>Backup fuse (series with raw dc)</p> <p>Adjustment range</p> <p>Regulation</p> <p>    Line and static load</p> <p>    Dynamic load #1</p> <p>    Dynamic load #2</p> <p>    Ripple and noise</p> <p>    1000 hour drift</p> <p>    Temperature (0-60°C)</p> <p>Real-Time Clock, LTC L</p> <p>Rated load</p> <p>Frequency</p> <p>Wave shape</p> <p>Pulse height</p> <p>Baseline</p> <p>Short circuit current</p> <p>BUS DC LO L and BUS AC LO L</p>	<p>17.5 - 20.5V (22V abs. max. output)</p> <p>6-8A (-.02A/°C)</p> <p>10A</p> <p>±5% min.</p> <p>±1%</p> <p>±2.5%</p> <p>±3%</p> <p>3% p-p</p> <p>±0.25%</p> <p>±1%</p> <p>Two TTL loads</p> <p>AC line</p> <p>Approximately square wave</p> <p>3.5 to 5.0V positive</p> <p>0 to 1.0V negative</p> <p>15 mA peak max.</p>
<p>Static Performance at Full Load</p> <p>(for 230V connection double below voltages)</p>	
<p>High state BUS DC LO L Goes to high</p> <p>High state BUS AC LO L Goes to high</p>	<p>74-80 Vac line voltage</p> <p>8-11V higher</p>



Table 1-1 (Cont)

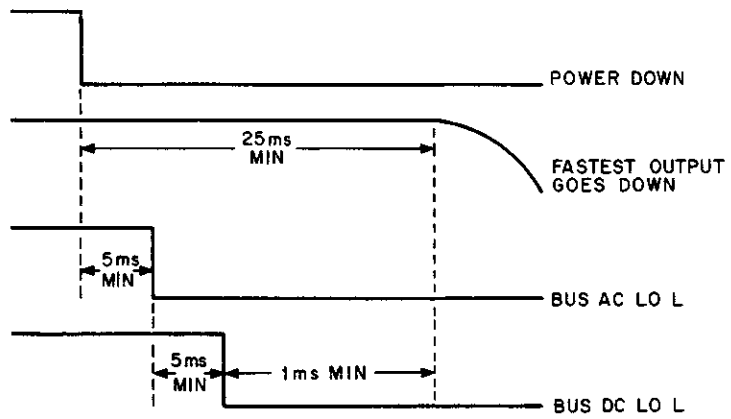
Specifications	Description
High state BUS AC LO L Drops to low	80-86 Vac line voltage
High state DC LO L Drops to low	7-10V lower
Hysteresis (contained in above specs)	3-4 Vac
Output voltages still good	70 Vac line voltage

Worst case on power up is high line, FL.



II-1094

Worst case on power down is low line, FL.



II-1099

Table 1-1 (Cont)

Specifications	Description
Output Characteristics	
<p>Open collector</p> <p>Pull-up voltage on Unibus</p> <p>Rise and fall times</p>	<p>50 mA sinking capability +0.4V max. offset</p> <p>5V nominal, 180 <math>\Omega</math> impedance</p> <p>1 <math>\mu</math>s max.</p> <p>Outputs shall remain in 0 state subsequent to power failure until power is restored, despite the fact that Unibus pull-up voltages may remain.</p>
Mechanical and Environmental Specifications	
<p>Weight</p> <p>    DC regulator</p> <p>    Power chassis assembly including DC regulator module</p> <p>Dimensions</p> <p>Cooling means</p> <p>Minimum cooling</p> <p>Requirements</p> <p>Rated heat sink temperature</p> <p>Shock, Non operating</p>	<p>7 lb approx.</p> <p>18 lb approx.</p> <p>16.50 in. length</p> <p>5.19 in. width</p> <p>3.25 in. height</p> <p>Integral 5 in. fan.</p> <p>375 LFM through heat sink</p> <p>250 LFM over caps, chokes and transformer</p> <p>95°C max.</p> <p>40G (duration 30 ms) 1/2 sine each of six orientations</p>

Table 1-1 (Cont)

Specifications	Description
Vibration, Non operating	1.89G RMS average, 8G peak; varying from 10 to 50 Hz, 8 db/ octave rolloff 50-200 Hz; each of six directions
Ambient temperature	0 to +60°C operating -40 to +71°C storage
Relative humidity	95% max. (without condensation)
Altitude	10K ft

Output parameters are specified at the pins of the 9-pin Mate-N-Lok connector (Figure 1-5), which plugs into the output connector on the 5409728 module. All output voltages are given with respect to the common ground pin on this connector. IR drops in the distribution wiring have been minimized to achieve good regulation at the load.

Pin 1 BUS AC LO L

Pin 2 Common

Pin 3 +5V output

Pin 4 LTCL (Clock Signal)

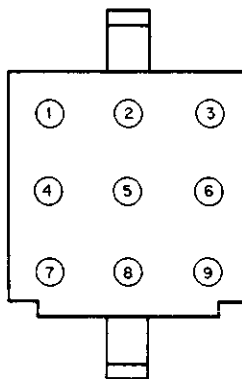
Pin 5 +15V output

Pin 6 BUS DC LO L

Pin 7 Not used

Pin 8 Not used

Pin 9 - 15V output



#### NOTES

1. The circuit connected to pins 7 and 8 is not used in the PDP-11.
2. Pin 2 is not connected to chassis within the power supply. Chassis ground is made at the backplane.

Figure 1-5 Output Connector, 5409728 Regulator Module

## 2.1 INTRODUCTION

The power supply is divided into two sections: the ac input circuit and the dc regulator module. A detailed description to the circuit level is provided for each section. The ac input circuit description discusses the power supply interconnections, power control, power switch, transformer, power control circuit breaker, and the power supply thermostat. The dc regulator module operation description discusses the generation at the circuit level of each of the five power supply outputs.

## 2.2 AC INPUT CIRCUIT

A detailed ac interconnection is shown in Figure 2-1. Figures 2-2 and 2-3 give this information in schematic form.

The line cord, single pole thermal breaker, RFI capacitors, and connections for transformer 115V or 230V wiring are contained in the power control. To select 115V input of 230V input, simply select the proper power control BC05HXX or BC05JXX, where XX denotes length in feet.

A three-section managed keyswitch is employed and is mounted on the console. One section interrupts the power to the transformer primary. A second section is wired to two 3-pin Mate-N-Loks; if the PDP-11 cabinet power control bus is plugged into one of these connectors, the keyswitch will turn on the whole cabinet as well as the computer. The other three-pin Mate-N-Lok is provided for daisy chaining in the cabinet power control system. The third section of the keyswitch is for Panel Lock and is described in Chapter 4 of Part 1.

