DIGITAL EDUCATION AND TRAINING

digital™

EY-9254E-SG-0001

PDP-11/70 Diagnostics and Module Level Repair
Student Guide

Educational Services

digital™

EY-9254E-SG-0001

PDP-11/70 Diagnostics and Module Level Repair

Student Guide

Printed in U.S.A.

PDP-11/70 DIAGNOSTICS
STUDENT GUIDE/LAB GUIDE

To the Student:

This outline is furnished as a guide to assist you in following the instruction presentations. It also has a list of references pertaining to subject matter.

DIGITAL is aware that students come from diversified backgrounds and vary in levels of experience, therefore, the instructor has the right to vary its contents in meeting individual requirements.

TABLE OF CONTENTS

COURSE OUTLINE

Day 1

Lecture

I. System Overview
   A. System Block Diagram
   B. Introduction to the UNIBUS
   C. System Addressing-Mapping Modes

II. UNIBUS Concepts
   A. Unibus Signals
   B. Interlocked Communications
   C. Data Transfers
   D. Interrupts
   E. Direct Memory Access DMA
   F. Priority
   G. Power Control
   H. Initialize

References

PDP-11/70 Processor Handbook, Chapters 2,4,10
(EB-19402-20,EB-26077-41)
PDP-11 Bus Handbook (EB-17525-20)
Student Guide/Laboratory Guide Assignment 1
KB11-C Processor Technical Manual, Section I
(EK-KB11C-TM-001)

Day 2

Lecture

III. Review Console Operations
- A. Load Address
- B. DEP/EXAM to Memory
- C. DEP/EXAM to GPRs
- D. Consecutive DEP/EXAMs
- E. Single Instruction Cycle
- F. Single Bus Cycle
- G. Identify Various LEDs and Switches

IV. Stack Operations
- A. Initialization
- B. violations
- C. Stack Limit Register

V. Interrupts/Traps/Aborts
- A. Hardware
- B. Software (PIRQ Register)

VI. Logic Symbology

References

PDP-11/70 Processor Handbook, Chapters 3&4
(EB-19402-20,EB-26077-41)
PDP-11/70 Processor Handbook, Chapters 4,5,10
(EB-19402-20,EB-26077-41)
KB11-C Processor Techical Manual, Section III (EK-KB11C-TM-001)
KB11-C Processor Techical Manual, Section II,Chapter 6
(EK-KB11C-TM-001)
Student Guide/Laboratory Guide Study Assignments 2,3

Laboratory

Console Familiarization Laboratory, Student Guide/Laboratory Guide

Day 3

Lecture

VII. DL11-W Interface Module
- A. Functional Description
  1. Control and Status Registers
  2. Data Buffers
  3. Interrupt Vector
  4. real Time Clock
- B. Logic Overview of Prints

VIII. M9312 Bootstrap Terminator
- A. General Description
  1. UNIBUS Termination
  2. Bootstrapping
  3. Diagnostics
  4. Console Emulator
- B. Description of Boot Procedures
  1. Powerup Boot Sequence
  2. External Boot Porcedure
  3. Front Panel ROM selection
- C. Physical Characteristics
  1. Switch Settings
  2. Jumper Selection
  3. Faston Tabs
- D. Overview of Logic

References

DL11-W Technical Manual (EK-DL11W-TM-001)
M9312 Technical Manual, Chapters 1-3, Appendix A (EK-M9312-TM-001)
M9312 Diagnostic Listing and Error Address List, Student Guide/ Laboratory Guide
PDP-11/45 & 11/70 Course Hardware Drawings
Student Guide/Laboratory Guide Studey Assignment 4

Laboratory

Do DL11-W Laboratory from Student Guide/Laboratory Guide.
Do M9312 Bootstrap terminator laboratory.
Student Guide/Laboratory Guide.
Will put in M9312 Bugs.
Toggle in Programs and Single Step Useing Front Panel Switches and Lights.

Day 4

Lecture

IX. PDP-11/70 Diagnostics
   A. Monitor Commands
   B. Execution and Interpretation

X. Power Supply System (CPU)
   A. AClow and DClow
   B. Switched and Unswitched Outlets
   C. Power Block Diagram
      1. 861 D/E Power Controller
      2. H7420 Power Regulators
      3. Backplane Connections
   D. Physical Layout
   E. Schematic Overview of Monitor Circuit, +15V regulator, +5V Regulator

References

PDP-11/70 Maintenance and Installation Manual, Chapter 3, Chapter 5 (EK-11070-MM-001)
PDP-11/70 Maintenance and Installation Manual, Chapter 4, (EK-11070-TM-001)
PDP-11/70 Maintenance Card, Page 1
PDP-1/45 & 11/70 Course Hardware Drawings
Student Guide/Laboratory Guide Study Assignment 5

Laboratory

Do power laboratory from Student Guide/Laboratory Guide
Do running XXDP laboratory from Student Guide/Laboratory Guide
Trouble Shoot power problems

Day 5

Lecture

XI. Central Processor Block Diagram
  A. Control Path
    1. Self-sequencing Control ROM (ROM word)
    2. Branch and Fork Logic
  B. Data Path
    1. Data Bus Connections
    2. General Purpose Registers
    3. ALU Section

XII. Timing Generator Overview
  A. Main Timing States
    1. T1-T5
    2. TS1-TS5
    3. Various Free Running Clocks

XIII. Flow Diagram Conventions and Symbology

XIV. Timing State Analysis of Flows 14 (Console Flows)
  A. Load Address
  B. DEP/EXAM to Main Memory
  C. DEP/EXAM to GPRs
  D. Single Instruction Cycle

XV. Overview of Instruction Flows (Flows 1-11)
  A. ROM Cycle Analysis of Following Instructions:
    1. CCOP
    2. Branch
    3. Single Operands*
    4. Double Operands*
    * Use both memory reference and nonmemory reference examples

References

PDP-11/70 Student Guide/Laboratory Guide
KB11-C Processor Technical Manual, Section II, Chapter 1 pp.1-11 Section II, Chapter 4, Section II, Chapter 1 p. 31, Section II, Chapter 1 pp.10-31 (EK-KB11C-TM-001)

Laboratory

Maintenance Card Exercise, Installation Manual, Chapter 5 pp.54-62
Do the Maintenance Card Laboratory For Flows 14 in the Student Guide/Laboratory Guide.
Troubleshooting Console Problems to Board Level using the Maintenance Card.

Day 6

Lecture

XVI. Overview of Service Flows (Flows 12-13)

A. Timing State Analysis of following Operation
   1. PUP Sequence
   2. External Interrupt

XVII. Review Major CPU functions Again using Troubleshooting Chart in PDP-11/70 Maintenance and Installation Manual Chapter 5 (EK-11070-MM-001)

A. Overview Important Functions Performed by each Module:
   1. RAC
   2. TIG
   3. GRA & DAP
   4. IRC
   5. TMC & UBC
   6. PDR

References

KB11-C Processor Manual Chapter 6 (EK-KB11C-TM-001)
PDP-11/70 Maintenance and Installation Manual, Chapter 5 p. 5, pp. 15-27, pp. 63,64
PDP-11/70 Maintenance Card p. 6

Laboratory

Service Flow Exercise, Student Guide/Laboratory Guide
Troubleshoot CPU problems to board level using EKBA-EKBB Diagnostics

Day 7

Lecture

XVIII. Memory Management
- A. Main Concepts
  1. Relocation (Virtual Addresses)
  2. Protection
  3. Statistical Information
  4. Mapping Modes
- B. Block Diagram Overview
  1. Control Section
  2. Selection of PAR/PDR Pair
  3. Relocation Section
  4. Abort/Trap Section
  5. Memory Management Registers
- C. Review Other Functions Performed
  1. Address Display to Front Panel
  2. Address Validity Check
  3. System Size Registers (p. 43 of Maintenance and Installation Manual (EK-11070-MM-001)
  4. Register Address Decode
  5. Console Cabling

XIX. UNIBUS Map Module
- A. Main Concepts
  1. Relocation (UNIBUS Addresses)
  2. Cache Register Operations
- B. Block Diagram Overview
  1. Accessing Map Registers
  2. Relocation Section
  3. Read/Write Paths
  4. Limit Comparator Jumpers

References

KB11-C Processor Manual, Section IV Chapters 1-6,9 (EK-KB11C-TM-001)
KB11-C Processor Manual, Section V (EK-KB11C-TM-001)
PDP-11/70 Maintenance and Installation Manual, Chapter 5 pp.27-32, Chapter 5 p. 65, Chapter 3 pp. 40-43 (EK-11070-MM-001)
PDP-11/70 Maintenance and Installation Manual, Chapter 5 pp. 33,65 (EK-11070-MM-001)
PDP-11/70 Maintenance Card pp. 7,8,9,12

Laboratory

Do Memory Management Exercise in Student Guide/Laboratory Guide.
Relocate Virtual Addresses to Physical Address from Front Panel using address switch.
Troubleshoot Memory Management problems to board level using Diagnostics EKBE.
Troubleshoot Unibus Map problems to board level using Diagnostics EKBF.

Day 8

Lecture

XX. Cache Memory
- A. Cache Concepts
  1. General Characteristics
  2. How Address is Processed
- B. Block Diagram Overview
  1. Address Section - ADM Module (Cabling)
  2. Data Section _ DTM Module
  3. Data Section Connecting to Main Memory CDP Module (Cabling)
- C. Cache Operations
  1. Read Hit
  2. Read Miss
  3. Write Hit
  4. PUP Sequence
  5. Parity Checking and Generation
- D. Cache Registers
  1. Error Address
  2. Control
  3. Hit/Miss
  4. Memory System Error
  5. Maintenance
- E. Diagnostics
  1. EKBC ---> CCB,CDP
  2. EKBD ---> DTM,ADM
  3. How to Run with Cache Disabled
- F. Overview of Cache Control Board (M8142)
  1. Arbitration
  2. PUP Sequencer
  3. Random Flip-Flop
  4. Timing Sequencer
  5. Valid Bit

Day 8 (Cont.)

Lecture

H. Summary of Cache Operations
J. 11/70 Trouble analysis

References

KB11-C Processor Manual, Section VI, Chapters 1-2
(EK-KB11C-TM-001)
PDP-11/70 Maintenance and Installation Manual, Chapter 5 pp.33-41,
p. 64 (EK-11070-MM-001)
KB11-C Processor Manual Section IV, Chapters 1-6,9
(EK-KB11C-TM-001)
PDP-11/70 Maintenance and Installation Manual, Chapter 5 pp. 27-32
Chapter 5 p. 65, Chapter 3 pp.40-43 (EK-11070-MM-001)
PDP-11/70 Maintenance Card pp. 5-8

Laboratory

Do the Cache Register Laboratory in the Student Guide/Laboratory Guide.
Troubleshooting Cache Problems to Boad Level Using Cache Diagnostic to Isolate Failures.

Day 9

Lecture

XXI. RH-70 Massbus Controllers

A. RH-70 Concepts
   1. Basic Read/Write Operations to Disk or Tape
B. Massbus Protocol
   1. Data Bus
   2. Control Bus
C. MBC/Cache Bus
D. Registers
   1. Bit Field Descriptions
   2. Local Register Read/Write Operations
   3. Remote Register Read/Write Operations
E. Data Paths
   1. FIFO Buffer - Command Data Paths
   2. Data Buffer Maintenance Operations

References

RH-70 Massbus Controller Handout

Laboratory

Troubleshoot the RH-70 to the Module Level useing the Diagnostics

Day 10

Lecture

XXII. Core Memory

A. Power Supply
   1. Physical Layout
   2. Voltage Test Points
B. Box Configuration
   1. MJ11-A vs. MJ11-B Backplanes
   2. Description of Stack Modules
   3. Configuration Error
   4. Mismatch Error
C. External Interleaving
   1. Description of Two-way Interleaving
   2. Switch Configuration of M8148/47
D. Cabling
   1. Between Cache and Main Memory
   2. Between Memory Boxes
   3. Basic Bus Signals
E. Diagnostics
   1. EMJA (Core only)
B[B 2. EMKA (Core or MOS)
   3. Interpretation of Configuration Map
G. Summary of Worst Case Conditions
   1. Bad Power
   2. Bad Cabling
   3. Bad Stack 0

XXIII. MOS Memory

A. Power Supply
   1. Physical Layout
   2. Voltage Test Points
B. Box Configutration
   1. Layout of Array Board and Controllers
   2. Description of Various Size MOS Boards
C. Box Controller
   1. Interleave Switch
      a. Two-way
      b. Four-way
   2. Starting Address Switch
   3. Description of Other Switches and LEDs Available on Controller

Day 10 (Cont.)

Lecture

D. Bus Cabling
   1. Between Cache and Main Memory
   2. Between Memory Boxes
   3. Basic Bus Signals
E. System Size Register
F. Diagnostics
   1. EMKA
   2. Interpretation of Configuration Map
G. Worst Case Conditions
   1. Bad Power
   2. Bad Cabling
   3. Bad Low Arrays

References

MJ11 Technical Manual (16K and 32K Core) (EK-MJ11-MM-002)
MK11 Technical Manual (MOS Memory) (EK-0MK11-TM-001)
MOS Memory Maintenance Card
PDP-11/70 Maintenance and Installation Manual, Chapter 3 pp. 34-60 and Chapter 4 pp. 21-67, 74-100 (EK-11070-MM-001)

Laboratory

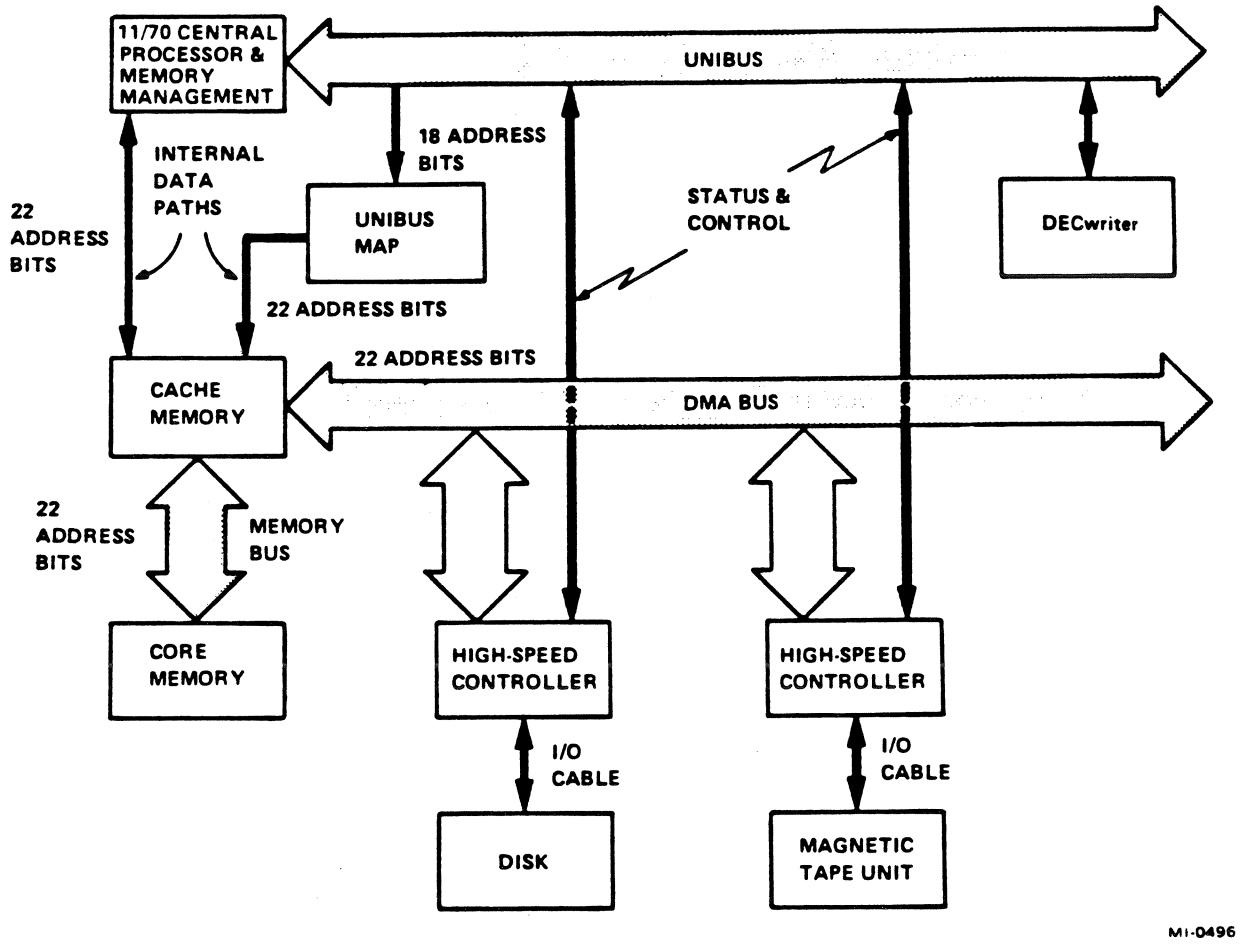
Troubleshoot the Memory to the Module Level using the Diagnostics

PDP-11/70 Diagnostics and Module Level Repair

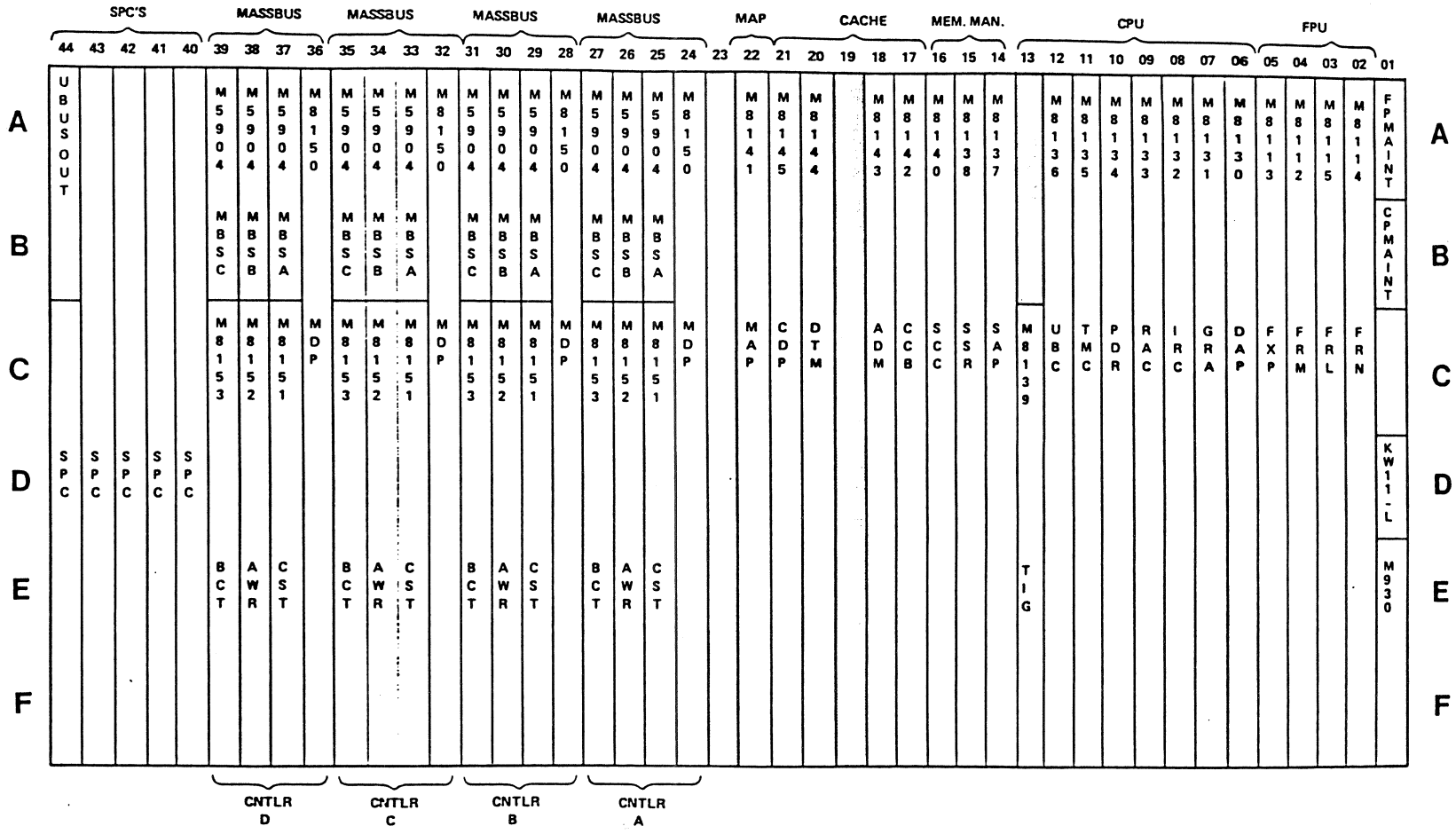
DAY ONE

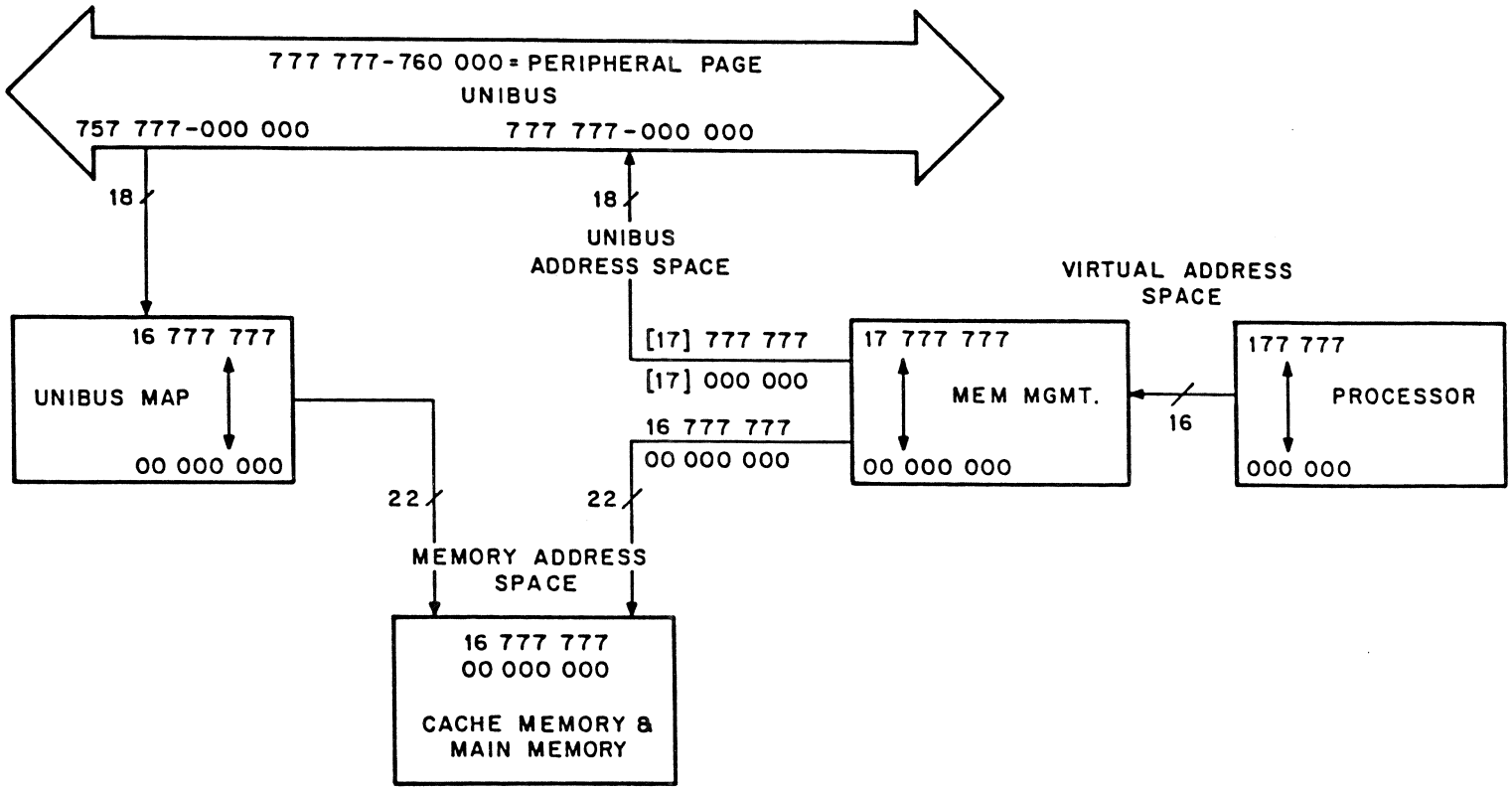
Major Characteristics of the 11/45, 11/50, 11/55, and 11/70