IV-2-2

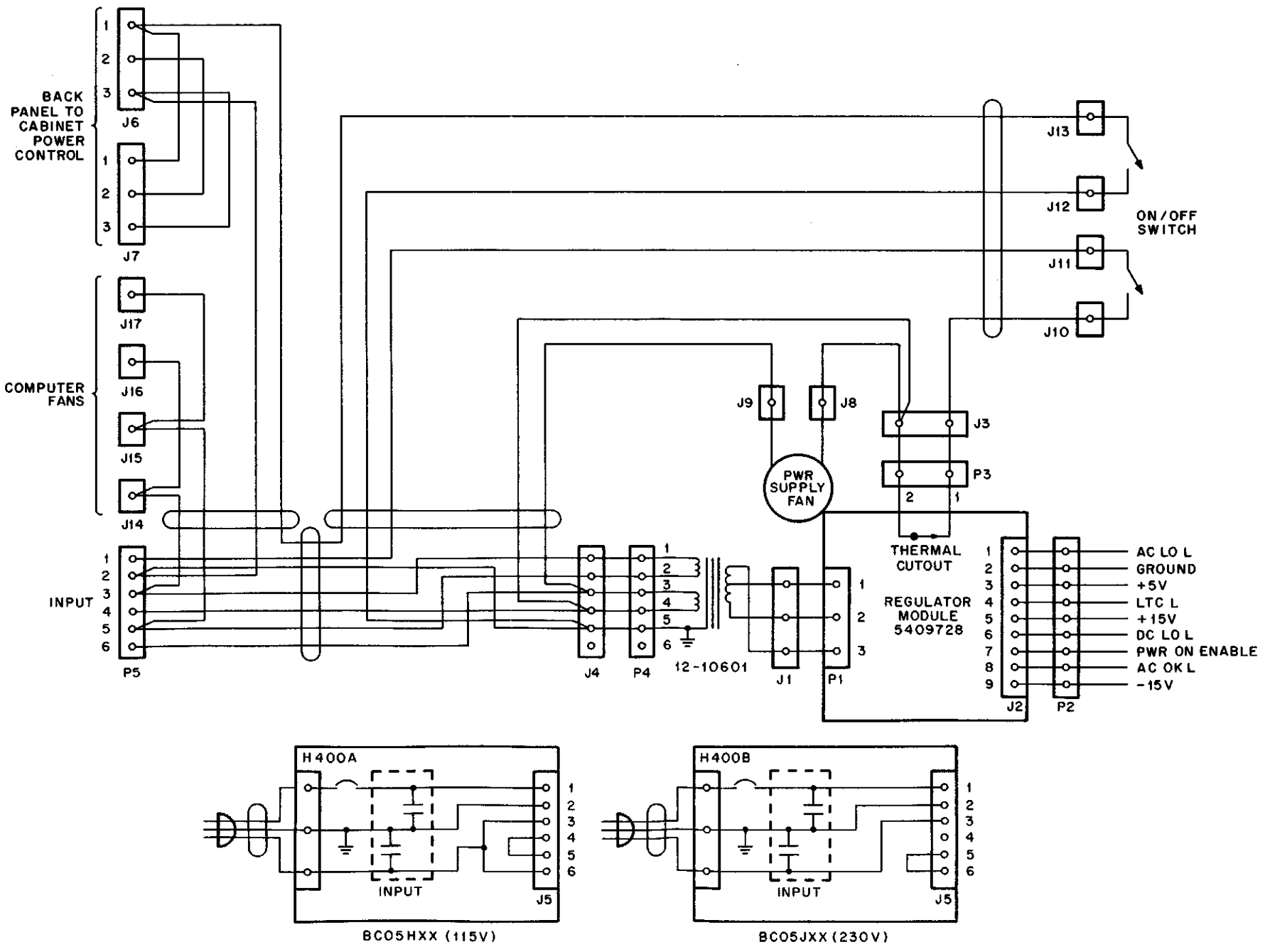
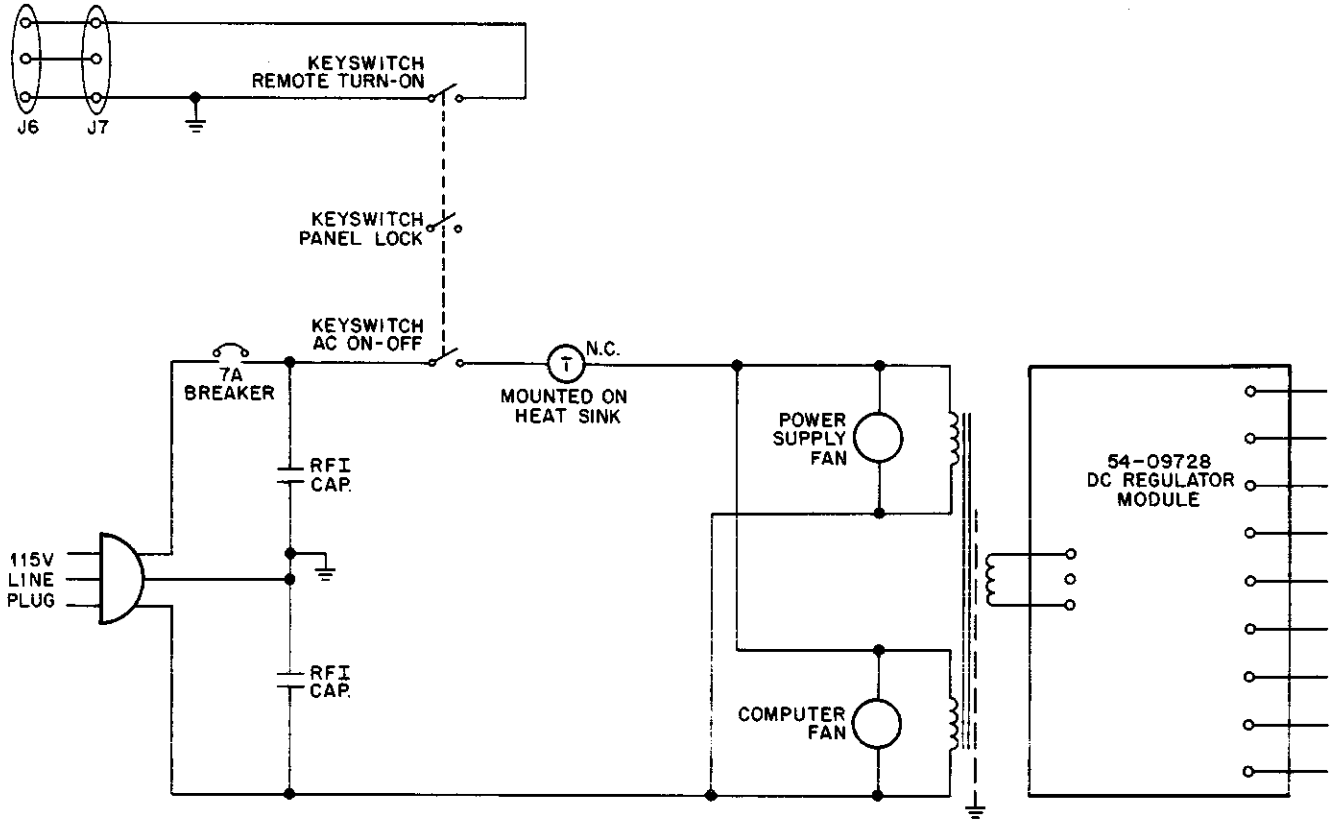
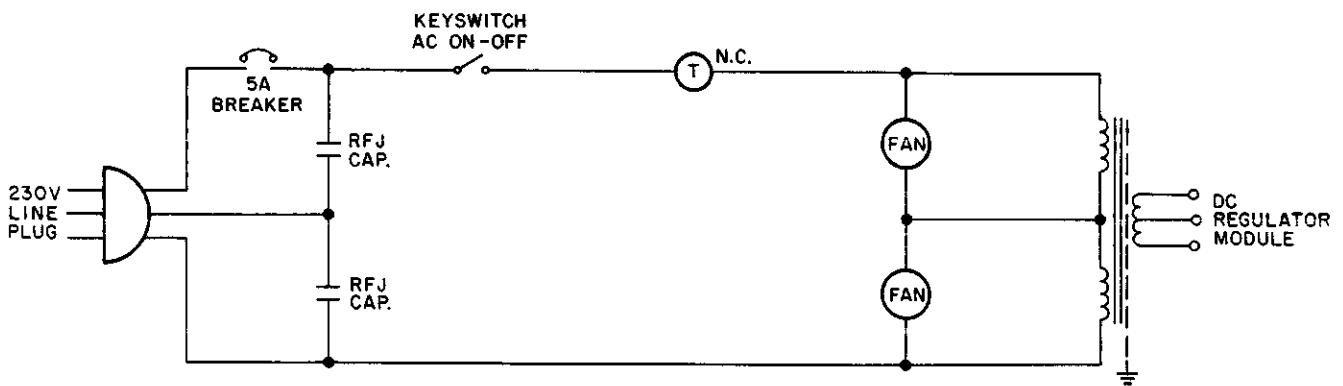


Figure 2-1 Detailed AC Interconnection Diagram



11-1136

Figure 2-2 115V Connections - Simplified Schematic Diagram



11-1137

Figure 2-3 230V Connection Diagram

The transformer is rated for 47–63 Hz and is equipped with two windings that are connected by the power control in parallel for 115V operation and in series for 230V. The fans are each connected across half of the primary so that they are always provided with 115V nominal. There is an electrostatic shield between primary and secondary of the transformer.

The power control circuit breaker contains a single pole thermal circuit breaker that protects against input overload and is reset by pressing a button on the rear of the computer.

The thermostat is mounted on the power supply heat sink. If the heat sink temperature rises to about 100°C, the thermostat will open one side of the primary circuit and de-energize the power supply. It will automatically reset at about 64°C.

### 2.3 DC REGULATOR MODULE OPERATION

The 5409728 DC Regulator Module block diagram is shown in Figure 2-4. (Refer to drawing D-CS-5409728-0-1 in the Engineering Drawing Manual for the complete, updated circuit schematic.) The centertapped output of the power isolation transformer is used to produce  $\pm 28\text{V}$  nominal, raw dc. The  $\pm 28\text{V}$  is unregulated but well filtered by the input storage capacitors.

Two regulators are powered by the +28V. Their outputs are +15V and +5V. A simple series regulator is employed for the +15V output; the +5V output is higher powered and an efficient switching regulator is employed. Both the positive outputs will current limit if shorted.

The +5V output is also protected against overvoltage by a crowbar circuit, which will limit at less than 7V and blow a fuse in case of a circuit failure. The -28V is used to power a switching regulator, which supplies -15V. The -15V is also overcurrent, overvoltage, and fuse protected.

The LTC L Real-Time Clock synchronizing signal is generated by a simple Zener clipper that is fed from the transformer secondary.

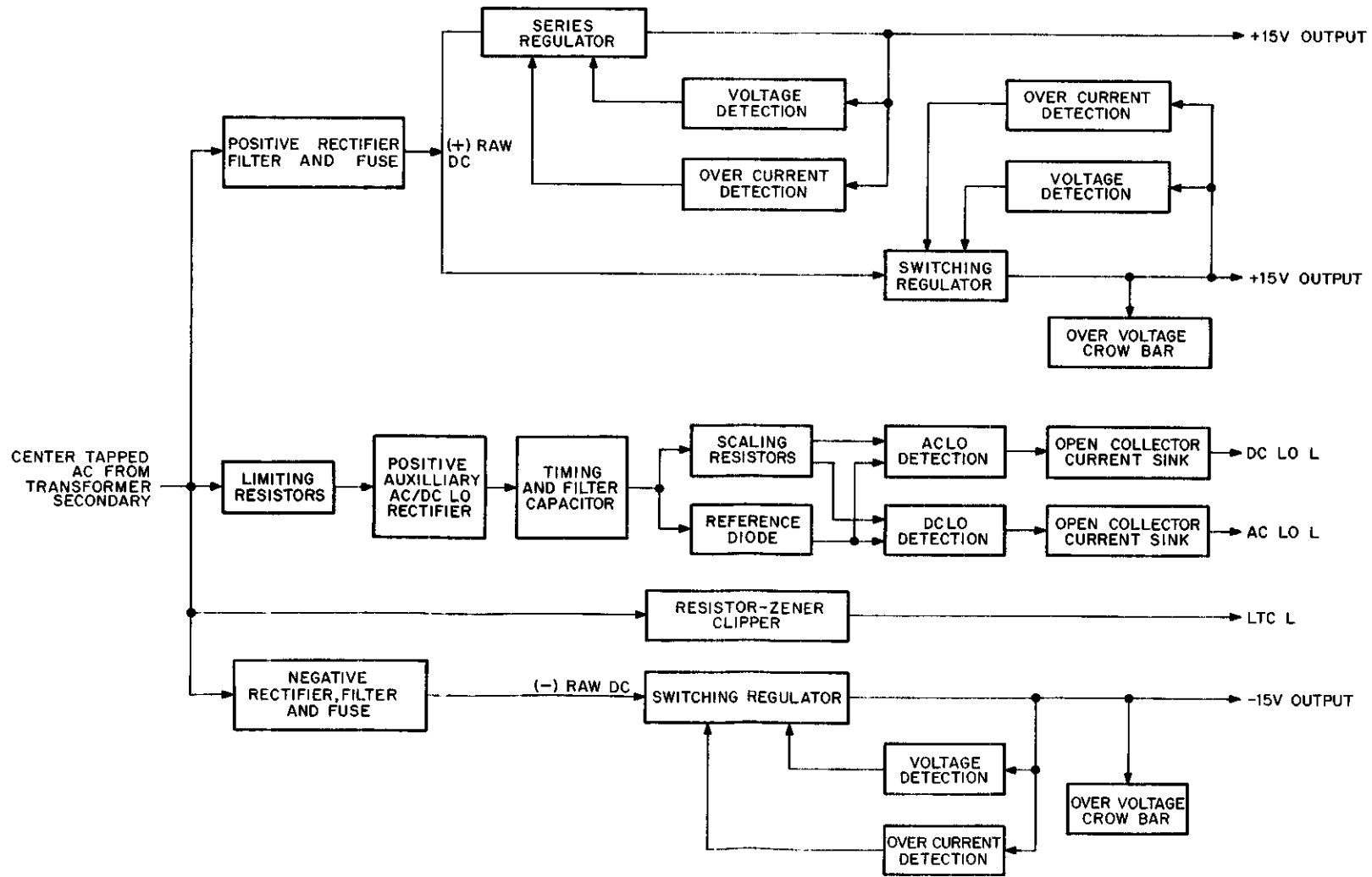


Figure 2-4 Regulator Module Block Diagram



The BUS AC LO L and BUS DC LO L signals are used to warn the Unibus of imminent power failure. Basically what is done is to detect the transformer secondary voltage and generate two timed TTL-compatible open-collector signals, which are used for power fail functions by devices on the Unibus. The detailed characteristics are discussed in Paragraph 1.3 and the circuit operation in Paragraph 2.3.3. Note that this circuit does not detect the regulated dc output voltages as is done in some of the other PDP-11 processors and peripherals.

### 2.3.1 Generation of $\pm$ Raw DC

As stated in the previous paragraph, the centertapped transformer secondary voltage must be rectified and filtered prior to being fed to the three dc regulators.

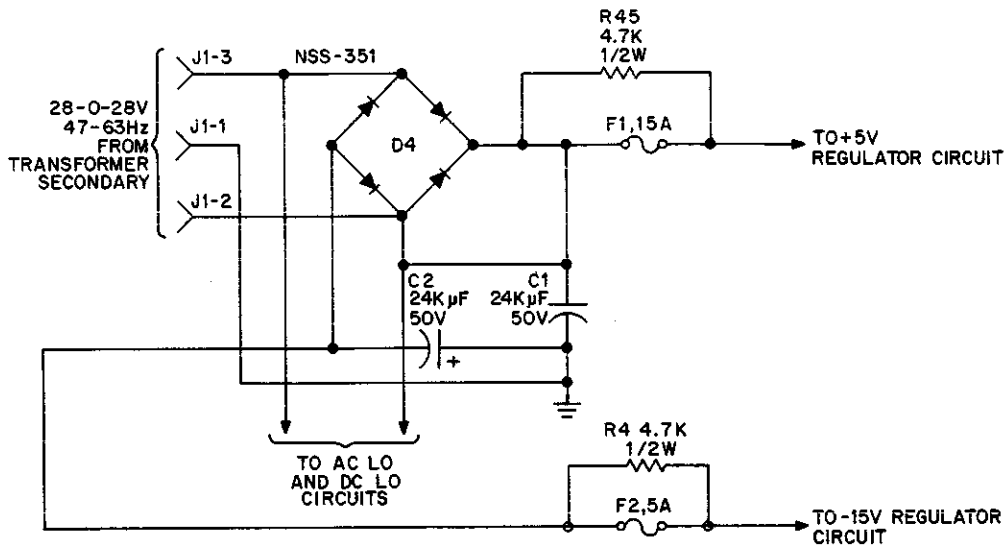
The circuitry involved is shown in Figure 2-5. The bridge rectifier D14 is mounted on the heat sink and the input capacitors C1 and C2 are mounted on the bottom of the regulator module. These capacitors filter the input dc and are large enough to provide at least 25 ms storage when the input power is shut off or fails.

A fuse is used on each output to protect the regulator and load during faults. The fuses will not normally blow when a regulator output is shorted because the three outputs are electronically overcurrent protected. However, the appropriate fuse will blow in case of +5V or -15V overvoltage crowbar or in case of failure in one of the overcurrent circuits.

The resistor across each fuse provides a slow (100 - 150 seconds) discharge of C1 or C2 after the power is turned off in case a fuse blows. The capacitors are placed ahead of the fuse to limit the energy in any fault and thus better protect the outputs.

### 2.3.2 LTC L Circuit

The LTC L Real-Time Clock synchronizing signal (Figure 2-3) is generated by a Zener clipper circuit. The output waveform is a square (clipped sine) wave at line frequency. For one polarity of output sine wave, D13 clips at about +3.9V and in the other polarity D13 clips at its forward voltage of -0.7V.



11-1177

Figure 2-5 Rectifier and LTC L Circuits

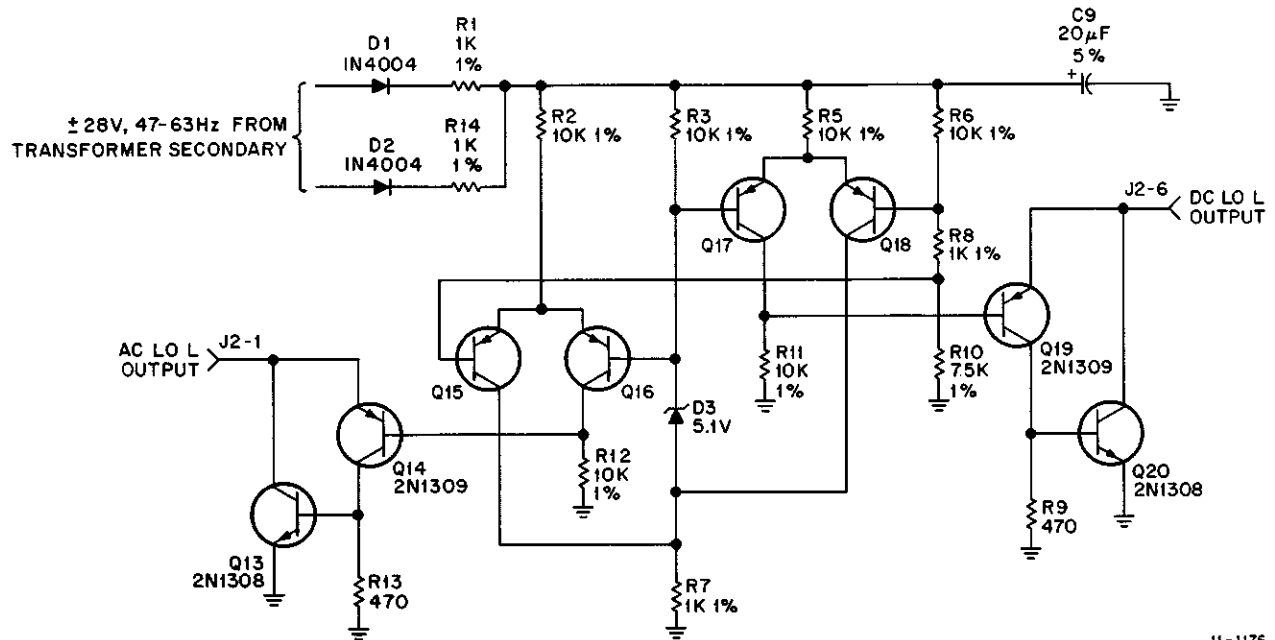
### 2.3.3 BUS AC LO L and BUS DC LO L Circuits

The circuitry shown in Figure 2-6 is employed to generate the timed Unibus power status signals specified in Paragraph 1.3. These are used for power fail functions. The transformer secondary voltage is rectified by D1 and D2 and filtered by C9 and R1, R14.

Circuit parameters are chosen so that the voltage across C9 will rise slower than the three regulated output voltages on power-up and will decay faster than the three regulated output voltages on power-down.

Two differential amplifier circuits are used to detect power status; C17, Q18 is used to generate BUS DC LO L and Q15, Q16 is used to generate BUS AC LO L. The differential amplifiers share a common reference Zener diode D3, which is fed approximately 1 mA by R3.

As C9 charges subsequent to power-up first Q17, Q18 and then Q15, Q16 change state; the reverse is true during power-down. When C9 starts to charge, Q17 and Q16 are on and Q16 and Q18 are not conducting. As C9 charges further Q18 starts to conduct into



11-1176

Figure 2-6 BUS AC LO and BUS DC LO Circuits

R7 and raises the voltage on D3 cathode. This acts as positive feedback and snaps Q17 off and Q18 on more solidly. A few milliseconds later the voltage across C9 has risen sufficiently for the same process to take place in differential amplifier Q15, Q16. The status of each differential amplifier is followed by the germanium transistor open-collector output stages Q19, Q20 for BUS DC LO L and Q13, Q14 for BUS AC LO L. These stages clamp the Unibus at about +0.4V until the differential amplifier circuits sequentially signal them across R11 and R12 that power is up. The outputs then rise to about +5V as dictated by the Unibus loading and pull-up termination resistors.

As previously stated the sequence is as follows:

power up → then BUS DC LO L = 1 → then BUS AC LO L = 1  
 0 = High (+3V)  
 power down → BUS AC LO L = 0 → BUS DC LO L = 0 1 = Low (+.8V)

There is sufficient storage in the regulator output capacitors C1 and C2 so that the regulated output voltages will be within specifications at any time that BUS DC LO L or BUS AC LO L = 1. Note that the open collector stages are designed to clamp the Unibus to 0.4V maximum, even when there is no ac input to the regulator. They are inherently biased on by R11 and R12 until the differential amplifiers signal that power is OK.

### 2.3.4 +15V Regulator Circuit

The +15V regulator shown in Figure 2-7 is a simple series regulator. The pass transistor Q1 is a high-gain power Darlington and is mounted on the heat sink. Base drive current is supplied to Q1 via R38. Q3 acts to limit the value of this current to the required value by shunting it away from the Q1 base. Q4, the voltage detector amplifier, biases on Q3 and thus limits current in Q1. The +15V output voltage is sampled on the viewing chain R34, 35, 36 and compared to the voltage across reference Zener D8, which is fed by R37. If the output should try to increase from the regulated value the emitter of Q4 is made relatively more negative than its base and conduction through Q4 increases. This increases the conduction through Q3 and causes Q1 to shut down sufficiently to restore the output voltage to the regulated value. Ambient temperature compensation of the voltage detector is essentially flat since D8 has a +2 mV/°C temperature coefficient and the base emitter junction of Q4 has a -2mV/°C temperature coefficient.

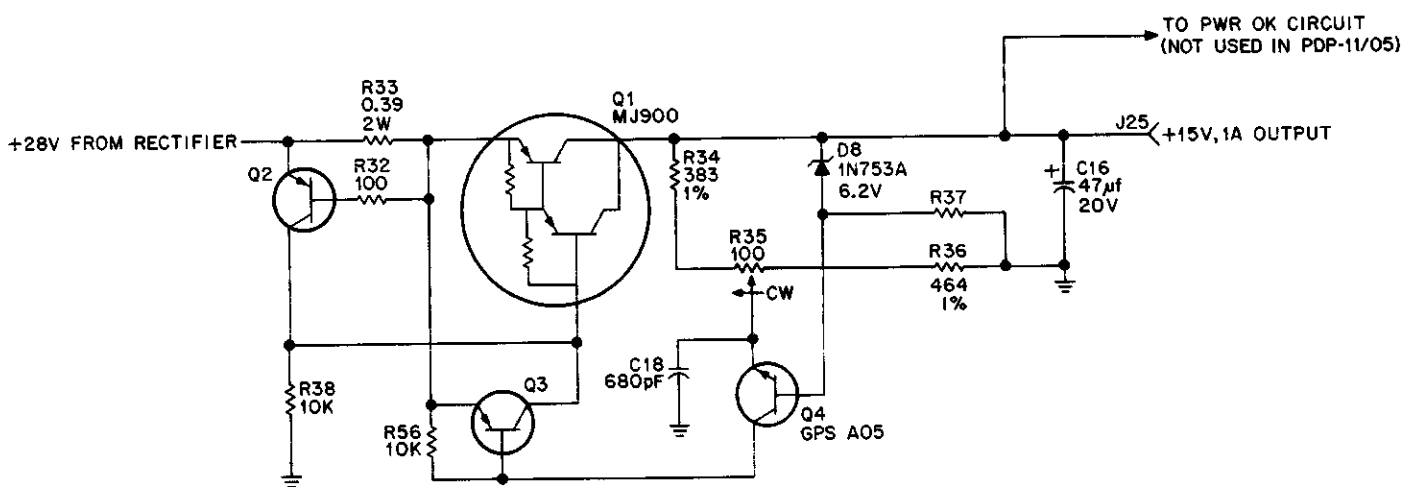


Figure 2-7 +15V Regulator Circuit



The viewing chain consists of R49, 50, 51 and the reference Zener is D9, which is fed by R44. Q10 is the detector amplifier. The pass transistor Q6 and first stage driver Q7 are mounted on the heat sink. The predriver Q8 is turned on by R46. The current is diverted from the base of Q8 by off-driver Q9, which is controlled by Q10. Thus far it can be seen that the +15V and +5V regulators are similar in operation, i.e., a tendency for the output voltage to rise results in more conduction through Q10 and resultant limiting of conduction through Q6.