	11/45 1 /50 & 11/55	11/70
PRINCIPAL MARKETS	OEM & End User	OEM & End User
PROCESSING MODES	Kernel Supervisor User	Kernel Supervisor User
NUMBER OF GPRs	16	16
NUMBER OF HARDWARE INTERRUPT LEVELS	4	4
NUMBER OF SOFTWARE INTERRUPT LEVELS	7	7
MAXIMUM ADDRESS SPACE (Words)	128K	2 million
MAXIMUM MEMORY SIZE (words)	124K	1.9 million
BUS STRUCTURE	Unibus & Fast Bus	Unibus DMA Bus Memory Bus
TYPES OF MAIN MEMORY	Core MOS Bipolar	Core & Bipolar Cache
NUMBER OF HARDWARE STACKS	3	3
MEMORY MANAGEMENT	Standard	Standard
STANDARD INSTRUCTION SET	Basic PDP-11 instruction set plus 11 new instructions	

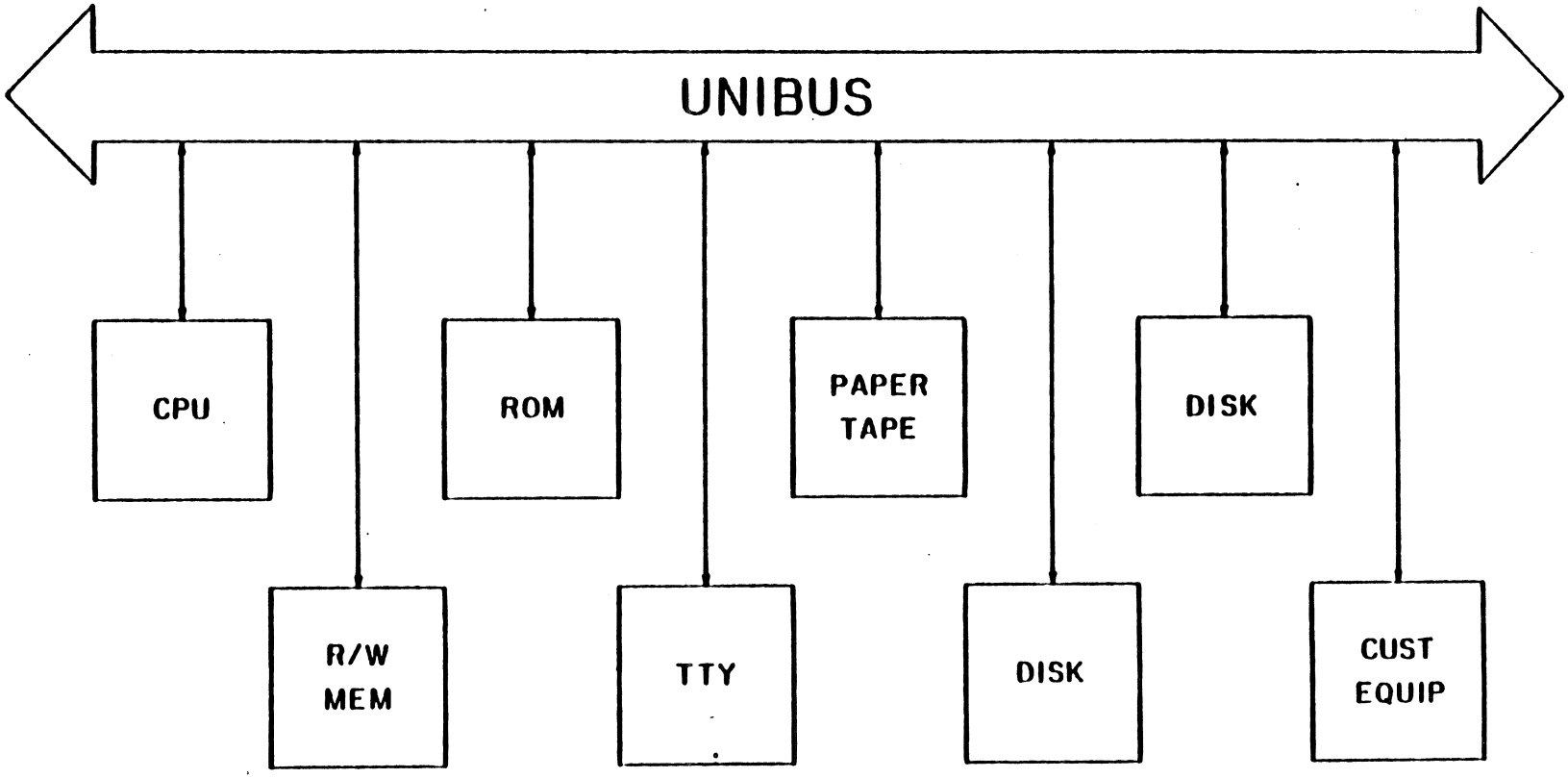


Typical PDP-11/70 Computer System


SPC'S
MASSBUS
MASSBUS
MASSBUS
MASSBUS
MAP
CACHE
MEM. MAN.
CPU
FPU
44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01
A
B
C
D
E
F
UBUSOUT
SPC
M8150
M8153
BCT
M8152
AWR
M8151
CST
MDP
M8141
MAP
M8145
CDP
M8144
DTM
M8143
ADM
M8142
CCB
M8140
SCC
M8138
SSR
M8137
SAP
M8139
TIG
M8136
UBC
M8135
TMC
M8134
PDR
M8133
RAC
M8132
IRC
M8131
GRA
M8130
DAP
M8113
FXP
M8112
FRM
M8115
FRL
M8114
FRN
FPMAINT
CPMAINT
KW11-L
M930
CNTLR D
CNTLR C
CNTLR B
CNTLR A


777 777-760 000 = PERIPHERAL PAGE
UNIBUS
757 777-000 000
777 777-000 000
18
18
UNIBUS
ADDRESS SPACE
VIRTUAL ADDRESS
SPACE
16 777 777
UNIBUS MAP
00 000 000
[17] 777 777
[17] 000 000
17 777 777
MEM MGMT.
00 000 000
16
177 777
PROCESSOR
000 000
16 777 777
00 000 000
22
22
MEMORY ADDRESS
SPACE
16 777 777
00 000 000
CACHE MEMORY &
MAIN MEMORY

11-4019

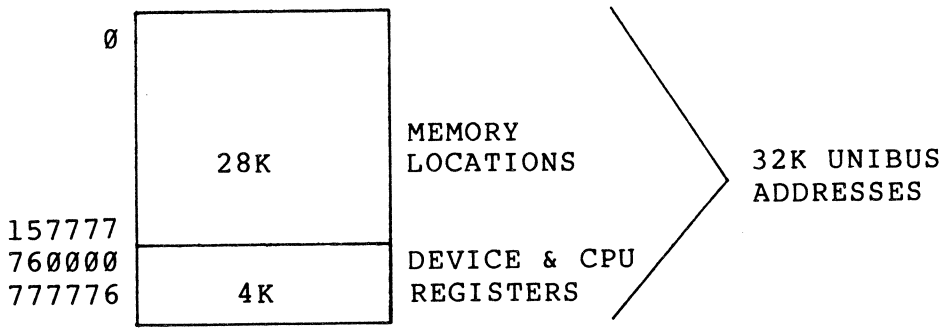


FEATURES

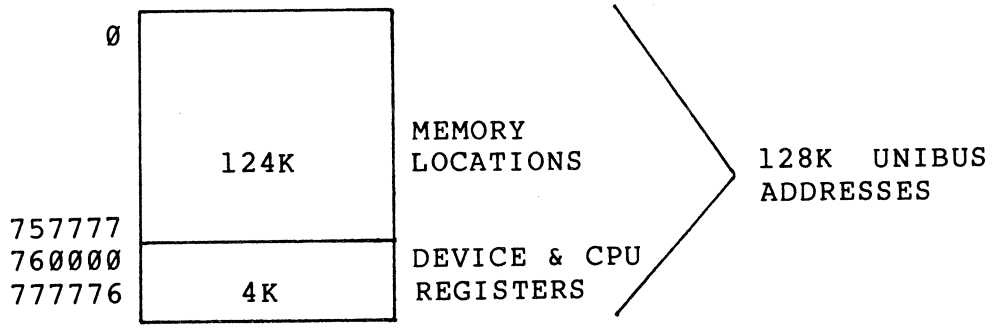
- SINGLE BI-DIRECTIONAL BUS
- ALL DEVICES INTERFACE TO UNIBUS
- ASYNCHRONOUS MASTER/SLAVE RELATIONSHIP
- ALL I/O DEVICE REGISTERS ADDRESSABLE AS MEMORY

UNIBUS ADDRESS SPACE

A 16 BIT ADDRESS used for addressing can address a maximum of 32K WORDS; however, only 28K is used for memory, the top 4K is reserved for DEVICE & CPU REGISTERS (I/O PAGE).

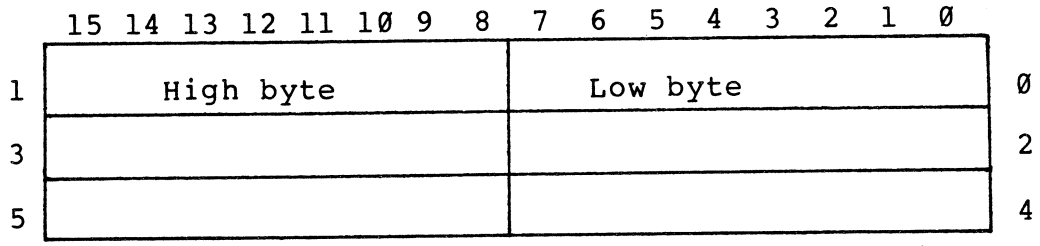


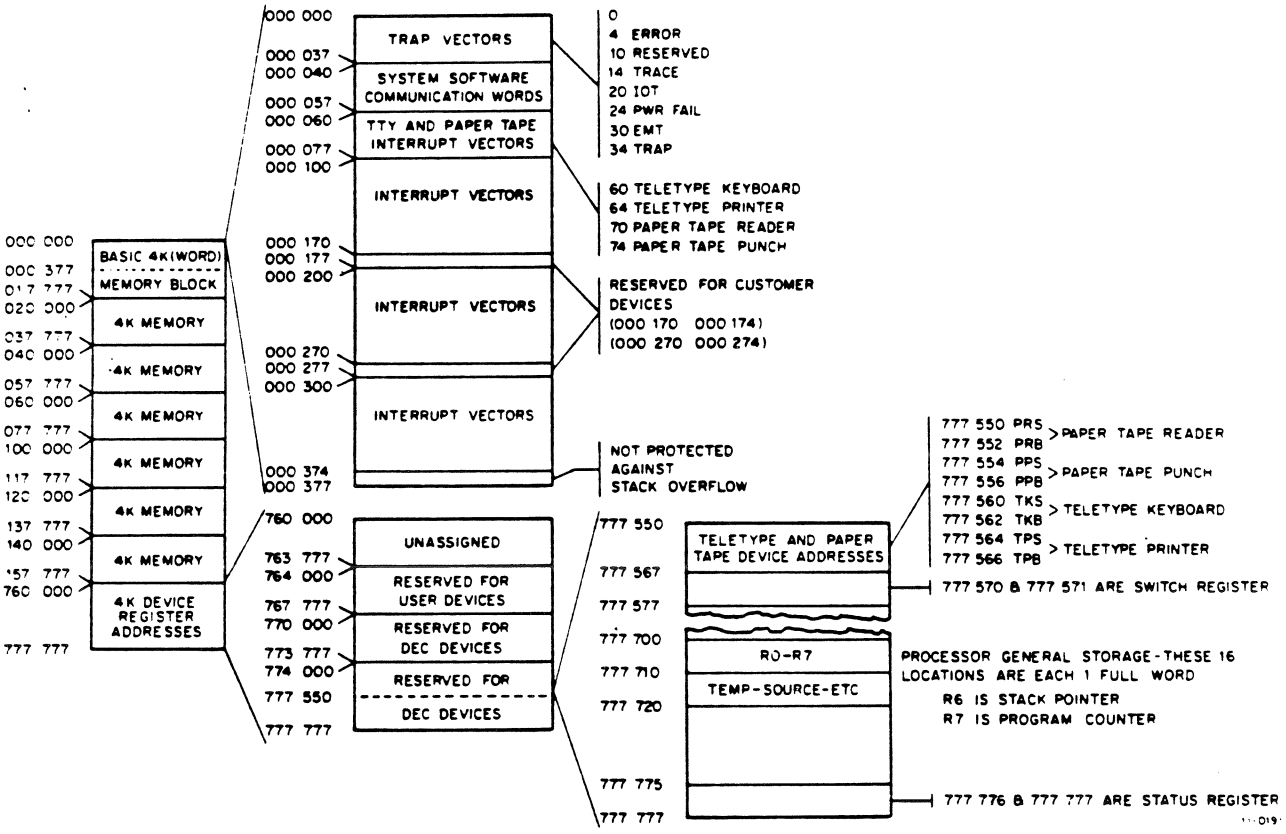
However, with the option of MEMORY MANAGEMENT, the processor can use an 18 BIT ADDRESS word which can access 128K WORDS. Again, 124K is used for MEMORY and the top 4K is reserved for DEVICE & CPU REGISTERS.



Each location can hold up to 16 data bits; one word WORD ADDRESSES ARE ALWAYS EVEN.

Memory is word orientated and byte (1/2 word) accessible. The HIGH BYTE <BITS 8:15> is stored in ODD ADDRESSES. The LOW BYTE <BITS Ø:7) is stored in EVEN ADDRESSES.




000 000
BASIC 4K(WORD) MEMORY BLOCK
000 377
017 777
020 000
4K MEMORY
037 777
040 000
4K MEMORY
057 777
060 000
4K MEMORY
077 777
100 000
4K MEMORY
117 777
120 000
4K MEMORY
137 777
140 000
4K MEMORY
157 777
760 000
4K DEVICE REGISTER ADDRESSES
777 777
000 000
TRAP VECTORS
000 037
000 040
SYSTEM SOFTWARE COMMUNICATION WORDS
000 057
000 060
TTY AND PAPER TAPE INTERRUPT VECTORS
000 077
000 100
INTERRUPT VECTORS
000 170
000 177
000 200
INTERRUPT VECTORS
000 270
000 277
000 300
INTERRUPT VECTORS
000 374
000 377
760 000
UNASSIGNED
763 777
764 000
RESERVED FOR USER DEVICES
767 777
770 000
RESERVED FOR DEC DEVICES
773 777
774 000
RESERVED FOR DEC DEVICES
777 550
777 777
0
4 ERROR
10 RESERVED
14 TRACE
20 IOT
24 PWR FAIL
30 EMT
34 TRAP
60 TELETYPE KEYBOARD
64 TELETYPE PRINTER
70 PAPER TAPE READER
74 PAPER TAPE PUNCH
RESERVED FOR CUSTOMER DEVICES
(000 170 000 174)
(000 270 000 274)
NOT PROTECTED AGAINST STACK OVERFLOW
777 550
777 567
777 577
777 700
777 710
777 720
777 775
777 777
TELETYPE AND PAPER TAPE DEVICE ADDRESSES
R0-R7
TEMP-SOURCE-ETC
777 550 PRS
777 552 PRB
> PAPER TAPE READER
777 554 PPS
777 556 PPB
> PAPER TAPE PUNCH
777 560 TKS
777 562 TKB
> TELETYPE KEYBOARD
777 564 TPS
777 566 TPB
> TELETYPE PRINTER
777 570 & 777 571 ARE SWITCH REGISTER
PROCESSOR GENERAL STORAGE - THESE 16 LOCATIONS ARE EACH 1 FULL WORD
R6 IS STACK POINTER
R7 IS PROGRAM COUNTER
777 776 & 777 777 ARE STATUS REGISTER
MR-019

Address Map

UNIBUS SIGNAL LINES

Unibus signal lines are divided into 3 interrelated sections:

1. Data Transfer Section
2. Priority Arbitration Section
3. Initialize Section

Name	Mnemonic	No. of Lines	Function	Assertion Level
DATA TRANSFER SECTION				
Address	A<17:00>	18	Selects slave address	Low
Data	D<15:00>	16	Information transfer	Low
Control	C0,C1	2	Type of Transfer	Low
Master Sync	MSYN	1	Command to DO IT	Low
Slave Sync	SSYN	1	Reply saying it is done	Low
Parity	PA,PB	2	Parity error	Low
Interrupt	INTR	1	MSYN to CPU	Low
PRIORITY ARBITRATION SECTION				
Bus Request	BR<7:4>	4	Bus request for interrupt	Low
Bus Grant	BG<7:4>	4	Grants use of the bus when free	High
Non-processor Request	NPR	1	Bus request for data transfer	Low
Non-processor Grant	NPG	1	Grants use of the bus when free	High
Selection Acknowledge	SACK	1	Acknowledges receipt of a bus grant	Low
Bus Busy	BBSY	1	Signifies the bus is in use	Low
INITIALIZE SECTION				
Initialize	INIT	1	System reset	Low
AC Low	ACLO	1	First warning that AC power is failing	Low
DC Low	DCLO	1	Signals "Too Late" a power fail has occurred	Low

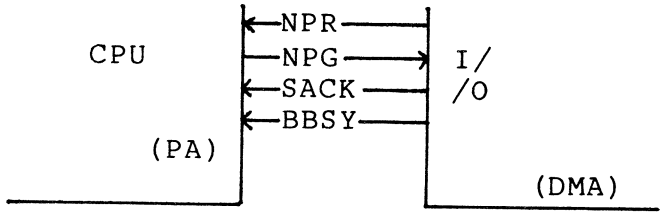
REQUESTING BUS CONTROL

2 TYPES OF REQUESTS

NPR---Non Processor Request
- a) To transact data
- b) Honored between bus cycles
- c) No CPU intervention

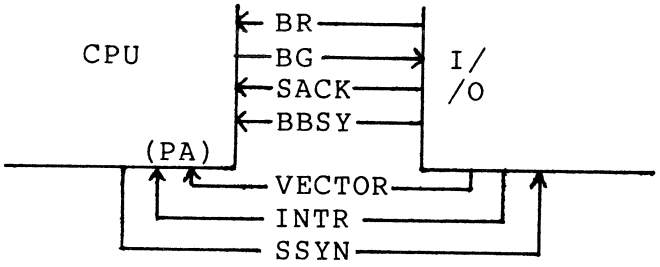
BR---Bus Request
- a) To interrupt the CPU
- b) Honored between instruction cycles

HONORING NPRs



1. Device asserts NPR
2. CPU sees NPR and asserts NPR
3. Device sees NPG, drops NPR
   Asserts SACK
   Then drops NPG
4. Device asserts BBSY (master)
   Drops SACK
   Does a data transfer

HONORING BRs



1. Device asserts BR
2. CPU sees BR and asserts BR
3. Device sees BG, drops BR
   Asserts SACK
   CPU drops BG
4. Device asserts BBSY (master)
5. Device sends the vector over D lines
6. Device sends interrupt
   Drops SACK
7. Strobes data and sends SSYN

NOTE:

WHEN SACK IS ASSERTED, THE DEVICE IS GUARANTEED TO BE NEXT BUS MASTER. SACK INHIBITS ANY GRANTS.

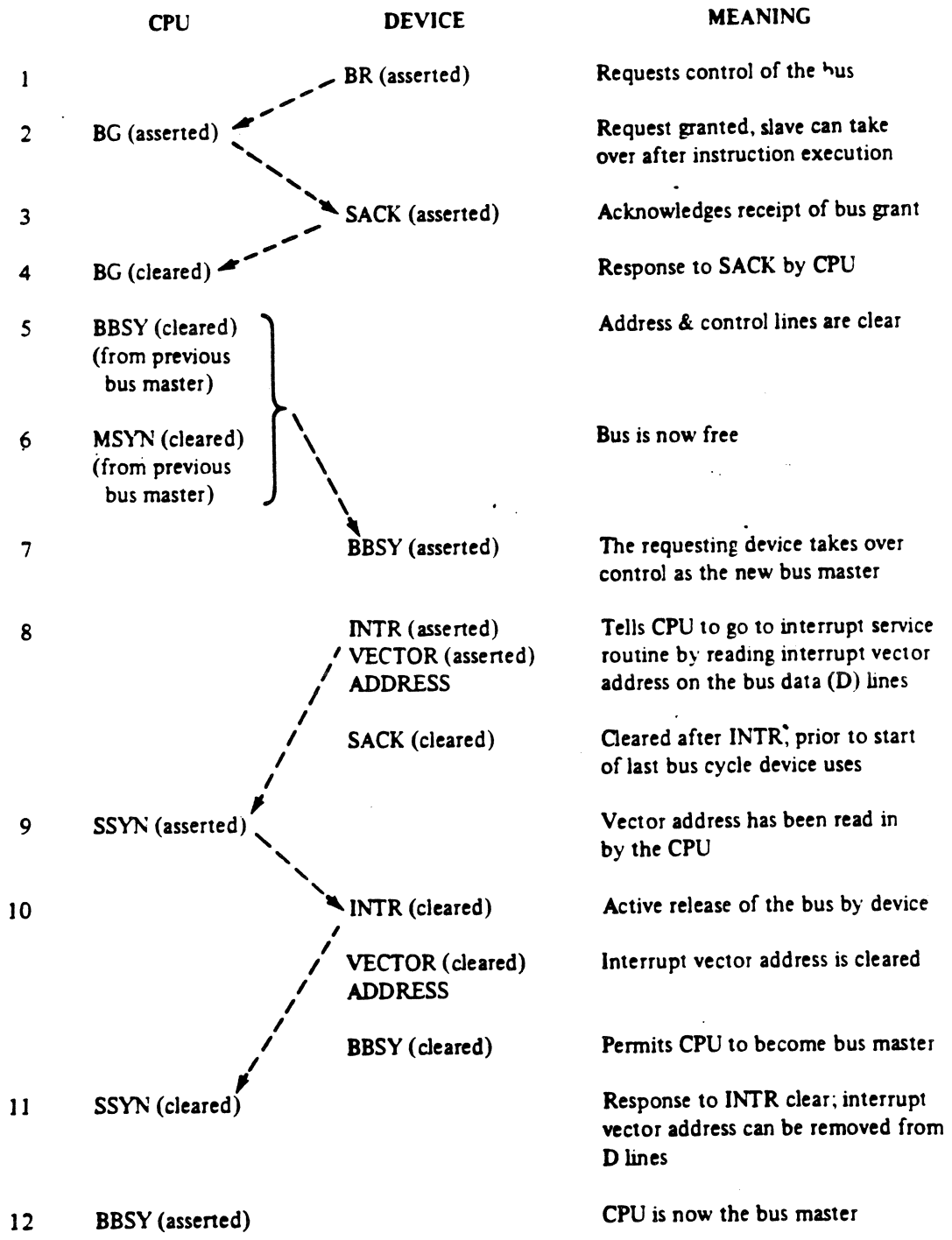
BBSY GAINS MASTERSHIP, NO OTHER DEVICE CAN GAIN CONTROL OF THE UNIBUS

BGs ARE GIVEN ONLY IF DEVICE'S PRIORITY IS HIGHER THAN THE CPU'S PROGRAM

NOTE:

FOR A MORE DETAILED UNDERSTANDING CONSULT PAGES 33 THRU 42 OF THE PDP-11 BUS HANDBOOK

INTERRUPT SEQUENCE



PRIORITY

In order for a device to receive or send information (data transfer) over the unibus, that device must become bus master. There are TWO distinct methods by which a device is able to do this, NPRs and BRs. The method by which the device gains control of the bus is determined by the type of device. Only DMA (Direct Memory Access) devices can do NPRs.

ASSIGNMENTS

a) OPERATING SPEED - Fastest devices get higher priority
b) DATA RECOVERY - Higher priority is given to devices whose data cannot be recovered easily
c) SERVICE REQUIREMENTS - Higher priority is given to devices that require less CPU intervention

ARBITRATION STRUCTURE

The priority logic is located in the CPU, but works independently of it (not CPU dependent).

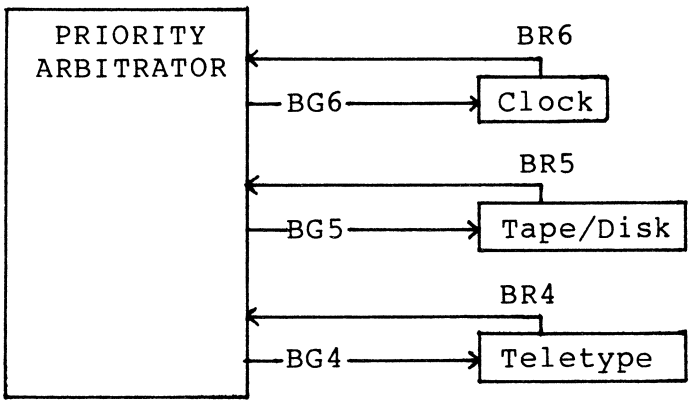
VERTICAL

There are five levels of requests: BR7, BR6, BR5, BR4 and NPR. NPR has the highest priority

BRs are fixed priority levels and *must be higher than the CPU program in order to be granted (BG7, BG6, BG5, BG4) bus master

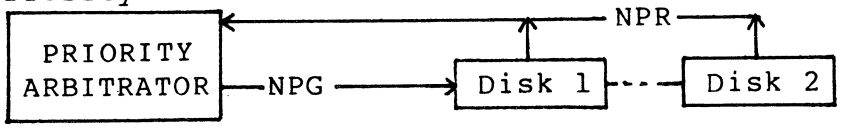
NOTE: THE HIGHER THE NUMBER OF THE REQUEST, THE HIGHER THE PRIORITY

*The CPU priority level is determined by the Processor Status Word which is located in the CPU register 777776, BITS 5, 6 & 7

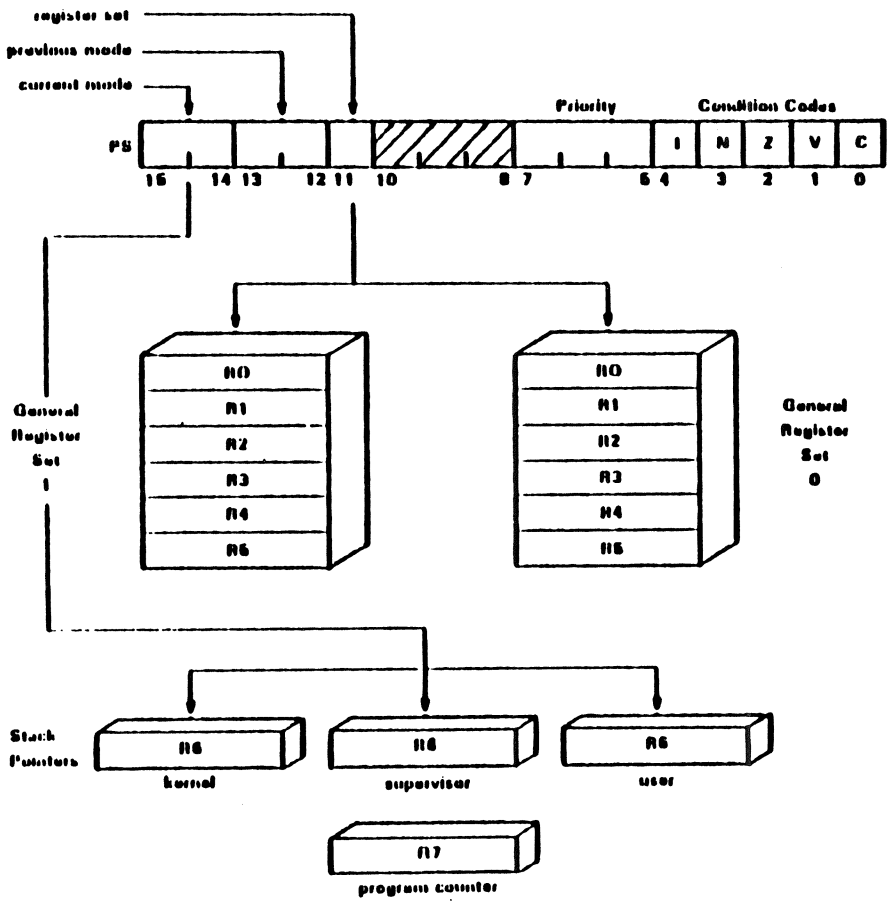


HORIZONTAL

When connected to the same vertical level, the device electronically closest to the CPU will have highest priority



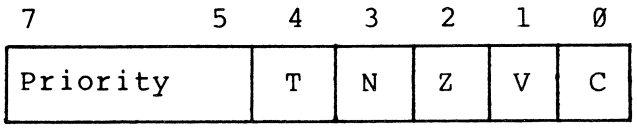
processor status word



- Multi-Programming Modes
  - 00 kernel: All instructions are legal
  - 01 Supervisor: } "Halt", illegal
  - 11 User: } "Reset", "SPL" ignored
- 16 General Registers
  - 2 sets of 6 (R0–R5)
    - 1 set for normal programming (or user mode)
    - 1 set for fast data collection (or kernel, supervisor modes)
  - 3 Processor Stack Pointers (R6) facilitate multi-programming through hardware protection of kernel-supervisor-user
  - 1 Program Counter (R7)

TR-1326

PROCESSOR STATUS WORD



The Status Register may be addressed at 777776.

PRIORITY — Specifies 1 of 8 levels of processor priority, which in turn determines the BR priority level(s) on which the bus may be gained

T — Trace bit causes an automatic trap at the completion of the instruction following the one that set the T-bit

N — A Condition Code indicating that the result of the last instruction affecting it was negative.

Bit 15 1/Ø Negative/Positive WORD
Bit 7 1/Ø Negative/Positive BYTE

Z — A Condition Code indicating that the result of the last instruction affecting it was zero

V — A Condition Code indicating that the result of the last arithmetic operation exhibited arithmetic overflow (a change in sign when quantities of like signs are added). Also used elsewhere as indicated in the PDP-11 Handbook. Exceptions also noted in the Handbook.

C — A Condition Code indicating that the result of the last instruction affecting it had a carry out of Bit 15 (bit 7 for a BYTE). Exceptions noted in the PDP-11 Handbook.

NOTE:

The Condition Codes in the Status Word are not implicitly affected by a mov #n, @# 177776 or by any other instruction whose purpose is to explicitly modify the Processor Status Word.

Study Questions:

1. The PDP-11/70 has _____ general purpose registers, of which _____ are available for use at any given time.

2. Describe the predefined functions of the following general purpose registers.

   a. R7

   b. R6

   c. R16

   d. R17

3. What is the function of the processor priority bits in the Processor Status Word?

4. What is an "Odd Addressing Error"?

5. List the following types of requests in descending order of priority.

a. _____	1. PIR7
b. _____	2. BR5
c. _____	3. PIR3
d. _____	4. BR 4
e. _____	5. NPR
f. _____	6. PIR 6
g. _____	7. BR7

6. Which of the following statements concerning the UNIBUS are true?

   a. It uses a single common bus concept.
   b. Each device on the bus uses the same form of communication.
   c. No special input or output instructions are used.
   d. Data can be transferred bidirectionally.
   e. More than one device can control the bus at the same time.
   f. Communication is independent of the physical bus length.
   g. Storage to storage transfers are not possible.
   h. Communication between devices on the bus has priority over communication between a device and the processor.

7. What is meant by "Interlocked Communication"?

8. UNIBUS priority arbitration occurs _____

   a. within the device requesting bus mastership.
   b. automatically on the UNIBUS itself.
   c. within the processor.
   d. when the previous data transfer is completed.

9. Describe what is meant by "word addressing" and "byte addressing".

10. With memory management disabled, a PDP-11/70 can directly access _____ memory locations.

    a. 4,096
    b. 28,672
    c. 32,769
    d. 126,976

11. a. Why is there a 150 ns minimum delay between the assertion of address and control information and the assertion of MSYN?

    b. Why is there a 75 ns minimum delay between the dropping of MSYN and the dropping of address and control information?

12. During the interrupt operation, the vector address is sent from the interrupting device to the CPU over the _____ lines of the Unibus.

13. Time-out can occur during the selection of the next bus master if _____ is not received. It can occur during the transfer of data if _____ is not received.

PDP-11/70
Hardware Introduction
Study Assignment 1
Answer Sheet

Study Questions:

1. The PDP-11/70 has 16 general purpose registers, of which 8 are available for use at any given time.

2. Describe the predefined functions of the following general purpose registers:

   a. R7 - **program counter**

   b. R6 - **Kernel stack pointer**

   c. R16 - **Supervisor stack pointer**

   d. R17 - **User stack pointer**

3. What is the function of the processor priority bits in the Processor Status Word?

   The processor priority bits determine the minimum priority level that will be honored of the pending requests for interrupt.

4. What is an "Odd Addressing Error"?

   An "Odd Addressing Error" is the designation of a word operand at an odd address -- half way between word boundaries.

5. List the following types of requests in descending order of priority.

a.	5	1. PIR7
b.	1	2. BR5
c.	7	3. PIR3
d.	6	4. BR 4
e.	2	5. NPR
f.	4	6. PIR 6
g.	3	7. BR7

6. Which of the following statements concerning the UNIBUS are true?

   a. It uses a single common bus concept.
   b. Each device on the bus uses the same form of communication.
   c. No special input or output instructions are used.
   d. Data can be transferred bidirectionally.
   e. More than one device can control the bus at the same time.
   f. Communication is independent of the physical bus length.
   g. Storage to storage transfers are not possible.
   h. Communication between devices on the bus has priority over communication between a device and the processor. **A,B,C,D,F,H**

7. What is meant by "Interlocked Communication"?

   "Interlocked Communication" is data transfer under the control of initiation, response control signals. For every initiation signal from a master device, a response from the slave device is required for the process to continue.

8. UNIBUS priority arbitration occurs ____C____

   a. within the device requesting bus mastership.
   b. automatically on the UNIBUS itself.
   c. within the processor.
   d. when the previous data transfer is completed.

9. Describe what is meant by "word addressing" and "byte addressing".

   "Word Addressing" - designation of a 16 bit operand
   - even addresses only

   "Byte Addressing" - designation of an 8 bit operand
   - even or odd addresses

10. With memory management disabled, a PDP-11/70 can directly access ___B___ memory locations.

    a. 4,096
    b. 28,672
    c. 32,769
    d. 126,976

11. a. Why is there a 150 ns minimum delay between the assertion of address and control information and the assertion of MSYN?

75 NS DELAY FOR PROPAGATION OF ADDRESS AND CONTROL INFORMATION.

75 NS DELAY TO INSURE THAT ADDRESS AND CONTROL INFORMATION IS STABLE BEFORE MSYN IS ASSERTED.

b. Why is there a 75 ns minimum delay between the dropping of MSYN and the dropping of address and control information?

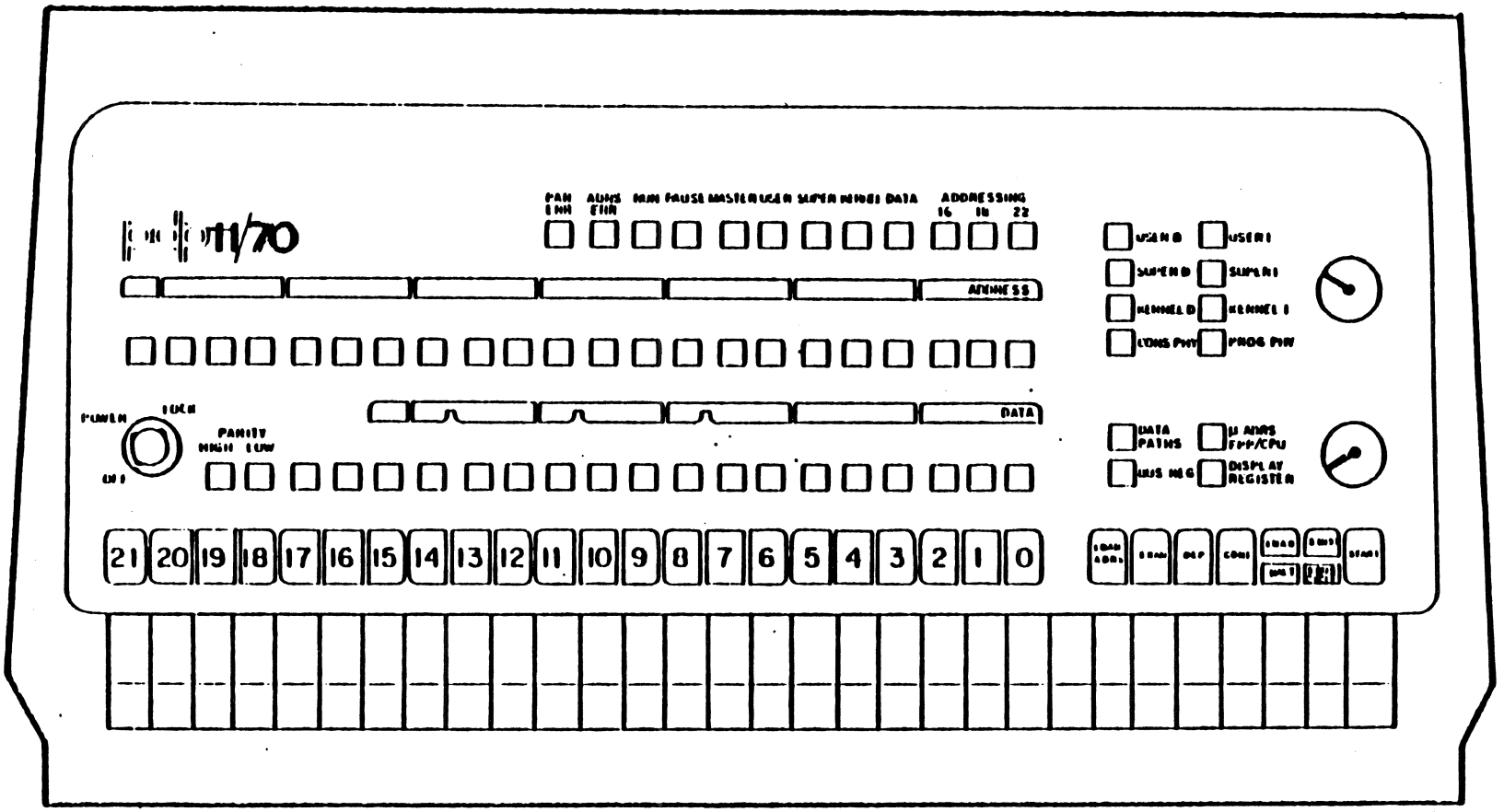
TO INSURE THAT MSYN IS NO LONGER IN TRANSITION WHEN THE ADDRESS AND CONTROL INFORMATION IS ALLOWED TO CHANGE.

12. During the interrupt operation, the vector address is sent from the interrupting device to the CPU over the **DATA** lines of the Unibus.

13. Time-out can occur during the selection of the next bus master if **SACK** is not received. It can occur during the transfer of data if **SSYN** is not received.

PDP-11/70 Diagnostics and Module Level Repair

DAY TWO


11/70
PAR ERR
ADRS ERR
MAINT
PAUSE
MASTER
USER
SUPER
KERNEL
DATA
ADDRESSING
16
18
22
ADDRESS
DATA
POWER
LOCK
PARITY
HIGH
LOW
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

TR-0985

Console Familiarization Laboratory
Study Assignment 2

This Worksheet is designed to familiarize the student with the general operation of the control console and to recognize the indicator displays. For switch and indicator descriptions of the PDP-11/70 refer to the following:

PDP-11/70 Processor Handbook, Chapter 11

Step 1: A. Insert Key into power switch on the left-hand side of the console and turn key to the POWER position.

B. Remove key from power switch.

C. Place the HALT/ENABL switch in the HALT position.

D. Place the address display switch in the CONS PHY position.

E. Place the data display switch in the DATA PATHS position.

F. Notice that the KERNEL indicator and MASTER indicator are on. When power was applied an INIT occurred which forced the processor into KERNEL MODE. The INIT also cleared devices on the Unibus which allowed the processor to become the Master of the Unibus.

Step 2: A. Place 17777706 in the Data switches (Up=1, Down=0).

B. Depress the LOAD ADRS switch. Notice that the address indicators display 17777706. This indicates that the contents of the Data switches were loaded into the processor.

C. Place 000456 into the Data switches.

D. Raise the DEP switch. Notice that the Data Indicators display 000456. This indicates that the contents of the Data switches were placed in the addressed location.

E. Refer to the KB11-B Processor Manual. Table 1-1, section 1.25, shows that 17777706 is the address of General Register 6. In step D above, 000456 was placed in R6.

F. Place 000123 in the Data switches.

G. Raise the DEP switch nine times. Notice that the Address Indicators do not increment this time; instead, they cycle back to 17777700, the address of R0. Consecutive deposits in general addresses will cause automatic cycling through the addresses 17777700-17777717.

TO LOAD A GENERAL REGISTER: Place the register address (17777700-17777717) in the Data switches, depress LOAD ADRS, place the value to be loaded (0-177777) in the Data switches, raise DEP. Consecutive registers can be loaded by placing a new value in the data switches and raising DEP. Consecutive deposits causes the addresses to cycle from the address of R0 through the address of R17 back to the address of R0.

Step 3: Load the following values into the general register indicated.

R0 = 123456
R1 = 052525
R2 = 070707

Step 4: A. Place 17777706 in the Data switches.

B. Depress LOAD ADRS.

C. Depress EXAM. Notice that 000456 is displayed in the Data Indicators. This is the original value that was placed in R6 in Step 2.

D. Depress EXAM again. Notice that the address increments to 17777707 and the data indicated in 000123.

TO EXAMINE THE CONTENTS OF A GENERAL REGISTER:

Place the register address (17777700-17777717) in the Data switches, depress LOAD ADRS, depress EXAM, the contents of the register will be displayed in the Data Indicators when the Data Display Switch is in the DATA PATHS position. Consecutive registers may be examined by consecutive depressions of the EXAM switch.

Step 5: Examine R0, R1, and R2 to confirm that they contain the values in them in Step 3.

Step 6: A. Place 001000 in the Data switches.

B. Depress LOAD ADRS.

C. Place 123456 in the Data switches.

D. Raise DEP. This has caused the value 123456 to be placed into memory location 1000. (Observe address and Data indicators)

E. Place 052525 in the Data switches.

F. Raise DEP. This has caused the value 052525 to be placed into memory location 1002. (Observe Address and Data indicators)

G. Raise DEP. Notice that the address automatically increments by two. Subsequent deposit operations will cause the contents of the Data switches to be placed into consecutive memory locations.

TO LOAD A MEMORY LOCATION: Place address in Data switches, depress LOAD ADRS, place value to be loaded in Data switches, raise DEP. Consecutive memory locations can be loaded by placing a new value in the Data switches and raising DEP.

Step 7: Place the following values in the indicated memory locations:

500 = 012345 504 = 162735
502 = 007700

Step 8: A. Place 001000 in the Data switches.

B. Depress LOAD ADRS.

C. Depress EXAM. Notice that 123456 is displayed in the Data indicators. This indicates that memory location 1000 contains 123456. (Observe Address and Data indicators).

D. Depress EXAM. Notice that the Address is automatically incremented in the same manner as it was in Step 6.

TO EXAMINE THE CONTENTS OF A MEMORY LOCATION: Place address in Data switches, depress LOAD ADRS, depress EXAM. Consecutive memory locations can be examined by depressing EXAM.

Step 9: Examine locations 500, 502, and 504 to confirm that they contain the values placed in them in Step 7.

Step 10: A. Place 001000 in the Data switches.

B. Depress LOAD ADRS.

C. Depress EXAM, Location 1000 contains the value 123456 placed there in Step 6. (Observe Address and Data Indicators).

D. Place 070707 in the Data switches

E. Raise DEP. Notice that the address did not change. 070707 was placed in location 1000.

F. Depress EXAM. Notice that the address did not change. The address will only be incremented if the EXAM or DEP switch is operated twice in succession.

TO VERIFY CONSECUTIVE MEMORY LOCATIONS: Examine consecutive locations until location is encountered which does not contain a proper value, place proper value in Data switches, raise DEP, depress EXAM, continue examining consecutive locations and correcting improper values until all locations desired have been verified.

Step 11: Place the following values in the registers and locations specified:

```
  R0 = 177777
  R1 = 000002
  R2 = 000002
1000 = 060102
1002 = 000000
1004 = 005000
1006 = 177777
1010 = 010200
1012 = 000000
```

Step 12: Verify that the registers and locations contain the following:

```
  R0 = 177777
  R1 = 000002
  R2 = 000002
1000 = 060102
1002 = 000000
1004 = 005000
1006 = 000000
1010 = 010200
1012 = 000000
```

Location 1006 = 000000 is correct. Change its contents to this value.