Here the similarity ends. The +5V circuit is a regulator that operates in the switching mode for increased efficiency. To get the regulator to switch, positive feedback is applied to the voltage detector input via R47 (R16 & C17 are used to improve the switching operation during short circuit current limiting). Thus the whole regulator acts as a power Schmitt trigger and is either completely turned on or turned off, depending on whether the output voltage is too high or too low. When Q6 is on, it supplies current through filter choke L1 to the output smoothing capacitor C7 and the load. When Q6 is off, the L1 current decays through commutating diode D10, which becomes forward biased by the back emf of L1. The waveform across D10 is a 30V nominal rectangular pulse train. The filtered output across C7 is thus +5 Vdc with about a 200 mV pp 10 kHz nominal sawtooth of super-imposed ripple. At the crest of the ripple, Q6 turns off and at the valley Q6 turns on. This switching mode of operation limits the dissipation in the circuit to the saturated forward losses of Q6 and D10 and the switching losses of Q6. The resultant high efficiency allows the heat sink to be small and the number of power semiconductors to be few.

R50 is the voltage adjustment potentiometer. R51 is a positive temperature coefficient wire-wound resistor that compensates for the fact that the Q10 base-emitter junction and the reference diode D9 both have negative voltage temperature coefficients. Q5 current, limited by R39, 40, detects the overcurrent signal generated across resistor R41, which is in series with the Q6 collector.

Output fault current is limited to a safe value because conduction of Q5 makes the reference voltage across D9 decrease to zero. This causes Q10 to conduct and shuts down the regulator. C5 is an averaging capacitor, which is necessary in the circuit because the current through R41 is pulsating.

High frequency bypass capacitors are used on input and output of the regulator, C3 and C6, respectively. C4 is used to slow down the turn-on of Q6 to allow D6 to recover from the on state without a large reverse current spike.

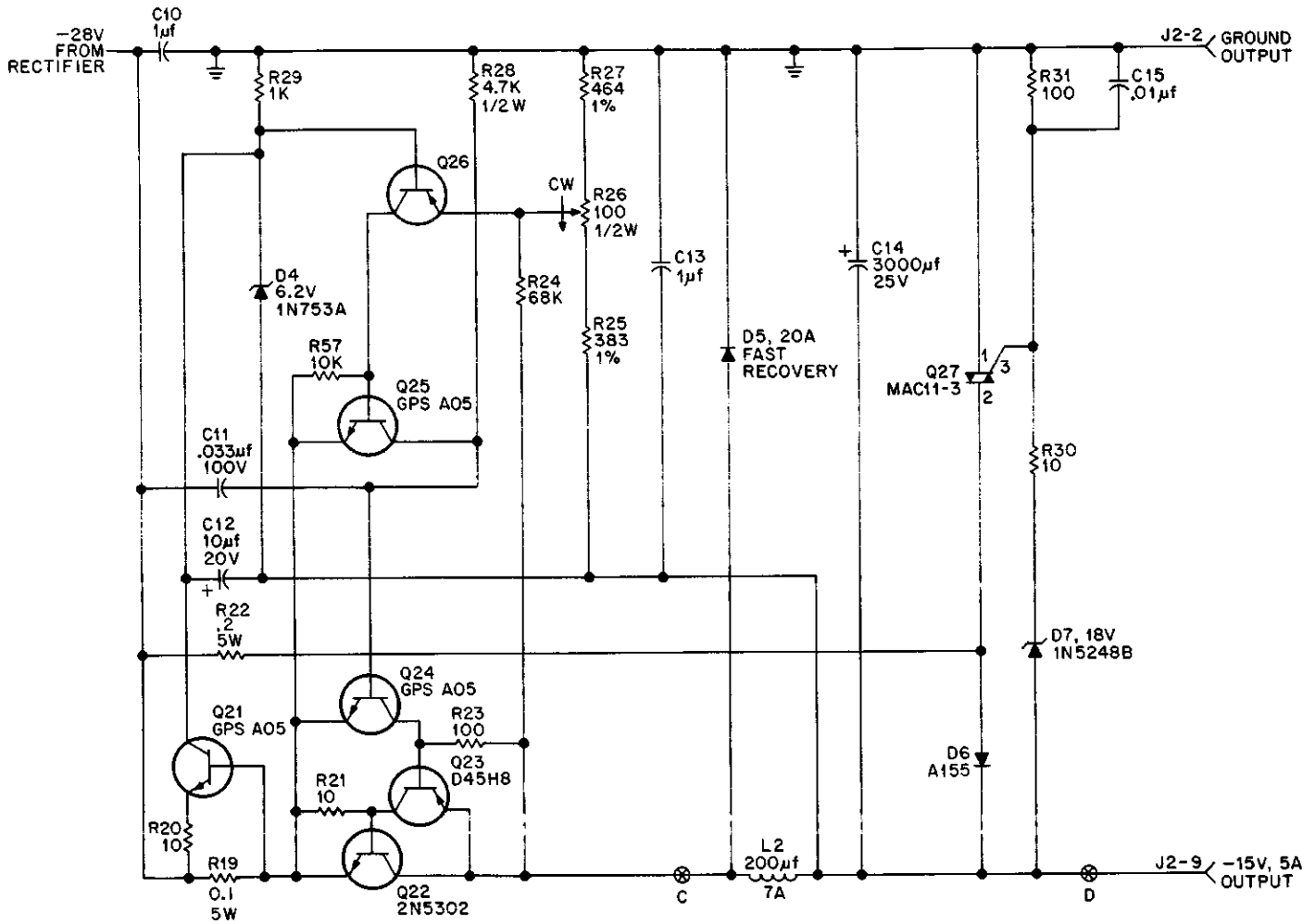
In the event that a malfunction causes the output voltage to increase beyond about 6.3V nominal, Zener diode D12 will conduct and fire silicon-controlled rectifier Q11. This will crowbar the output voltage to a low value through D11 and will blow fuse F1 through R52.

#### 2.3.6 -15V Regulator Circuit

The -15V regulator circuit is shown in Figure 2-9. It is essentially the complement of the +5V regulator circuit and differs only in minor detail.

In particular: the crowbar device is a Triac Q27 instead of an SCR; no temperature compensating resistor is required because Q26 and D4 track each other, as in the +15V regulator (Paragraph 2.3.4); the detailed interconnection of the drivers and the circuit values are different.

The -15V output voltage is adjusted by potentiometer R26.



11-0970

Figure 2-9 -15V Regulator Circuit



## CHAPTER 3 MAINTENANCE

### 3.1 INTRODUCTION

Information is provided in this chapter to maintain the power supply. This consists of adjustments, circuit waveforms, troubleshooting, and parts identification. The adjustments consist of three output potentiometers. The circuit waveforms provide a guide to proper operation at various places in the circuit. The troubleshooting section provides rules, hints, and a troubleshooting chart as a maintenance aid in isolating power supply malfunctions. Finally, the parts identification section provides a directive to obtaining parts information for the entire power supply unit through a parts location directory to the mechanical engineering drawings in the Engineering Drawing Manual.

### 3.2 ADJUSTMENTS

There are only three adjustments to the power supply. These adjust the three dc output voltages: +15V, +5V, and -15V. A small screwdriver is all that is required. Clockwise adjustment of any of the potentiometers increases voltage; and the potentiometers are located on the top side of the dc regulator module. The potentiometer designations are:

1. R35 - +15V
2. R50 - +5V
3. R26 - -15V

In performing any of these adjustments note the following:

#### CAUTIONS

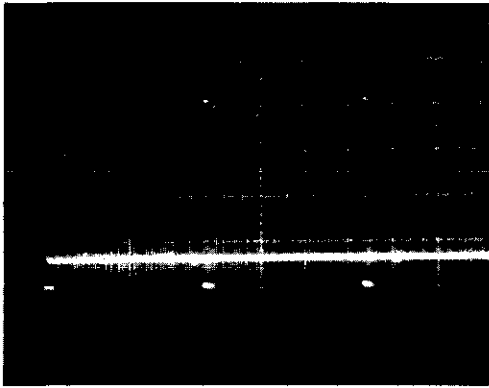
1. Do not adjust voltages beyond their 105 percent rating and adjust slowly in order to avoid over-voltage crowbar, which will blow dc output fuses.
2. Do use a calibrated voltmeter; preferably a digital voltmeter. Voltages should be adjusted to their center values: +15.0, +5.0, and -15.0, all under load at the dc cable termination on the system unit.

### 3.3 CIRCUIT WAVEFORMS

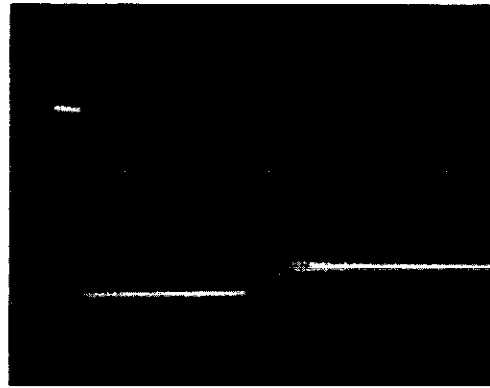
The two basic regulator circuits used on the dc regulator module generate +5V and -15V. Figure 3-1 shows six waveforms of the +5V regulator circuit taken at two points (A and B) in the circuit (Figure 2-6). Waveforms a, b, and c are taken at point A, which is the +5V circuit, Q6 transistor output. Waveforms d, e, and f are taken at point B, which is +5V power supply output (J2-3). Figure 3-1 also indicates the load conditions and time scales for each waveform. Figure 3-2 shows six waveforms of the -15V regulator circuit taken at two points (C and D) in the circuit (Figure 2-7). Waveforms a, b, and c are taken at point C, which is the -15V circuit, Q22 transistor output. Waveforms d, e, and f are taken at point D, which is the -15V power supply output (J2-9). The load conditions and time scales of the respective waveforms are indicated in Figure 3-2. These waveforms were taken on a Tektronix Model 453 oscilloscope. To locate the circuit test points on the dc regulator module refer to Paragraph 3.5. All waveforms are with respect to J2-2, power common.

### 3.4 TROUBLESHOOTING

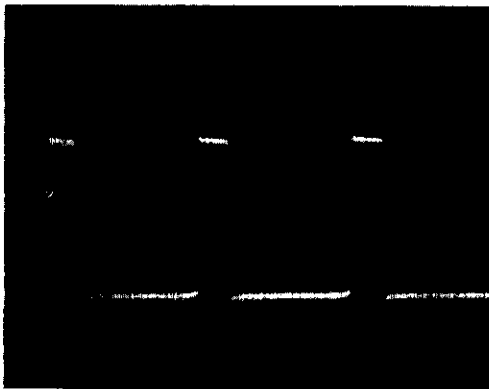
Troubleshooting information for the power supply consists of troubleshooting rules, hints, and a troubleshooting chart. This information provides a maintenance aid to isolating power supply malfunctions (drawing D-CS-5409728-0-1).