Step 12: cont. Locations 1000 thru 1012 contain machine instructions which will perform the following:

```
1000 ADD %1, %2 Add the contents of R1 to R2 and place
the result in R2.
1002 HALT            Stop the processor
1004 CLR %0          Place zeros in R0
1006 HALT            Stop the processor
1010 MOV %2, %0      Move the contents of R2 to R0
1012 HALT            Stop the processor
```

Step 13:

A. Place 001000 in Data switches.

B. Depress LOAD ADRS.

C. Depress START. Notice that the Address does not change. Depressing START with the HALT/ENABL switch in the HALT position will generate an INIT, but it will NOT cause the processor to start executing the program.

TO START A PROGRAM: Place starting address in Data switches, depress LOAD ADRS, place HALT/ENABL switch in ENABL position, depress START.

TO RESET THE PROCESSOR: Place HALT/ENABL in HALT position, depress START.

D. Place HALT/ENABL switch in ENABL position.

E. Depress START. Note that the Address indicates display 1004, which is one location past the HALT instruction. Also note that the Data indicates display 177777, which is the contents of R0.

WHEN THE PROCESSOR STOPS AS A RESULT OF A HALT INSTRUCTION, THE ADDRESS DISPLAYED WILL BE ONE LOCATION PAST THE HALT INSTRUCTION AND R0 WILL BE DISPLAYED IN THE DATA PATHS.

F. Depress CONT. The program will continue execution until the next HALT instruction. Note that the Address indicators display 1010 and the Data Indicators display 000000. This is because the instruction at location 1004 placed zeros into R0 and location 1006 contained a HALT instruction.

Step 13 cont. G. Depress CONT. Explain the contents of the Address and Data indicators._____

_____

_____

TO CONTINUE AFTER A HALT INSTRUCTION: Depress CONT.

Step 14: A. Place 000240 in locations 1002 and 1006. This is a NOP or do nothing instruction. Reload R0, R1, and R2 to their original values.

B. START the program. Notice that the processor stops with the address indicators containing 001014 and the Data indicators displaying 000004. Why?

_____

_____

_____

Step 15: A. Place HALT/ENABL switch in HALT position.

B. Reload R0, R1, and R2 to their original values.

C. Place 001000 in Data switches.

D. Depress LOAD ADRS.

E. Place S INST/S BUS CYCLE in S INST position.

F. Depress START. Notice that the processor stops at address 1000.

G. Depress CONT. Notice that the processor stops at address 1002.

H. Depress CONT again. Notice that the processor stops at address 1004. By depressing CONT with the control switches in the HALT and S INST positions, the processor will perform one instruction and stop.

TO PERFORM A PROGRAM ONE INSTRUCTION AT A TIME: Place HALT/ENABL switch in HALT position, place starting address in Data switches, depress LOAD ADRS, place S INST/S BUS CYCLE switch in S INST position, depress START, depress CONT to perform one instruction, depress CONT to perform subsequent instructions.

Step 16:

A. Place HALT/ENABL switch in HALT position.

B. Reload R0, R1, and R2 to their original values.

C. Place 001000 in Data switches.

D. Depress LOAD ADRS

E. Place S INST/S BUS CYCLE switch in S BUS CYCLE position.

F. Depress START.

G. Depress CONT. Notice that the processor stops with the PAUSE light on and the Address indicators displaying 1000. The processor has stopped at the end of the first bus cycle.

H. Place the Data Display switch in the BUS REGISTER position. Notice that the Data indicators display 060102, which is the first instruction. It is being obtained from memory via the BR.

I. Depress CONT. Observe the PAUSE indicator, the Address indicators and the Data indicators. What is being displayed in the Data indicators?

_____

_____

_____

J. Depress CONT. Observe all indicators. What is being displayed and why?

_____

_____

_____

K. Keep depressing CONT and observing the indicators until the Address indicators display 1014. Notice that the PAUSE light is not on, why?

_____

_____

_____

Step 16: cont. TO PERFORM A PROGRAM ONE BUS CYCLE AT A TIME: Place HALT/ENABL switch in HALT position, place starting address in Data switches, depress LOAD ADRS, place S INST/S BUS CYCLE switch in S BUS CYCLE position, depress START, depress CONT, processor will stop at end of first bus cycle, depress CONT to perform subsequent bus cycles.

Step 17:

A. Place 001001 in the Data switches.

B. Depress LOAD ADRS.

C. Depress EXAM. Notice that ADRS ERR indicator is on.

AN ODD ADDRESS ERROR WILL OCCUR WHEN ATTEMPTING TO DEPOSIT OR EXAM AN ODD ADDRESS LOCATION.

D. Reset the processor.

Step 18: Locate white switch to the left of the LOAD ADRS switch. Raise it. This is the lamp test switch and can be used to verify whether or not the indicators are working.

Step 19:

A. Place key in power switch.

B. Turn to LOCK position. All console control switches are now disabled. Attempt various console operations to verify this.

C. Return power switch to OFF position and remove key.

Rotate 1

The purpose of this program 1 is to get a single light in the data display to rotate from end to end. You light the data lights by writing data to the console address and display register (17 777 570). When the program runs the data display rotational switch show be in the console display setting.

```
1000/ MOV #1,R0          ! Move the decimal number 1 into general register 0
1002/ 1                  ! The decimal number 1 going to the general purpose
                         ! register 0

1004/ CLR R5             ! Clear the contents of general purpose register 5

1006/ MOV R0,@#17777570  ! Move the contents of general purpose register 0
1010/ 17777570           ! to the light register who's address is 17777570

1012/ INC R5             ! Increment the value in general purpose register
                         ! 5 by the value of a decimal 1

1014/ BNE 1012           ! Branch backward to address 1012 if general purpose
                         ! register 5 is not equal to zero,the zero bit of the
                         ! processor status register address (17777776)"PSW"
                         ! bit number two (2) = 0

1016/ ROL R0             ! Rotate left the contents of general purpose register
                         ! 0

1020/ BR 1006            ! Branch backward  to address 1006 all the time
```

The way to deposit the above program into memory

```
1000 Load address switch
     012700   Deposit switch/Address lights will update by two(2)
     1        Deposit switch/Address lights will update by two(2)
     005005   Deposit switch/Address lights will update by two(2)
     010037   Deposit switch/Address lights will update by two(2)
     17777570 Deposit switch/Address lights will update by two(2)
     005205   Deposit switch/Address lights will update by two(2)
     001376   Deposit switch/Address lights will update by two(2)
     006100   Deposit switch/Address lights will update by two(2)
     000772   Deposit switch/Address lights will update by two(2)
```

Light bounce

This program 2 will bounce a light through the data display lights
from one end to the other.

```
2000/ MOV #1,R0          ! Move the decimal number 1 into general register 0
2002/ 1                  ! The decimal number 1 going to the general purpose
                         ! register 0

2004/ CLR R5             ! Clear the contents of general purpose register 5

2006/ MOV #17777570,R4   ! Move the octal value of 17777570 into general register 4
2010/ 17777570           ! The octal number 17777570 going to the generl purpose
                         ! register 4

2012/ MOV R0,(R4)        ! Move the contents of general purpose register 0 to the
                         ! address contained in general purpose register 4 the
                         ! address is 17777570 the address of the light register

2014/ INC R5             ! Increment the value in general purpose register 5
                         ! by the value of a decimal 1

2016/ BNE 2014           ! Branch backward to address 2014 if general purpose
                         ! register 5 is not equal to zero, the zero bit of
                         ! the processor status register address (17777776)
                         ! "PSW" bit number two (2) = 0

2020/ NOP                ! No operation just allow time to pass so that everything
                         ! in the PDP-11/70 will be ready for the next instruction

2022/ ROL R0             ! Rotate left the contents of general purpose register 0

2024/ BCC 2012           ! Branch backward to address 2012 if there was no
                         ! carry when you rotated left general purpose register
                         ! 0 the carry bit of the processor status register
                         ! address (17777776) "PSW" bit number zero (0) = 0

2026/ ROR R0             ! Rotate right the contents of general purpose register 0

2030/ BCS 2020           ! Branch backward to address 2020 if there was a
                         ! carry when you rotate right general purpose register
                         ! 0 the carry bit of the processor status register
                         ! address (17777776) "PSW" bit number zero (0) = 1

2032/ MOV R0,(R4)        ! Move the contents in general purpose register 0 to the
                         ! address contained in general purpose register 4 the
                         ! address is 17777570 the address of the light register

2034/ INC R5             ! Increment the value in general purpose register 5
                         ! by the value of a decimal 1

2036/ BNE 2034           ! Branch backward to address 2034 if general purpose
                         ! register 5 is not equal to zero, the zero bit of
                         ! the processor status register address (17777776)
                         ! "PSW" bit number two (2) = 0

2040/ BR 2026            ! Branch backward to address 2026 all the time
```

The way to deposit the above program into memory

```
2000 Load address switch
     012700   Deposit switch/Address lights will update by two(2)
     1        Deposit switch/Address lights will update by two(2)
     005005   Deposit switch/Address lights will update by two(2)
     012704   Deposit switch/Address lights will update by two(2)
     17777570 Deposit switch/Address lights will update by two(2)
     010014   Deposit switch/Address lights will update by two(2)
     005205   Deposit switch/Address lights will update by two(2)
     001376   Deposit switch/Address lights will update by two(2)
     000240   Deposit switch/Address lights will update by two(2)
     006100   Deposit switch/Address lights will update by two(2)
     103372   Deposit switch/Address lights will update by two(2)
     006000   Deposit switch/Address lights will update by two(2)
     103773   Deposit switch/Address lights will update by two(2)
     010014   Deposit switch/Address lights will update by two(2)
     005205   Deposit switch/Address lights will update by two(2)
     001376   Deposit switch/Address lights will update by two(2)
     000772   Deposit switch/Address lights will update by two(2)
```

Switch change light bounce

The purpose of this program 3 is to bounce the data lights between any switches that are set on the front panel data/address switches.

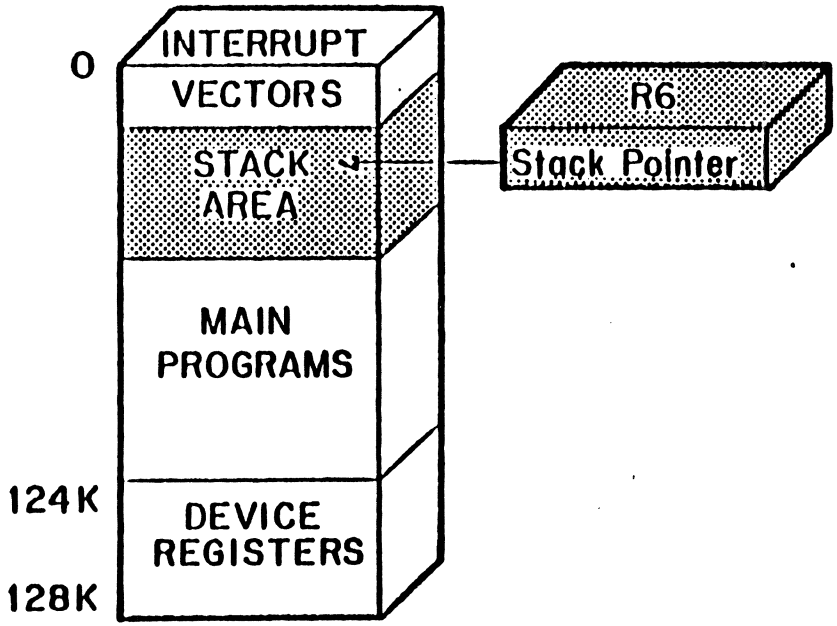
```
1000/ CLR R5              ! Clear the contents of general purpose register 5

1002/ MOV #1,R0           ! Move the decimal number 1 into general register 0
1004/ 1                   ! The decimal number 1 going to the general purpose
                          ! register 0

1006/ MOV #17777570,R4!Move the octal value of 17777570 into general register 4
1010/ 17777570            ! The octal number 17777570 going to the general purpose
                          ! register 4

1012/ MOV R0,(R4)         ! Move the contents of general purpose register 0 to
                          ! the address contained in general purpose register 4
                          ! the address is 17777570 the addres of the light
                          ! register

1014/ BIT (R4),R0         ! Test the bit at the address contained in general
                          ! purpose register 4 (17777570) switch register
                          ! address. See if any switches have been moved. With
                          ! contents of general purpose register 0 a logical
                          ! AND function

1016/ BNE 1032            ! Branch foreward to address 1032 if the results of
                          ! the bit test (AND function) was not equal to zero,
                          ! the zero bit of the processor status register
                          ! address (17777776) "PSW" bit number two (2) = 0

1020/ ROL R0              ! Rotate left the contents of general purpose register
                          ! 0

1022/ INC R5              ! Increment the value in general purpose register 5
                          ! by the value of a decimal 1

1024/ BNE 1022            ! Branch backward to address 1022 if general purpose
                          ! register 5 is not equal to zero, the zero bit of
                          ! the processor status register address (17777776)
                          ! "PSW" bit number two (2) = 0

1026/ NOP                 ! No operation just allow time to pass so that every
                          ! thing in the PDP-11/70 will be ready for the next
                          ! instruction

1030/ BR 1012             ! Branch backward to address 1012 all the time

1032/ ROR R0              ! Rotate right the contents of general purpose register
                          ! 0

1034/ MOV R0,(R4)         ! Move the contents of general purpose register 0 to
                          ! the address contained in general purpose register 4
                          ! the address is 17777570 the address of the light
                          ! register
```

```
1036/ BIT (R4),R0       ! Test the bit at the address contained in general
                        ! purpose register 4 (17777570) switch register
                        ! address. See if any switches have been moved. With
                        ! the contents of general purpose register 0 a
                        ! logical AND function

1040/ BNE 1020          ! Branch backward to address 1020 if the results of
                        ! bit test (AND function) was not equal to zero,
                        ! the zero bit of the processor status register
                        ! address (17777776) "PSW" bit number two (2) = 0

1042/ INC R5            ! Increment the value in general purpose register 5
                        ! by the value of a decimal 1

1044/ BNE 1042          ! Branch backward to address 1042 if general purpose
                        ! register 5 is not equal to zero, the zero bit of
                        ! the processor status register address (17777776)
                        ! "PSW" bit number two (2) = 0

1046/ BR 1032           ! Branch backward to address 1032 all the time
```

The way to deposit the above program into memory

```
1000 Load address switch
     005005      Deposit switch/Address lights will update by two(2)
     012700      Deposit switch/Address lights will update by two(2)
     1           Deposit switch/Address lights will update by two(2)
     012704      Deposit switch/Address lights will update by two(2)
     17777570    Deposit switch/Address lights will update by two(2)
     010014      Deposit switch/Address lights will update by two(2)
     031400      Deposit switch/Address lights will update by two(2)
     001005      Deposit switch/Address lights will update by two(2)
     006100      Deposit switch/Address lights will update by two(2)
     005205      Deposit switch/Address lights will update by two(2)
     001376      Deposit switch/Address lights will update by two(2)
     000240      Deposit switch/Address lights will update by two(2)
     000770      Deposit switch/Address lights will update by two(2)
     006000      Deposit switch/Address lights will update by two(2)
     010014      Deposit switch/Address lights will update by two(2)
     031400      Deposit switch/Address lights will update by two(2)
     001366      Deposit switch/Address lights will update by two(2)
     005205      Deposit switch/Address lights will update by two(2)
     001376      Deposit switch/Address lights will update by two(2)
     000771      Deposit switch/Address lights will update by two(2)
```

Last-in First-out Lists



- Stacks

  provide common storage and transfer of variables for all routines.

- Processor Stack

  subroutines---save the contents of the link register (JSR) and restore them (RTS) automatically

  interrupts---automatically save the status of the machine (PC and PS) and restore them on completion (RTI, RTT).

Architecture allows unlimited nesting of subroutines and interrupts.

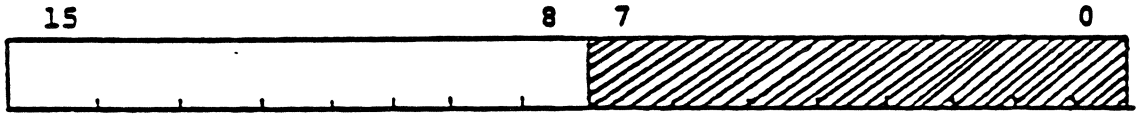
- Stack Limit Register provides a variable stack boundary for the kernel. Inhibits against overflow >16 words.

STACK LIMIT REGISTER OPERATION

The Stack Limit allows program control of the lower limit for permissable kernel stack addresses. This limit may be varied in increments of $(400)_8$ bytes or $(200)_8$ words, up to a maximum address of 177 400 (almost the top of a 32K memory).

The normal boundary for stack addresses is 400. The Stack Limits Register allows this lower limit to be raised, providing more address space for interrupt vectors or other data that should not be destroyed by the program.

There is a Stack Limit Register, with the following format:



The Stack Limit Register can be addressed as a word at location [17] 777774, or as a byte at location [17] 777775. The register is accessible to the processor and console, but not to any bus device.

The 8 bits, 15 through 8, contain the stack limit information. These bits are cleared by System Reset, Console Start, or the RESET instruction. The lower 8 bits are not used. Bit 8 corresponds to a value of $(400)_8$ or $(256)_{10}$.

STACK LIMIT VIOLATIONS

When instructions cause a stack address to exceed (go lower than) a limit set by the programmable Stack Limit Register, a Stack Violation occurs. There is a Yellow Zone (grace area) of 16 words below the Stack Limit which provides a warning to the program so that corrective steps can be taken. Operations that cause a Yellow Zone Violation are completed, then a bus error trap is effected. The error trap, which itself uses the stack, executes without causing an additional violation, unless the stack has entered the Red Zone.

A Red Zone Violation is a Fatal Stack Error. (Odd stack or non-existent stack are the other Fatal Stack Errors.) When detected, the operation causing the error is aborted, the stack pointer is re-positioned to address 4, and a bus error occurs. The old PC and PS are pushed into location 0 and 2, and the new PC and PS are taken from locations 4 and 6.

STACK LIMIT ADDRESSES

The contents of the Stack Limit Register (SL) are compared to the stack address to determine if a violation has occurred. The least significant bit of the register (bit 8) has a value of $(400)_8$. The determination of the violation zones is as follows:

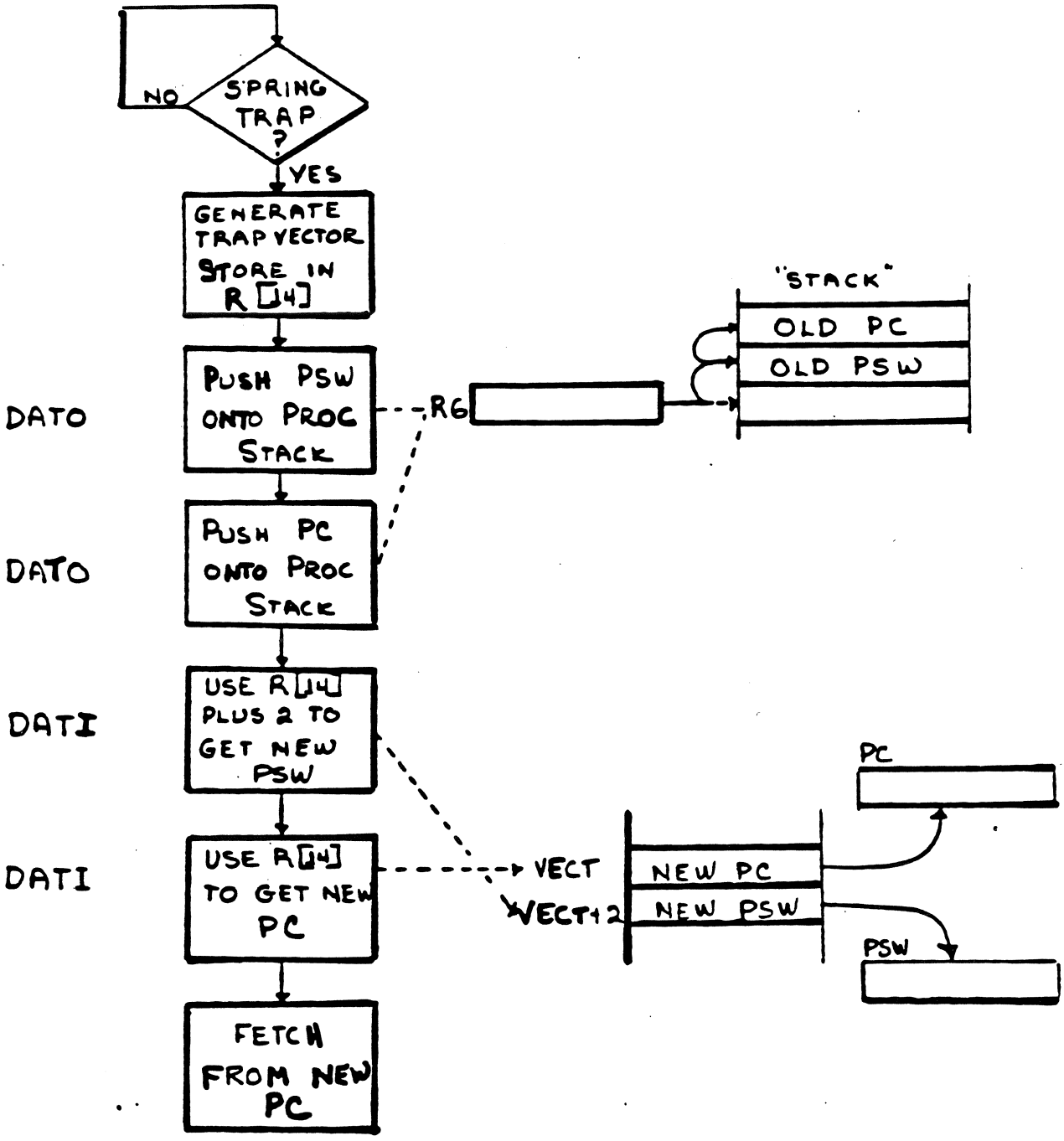
Yellow Zone = (SL) + $(340 \text{ through } 377)_8$ execute, then trap

Red Zone $\leq$ (SL) + $(337)_8$ abort, then trap to location 4

If the Stack Limit Register contents were zero:

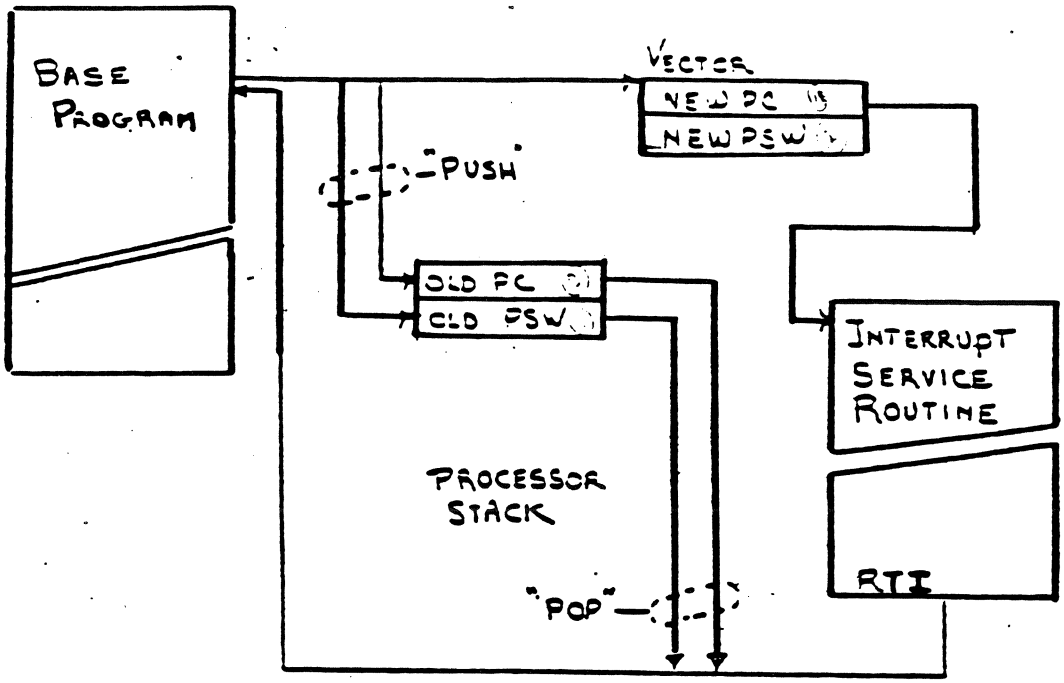
Yellow Zone = 340 through 377
Red Zone = 000 through 337

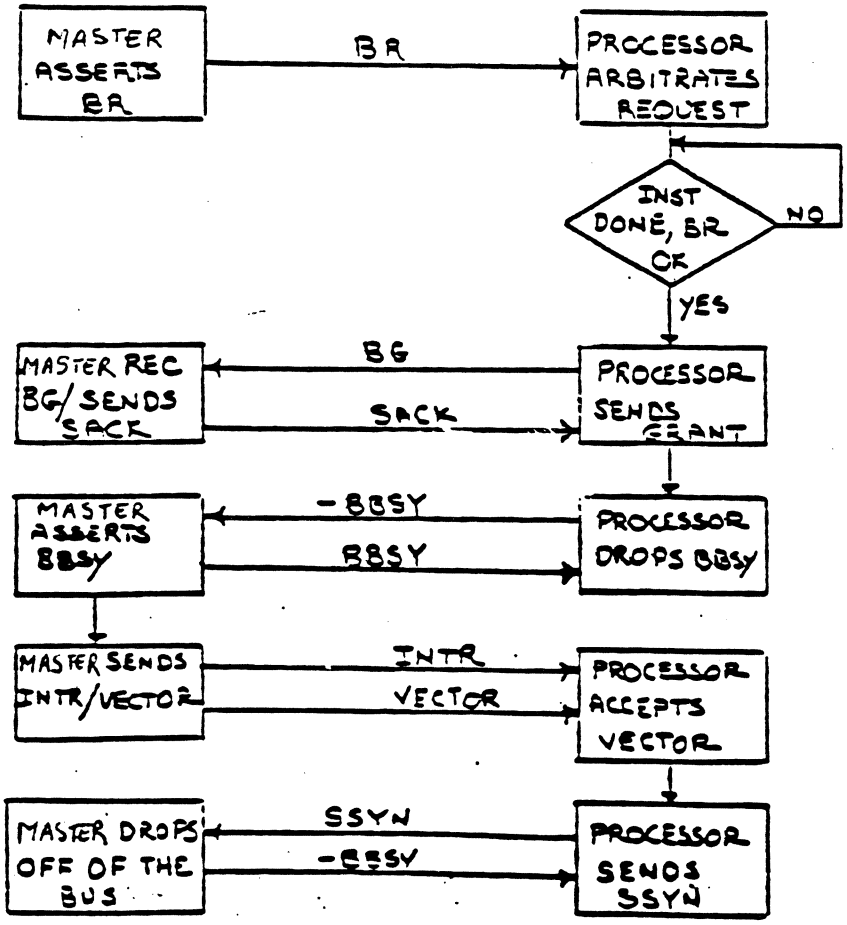
TRAP ABORT FLOW CHART



NOTE: All traps operate the same, the only difference is:

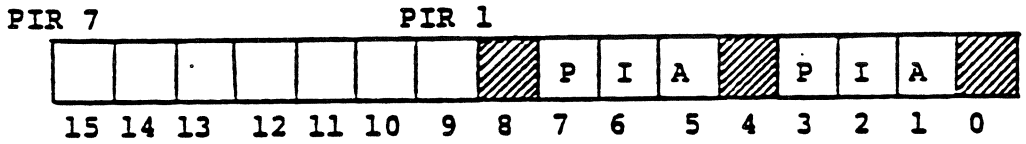
1. Cause of Trap
2. Vector Address


BASE PROGRAM
VECTOR
NEW PC
NEW PSW
"PUSH"
OLD PC
OLD PSW
INTERRUPT SERVICE ROUTINE
PROCESSOR STACK
"POP"
RTI


MASTER ASSERTS BR
BR
PROCESSOR ARBITRATES REQUEST
INST DONE, BR OK
NO
YES
MASTER REC BG/ SENDS SACK
BG
SACK
PROCESSOR SENDS GRANT
MASTER ASSERTS BBSY
-BBSY
BBSY
PROCESSOR DROPS BBSY
MASTER SENDS INTR/VECTOR
INTR
VECTOR
PROCESSOR ACCEPTS VECTOR
MASTER DROPS OFF OF THE BUS
SSYN
-BBSY
PROCESSOR SENDS SSYN

Program Interrupt Requests

A request is booked by setting one of the bits 15 through 9 (for PIR 7 = PIR 1) in the Program Interrupt Register at location (17)777772. The hardware sets bits 7--5 and 3--1 to the encoded value of the highest PIR bit set. This Program Interrupt Active (PIA) should be used to set the Processor Level and also index through a table of interrupt vectors for the seven software priority levels. The figure below shows the layout of the PIR register.



Program Interrupt Request Register

When the PIR is granted, the Processor will Trap to location 240 and pick up PC in 240 and the PSW in 242. It is the interrupt service routine's responsibility to queue requests within a priority level and to clear the PIR bit before the interrupt is dismissed.

The actual interrupt dispatch program should look like:

```
MOVE PIR,PS                ; placed Bits 5--7 in PSW Priority
                             Level Bits

MOV R5, --(SP)             ; save R5 on the stack

MOV PIR,R5

BIC #177761,R5             ; Gets Bits 1--3

JMP @DISPAT(R5)            ; use to index through table which
                             requires 15 core locations.
```

Example:

```
                                                    Error Trap
Load Address = 500/START/ASSUME 4K CORE             ==========
                                                    (Bus Time Out)

        500 - MOV # 1500,%6
        502 - 1500
        504 - MOV #5000,@#4
        506 - 5000
        510 - 4
        512 - CLR @#6
        514 - 6
        516 - MOV #22,@#777550  *Non-Existent Address
        520 - 22
        522 - 777550
        524 - HALT

       5000 - HALT
```

When program HALTS, what should be contents of following:

R[VECT] =

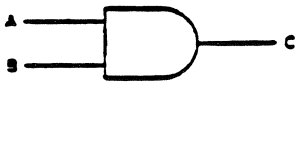
R6 =

Loc 1474 =

Loc 1476 =

R7 =

PSW =



A	B	C
L	L	L
L	H	L
H	L	L
H	H	H

wi-0079

Positive AND Gate
(And Function)

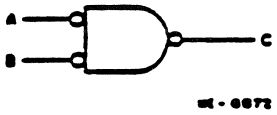
wi-0073

Positive AND Gate
(Or Function)

A	B	C
L	L	L
L	H	H
H	L	H
H	H	H

wi-0077

Positive OR Gate
(Or Function)



wi-0072

Positive OR Gate
(Or Function)

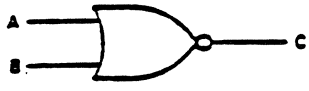
A	B	C
L	L	H
L	H	H
H	L	H
H	H	L

wi-0074

Positive NAND Gate
(And Function)

wi-0078

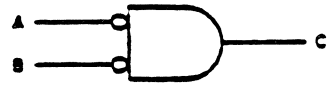
Positive NAND Gate
(Or Function)



A	B	C
L	L	H
L	H	L
H	L	L
H	H	L

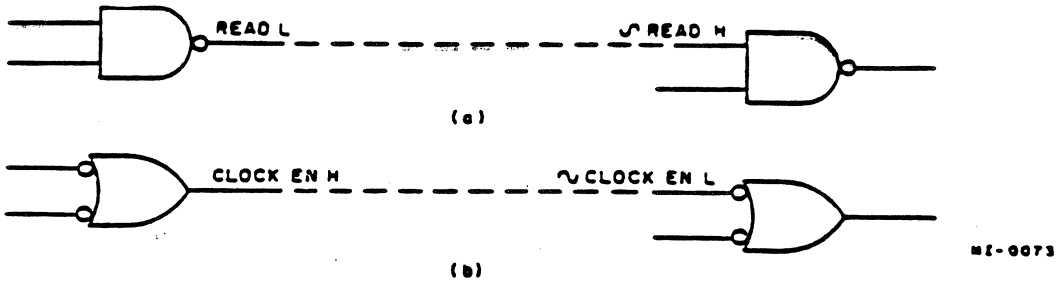
wi-0075

Positive NOR Gate
(Or Function)

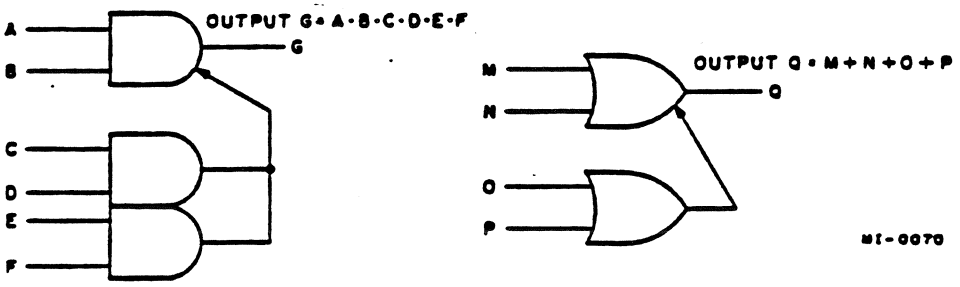


wi-0080

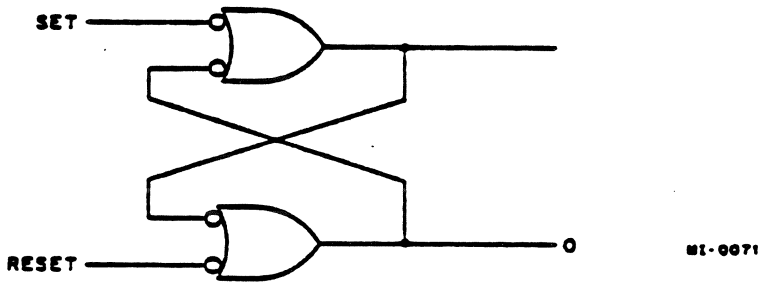
Positive NOR Gate
(And Function)



AND/OR Gate Signal Names

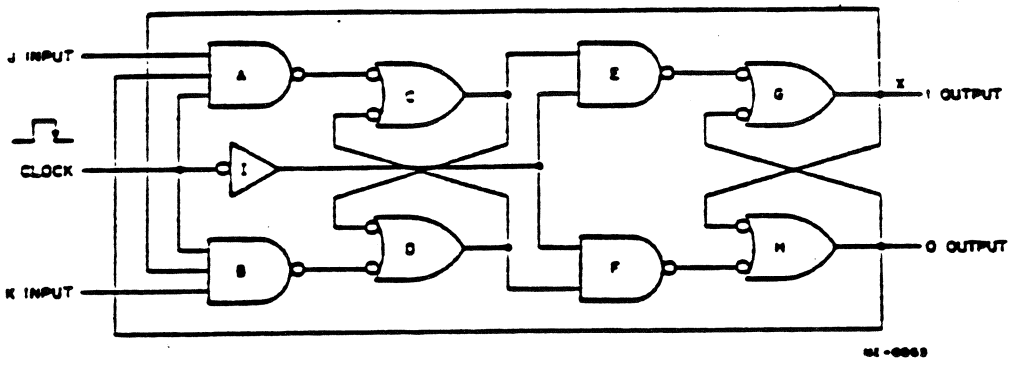


Expanded Gate



Previous State		Input Condition		Result	
1	0	SET	RESET	1	0
L	H	L	H	H	L
H	L	H	L	L	H
L	H	H	H	no change	
H	L	H	H	no change	
H	L	L	H	no change	
L	H	H	L	no change	
L	H	L	L	H	H*
H	L	L	L	H	H*

*Ambiguous state: the input that stays low longest will assume control.

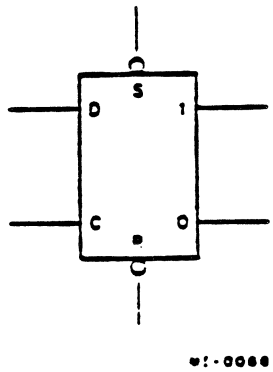


Master/Slave J-K Flip-Flop

Initial Conditions				Final Conditions	
Outputs		Inputs		Outputs	
1	0	J	K	1	0
LO	HI	LO	LO	LO	HI
LO	HI	LO	HI	LO	HI
LO	HI	HI	LO	HI	LO
LO	HI	HI	HI	HI	LO
HI	LO	LO	LO	HI	LO
HI	LO	LO	HI	LO	HI
HI	LO	HI	LO	HI	LO
HI	LO	HI	HI	LO	HI

D-Type Flip-Flop

The D-type flip-flop has four inputs: SET (S), RESET (R), DATA (D), and CLOCK (C). Each flip-flop has two outputs: 1 and Ø. The flip-flop is bistable; i.e., it remains in one of its two logic states (1 or Ø) until an input condition causes it to change.



D-Type Flip-Flop

A flip-flop is set to the 1 state if either of the following conditions occur:

a. A negative-going pulse appears at the SET input.

b. The DATA input ig high and a positive-going pulse is applied to the CLOCK input.

A flip-flop is reset (set to the Ø state) if either of the following conditions occur:

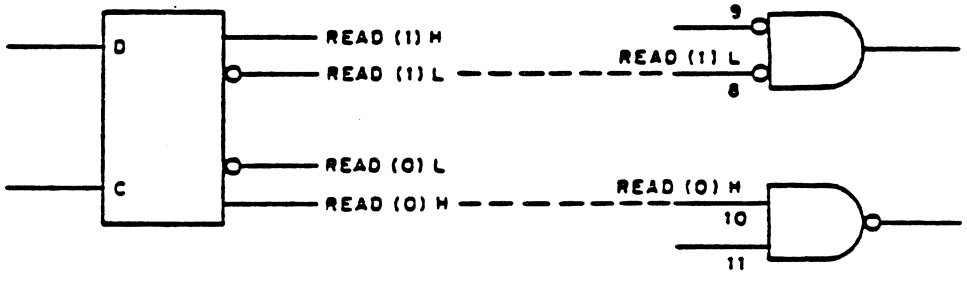
a. A negative-going pulse appears at the RESET input.

b. The DATA input is low and a positive-going pulse appears at the CLOCK input.

When a flip-flop is in the 1 state, the 1 output is high and the Ø output is low. When a flip-flop is in the Ø state, the Ø output is high and the 1 output is low.

In certain cases, flip-flops on DEC logic prints may be redefined flip-flops whereby the Ø and 1 output functions are reversed; in other words, the conditions that set a normal flip-flop to the 1 state will set a redefined flip-flop to the Ø state, and vice versa.

Flip-flops on DEC logic prints are shown as indicated.

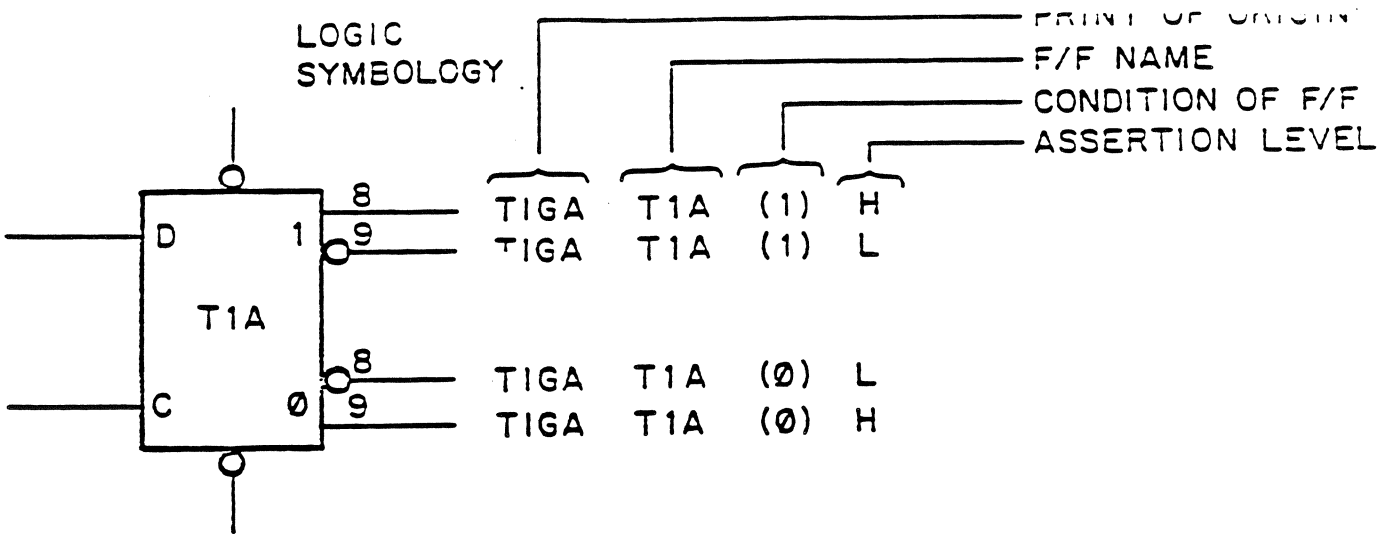


NOTE

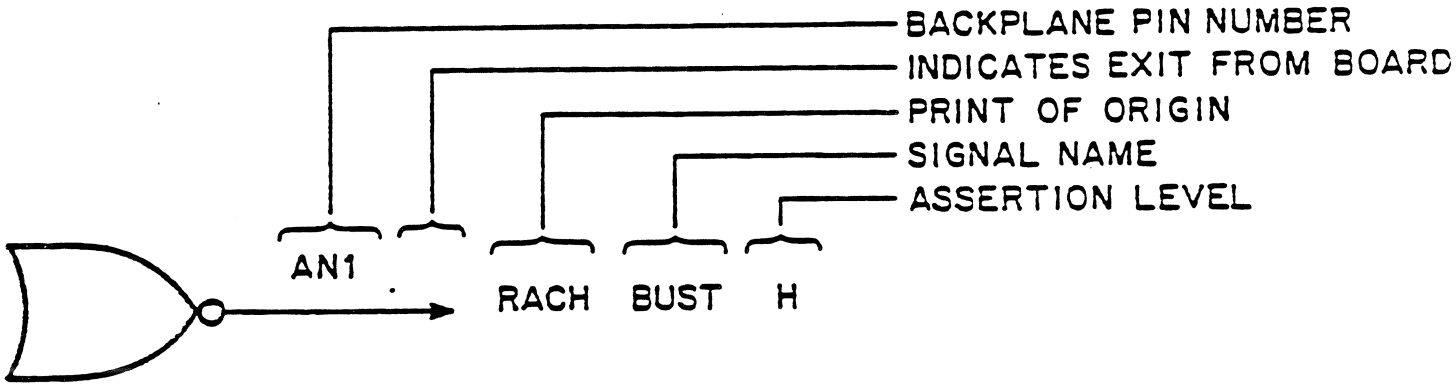
The output pin for READ (1) L is physically and electrically the same pin as the output pin for READ (0) H and vice-versa.