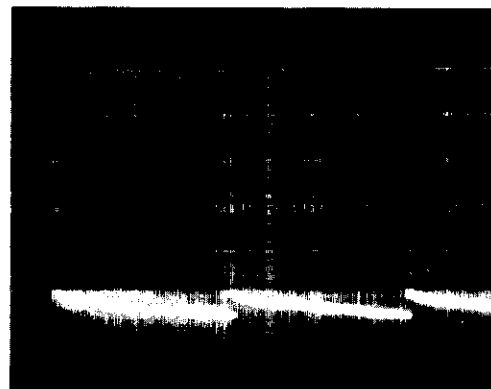
a) Point A, No load, 2 ms/div,  
and 10V/div.



b) Point A, No load, 20  $\mu$ s/div,  
and 10V/div.

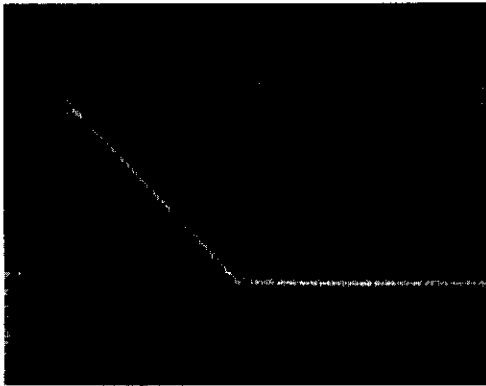


c) Point B, 20A load, 20  $\mu$ s/div,  
and 10V/div.

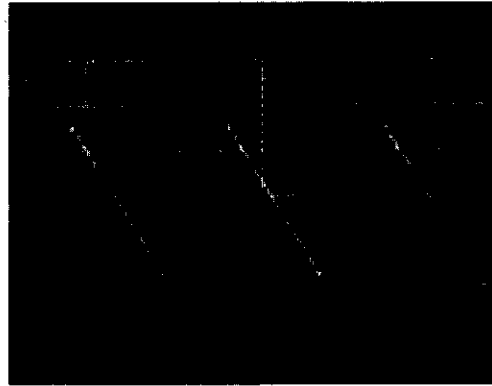


d) Point B, No load, 2 ms/div,  
and 50 mV/div.

Figure 3-1 +5V Regulator Circuit Waveforms

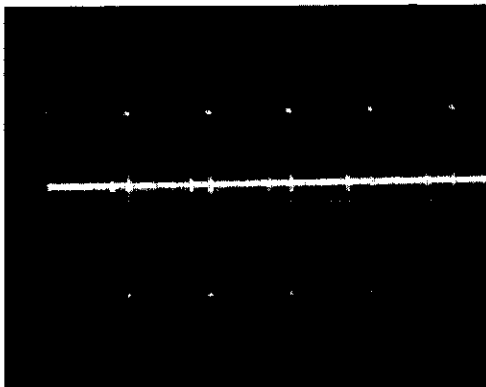


e) Point B, No load, 20  $\mu\text{s}/\text{div}$ , and 50 mV/div.

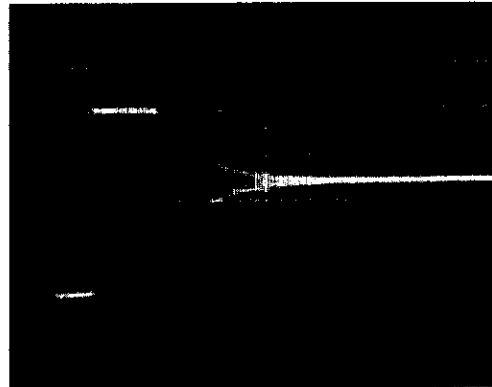


f) Point B, 20A load, 20  $\mu\text{s}/\text{div}$ , and 5 mV/div.

Figure 3-1 +5V Regulator Circuit Waveforms (Cont)

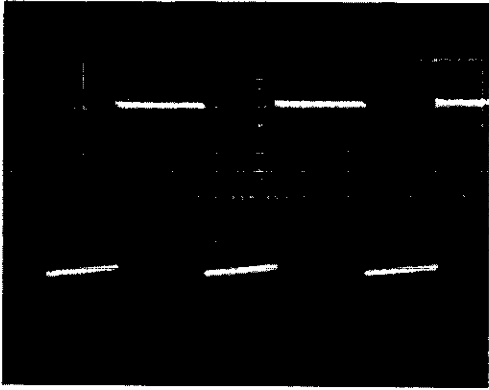


a) Point C, No load, 5 ms/div, and 10V/div.



b) Point C, No load, 50  $\mu\text{s}/\text{div}$ , and 10V/div.

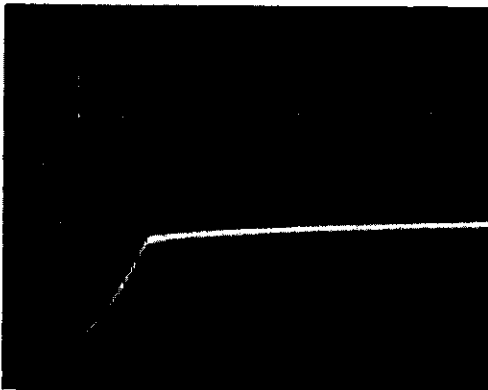
Figure 3-2 -15V Regulator Circuit Waveforms



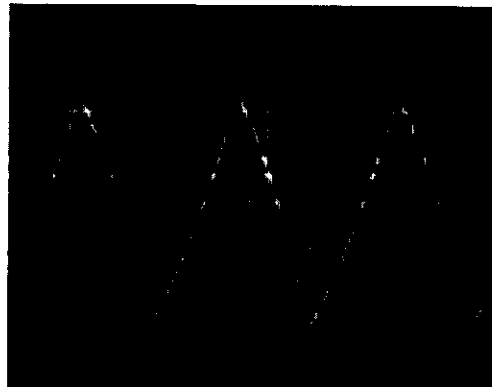
c) Point C, 5A load, 50  $\mu$ s/div, and 10V/div.



d) Point D, No load, 5 ms/div, and 50 mV/div.



e) Point D, No load, 5  $\mu$ s/div, and 50 mV/div.



f) Point D, 5A load, 50  $\mu$ s/div, and 5 mV/div.

Figure 3-2 -15V Regulator Circuit Waveforms (Cont)

### 3.4.1 Troubleshooting Rules

Troubleshooting rules for the power supply are listed as follows:

1. Make certain that power is turned off and unplugged before servicing the power supply.
2. Ensure that input capacitors C1 and C2 are discharged before servicing the power supply. A 10 to 100  $\Omega$ , 10W resistor can be used to hasten the discharge of the capacitors. (Be sure power is off.)
3. The dc regulator module is not internally grounded to the chassis; therefore, shorts to ground can be located after disconnecting the dc output cable to the system unit.
4. The dc output fuses F1 and F2 can be replaced without removing the dc regulator module. Before unsoldering fuses, observe cautions described in steps 1 and 2.
5. For proper operation, all hardware must be secured tightly to about 12 inch/pounds (i.e., capacitors, chokes, semiconductors). All hardware should be replaced with identical hardware replacement parts.
6. The dc regulator module may be removed from the top of the power chassis assembly while the latter is still bolted to the computer chassis. The dc regulator module is held in place by six screws.
7. When replacing power semiconductor components that are secured to the heat sink, apply a thin coat of Wakefield #128 compound or Dow Silicone Grease to the heat sink contact side (bottom) of the semiconductor. Insulating wafers are not required.

### 3.4.2 Troubleshooting Hints

#### CAUTION

Unplug computer before servicing.

The most likely source of power supply malfunction is the dc regulator module. A quick remedy for a malfunction may be to replace this entire module. The problem, however, could be a short in the system unit or possibly a defective component or other problem in the ac input circuit.

1. The +5V and -15V regulators contain overvoltage detection circuitry. If R50 or R26 are adjusted too far clockwise, the corresponding crowbar circuit will trip and blow fuses. To correct this condition: adjust the potentiometer fully counterclockwise, replace the blown fuse, and re-adjust as per Paragraph 3.2.
2. Make a visual examination of the circuitry. Check for burnt resistors, cracked transistors, burnt printed circuit board etch, oil leaking from capacitors, and loose connections. A visual check can be a quick method of locating the cause of a malfunction.

### 3.4.3 Troubleshooting Chart

In checking the various areas of the power supply, the rules listed in Paragraph 3.4.1 should be followed. The waveforms shown in Paragraph 3.3 provide a comparison for the troubleshooting readings. Table 3-2 provides the dc regulator troubleshooting chart.

Table 3-2  
Troubleshooting Chart

Problem	Cause
No +5V and +15V output	F1 opened D14 or transformer opened +5V adjusted too high *
+5V Output Too Low	Q5, D9, Q10, Q9, Q11, D12, or D10 Shorted C5 or C7 shorted R49, R50, R46, or R44 opened Q6, Q7, Q8, or D11 shorted A9, Q10, or D9 opened * R51, or R50 opened
+15V Output Too High	Q1 shorted E8 opened R35 or R36 opened

\* This set of causes makes the crowbar fire, which in turn blows the appropriate fuse.

Table 3-2 (Cont)  
 Troubleshooting Chart

Problem	Cause
-15V Output Too Low	F2 opened D14 or transformer opened Q25, D4, Q26, Q21, Q27, D7 or D5 shorted C14 or C12 shorted R22, R26, R25, or R29 opened Q22, Q23, Q24, or D6 shorted Q25, Q26, or D4 opened R26 or R27 opened * -15V adjusted too high *
AC LO L Won't Go High	Q13, Q14, or Q15 shorted Q16 or D3 opened R7, R3, R6, or R8 opened C9 shorted
AC LO L Won't Go Low and/or acts erratically on Power-On/Power-Off	Q13, Q14, or Q16 opened Q15 or D3 shorted R12, R13, R7, or R10 opened
DC LO L Won't Go High	Q19, Q20, or Q12 shorted Q17 or D3 opened R7, R2, or R6 opened C9 shorted
DC LO L Won't Go Low	Q19, Q20, or Q17 opened Q17 or D3 opened R7, R3, or R6 opened C9 shorted
DC LO L Won't Go Low and/or acts erratically on Power-On/Power-Off	Q19, Q20, or Q17 opened Q18 or D3 shorted R9, R10, R11, or R8 opened
No LTC L Signal	R55 opened D13 shorted
LTC L Going Too High	D13 opened

\* This set of causes makes the crowbar fire, which in turn blows the appropriate fuse.