MI-0067

Flip-Flop Signal Destinations



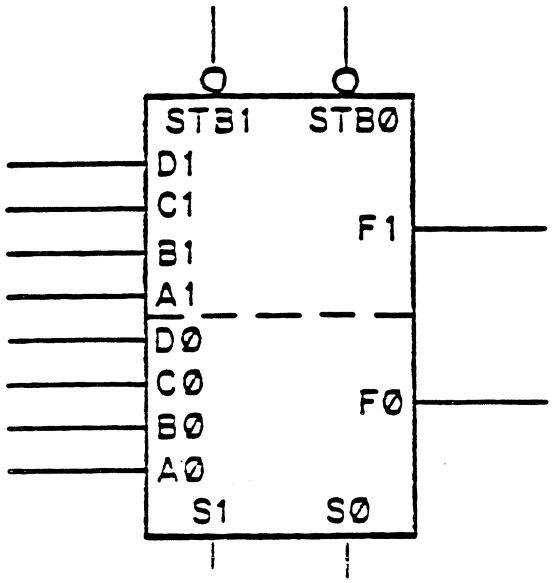
TR-1328



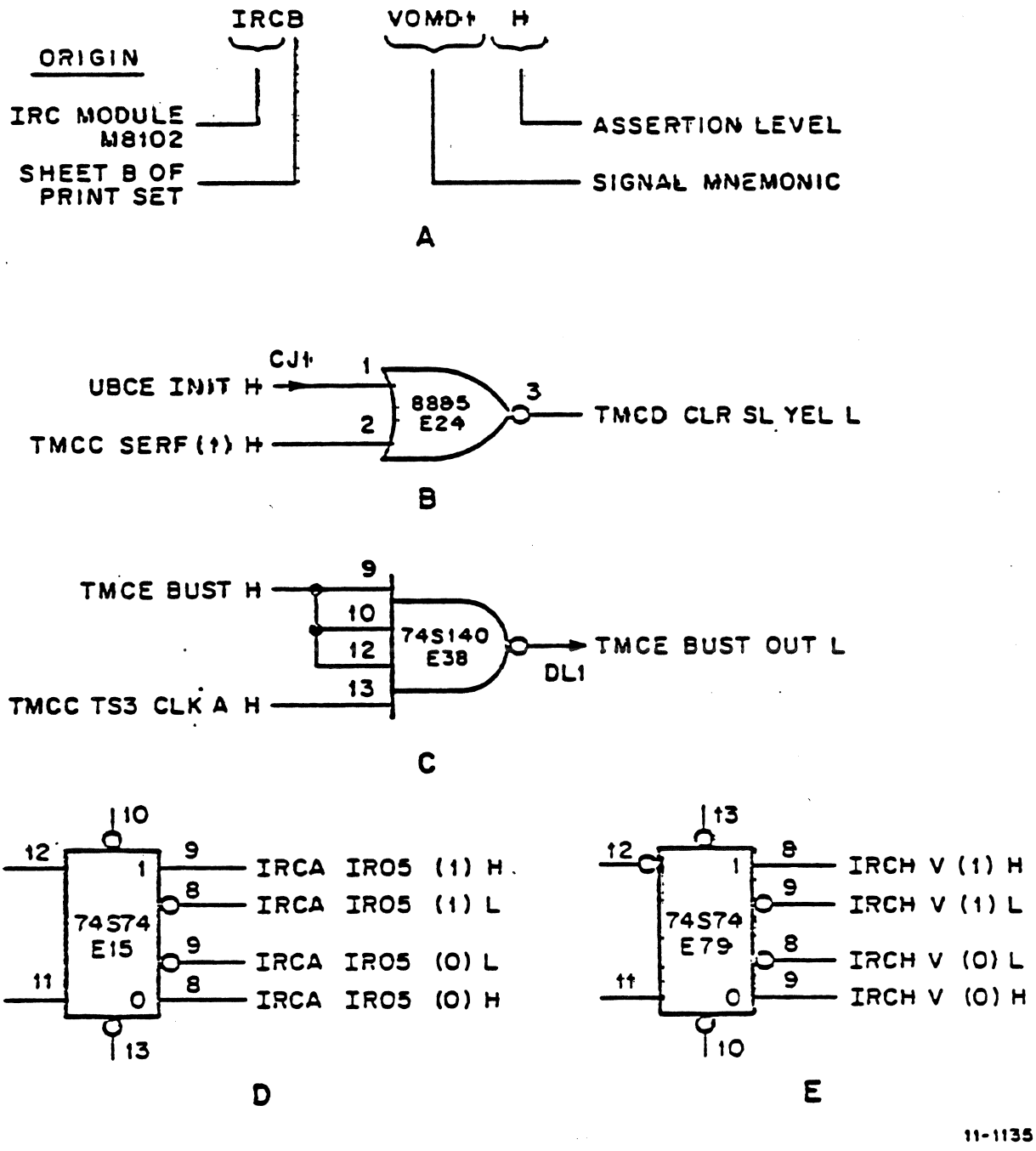
TR-1329

GATE OUTPUT NAMING STANDARD

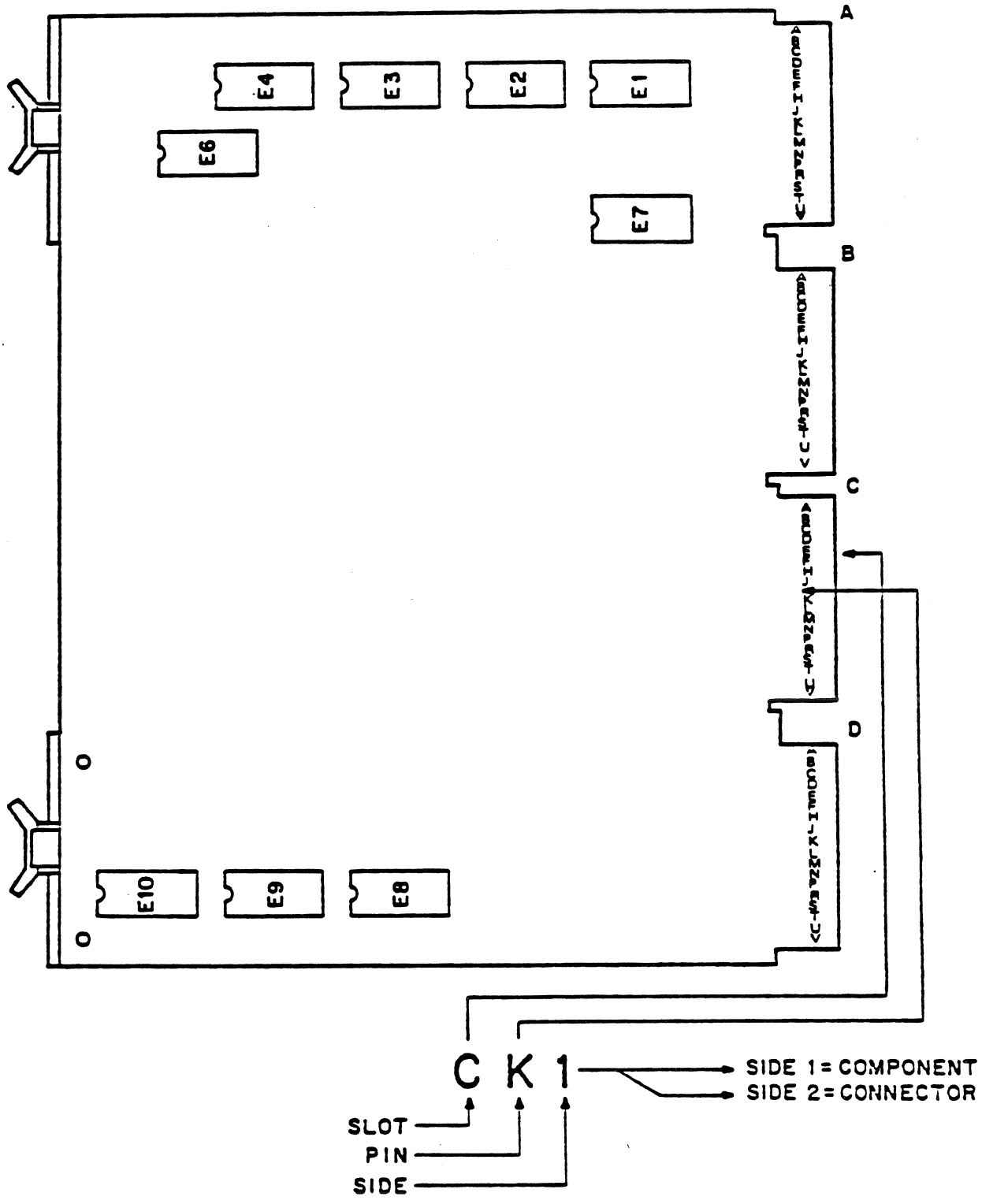
74S153 DUAL FOUR-TO-ONE MULTIPLEXER



S1	SØ	F
L	L	A
L	H	B
H	L	C
H	H	D



PDP-11/45, 11/50 Drawing Convention Examples


A
B
C
D
E1
E2
E3
E4
E6
E7
E8
E9
E10
C K 1
SIDE 1 = COMPONENT
SIDE 2 = CONNECTOR
SLOT
PIN
SIDE

TR-1325

Study Questions:

1. Which bit of the PIR register must be set to generate the request PIR2?

2. If the PIR has bits 15, 13, and 10 set, what will be the value in bits <7:5> ?

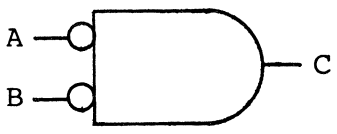
_____

3. When an interrupt occurs, PC and PSW are pushed onto the stack. Which will appear in the lower numbered address?

4. If a BR7 interrupt is in progress and a PIR7 request is generated, will the PIR7 interrupt take place immediately, before the BR7 interrupt is finished?

_____

5. Complete the truth table for the following logic gate:



A	B	C
L	L	
L	H	
H	L	
H	H	

6. The output of a flip flop has the following name:

UBCB MSYN (1) H

This outout is equivalent to _____.

A. UBCB MSYN (0)H
B. UBCB MSYN (0)L
C. UBCB MSYN (1)L
D. UBCB MSYN H

7. A signal is found on the pin BC1. Which of the following statements about that signal is true?

   A. It is connected to a pin in backplane row C.
   B. It is connected to the component side of a module.
   C. It is found in slot 1, and only slot 1.
   D. It is connected to the solder side of a module.

8. Backplane pins are named with the letters of the alphabet from A through V, with the letters _____ omitted.

PDP-11/70
Hardware Introduction
Study Assignment 3
Answer Sheet

Study Questions:

1. Which bit of the PIR register must be set to generate the request PIR2?

BIT 1Ø

2. If the PIR has bits 15, 13, and 1Ø set, what will be the value in bits <7:5> ?

111 (7)

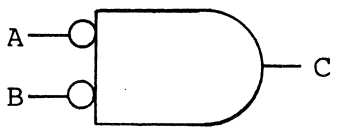
3. When an interrupt occurs, PC and PSW are pushed onto the stack. Which will appear in the lower numbered address?

PC

4. If a BR7 interrupt is in progress and a PIR7 request is generated, will the PIR7 interrupt take place immediately, before the BR7 interrupt is finished?

NO

5. Complete the truth table for the following logic gate:



A	B	C
L	L	H
L	H	L
H	L	L
H	H	L

6. The output of a flip flop has the following name:

UBCB MSYN (1) H

This outout is equivalent to B.

A. UBCB MSYN (Ø)H
B. UBCB MSYN (Ø)L
C. UBCB MSYN (1)L
D. UBCB MSYN H

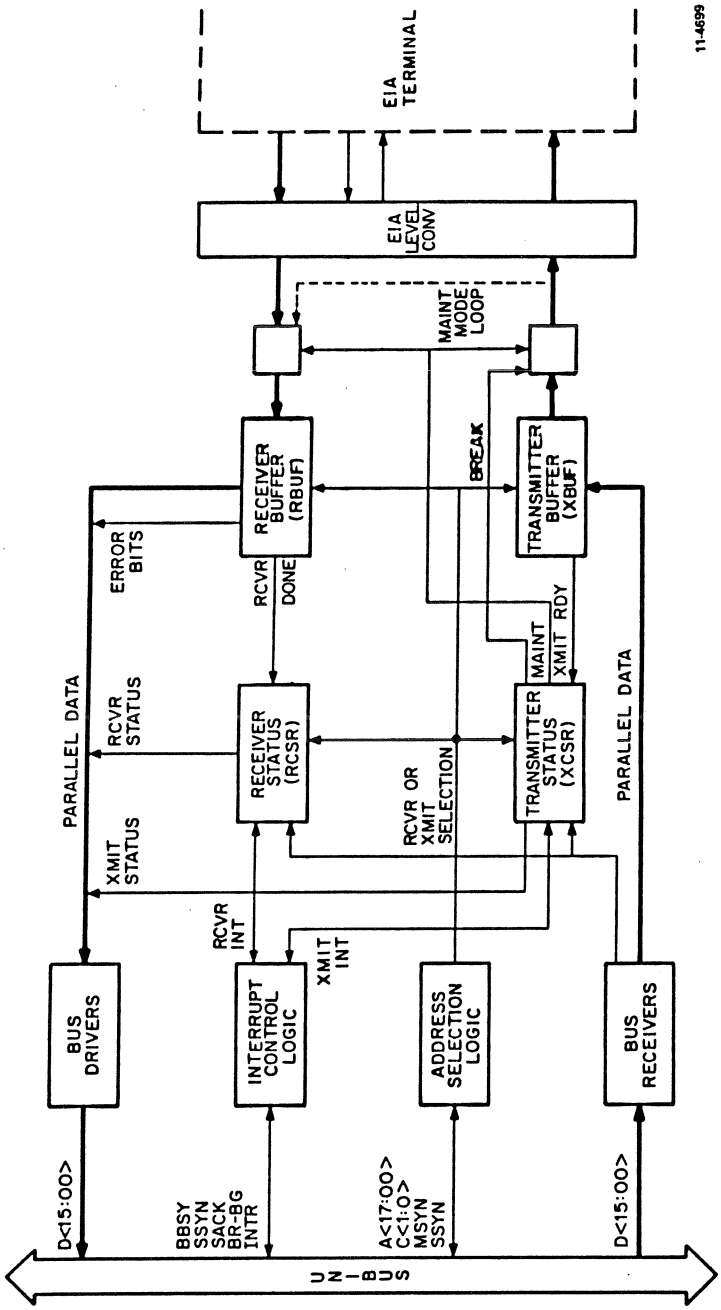
7. A signal is found on the pin BC1. Which of the following statements about that signal is true?

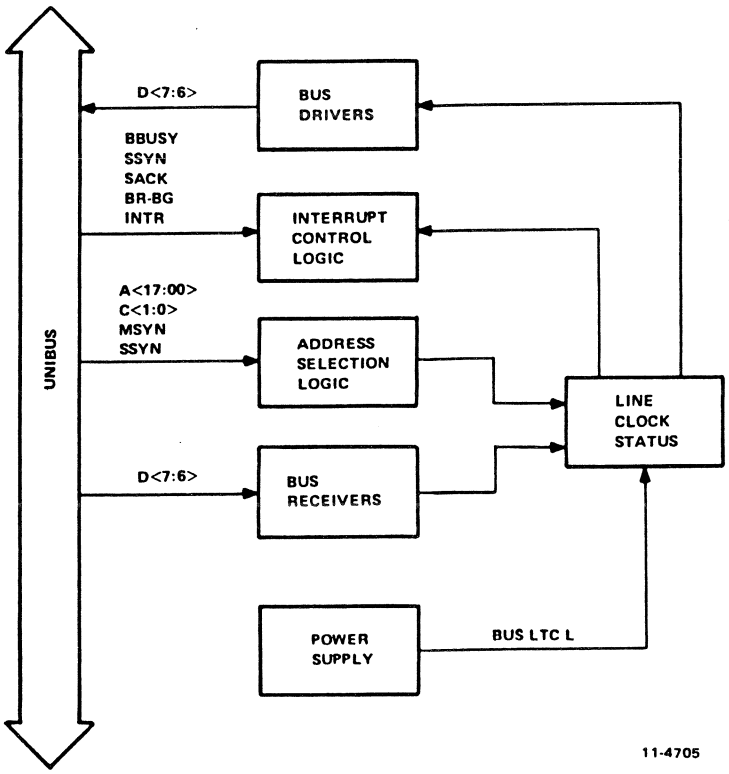
   A. It is connected to a pin in backplane row C.
   B. It is connected to the component side of a module.
   C. It is found in slot 1, and only slot 1.
   D. It is connected to the solder side of a module.

8. Backplane pins are named with the letters of the alphabet from A through V, with the letters G,I,O,Q omitted.

PDP-11/70 Diagnostics and Module Level Repair

DAY THREE

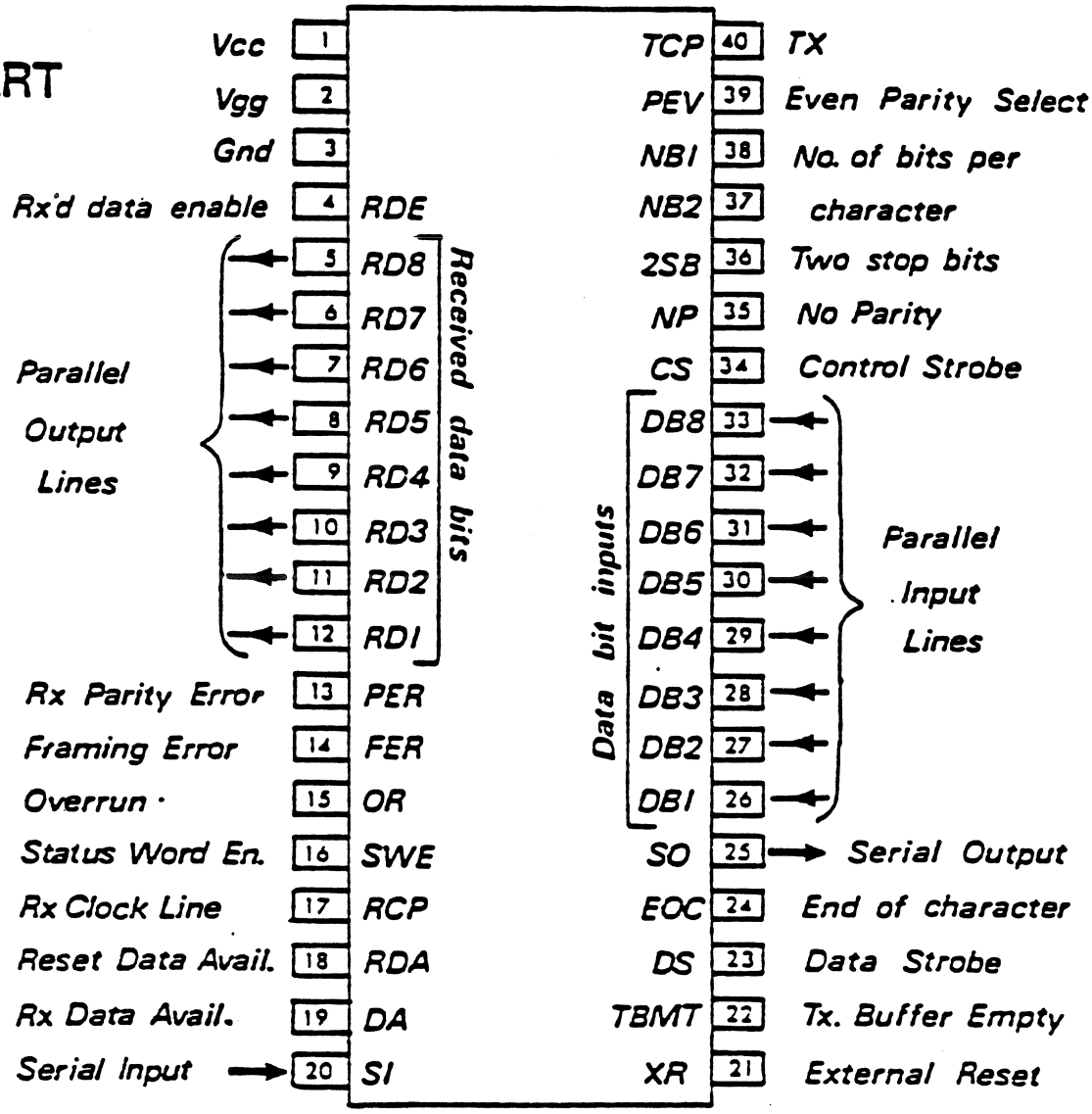

EIA TERMINAL
EIA LEVEL CONV
MAINT MODE LOOP
BREAK
RECEIVER BUFFER (RBUF)
TRANSMITTER BUFFER (XBUF)
ERROR BITS
RCVR DONE
XMIT RDY
MAINT
PARALLEL DATA
RCVR STATUS
RECEIVER STATUS (RCSR)
TRANSMITTER STATUS (XCSR)
RCVR OR XMIT SELECTION
XMIT STATUS
RCVR INT
XMIT INT
BUS DRIVERS
INTERRUPT CONTROL LOGIC
ADDRESS SELECTION LOGIC
BUS RECEIVERS
D<15:00>
BBSY
SSYN
SACK
BR-BG
INTR
A<17:00>
C<1:0>
MSYN
SSYN
D<15:00>
U N - B U S
11-4699


UNIBUS
D<7:6>
BUS
DRIVERS
BBUSY
SSYN
SACK
BR-BG
INTR
INTERRUPT
CONTROL
LOGIC
A<17:00>
C<1:0>
MSYN
SSYN
ADDRESS
SELECTION
LOGIC
LINE
CLOCK
STATUS
D<7:6>
BUS
RECEIVERS
POWER
SUPPLY
BUS LTC L
11-4705

1808

19-10459

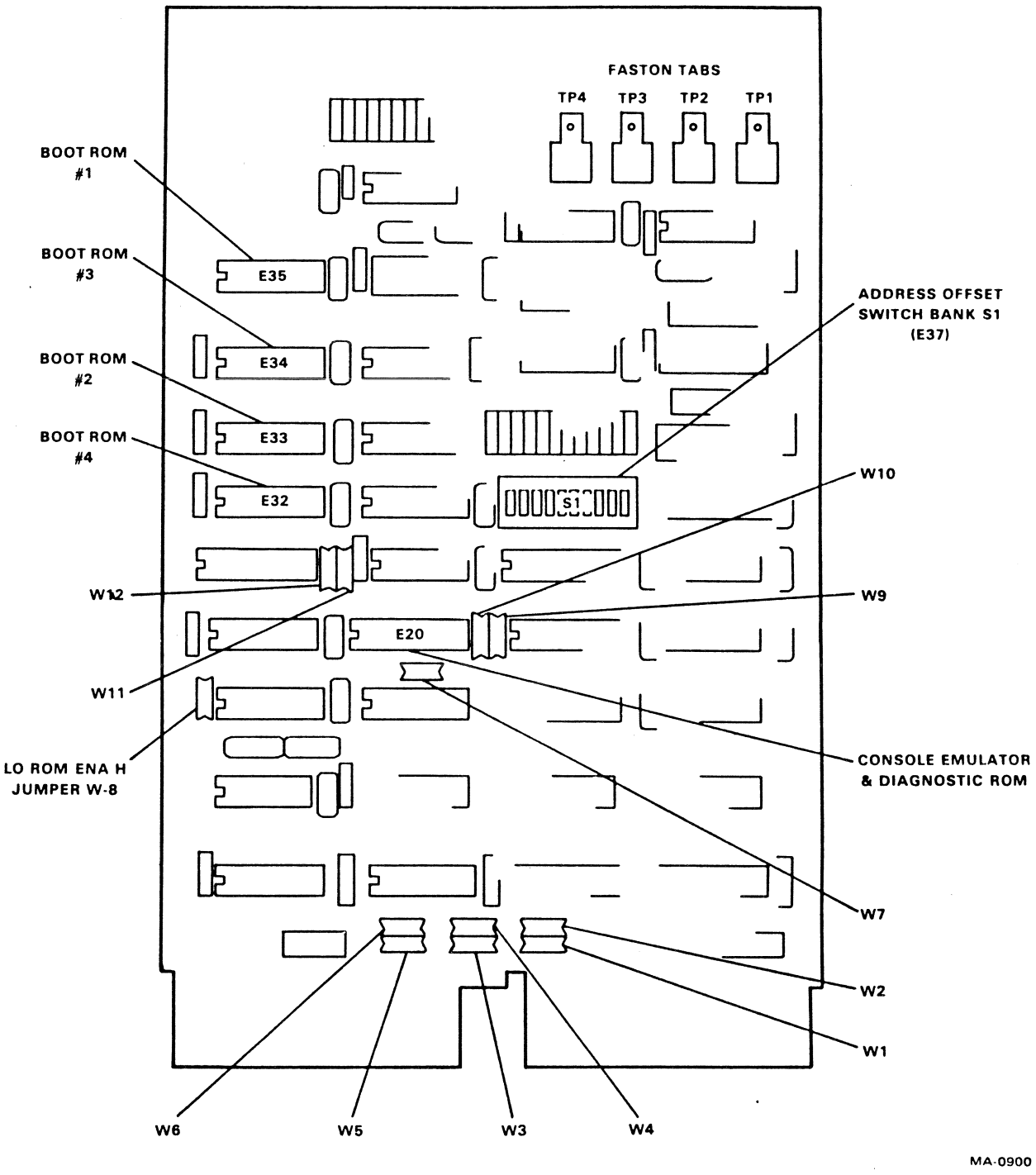
UART



PIN NO.	SYMBOL	NAME	FUNCTION
1	Vcc	Vcc Power Supply	+5v. supply
2	Vgg	Vgg Power Supply	-12v. supply
3	Gnd	Ground	Ground
4	RDE	Received Data Enable	A low on the receiver enable line placed the received data onto the output lines.
5-12	RD8-RD1	Received Data Bits	These are the 9 data output lines. These lines may be "Wire-Ored". When 5, 6, or 7 level code is selected the most significant unused bits are Low. Character will be right justified into the least significant bits. RD1 (Pin 12) is the least significant bit, RD8 (Pin 5) is the most significant bit. A High indicates a Mark.
13	PER	Receive Parity Error	This line goes to a High if the received character parity does not agree with the selection (Pin 39).
14	FER	Framing Error	This line goes to a High if the received character has no valid Stop bit, i.e., the bit following the Parity bit is not marking.
15	OR	Overrun	This line goes to a High if the previously received character is not read (DA line not Reset) before the present character is transferred to the receiver holding register.
16	SWE	Status Word Enable	A Low on this line placed the Status Word bits (PE, DA, TBMT, FE, OR) onto the output lines.
17	RCP	Receiver Clock Line	Requires a clock 16 times required Rx baud rate.
18	RDA	Reset Data Available	A Low on this line will reset the DA line.

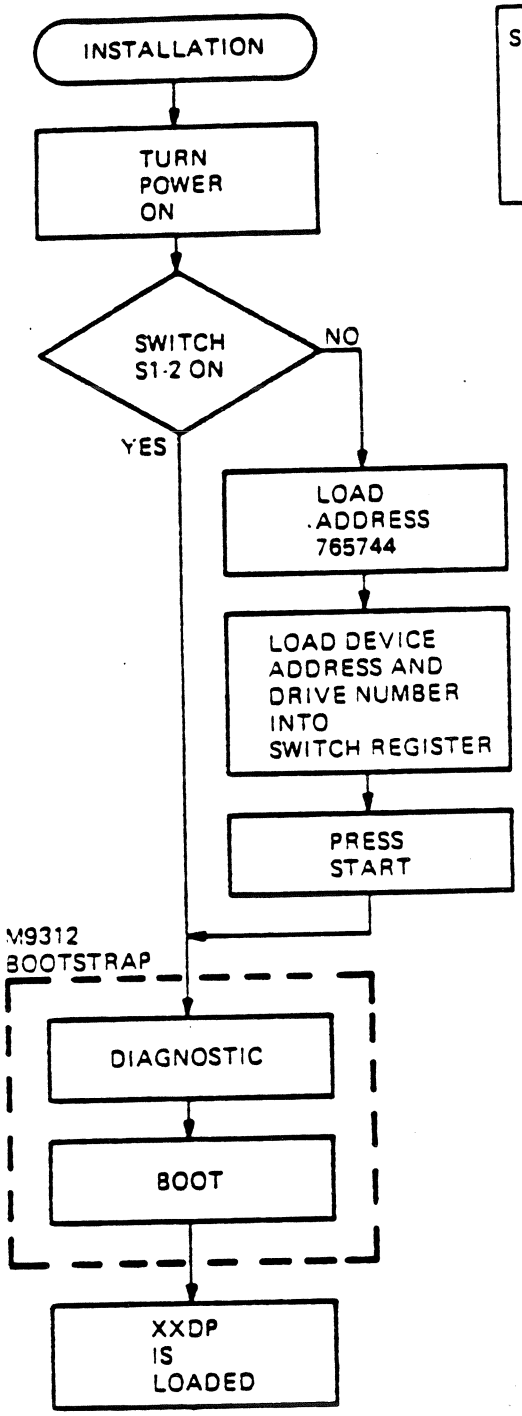
PIN NO.	SYMBOL	NAME	FUNCTION
34	CS	Control Strobe	A high on this lead will enter the control bits (POE,NB1,NB2,SB,NP) into the control bits Holding register. This line can be strobed or hard wired to a High level.
35	NP	No Parity	A High on the lead will eliminate the parity bit from the transmitted and received character. The stop bits will immediately follow the last data bit on transmission. The receiver will not check parity or reception. It will, when asserted, also clamp the PE to a Low.
36	2SB	Two Stop Bits	This lead will select the number of stop bits. 1 or 2 to be appended immediately after the parity bit. A Low will insert 1 stop bit and a High will insert 2 stop bits.
37-38	NB2, NB1	Number of Bits/ Charact.	37 38 Bits/Character L L 5 L H 6 H L 7 H H 8
39	PEV	Even Parity Select	L Inserts/checks odd H Inserts/checks even
40	TCP	Trans-mitter	Requires clock freq. 16 times required Tx baud rate.

PIN NO.	SYMBOL	NAME	FUNCTION
19	DA	Received Data	This line goes to a High when an entire character has been received and transferred to the receiver Holding register.
20	SI	Serial Input	This line accepts the Serial bits input stream. A High must be present when Data is not being received. High is a Mark. Low is a Space.
21	XR	External Reset	Should be pulsed after Power turn on to a High. Reset all registers. Sets Serial Output line to a High.
22	TBMT	Transmitter Buffer Empty	The Transmitter Buffer Empty flag goes to a High when the data bits Holding Register may be loaded with another character.
23	DS	Data Strobe	A Low to High transition on this line will enter the data bits into the Data Bits Holding Register. Data loading is controlled by the rising edge of DS.
24	EOC	End of Character	This line goes to a High each time a full character including Stop bits is transmitted. It remains at this level until the start of transmission which is the mark to space transition of the Start bit. It will remain as a High when data is not being transmmitted.
25	SO	Serial Output	This line will serially, by bit, provide the entire transmitted character. It will remain at a High when no data is being transmitted. High is a Mark, Low is a Space.
26-33	DB1-DB8	Data Bit Inputs	These are the 8 parallel data input lines. If 5, 6, 0r 7 bits are transmitted the least most significant bits are used. DB1 is the least significant bit (Pin 26). DB8 is the most significant bit, (Pin 33). A High input will cause a mark (High) to be transmitted.



M9312 Bootstrap/Terminator Module

M9312 BOOT PROCEDURE & R.O.M. IDENTIFICATION



SELECT BOOT FUNCTIONS VIA MICROSWITCHES (S1)

S1-1	OFF
S1-2	POWER-UP BOOT ENABLE
S1-3 & 4	BOOT ROM SOCKET NUMBER
S1-5:9	DEVICE ROM ADDRESS
S1-10	DIAGNOSTIC ON/OFF SWITCH

DEVICE ADDRESS SW REG<08:00>

X12	RL01, RK06/07, RX01, RX02, RP02/RP03, RK03/5/5J, TU16/E16, TM02/03, RS03/04, PC05, TU60, RS11, CR11
X42	TU55/56, DL11A/W
X56	RP04/516, RM02/03, RS64

X = 0 FOR BOOT ROM SOCKET 1
X = 2 FOR BOOT ROM SOCKET 2
X = 4 FOR BOOT ROM SOCKET 3
X = 6 FOR BOOT ROM SOCKET 4

DRIVE NUMBER SW REG<11:09>

* ROM SOCKET # CAN BE FOUND BY EXAMINING I.D. CODE OF EACH ROM's I.D. LOCATION 17 773 xxx : WHERE xxx= 000/ROM#1; 200/#2; 400/#3; & 600/#4. BELOW IS A LIST OF I.D. CODES AND SUPPORTED DEVICES:

	ID CODES	DEVICES
	042114	RL01
	042115	RK06,07
	042130	RX01
	042131	RX02
**	042120	RP02,03, RP04/516, RM02,03
	042113	RK03,05/05J, TU55/56
**	046515	TU16/E16, TM02/03
	046524	TU10/TE10, TS03
	042123	RS03/04
	050122	PC05, DL11A/W
	041524	TU60
	042106	RS11, RS64
	041522	CR11

** = FREQUENTLY USED 1170 LOAD DEVICES. TK-1458

M9312 DIAGNOSTIC ROM
ERROR HALTS

Address Displayed	Test No	Failing Subroutine
165052	1	Branch always
165070	2	CLR, BMI, BVS, BHI, BLOS, BLT
165104	3	DEC, BPL, BEQ, BGE
165116	4	ROR, BVC, BHIS, BNE
165146	5	Test register data path
165156	6	ROL, BCC, BLT
165174	7	ADD, INC, COM, BCS, BLE
165214	10	ROR, DEC, BIS, ADD, BLO
165234	11	COM, BIC, BGT, BLE
165256	12	SWAB, CMP, BIT, BNE, BGT
165310	13	MOUB, SOB, CLR, TST, BPL, BNE
165362	14	JSR, RTS, RTI, JMP
165456	15	Test main memory from virtual 1000 to last address. RØ = address R1 = bad data R2 = good data
165500	15	Data error RØ = address R1 = bad data R2 = good data
165544	16	Cache data test. Cache data didn't match. RØ = address R4 = expected data
165554	16	Cache failed to hit RØ = address that was referenced
165664	17	Test memory with cache on. Data didn't compare RØ = address + 2 that failed
165676	17	Hit failed to occur RØ = address + 2 that failed
165716	17	Cache or memory parity error. Check cache control register to determine cause of failure.

Errors - M9312 – The following is a list of diagnostic error halt addresses and the corresponding failing test. Most of the errors are hard failures and there will be no recovery from them. If one of the cache memory tests fails (test 23 at location 165564 or test 24 at location 165704), attempt to boot the system by pressing continue. This will cause the program to force misses in both groups of cache before going to the bootstrap portion of the program.

PDP 11/70 DIAGNOSTICS

All 1170 diagnostics load and start at address 200_8. Here is a list of the diagnostics, their functions, special warnings, adn the correct test sequence.

EKBA?? CPU Part 1
Test traditional PDP 11 instructions

EKBB?? CPU Part 2
Test advanced PDP 11 instructions

EKBC?? CACHE Part 1
Basic CACHE diagnostics

EKBD?? CACHE Part 2
Advanced cache diagnostics.
Warning - the diagnostics will use any write enabled RP's, RS's, or RK05's so always remove any volatile media.

EKBE?? Memory Management Test

EKBF?? Unibus Map Test

EMKA?? MOS/Core Memory Test
Checks out both MJ11 and/or MK11 memory.

EMJA?? MJ11 Memory Test
Test MJ11 core. Always use EMJA?? after EMKAB0 on MJ11 systems to detect errors missed by EMKAB0/A0. Warning - EMJA?? should always run with CACHE disabled. EMJAD0 does that automatically, but older versions require depositing 14(8) at 17777746 (CACHE Control Register) after diagnostic startup, using HALT, DEPOSIT, and CONTINUE.

EQKC?? 1170 CPU Instruction Exerciser
Comprehensive test of CPU/memory/cache and I/O
Switch Register Bit 6 = 1 will inhibit relocation above 16K. SWR Bit 8 = 1 is required for disk or tape relocation. Use of SWR Bit 4 = 1 (Inhibit Random Disk relocation) has been found to allow successful relocation with RM03's, currently unsupported under EQKC??.

EMKAB0
MOS/CORE MEMORY TEST

SWITCH REGISTER SETTINGS

15	(100000)	HALT ON ERROR
14	(40000)	LOOP ON TEST
13	(20000)	INHIBIT ERROR TYPEOUTS
12	(10000)	INHIBIT PROGRAM RELOCATION
11	(4000)	QUICK VERIFY/INHIBIT MARGINS
10	(2000)	BELL ON ERROR
9	(1000)	LOOP ON ERROR
8	(400)	HALT PROGRAM. Program should always be halted this way if MOS to flush possible DBE's.
7	(200)	DETAILED ERROR REPORT
6	(100)	PRINT CONFIGURATION MAP
5	(40)	LIMIT MAX ERRORS PER BANK
4	(20)	FAT TERMINAL. 132 columns
3 thru 1		TEST MODE
	(0)	0 BAFPAF banks fwd, patterns fwd
	(2)	1 BAFPAR banks fwd, patterns rev
	(4)	2 BAWPAF banks worst first, patterns fwd
	(6)	3 BAWPAR banks worst first, patterns rev
	(10)	4 PAFBAF patterns fwd, banks fwd
	(12)	5 PAFBAW patterns fwd, banks worst first
	(14)	6 PARBAF patterns rev, banks fwd
	(16)	7 PARBAW patterns rev, banks worst first
0	(1)	DETECT SINGLE BIT ERRORS.

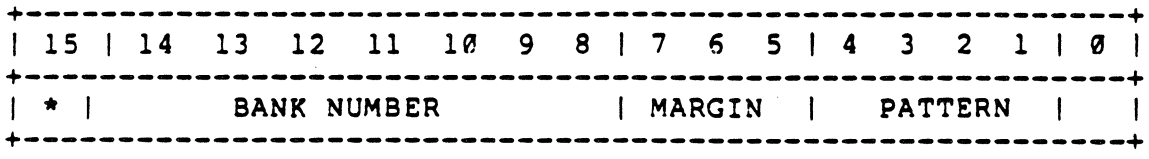
EMKABØ
MOS/CORE MEMORY TEST

TERMINAL COMMANDS

CONTROL-C	Cleans up program, exits neatly, but tries to boot DKØ. If boot fails, then tries DK1. Flushes Double Bit Errors in MK11.
CONTROL-D	Enter a modified version of ODT. (Not advised when program is relocated.)
CONTROL-E	Exit from ODT.
CONTROL-T	Tell me what's happening. Gives current bank, pattern under test.
CONTROL-F	Field Service Mode.
Ø	Exit field service mode.
1	Read MK11 CSR.
2	Load MK11 CSR.
3	Examine memory.
4	Modify memory.
5	Select BANK, MARGIN, PATTERN for testing.
6	Type configuration map.
7	Battery Backup test.
8	SOB-A-Long test.
9	Super tight scope loop.
1Ø	Error summary.
11	Refresh test.
12	Set fill count.
13	Enter KAMIKAZE mode. (failures in this mode may crash the program)
14	Exti KAMIKAZE mode.
15	Turn off CACHE.
16	Turn on CACHE.
17	Run only multi-port tests.
18	Resume running both single and multi-port tests.
19	Test only selected banks.
2Ø	Resume testing all banks.

EMKAB0
MOS/CORE MEMORY TEST

DISPLAY REGISTER



* - 1=RELOCATED
0=NOT RELOCATED

MARGINS

	MJ11	MK11
0	NORMAL	SAME
1	FORCE "WRONG" ADD. PAR. ERR.	SAME
2	EARLY STROBE	EARLY MDR LOAD
3	LATE STROBE	LATE REFRESH
4	LOW CURRENT	RESERVED
5	HIGH CURRENT	RESERVED

11/70 MEMORY "CLEAR" ROUTINE

FOR ALL OF MEMORY: 0-TO-SYSTEM SIZE

This program will write any pattern loaded in R5 throughout all of memory from location 0 to system size register. It executes from superviser "Instruction and Data Space" PAR's, on Unibus I/O page.

R5 contains pattern to write, or "0" to clear memory.

Cache and Unibus map will not be used.

This routine is primarily intended to clear MJ11 core memory, which may have bad data parity from previous leftover errors or stack module (H217C/H224-C) replacements. Each MK11 box can be cleared by interrupting box AC/BATTERY Power switch. However, this routine is also useful to write known patterns.

NOTE

> Do not use "CLR" instruction when clearing 11/70 memory as CPU does a DATIP/DATO sequence. The "DATIP" will cause parity traps from CACHE on MJ/MK11 errors. Use the "MOV" instruction.

This routine can be used to scan memory for leftover Data Parity Errors from software crashes by replacing "MOV R5,(R0)+" at 17772246 ($1 symbol) with a "TST (R0)+". Also put a TRAP catcher at the Parity Trap Vector:

LOCATION 114 DEPOSIT 116
LOCATION 116 DEPOSIT 0

This will halt the system on at 120 on a Parity Error Trap. Use the Memory System Error Register and HI/LO Address registers to locate the memory location with bad parity.

USE THE FOLLOWING STEP-BY-STEP PROCEDURE

1. Press HALT, then START to initialize CPU and system.

2. Ensure system size register (17777760) accurately reflects actual memory size. Note that it should equal last memory address right-shifted two (2) octal positions.

3. Load following values and program into indicated Memory Management and General Registers. Remember to use "Console Physical" address select mode and <u>full</u> 22-bit address when loading addresses.

```
LOCATION        CONTENTS        INSTRUCTIONS & NOTES

17772300        077406          KIPDR 0
17772316        077406          KIPDR 7 = I/O page for program
17772340        000000          KIPAR 0; Load 200 if using
                                trap catchers.
17772356        177600          KIPAR 7 = I/O page for program
17777700        000000          R0 = Start Virtual Address
17777701        172340          R1 = KIPAR 0 Addresses
17777702        177572          R2 = MMR0 Address
17777703        177760          R3 = System Size Reg Address
17777704        172516          R4 = MMR3 Address
17777705        (PATTERN)       R5 = Desired Pattern or "0"
17777706        177676          R6 = Stack Pointer
17777776        000000          PSW = 0
17772240        012714          MOV #20, (R4); Enable 22-bit
                                Mapping
17772242        000020
17772244        005212          INC (R2); Enable Memory
                                Management Relocation
17772246        010520  1$:     MOV R5, (R0)+; Data to Memory
17772250        020027          CMP R0, #17776; Top of Page 0?
17772252        017776
17772254        003774          BLE   1$
17772256        062711          ADD #200,(R1); Step Page
17772260        000200
17772262        021311          CMP (R3),(R1); Top of Memory?
17772264        003402          BLE   2$
17772266        005000          CLR R0; Start at Virtual
                                Address 0.
17772270        000766          BR    1$
17772272        005312  2$:     DEC (R2); Disable Relocation
17772274        000000          HALT
```

3. Load Address 172240; Raise HALT; and Press START.

4. Program will HALT when all of memory has been referenced.

5. If program does not HALT, start by examining CPU Error, Memory System Error, and HI/LO Error Address Register. Trap Catchers may be used by depositing 200 in KIPAR 0 (17772340) instead of zero, and setting up vector locations with their address+2 and vector+2 with a zero.

digital

INTEROFFICE MEMORANDUM

TO: All F.S. Branch Unit Managers
cc: Joe DeMaria
Charlie Kapper
Larry Gerrity
George Dollish, PT
Larry Owens, MJ
SUBJECT: Jim Hill, CH
Jesse Simmerman, CH
Mike Babcock, PT
Felix Bodzenski, PT
Bob Roxby, PT

DATE: October 2, 1980
FROM: Bob Betz
DEPT: PHD F.S. Support
EXT: 2320
LOC/MAIL STOP: PH - Blue Bell

1170 DIAGNOSTICS THAT UTILIZE DISKS

Diagnostics DEKBBCO, DO or EO (CPU, Part 2) and DEKBDCO, DO, EO (Cache, Part 2) have the ability to access disk drives on an "RH" controller or an RK05 disk system. This fact can be a benefit in troubleshooting or a catastrophe to the account representative, if he fails to write protect his diagnostic pack or leaves the customer pack on line and write enabled. Following is a synopsis of both diagnostics which may prove useful:

1 - DEKBBCO, DO, EO (CPU, PART 2)

Switch register bit 5 either allows or inhibits BR5 testing. If S.R.5 = 0 allows testing, S.R. 5 = 1 inhibits testing. If S.R.5 = 0, diagnostic will go out and look for a disk drive that is on line, write enabled, and is a BR5; i.e., (RP04, 5, 6, or RK05). Drive will access disk and possibly write on pack.

2 - DEKBDCO, DO, EO (CACHE, PART 2)

Test 31 of "DEKBDCO" checks out "MBC" and will write on disk drive that is on line, write enabled.

Test 35 of "DEKBDDO, EO" checks out "MBC" and will write on disk device that is on line, write enabled.

The following paragraph contains some helpful hints to be followed in order to utilize these diagnostic aids for troubleshooting rather than an aid in zeroing disk packs:

1. Remove customer data and system packs from all disk drives.

2. Write protect the diagnostic pack at all times.

3. If you want to utilize this function of diagnostics, after loading diagnostic, but prior to running, put a scratch pack in a second drive, write enabled and on line and make sure you either take your diagnostic pack off-line or write protect it.

4. In the case of DEKBB?? testing may be inhibited by setting S.R.5 = 1.

5. Always review your diagnostic write ups prior to running diagnostic.

These two diagnostics are the only diagnostics with the exception of the "1170 CPU Instruction Exerciser" (DEOKC??) that utilize a disk drive to perform some testing functions incorporated in the diagnostic.

Please make copies for all your account representatives and contact me if you have questions.

digital

INTEROFFICE MEMORANDUM

TO: All M.A.R. District Support Managers/Supervisors

CC: John Ogden
Larry Fernan
Harold Holmes

DATE: March 6, 1981
FROM: Bob Betz/Scott Swist
DEPT: MAR Product Support/PHD Support
EXT: 294
LOC/MAIL STOP: PC

SUBJ: Running 1170/MJ11/MK11 Memory Margins Utilizing New DECX11

There arises from time to time a need to run memory margins with DECX11 to simulate customer operation when troubleshooting intermittent system problems. DECX11 does not allow this to be dones because it clears the cache maintenance register (17777750) when it is loaded. Following is a procedure that we found that will allow you to run margins utilizing the new DECX11 with either XMONAO,BO,CO.

The instruction that clears the maintenance register is in the "BLOG70" module of the DECX11 map. Using the "E" monitor of all three libraries, it would be located at relative location 136 of the "BLOG70" module for AO and BO and relative location 154 of CO. Following are the absolute locaitons for each library, the contents of the locations and how they should be altered.

XMONAO: ("E" Monitor)

Location	Is	S/B
6514	5077	240 NOP
6516	172502	240 NOP
6520		207 (This Location Used As Reference Only, Do Not Alter)

XMONBO: ("E" Monitor)

Location	Is	S/B
6356	5077	240 NOP
6360	XXXX	240 NOP
6362	207	207 (Reference Only, Do Not Alter)

XMONCO: ("E" Monitor)

Location	Is	S/B
7032	5077	240 NOP
7034	XXXX	240 NOP
7036	12600	(Reference Only, Do Not Alter)

The following procedure is recommended for implementing the above changes and running margins:

1. Load and start your individual DECX11.

2. After CMD> prompt utilize the "MOD" command to implement above changes.

3. Also utilize the "MOD" command to set the desired maintenance register memory margins:

   CMD> MOD 177750 (CR)
   (Use following values)

	MJ11	MK11
4 =	Early Strobe	Early MDR Load
6 =	Late Strobe	Late Refresh
10 =	Low Current	Normal
12 =	High Current	Normal

4. At this point you can now run DECX11 with the desired margins.

5. If you suspect a particular area of memory, utilize the "RUNL" command to lock DECX11 into that particular area.

NOTE: This is to be done when normal diagnostics fail to isolate a problem that is intermittent in nature. Do not patch DECX11 when running it as a confidence check. The patches are an aid to be used only as a further troubleshooting aid.

PROCEDURE TO CHECK OUR THE OPERATION OF
MEMORY MANAGEMENT FROM THE CONSOLE

1. Address Select Switch = Program Physical Position

   Deposit 16 in kernal PDRØ
   17772300/16

   Deposit 50 in kernal PARØ/Base = 5000
   17772340/50

   Deposit 1 in MMRØ/Turn on KT
   17777572/1

2. Address Select Switch = KI Position

   Load address - and
   deposit pattern (IE 125252)

3. Address Select Switch to Program Physical Position

   Address lights should be 5000

4. Load address 5000

   Examine 5000 for correct data

Study Questions:

1. In a standard Control and Status Register, bit 6 is used to _____ and bit 7 indicates _____.

2. All peripheral devices have in their interfaces a _____ _____ register to send or receive data, and a _____ register to determine when the device is ready to send or receive it.

3. Assume that data is to be input to the system from the LA30 via interrupts. The interrupts service routine begins at Location 1000. Show the correct vector addresses and their contents.

Vector Addresses	Contents
_____	_____
_____	_____

Match the statements in Column A with the items in Column B.

"A"

4. Holds the character to be transferred to the external device.
5. Includes the interrupts enable bit that can be used to initiate interrupt sequences when RGVR DONE sets.
6. This bit, when set advances the paper-tape reader.
7. This signal comes up to indicate that a character has been transmitted.
8. This logic is a full duplex receiver/ transmitter.
9. This signal indicates that an incoming character has been assembled in RBUF.

"B"

a. RCSR
b. RBUF
c. UART
d. XBUF
e. XCSR
f. R DONE
g. RDR ENB
h. X CLK
i. XRDY
j. RCVR INT ENB

10. In order to jumper program DL-11 with the XBUF address 777346, jumpers _____ must be removed and jumpers _____ must be installed.