### 3.5 PARTS IDENTIFICATION

Parts identification for the power supply is provided in the Engineering Drawing Manual. This includes the assembly drawings with associated parts lists, which list the respective unit parts, their part designations, and their DEC part numbers. These drawings and the respective drawing numbers are listed as follows:

1. Power Supply Chassis: E-1A-5309816-0-0
2. Power Control Board 115V: C-1A-5409824-0-0  
230V: C-1A-5409825-0-0
3. DC Regulator Module: E-1A-5409728-0-0  
D-CS-5409728-0-1 (schematic)
4. Power Supply Assembly and Fan: D-AD-7003731-0-0
5. AC Input Box Assembly: D-UA-H400-0-0
6. Line Set 115 Vac 7A: C-UA-BC05H-0-0  
230 Vac 5A: C-UA-BC05J-0-0

**PART 5**  
**Appendices**

APPENDIX A TO BE SUPPLIED

## APPENDIX B DETAILS OF KD11-B IR DECODE

### B.1 INTRODUCTION

Instruction decoding in the KD11-B is divided into two sections, microroutine selection and auxiliary ALU control. Paragraph B.2 describes the process of microroutine selection and the method of breaking down the instruction word used in the KD11-B. Paragraph B.3 describes the auxiliary ALU control. The division of IR decode into two distinct parts occurs because of the large number of PDP-11 instructions that require source-destination calculations.

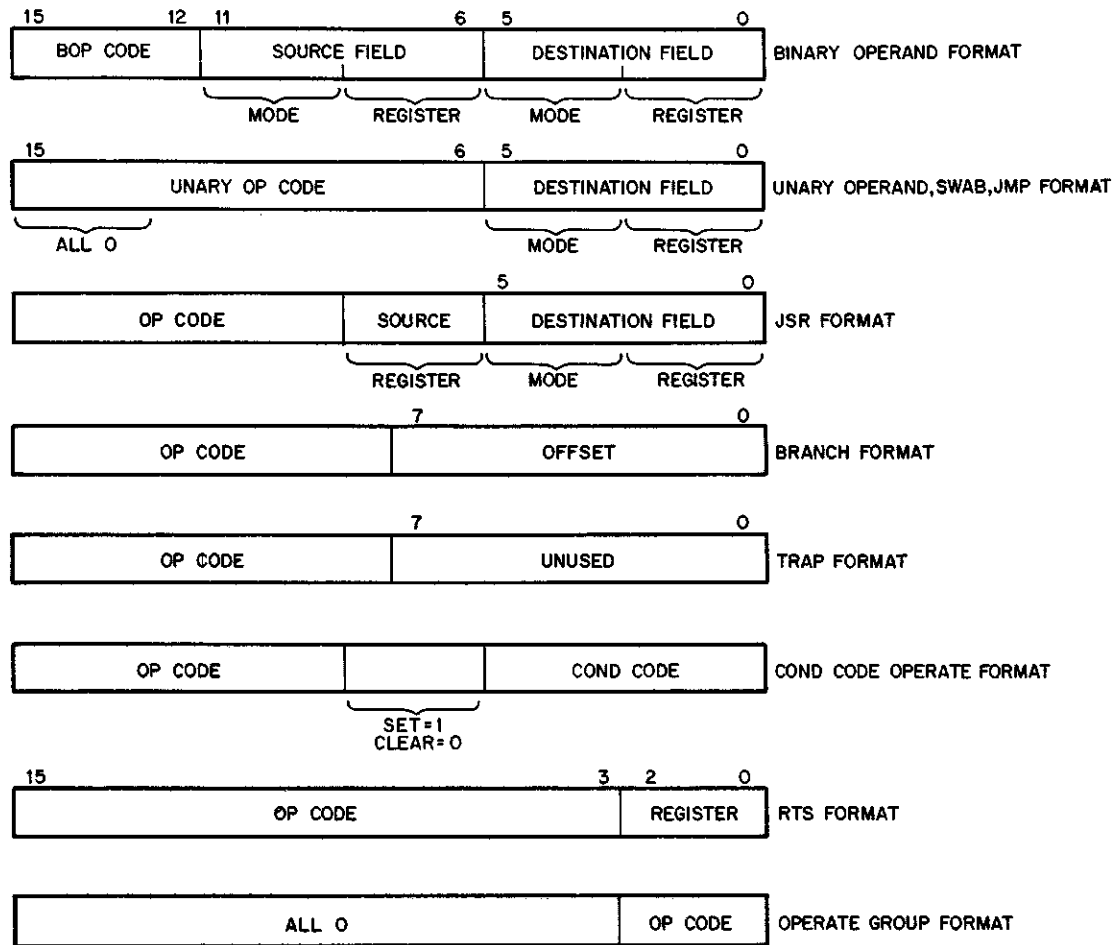
### B.2 MICROROUTINE SELECTION

To understand the decoder logic implementation it is necessary to be familiar with the format of the PDP-11 instructions and the method of microbranching used in the KD11-B. For the discussion of the KD11-B control store implementation, refer to Chapter 5 of Part II. A brief discussion of the PDP-11 instruction format follows below.

The following list contains some general rules for designing decoders for PDP-11 instructions:

1. In general, the PDP-11 operation code is variable from 4 to 16 bits.
2. Instructions are decoded from the most significant part of the word towards the least significant part of the word beginning with the most significant 4 bits.
3. There are a number of instructions that require two address calculations and a larger number that require only one address calculation. There are also a number of instructions that require address calculations, but do not operate on data.

4. All OP codes that are not implemented in the KD11-B must be trapped.
5. There are illegal combinations of instructions and address modes that must be trapped.
6. There exists a list of exceptions in the execution of instructions having to do with both the treatment of data and the setting of condition codes in the program status word.



11-1184

Note that the source fields and destination fields are common to a large number of instructions; thus, the process of decoding the instruction address format must be separated from the process of decoding the particular instruction operation. The KD11-B microprogram fetches the instruction, fetches the operands, allows an operation to occur on the operands, and then returns the modified destination operation (Chapter 5, Part II). The combinational logic that decodes the operation type for single and double operand instructions is shown on print DPF.

### B.2.1 Double Operand Instructions

Binary operand instructions are decoded by E066 shown on print DPG. If the instruction is of the binary operand class, the signal DPG CAL SOURCE L is asserted. This signal (ANDed with CONE BUT IR DECODE L) enables IR\* <11:9> to cause a microcode branch via the 74H01 gates contained in integrated circuit E072 as shown on the lower center of print DPG. The signal DPG CMP + BIT L, generated by E066, indicates that the particular binary operand instruction does not modify the destination operand. The signals DPG MOVE L and DPG MOVE BYTE L are used by the logic shown on print CONE. The following list of exceptions explains the use of the last three signals mentioned above.

INSTRUCTION	INDICATOR	EFFECT	EXCEPTION
CMP	DPG CMP + BIT L	set condition codes	Destination is not modified; therefore, DATIP not required.
BIT	DPG CMP + BIT L	set condition codes	Destination is not modified; therefore, DATIP not required.
MOVB	DPG MOVE L DPG BYTE L		If the destination is a register, i.e., destination mode 0, the result is sign extended. That is the sign of the low order byte is extended through the upper byte.
(ANY) BYTE	DPG BYTE L		Bit 0 of the address word must be used in determining which microroutine to use to position source and destination data. See Chapter 5, Part II for details.

The other signals generated by E066 do not affect the MPC, but affect the ALU and the condition does. The signal DPF CODE L is used on print DPF to switch the algorithm for setting the C and V bits.

DPF      CODE 0 L      ASSERTED  
UNASSERTED

\*IR <11:9> = bits 9 through 11 of the instruction register.

Note from Chapter 5, Part II and the flows that for a binary operand instruction the source operand is stored in R10 and the destination operand is temporarily stored in the B register. Then the control step  $B \leftarrow R10 \text{ OP } B$  is performed. Note also that the ALU can perform the operation A-leg minus B-leg, but not the converse. The CMP instruction requires the operation SOURCE minus DESTINATION, which is equivalent to A-leg minus B-leg. However, the SUB instruction requires the operation DESTINATION minus SOURCE. This is accomplished by storing the complement of the SOURCE in R10 for the SUB instructions only. Note that the signal CONE BUT DESTINATION L is an input to E066. The microprogram issued CONE BUT DESTINATION L, whenever the SOURCE operand is stored into R10. If the current instruction is a SUB, E066 issues the signals DPG DIS ALU S BITS H, CONF ALU SO L, and CONF ALU S2 L. This causes the complement of the B Reg to be stored in R10. When control step  $B \leftarrow R10 \text{ OP } B$  is performed for subtract instructions, the ALU operation is A-leg plus B-leg plus ONE, which is equivalent to DESTINATION minus SOURCE.

When the microprogram has completed the SOURCE calculation and retrieved the SOURCE OPERAND for a binary operand instruction, it generates the signal CONE BUT DESTINATION L. This signal is ORed and inverted to produce CONE BUT DESTINATION H. The MOV, MOVB, BYTE AND CMP + BYTE instructions are detected at the control steps listed below:

BIT PATTERNS	INSTRUCTION CLASS	ASSERTED SIGNALS
$\langle 11 \rangle = \langle 9 \rangle = \langle 8 \rangle = 1 +$ $\langle 10 \rangle = 0$	UNARY POTENTIAL TST	DPG CAL DEST L + DPG 54 L
$\langle 10:8 \rangle = 0 + \langle 11 \rangle = 0$	BRANCH	DPG CAL BRANCH L
$\langle 15:8 \rangle = 0$	OTHER	DPG ODD BYTE = OL

Two instructions in the other class require destination calculations: JMP and SWAB. These instructions are detected by F074 shown in the lower left-hand corner of DPG. Standard Unary instructions that affect or test the destination (with the exception of SWAB) are treated like binary instructions, i.e., the instruction is fetched, the operand is fetched, the operation is performed, and the operand is returned. The logic that decodes the

operation for  $B \leftarrow R10 \text{ OP } B$  is shown on print DPF. Note that for UNARY operand instructions the destination operand is copied into both R10 and B.

### B.2.2 Branch On Unary

There are three formats of instructions that require destination address calculations. The majority of the microcode destination routines are shared by all of the instructions that have destination fields. The ROM E071, shown in the upper right-hand corner of print DPG, is used to differentiate between the various instructions that use the microcode destination routines.

E071 is also used to detect illegal instruction combinations, which are defined as JMP or JSR and used with destination mode 0. The microcode flow chart for the KD11-B shows that in microstep D0-2 a test is made for Unary and illegal instruction by asserting the signal CONE BUT UNARY L. CONE BUT UNARY L produces the signal CONE ENAB UNARY L, which enables E071 (print DPG) to cause a microprogram branch. At other points in the microprogram such as D2-3, a test is made for a legal JSR or JMP instruction by the assertion of the signal CONE JMP + JSR L. The asserted signal CONE JMP + JSR L alters the input to E071 such that microroutines for legal JSR and JMP instructions are used. Note that CONE JMP + JSR L also causes the generation of the signal CONE ENAB UNARY L, which enables E071. The effect of E071 is shown in the following truth table:





### B.2.3 PDP-11 Branch Instruction

PDP-11 conditioned branch instructions are completely decoded by F059 shown on print DPG. E059 is enabled by the signal DPG CAL BRANCH L, which is asserted by E069 according to a previously discussed algorithm. IR <15> and IR <10:8> along with the condition codes N, Z, V, and C completely determine the branch instruction disposition. Note that the offset of a branch instruction is sign extended in microstep F-5 and shifted left one place in microstep B-1. Note that all successful branch instructions are interpreted by the microroutine that begins in B-1, while all unsuccessful branch instructions are interpreted by the microroutine that begins in B2-1.

### B.2.4 Operate Instructions

Operate instructions and instructions that set and clear condition codes are decoded by E074 and E064. NOPS, set condition code instruction, and clear condition code instructions all proceed from step F-5 to step CCM-1 in the microprogram. At step CCM-2, the microprogram performs a BUT DESTINATION to examine IR <4>. Set condition code instructions and the NOP-260 proceed with step SC-1 while clear condition code instructions and the NOP-240 proceed with step CC-1. Also in step CCM-1, the B Register is loaded with the contents of the instruction ANDed with  $17_8$ . This procedure zeroes all but the least significant 4-bits of the instruction copy contained in the B Register. Remember that the instruction is loaded into both the IR and B Register in step F-4. If the instruction is a SET COND CODE type, the operation is  $PSW \leftarrow B$  or  $PSW$  in step SC-1. Similarly for clear condition code instructions,  $PSW \leftarrow B$  and not  $PSW$  is performed in step CC-1. Note that even though the entire PSW is reloaded only the least significant 4 bits are affected by the sequence just described.

Other operate instructions such as WAIT, RTI, and HALT are decoded completely when BUT IR DECODE is issued during microstep F-5.

### B.3 AUXILIARY ALU CONTROL

The auxiliary ALU control consists of the ROMs E053, E061, and E068 shown on print DPF. These ROMs determine the operation to be performed whenever the microcode executes the action  $B \leftarrow R10 \text{ OP } B$ . E053 decodes binary operand instructions while the other two ROMs decode Unary operand instructions. Table B-1 tabulates the auxiliary control outputs for each binary and unary instructions.

Table B-1 Auxiliary Control for Binary and Unary Instructions

Instruction	Condition Codes			ALU Function	CIN	B	SP
	N and Z	V	C				
MOV (B)	LOAD	CLEAR	not affected	A Logical	∅	LOAD	
CMP (B)	LOAD	LOAD like SUBTRACT	LOAD like SUBTRACT	A - B - 1	+1	LOAD	
BIT (B)	LOAD	CLEAR	not affected	A + B	∅	LOAD	
BIC (B)	LOAD	CLEAR	not affected	~A + B	∅	LOAD	
BIS (B)	LOAD	CLEAR	not affected	AB	∅	LOAD	
ADD	LOAD	OP's same sign and result different	SET if carry out	A plus B	∅	LOAD	
SUB	LOAD	+ - (-) = - - (-) (+) = + SET	SET if $\bar{C}$ arry	A plus B	+1	LOAD	
CLR (B)	LOAD	CLEAR [ like ADD]	CLEAR	∅	∅	LOAD	
COM (B)	LOAD	CLEAR	SET	~B Logical	∅	LOAD	
INC (B)	LOAD	SET if rest held was 1∅∅ ∅∅∅ before OP	not affected		+1	LOAD	

B-9

Table B-1 (Continued)

## Auxiliary Control for Binary and Unary Instructions

Instruction	Condition Codes			ALU Function	CIN	B	SP
	N and Z	V	C				
NEG (B)	LOAD	SET if result is <del>100 000</del>	CLEARED if res=0 SET otherwise	A - B - 1	+1	LOAD	∅
ADC (B)	LOAD	SET if DET was <del>077777</del> and C = 1	SET if DST was 177777 and C = 1	A Arithmetic	+C	LOAD	
SBC (B)	LOAD	SET if DST and 100 000 and C = 1	CLEARED if result = ∅ and C = 1 - SET otherwise	A - B	+NC		
TST (B)	LOAD	CLEAR	CLEAR	A Logical	∅	LOAD	
ROR (B)	Z ← < C: ∅1> N ← C	N ⊕ C	<0>			SHIFT RIGHT	
ROL (B)	Z < 14: C> N ← <14>	N ⊕ C	<15> B <7>			SHIFT LEFT	
ASR (B)	Z ← <15: ∅1> N ← N	N ⊕ C	C ← 15			SHIFT RIGHT	
ASL (B)	2 ← <14: 1> N ← <14>	N ⊕ C	C ← 15			SHIFT LEFT	

APPENDIX C  
COMPUTER CONNECTORS

Table C-1 lists the computer connectors, the connector type, part number, pin and signal designations, and the associated connector cable. This includes the connectors for the SCL cable that interfaces the computer (Berg) connector to an LA30 or Model 33 ASR Teletype equivalent (Mate-N-Lok) connector. The power supply connectors are described in Part 4 of this manual.

Table C-1  
Connectors

Connector	Type	Part Number	Designations		Cable
			Pin	Signals	
SCL Connector	40 pin BERG	549949 (Female) 1270090-0 (Male)	BB	-15 V	8820
			V	SER 0+ (20 mA)	
			T	CLK IN (TTL)	
			DD	SER IN - (20 mA)	
			R	READER RUN - (20 mA)	
			N	CLK DISAB (TTL)	
			L	SER 0 - (20 mA)	
			C	+ 5V	
			D	SER 0 (TTL)	
			F	READER RUN + (20 mA)	
			RR	SER IN (TTL)	
			NN	20 mA INTERLOCK	
			LL	SERIAL IN + (20 mA)	
TELETYPE or LA30 Connector	p pin MATE-N-LOK (Female)	1209340	2	SER 0 -	83600
			3	-15 V	
			4	-15 V	
			5	SER 0 +	
			6	READER RUN	
			7	SER IN	

Table C-1 (Cont)  
Connectors

Connector	Type	Part Number	Designations		Cable
			Pin	Signals	
CONSOLE	40-pin Berg connector	549949 (Female) 1270090-0	PP	DAK H	BC08R-03
			BB	SW 15 (1) H	
			DD	SW 14 (1) H	
			FF	SW 13 (1) H	
			JJ	SW 12 (1) H	
			LL	SW 11 (1) H	
			NN	SW 10 (1) H	
			RR	SW 09 (1) H	
			TT	SW 08 (1) H	
			J	SW 07 (1) H	
			L	SW 06 (1) H	
			N	SW 05 (1) H	
			R	SW 04 (1) H	
			T	SW 03 (1) H	
			V	SW 02 (1) H	
			X	SW 01 (1) H	
			Z	SW 00 (1) H	
			HH	SCAN ADRS 01 (1) L	
			KK	SCAN ADRS 02 (1) L	
			MM	SCAN ADRS 03 (1) L	
			SS	SCAN ADRS 04 (1) L	
			CC	PUP L	
			C	RUN L	
			E	KEY LOAD ADRS (1) L	
			H	KEY EXAM (1) L	
			K	KEY CONT (1) L	
			M	KEY HLT ENB (1) L	
P	KEY START (1) L				
S	KEY DEP (1) L				
UNIBUS	M920 or M930	-----	AA1	INIT L	
			AA2	POWER (+5V)	
			AB1	INTR L	
			AB2	GROUND	
			AC1	D00 L	
			AC2	GROUND	
			AD1	D02 L	
			AD2	D01 L	
			AE1	D04 L	
			AE2	D03 L	

Table C-1 (Cont)  
Connectors

Connector	Type	Part Number	Designation		Cable
			Pin	Signals	
UNIBUS (cont)			AF1	D06 L	
			AF2	D05 L	
			AH1	D08 L	
			AH2	D07 L	
			AJ1	D10 L	
			AJ2	D09 L	
			AK1	D12 L	
			AK2	D11 L	
			AL1	D14 L	
			AL2	D13 L	
			AM1	PA L	
			AM2	D15 L	
			AN1	GROUND	
			AN2	PBL	
			AP1	GROUND	
			AP2	BBSY L	
			AR1	GROUND	
			AR2	SACK L	
			AS1	GROUND	
			AS2	NPR L	
			AT1	GROUND	
			AT2	BR 7 L	
			AU1	NPG H	
			AU2	BR 6 L	
			AV1	BG 7 H	
			AV2	GROUND	
			BA1	BG 6 H	
			BA2	POWER( +5V)	
			BB1	BG 5 H	
			BB2	GROUND	
			BC1	BR 5 L	
			BC2	GROUND	
			BD1	GROUND	
			BD2	BR 4 L	
		BE1	GROUND		
		BE2	BG 4 H		
		BF1	ACL0L		
		BF2	DCL0L		
		BH1	A01L		



Table C-1 (Cont)  
Connectors

Connector	Type	Part Number	Designation		Cable
			Pin	Signals	
UNIBUS (cont)			BH2	A00L	
			BJ1	A03L	
			BJ2	A02L	
			BK1	A05L	
			BK2	A04L	
			BL1	A07L	
			BL2	A06L	
			BM1	A09L	
			BM2	A08L	
			BN1	A11L	
			BN2	A10L	
			BP1	A13L	
			BP2	A12L	
			BR1	A15L	
			BR2	A14L	
			BS1	A17L	
			BS2	A16L	
			BT1	GROUND	
			BT2	C1 L	
			BU1	SSYN L	
BU2	C0 L				
BV1	MSYN L				
BV2	GROUND				
AC Remote Power Turn-On Connector	2-3 pin Mate-N- Loks (J6 and J7)	DEC 2- 09350-03 (Plug is DEC12-09351	1 2 3	Power Request Emergency shutdown Ground	Power Supply AC Cable
Line Cord Connector	AC Line Plug				(110V) BC05H (230V) BC05J

## APPENDIX D INTERFACE CIRCUITS AND HARDWARE

### D.1 INTRODUCTION

The specific circuits, modules, and hardware used for interfacing devices to the Unibus are described in this Appendix.

### D.2 CIRCUITS

The Unibus high-speed data transmission facility imposes certain restrictions when attaching other devices to it. The actual bus is a matched and terminated transmission line that must be received and driven with devices designed for that specific application. The following paragraphs describe bus transmission, bus signal levels, bus length, and bus receiver and transmitter circuits.

#### D.2.1 Unibus Transmission

The actual bus medium consists of several types of cable. The standard cabling comprises short jumper modules (M920 modules) that interconnect system units within a mounting box. Critical ground signals are also carried on these jumper modules. The cables interconnecting BA11 Mounting Boxes consist of a Mylar<sup>®</sup> cable assembly with alternating signals and ground. The characteristics necessary for proper Unibus transmission are:

Characteristic Impedance:	120 $\Omega$ $\pm$ 15%
Resistance:	0.13 $\Omega$ /ft. maximum

<sup>®</sup> Mylar is a registered trademark of E. I. DuPont deNemours.

Either twisted pair or coaxial cable laid for minimum crosstalk is recommended for long cable lengths and for applications requiring extreme physical durability of the cable.

The Unibus is terminated at each end by a resistive divider for each signal except the grant signals (Figure D-1). The grant signals are terminated with a single resistor. Two M930 Terminator Modules are included in every system to provide these functions.

### D.2.2 Unibus Signal Levels

The rest state for all Unibus signal lines, except the grant lines BG <7:4> and NPG, is a logic 0 of +3.4V. The asserted state (logic 1) is between ground and +0.8V, which is the saturation voltage of the device driving the bus. The rest state for the grant signals is ground (logic 0), and the asserted state (logic 1) is +3.4V. To guarantee operation under worst-case conditions, receivers should have a switching threshold of approximately 2V.

DEC uses standard terminology to name signal lines to aid the reader in determining their active state. Either an H or L follows the signal name mnemonic and is separated by a space. This letter indicates the asserted (logic 1) state of the signal to be either high (approximately +3V) or low (ground). Thus, a Unibus data line is called BUS D00 L, and a grant line is called BUS BG4 H. All signals that are not Unibus signals are characterized in terms of standard transistor-transistor logic (TTL) loads. These devices, which are similar to the 7400 Series, have a low state input load of -1.6mA and a high state leakage current of 40mA. Outputs are characterized by the number of inputs they can drive (called fanout). A standard TTL gate (as used in the M113) can drive 10 unit loads.

### D.2.3 Unibus Length and Loading

The maximum length of the Unibus is a complex relationship involving the type of cable, the bus loading, and distribution of receiver and transmission taps on the bus. Since the Unibus is a transmission line, and the transfers are asynchronous and interlocked, the electrical delay imposed by length is not a factor.

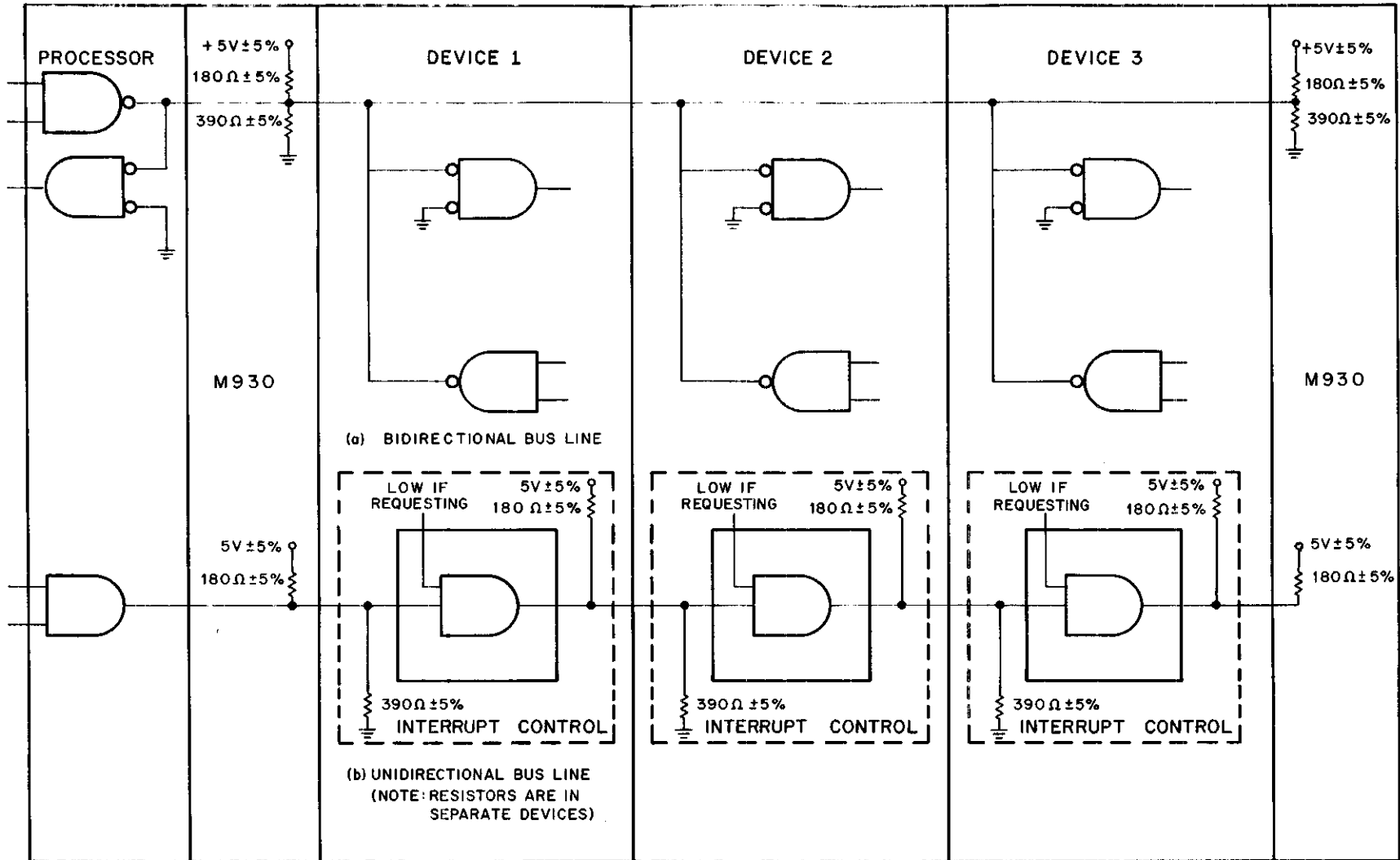


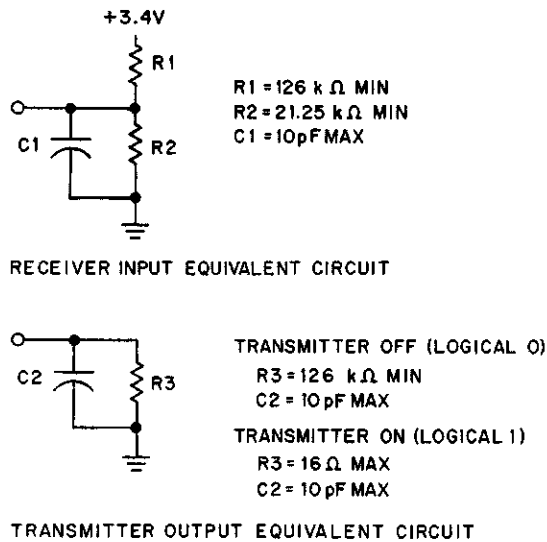
Figure D-1 Bus Terminations for Bidirectional (a), and Unidirectional (b), Bus Lines

With Flexprint cable (Tape Cable S-1680), the maximum reasonable length is 50 feet, minus the combined length of all stubs or taps, which are those wires from the actual bus to the receivers and transmitters. This maximum length is obtainable only if the individual tap lengths are less than 18 inches, and if the loading is not more than a standard of one receiver and two transmitters. If loads are concentrated at one end of the Unibus and a single load is at a distant point, the maximum length could change, provided that the crosstalk of the employed cable is low enough.

The Unibus is limited to a maximum of 20 unit loads. This limit is imposed because of the loading of receivers and leakage of drivers at the high state. This limit is set to maintain a sufficient noise margin. For more than 20 unit loads, a Unibus repeater option (DB11-A) may be used.

#### D.2.4 Bus Receiver and Transmitter Circuits

The equivalent circuits of the standard UNIBUS receivers and transmitters are shown in Figure D-2. Any device that meets these requirements is acceptable. To perform these functions, DEC uses two monolithic integrated circuits with the characteristics listed in Table D-1. Typical transmitter and receiver circuits are shown in Figure D-3.



11-0031

#### Transmitter Output Equivalent Circuit

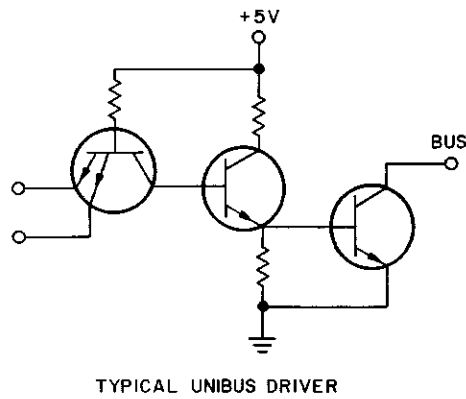
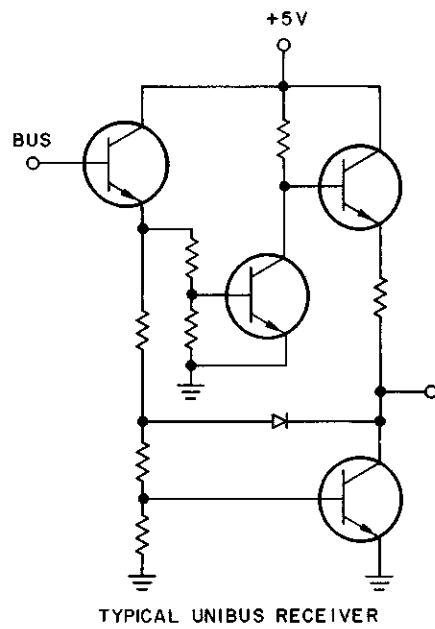
Figure D-2 Transmitter and Receiver Equivalent Circuits

Table D-1  
Unibus Receiver and Transmitter Characteristics

Device	Characteristics	Specifications	Notes	
Receiver (DEC 380A)	Input high threshold	V <sub>IH</sub> 2.5V min.	1	
	Input low threshold	V <sub>IL</sub> 1.4V max.	1	
	Input current @2.5V	I <sub>IH</sub> 160 μA max.	1,3	
	Input current @ 0V	I <sub>IL</sub> ±25 μA max.	1,3	
	Output high voltage	V <sub>OH</sub> 3.5 min.	2	
	Output high current	I <sub>OH</sub> -2 mA	2,3	
	Output low voltage	V <sub>OL</sub> 0.6V max.	2	
	Output low current	I <sub>OL</sub> -12.5 μA	2,3	
	Propagation delay to high state	TP <sub>DH</sub> 10 ns min. 45 ns max.	4,5	
	Propagation delay to low state	TP <sub>DL</sub> 10 ns min. 35 ns max.	4,5	
	Transmitter (DED 8881)	Input high voltage	V <sub>IH</sub> 2.0V min.	6
		Input low voltage	V <sub>IL</sub> 0.8V max.	6
Input high current		I <sub>IH</sub> 60 mA max.	6	
Input low current		I <sub>IL</sub> -2.0 mA max.	6	
Output low voltage @ 50 mA sink		V <sub>OL</sub> 0.8V max.	1	
Output high leakage current @ 3.5V		I <sub>OH</sub> 25 mA max.	1,3	
Propagation delay to low state		TP <sub>DL</sub> 25 ns max.	5,7	
Propagation delay to high state		TP <sub>DH</sub> 35 ns max.	5,8	

NOTES

1. This is a critical parameter for use of the Unibus. All other parameters are shown for reference only.
2. This is equivalent to being capable of driving seven unit loads of standard 7400 Series TTL integrated circuits.
3. Current flow is defined as positive if into the terminal.
4. Conditions of load are 375 ohms to +5V and 1.6 ohms in parallel with 15 pf to ground.
5. Times are measured from 1.5V level on input to 1.5V level on output.
6. This is equivalent to 1.25 standard TTL unit loading of input.
7. Conditions of 100 ohms to +5V, 15 pf to ground on output.
8. Conditions of 1K ohms to ground on output.



11-0030

Figure D-3 Transmitter and Receiver Typical Circuits

**digital**

**DIGITAL EQUIPMENT CORPORATION  
MAYNARD, MASSACHUSETTS 01754**