11. If both the receive and transmit halves of a DL-11 are requesting an interrupts, which will be serviced first?

PDP-11/70
Hardware Introduction
Study Assignment 4
Answer Sheet

Study Questions:

1. In a standard Control and Status Register, bit 6 is used to ENABLE INTERRUPTS and bit 7 indicates DONE OR READY.

2. All peripheral devices have in their interfaces a _____ DATA BUFFER _____ register to send or receive data, and a CONTROL AND STATUS _____ register to determine when the device is ready to send or receive it.

3. Assume that data is to be input to the system from the LA30 via interrupts. The interrupts service routine begins at Location 1000. Show the correct vector addresses and their contents.

Vector Addresses	Contents
60	1000
62	200

Match the statements in Column A with the items in Column B.

"A"		"B"
D	4. Holds the character to be transferred to the external device.	a. RCSR b. RBUF c. UART d. XBUF e. XCSR f. R DONE g. RDR ENB h. X CLK i. XRDY j. RCVR INT ENB
A	5. Includes the interrupts enable bit that can be used to initiate interrupt sequences when RGVR DONE sets.	
G	6. This bit, when set advances the paper-tape reader.	
I	7. This signal comes up to indicate that a character has been transmitted.	
C	8. This logic is a full duplex receiver/ transmitter.	
F	9. This signal indicates that an incoming character has been assembled in RBUF.	

PDP-11/70
Hardware Introduction
Study Assignment 4
Answer Sheet

10. In order to jumper program DL-11 with the XBUF address 777346, jumpers **A10, A9, A7, A6, and A5** must be removed and jumpers **A8, A4, and A3** must be installed.

11. If both the receive and transmit halves of a DL-11 are requesting an interrupts, which will be serviced first?

RECEIVE

DL11-W LAB

I.ADDRESS AND VECTOR SELECTION

1.NOTE: Before you begin this lab make sure that you note the position of the switches on the 5 switch pacs so they can be returned to the normal configuration when you complete the lab.

2.Using the illustration on page 2-3 and 2-4 of the technical manual , change the address of the terminal to 776510 and Mode 2.

3.Use one of the I/O programs you have already written to test whether the address selection was done correctly. If it was then you should be able to communicate with the console terminal.Without modifying your program try to communicate with the console now that the switches have been changed. YOUR PROGRAM SHOULD TRAP THROUGH VECTOR 4 DUE TO A UNIBUS TIMEOUT. This can be checked by setting up TRAP CATCHERS and examining the CPU ERROR REGISTER.(17 777 766)

Now change the lines in the program which reference the new DL11 addresses and rerun the program.It should work with no errors.

4.OPTIONAL You may change the BAUD RATE,PARITY,or any other parameters that you wish to test.

NOTE !!! RETURN ALL SWITCHES TO THE ORIGINAL CONFIGURATION. Test the terminal by running the ECHO TEST program,.

II.RUN XXDP DIAGNOSTICS

Use the following procedure to run diagnostics on the DL11

Insert the floppy into drive 0

1. LA 173004- This is the starting address of the BOOT program in ROM.

2. Lift HALT KEY and HIT START -This boots the XXDP + diagnostics from the floppy.

3. R ZLAC?? - runs the DL11 Diagnostics

The DL11 diagnostics are divided into two parts the Interface section and the terminal section. The Interface section will produce no printout unless an error occurs. If you wish to see a printout ask your instructor to insert a bug that will cause an error. The terminal section test the mechanical section of the terminal and will produce a printout for each test that it runs.

9312 BOOTSTRAP TERMINATOR LAB

I.BOOTING THE DEVICE USING LOAD AND START SEQUENCES

A.BOOTING FROM DRIVE 0
Use the table on page 24 of the PRINT SET to do the following

1.Boot the RXO1 from ROM 1	-no diagnostics	173004
2. " " "	-diagnostics enabled	173006

B.BOOTING FROM A DRIVE USING THE SWITCH REGISTER CODE

1.Boot the RX01 Drive 0 from ROM 1	165744 Switches = 0012
2.Boot the RX01 Drive 1 from ROM 1	165744 switches = 1012

C.XXDP PROMPT
Whenever you boot the system should respond with

```
CHMDXDO XXDP+ DX MONITOR
BOOTED VIA UNIT 0
28K UNIBUS SYSTEM

ENTER DATE (DD-MMM-YY):

RESTART ADDRESS: 152010
THIS IS XXDP+. TYPE "H" OR "H/L" FOR HELP.
```

Print ASCII Characters all the time

The function of this program 4 is to print a continuous lines of ASCII characters.

```
1000/ CLR R0            ! Clear the contents of general purpose register 0

1002/ TSTB @#177777564! Test the most snificate bit of the low byte
1004/ 17777564          ! transmit control status register to see if
                        ! bit 7 = 1

1006/ BPL 1002          ! Branch backward to address 1002 if bit 7 of
                        ! the control and status register address
                        ! 17777564 is = 0 the negative bit of the
                        ! processor status register address (17777776)
                        ! "PSW" bit number three (3) = 0

1010/ MOVB R0,@#        ! Move the contents of general purpose register 0
17777566                ! to the transmit data buffer of your terminal address
1012/ 17777566          ! 17777566 to be printed at your terminal

1014/ INCB R0           ! Increment the byte value in general purpose register
                        ! 0 by the value of a decimal 1

1016/ JMP @#1002        ! Jump to memory address location 1002 all the time
1020/ 1002
```

The way to deposit the above program into memory

```
1000 Load address switch
     005000   Deposit switch/Address lights will update by two(2)
     105737   Deposit switch/Address lights will update by two(2)
     17777564 Deposit switch/Address lights will update by two(2)
     100375   Deposit switch/Address lights will update by two(2)
     110037   Deposit switch/Address lights will update by two(2)
     17777566 Deposit switch/Address lights will update by two(2)
     105200   Deposit switch/Address lights will update by two(2)
     000137   Deposit switch/Address lights will update by two(2)
     1002     Deposit switch/Address lights will update by two(2)
```

Echo the character

This program 5 will echo back any character you type on the console keypad.

```
2000/ TSTB @#17777560   ! Test the most snificate bit of the low byte
2002/ 17777560          ! of the recieve control and status register to
                        ! see bit seven (7) = 1

2004/ BPL 2000          ! Branch backward to address 2000 if bit 7 of the
                        ! the recive control and status register address
                        ! 17777560 is = 0 the negative bit of the processor
                        ! status register (17777776) "PSW" bit
                        ! number throo (3) = 0

2006/ TSTB @#17777564   ! Test the most snificate bit of the low byte
2010/ 17777564          ! of the transmit control and status register to
                        ! see bit seven (7) = 1

2012/ BPL 2006          ! Banch backward to address 2006 if bit seven (7)
                        ! of the transmit control and status register address
                        ! 17777564 is = 0 the negative bit of the processor
                        ! status register (17777776) "PSW" bit number
                        ! three (3) = 0

2014/ MOV@#17777562,    ! Move the Character that is in the recieve data
      @#17777566        ! buffer to the transmit data buffer to be
2016/ 17777562          ! printed out at the page printer.
2020/ 17777566

2022/ BR 2000           ! Branch backward to address 2000 all the time
```

The way to deposit the above program into memory

```
2000 Load address switch
     105737   Deposit switch/Address lights will update by two(2)
     17777560 Deposit switch/Address lights will update by two(2)
     100375   Deposit switch/Address lights will update by two(2)
     105737   Deposit switch/Address lights will update by two(2)
     17777564 Deposit switch/Address lights will update by two(2)
     100375   Deposit switch/Address lights will update by two(2)
     013737   Deposit switch/Address lights will update by two(2)
     17777562 Deposit switch/Address lights will update by two(2)
     17777566 Deposit switch/Address lights will update by two(2)
     000766   Deposit switch/Address lights will update by two(2)
```

Message Program

This program 6 will store all the characters that you type until you hit the $ key on the keypad.

```
3000/ CLR R0           ! Clear the contents of general purpose register 0

3002/ CLR R1           ! Clear the contents of general purpose register 1

3004/ MOV#4000,R1      ! Move the octal number 4000 into general purpose
3006/ 4000             ! register 1 use to point to memory


3010/ TSTB @#17777560  ! Test the most snificate bit of the low byte
3012/ 17777560         ! of the Recieve control and status register
                       ! address (17777560) to see if bit seven (7) = 1

3014/ BPL 3010         ! Branch backward to address 3010 if bit seven (7)
                       ! of the control and status register address
                       ! 17777560 is = 1 the negative bit of the processor
                       ! status register address (17777776) "PSW" bit
                       ! number three (3) = 0

3016/ CMP @#17777562,  ! Compare to see if the character in the receive
      #244             ! data buffer address (17777562) is = $
3020/ 17777562         ! The address of the receive data buffer
3022/ 244              ! The ASCII character for a $

3024/ BEQ 3036         ! Branch foreward to address 3036 if the character
                       ! in the data compare is equal to the $ and will
                       ! branch if the Zero bit of Processor status register
                       ! address (17777776) "PSW" bit number two (2) = 1

3026/ MOV @#17777562,  ! Move the character that is in the receive data buffer
      (R1)+            ! address (17777562) to the address in general purpose
3030/ 17777562         ! register 1 and then increment to address in general
                       ! purpose register 1 by two (2)

3032/ INC R0           ! Increment the contents of general purpose register
                       ! 0 by the decimal 1

3034/ BR 3010          ! Branch backward to address 3010 all the time

3036/ MOV #4000,R1     ! Move the octal value of 4000 to general purpose
3040/ 4000             ! register 1 this is to point to memory address

3042/ TSTB @#17777564  ! Test the most snificate bit of the low byte
3044/ 17777564         ! of the transmit control and status register
                       ! address (17777564) to see if bit seven(7) = 1

3046/ BPL 3042         ! Branch backward to address 3042 if bit seven (7)
                       ! of the control and status register address (17777564)
                       ! is = 1 the negative bit of the processor status register
                       ! address (17777776) "PSW" bit number three (3) = 0
```

```
3050/ MOV (R1)+,        ! Move the character that is at the address in general
      17777566          ! purpose register 1 to the receive data buffer address
3052/ 17777566          ! (17777566) to be printed at the page printer

3054/ DEC R0            ! Decrement the contents of general purpose register
                        ! 0

3056/ BNE 3042          ! Branch backward to address 3042 as long as the contents
                        ! of general purpose register 0 are not equal to zero
                        ! the zero bit of the processor status register address
                        ! (17777776) "PSW" bit number two (2) = 0

3060/ JMP @#3000        ! Jump to address 3000 all the time
3062/ 3000
```

The way to deposit the above program into memory

```
3000 Load address switch
     005000   Deposit switch/Address lights will update by two(2)
     005001   Deposit switch/Address lights will update by two(2)
     012701   Deposit switch/Address lights will update by two(2)
     4000     Deposit switch/Address lights will update by two(2)
     105737   Deposit switch/Address lights will update by two(2)
     17777560 Deposit switch/Address lights will update by two(2)
     100375   Deposit switch/Address lights will update by two(2)
     023727   Deposit switch/Address lights will update by two(2)
     17777562 Deposit switch/Address lights will update by two(2)
     244      Deposit switch/Address lights will update by two(2)
     001404   Deposit switch/Address lights will update by two(2)
     013721   Deposit switch/Address lights will update by two(2)
     17777562 Deposit switch/Address lights will update by two(2)
     005200   Deposit switch/Address lights will update by two(2)
     000765   Deposit switch/Address lights will update by two(2)
     012701   Deposit switch/Address lights will update by two(2)
     4000     Deposit switch/Address lights will update by two(2)
     105737   Deposit switch/Address lights will update by two(2)
     17777564 Deposit switch/Address lights will update by two(2)
     100375   Deposit switch/Address lights will update by two(2)
     012137   Deposit switch/Address lights will update by two(2)
     17777566 Deposit switch/Address lights will update by two(2)
     005300   Deposit switch/Address lights will update by two(2)
     001371   Deposit switch/Address lights will update by two(2)
     000137   Deposit switch/Address lights will update by two(2)
     3000     Deposit switch/Address lights will update by two(2)
```

Interrupt Service Program

This program 7 will execute an interrupt every time the keyboard is hit. The job of the service routine is to store all the characters that are hit until the $ key is hit. The CPU base program will be printing out ASCII code. The first few characters that you type should be CR,LF,CR. It would be a good idea to finsish with these characters as well.

```
1000/ MOV #0, @#17777776 ! Move a octal value of 0 to the processor
1002/ 0                  ! status register address (17777776) to make
1004/ 17777776           ! sure no bits are set

1006/ MOV #0,R5          ! Move a octal value of 0 to general purpose
1010/ 0                  ! register 5

1012/ MOV #500,R6        ! Move a octal value of 500 to general prupose
1014/ 500                ! register 6 . General purpose register 6 will
                         ! be used for the Kernal Stack Pointer and 500
                         ! is the first memory location to decrement from.

1016/ MOV #4000,R1       ! Move the octal value of 4000 to general purpose
1020/ 4000               ! register 1 this will be the memory location to
                         ! put the ASCII characters when we type at the
                         ! keyboard.

1022/ MOV #5000,@#60     ! Move the octal value of 5000 to memory location
1024/ 5000               ! 60 which the vector address of the local console
1026/ 60                 ! terminal the 5000 will be the program couter value
                         ! of the service routine for this program.

1030/ MOV #200,@#62      ! Move the octal value of 200 to memory location
1032/ 200                ! 62 which is the processor status word location
1034/ 62                 ! "PSW" for the local console terminal service
                         ! routine 200 octal will set bit seven (7) of
                         ! of processor status register address (17777776)
                         ! which are the priority bits of the register.

1036/ MOV #100,          ! Move the octal value of 100 to the Receive
      @#17777560         ! control and status register address (17777560)
1040/ 100                ! which will turn on bit six(6) which is the software
1042/ 17777560           ! interrupt enable bit of the register so that we will
                         ! vector to memory address 60 which is the local
                         ! console terminal vector address

1044/ MOV #0,R4          ! Move the octal value of 0 to general purpose
1046/ 0                  ! register 4 will be a ASCII buffer

1050/ TSTB @#17777564    ! Test the most snificate bit of the low byte
1052/ 17777564           ! of the transmit control and status register
                         ! address (17777564) to 800 if bit seven(7) = 1
```

```
1054/ BPL 1050          ! Branch backward to address 1050 if bit seven(7)
                        ! of the control and status register address
                        ! (17777564) is = 1 the negative bit of the processor
                        ! status register address (17777776) "PSW" bit
                        ! number three (3) = 0

1056/ MOVB R4,          ! Move the low byte contents of general purpose
      @#17777566        ! register 4 a ASCII character to the transmit
1060/ 17777566          ! data buffer to be printed on the page printer

1062/ INC R4            ! Increment the contents of general purpose
                        ! register 4 by a decimal 1

1064/ CMPB R4,#200      ! Compare byte the contents of general purpose
1066/ 200               ! register 4 to the NULL ASCII character

1070/ BNE 1050          ! Branch backward to address 1050 as long as
                        ! the compare byte in general purpose register 4
                        ! are not equal to zero the zero bit of the
                        ! processor status register address (17777776)
                        ! "PSW" bit number two(2) = 0

1072/ BR 1044           ! Branch backward to address 1044 all the time
```

```
5000/ TSTB @#17777560     ! Test the most snificte bit of the low byte
5002/ 17777560            ! of the receiver control and status register
                          ! address (17777560) to see if bit seven (7)
                          ! = 1

5004/ BPL 5000            ! Branch backward to address 5000 if bit seven (7)
                          ! of the control and status register address
                          ! (17777560) is = 1 the negative bit of the
                          ! processor status register address (17777776)
                          ! "PSW" bit number three (3) = 0

5006/ MOVB @#17777562,    ! Move byte that is in the receive data buffer
      (R1)                ! address (17777562) into the address pointed
5010/ 17777562            ! to by general purpose register 1

5012/ CMPB (R1),#244      ! Compare byte at the address in general purpose
5014/ 244                 ! register 1 to see if it is ASCII $ (244)

5016/ BEQ 5026            ! Branch foreward to address 5026 if the character
                          ! in the data compare is equal to the $ ASCII 244
                          ! and will branch if the Zero bit of processor status
                          ! register address (17777776) "PSW" bit number
                          ! two (2) = 1

5020/ INC R1              ! Increment the value in general purpose register
                          ! 1 by the decimal 1 used to update to the next byte
                          ! address that is contained in general purpose
                          ! register 1

5022/ INC R5              ! Increment the value in general purpose register
                          ! 5 by the a decimal 1

5024/ RTI                 ! Return from trap or interrupt. Return the service
                          ! routine

5026/ MOV #4000, R1       ! Move the octal value of 4000 to general purpose
5030/ 4000                ! regiser 1 memory location 4000 where the input
                          ! ASCII characters are stored

5032/ TSTB @#17777564     ! Test the most snificatre bit of the low byte
5034/ 17777564            ! of the transmit control and status register
                          ! address (17777564) to see if bit seven (7) = 1

5036/ BPL 5032            ! Branch backward to address 5032 if bit seven (7)
                          ! of the control and status register address
                          ! (17777564) is = 1 the negative bit of the
                          ! processor status register address (17777776)
                          ! "PSW" bit number three (3) = 0

5040/ MOVB (R1)+,         ! Move byte at the address contained in general
      @#17777566          ! purpose register 1 to the transmit data buffer
5042/ 17777566            ! address (17777566) and then increment the address
                          ! contained in general purpose register 1

5044/ DEC R5              ! Decrement the contents of general purpose register
                          ! 5 by a decimal 1
```

```
5046/ BNE 5032          ! Branch backward to address 5032 al long as the
                        ! contents of general purpose register 5 are not
                        ! equal to zero the zero bit of the processor status
                        ! register address (17777776) "PSW" bit number
                        ! two (2) = 0

5050/ MOV #4000,R1      ! Move the octal value of 4000 to general purpose
5052/ 4000              ! register 1 the memory location where the ASCII
                        ! characters are stored

5054/ MOV #0,R5         ! Move the octal value of 0 to general purpose
5056/ 0                 ! register 0

5060/ RTI               ! Return from trap or interrupt. Return from
                        ! subroutine
```

The way to deposit the above program into memory

```
1000 Load address switch
     012737      Deposit switch/Address lights will update by two(2)
     0           Deposit switch/Address lights will update by two(2)
     17777776    Deposit switch/Address lights will update by two(2)
     012705      Deposit switch/Address lights will update by two(2)
     0           Deposit switch/Address lights will update by two(2)
     012706      Deposit switch/Address lights will update by two(2)
     500         Deposit switch/Address lights will update by two(2)
     012701      Deposit switch/Address lights will update by two(2)
     4000        Deposit switch/Address lights will update by two(2)
     012737      Deposit switch/Address lights will update by two(2)
     5000        Deposit switch/Address lights will update by two(2)
     60          Deposit switch/Address lights will update by two(2)
     012737      Deposit switch/Address lights will update by two(2)
     200         Deposit switch/Address lights will update by two(2)
     62          Deposit switch/Address lights will update by two(2)
     012737      Deposit switch/Address lights will update by two(2)
     100         Deposit switch/Address lights will update by two(2)
     17777560    Deposit switch/Address lights will update by two(2)
     012704      Deposit switch/Address lights will update by two(2)
     0           Deposit switch/Address lights will update by two(2)
     105737      Deposit switch/Address lights will update by two(2)
     17777564    Deposit switch/Address lights will update by two(2)
     100375      Deposit switch/Address lights will update by two(2)
     110437      Deposit switch/Address lights will update by two(2)
     17777566    Deposit switch/Address lights will update by two(2)
     005204      Deposit switch/Address lights will update by two(2)
     020427      Deposit switch/Address lights will update by two(2)
     200         Deposit switch/Address lights will update by two(2)
     001365      Deposit switch/Address lights will update by two(2)
     000763      Deposit switch/Address lights will update by two(2)
```

```
5000 Load address switch
     105737      Deposit switch/Address lights will update by two(2)
     17777560    Deposit switch/Address lights will update by two(2)
     100375      Deposit switch/Address lights will update by two(2)
     113711      Deposit switch/Address lights will update by two(2)
     17777562    Deposit switch/Address lights will update by two(2)
     121127      Deposit switch/Address lights will update by two(2)
     244         Deposit switch/Address lights will update by two(2)
     001403      Deposit switch/Address lights will update by two(2)
     005205      Deposit switch/Address lights will update by two(2)
     000002      Deposit switch/Address lights will update by two(2)
     012701      Deposit switch/Address lights will update by two(2)
     4000        Deposit switch/Address lights will update by two(2)
     105737      Deposit switch/Address lights will update by two(2)
     17777564    Deposit switch/Address lights will update by two(2)
     100375      Deposit switch/Address lights will update by two(2)
     112137      Deposit switch/Address lights will update by two(2)
     17777566    Deposit switch/Address lights will update by two(2)
     005305      Deposit switch/Address lights will update by two(2)
     001371      Deposit switch/Address lights will update by two(2)
     012701      Deposit switch/Address lights will update by two(2)
     012705      Deposit switch/Address lights will update by two(2)
     000002      Deposit switch/Address lights will update by two(2)
```

PDP-11/70 Diagnostics and Module Level Repair

DAY FOUR

PDP 11/70 DIAGNOSTICS

Each diagnostic test assumes successful completion of preceding tests.

Diagnostics should be run in the following sequence, otherwise errors will not be detected or error messages will be misleading.

MD-11-DEKBH-A	Diagnostic ROM (M9301-YC)
MD-11-DEKBA-A	CPU - Part 1
MD-11-DEKBB-A	CPU - Part 2
MD-11-DEKBC-A	Cache - Part 1
MD-11-DEKBD-A	Cache - Part 2
MD-11-DEKBE-A	Memory Management
MD-11-DEKBF-A	UNIBUS Map
MD-11-DEKBG-A	Power Fail
MD-11-DEMJA-A	Main Memory
MD-11-DEQKC-A	Test T15
MD-11-DERHA-A	RH-70 Massbus Control
MD-11-DZTUA-A thru DZTUF-A	TU16
MD-11-DERSA-A thru DERSD-A	RS03/RS04
MD-11-DERPK-1 thru DERPN-A and DERPS-A thru DERPV-A	RP04

PDP-11/70 Diagnostics are stored on two disks MAINDEC 11-DZZA-A-HB
MAINDEC 11-DZZB-A-HB

MD-11-DEKBI-A is a paper tape diagnostic to check the M9301-YC.

Some for the commands available under the XXDP Resident Monitor. That we can use to help run diagnostics and use to keep our diagnostics up to date.

D <cr>	Directory of the XXDP media comes to the Local TTY.
D/F <cr>	Short directory of the XXDP media on the local TTY.
D/L <cr>	Directory of the XXDP media on the line printer.
R Copy <cr>	Start the copy program.
R Filename <cr>	Loads and starts the indicated program.
L Filename <cr>	Loads the indicated program.
S Filename <cr>	Starts the desired program that was loaded with "L" command.
S Address <cr>	Starts the desired program at specified address normally Octal 200.
C Filename <cr>	Runs desired chain table.
C Filename/QV <cr>	Runs desired chain table in quick verify.

After running Update here is a couple of useful commands.

Copy and verify commands must be used with like mediums.

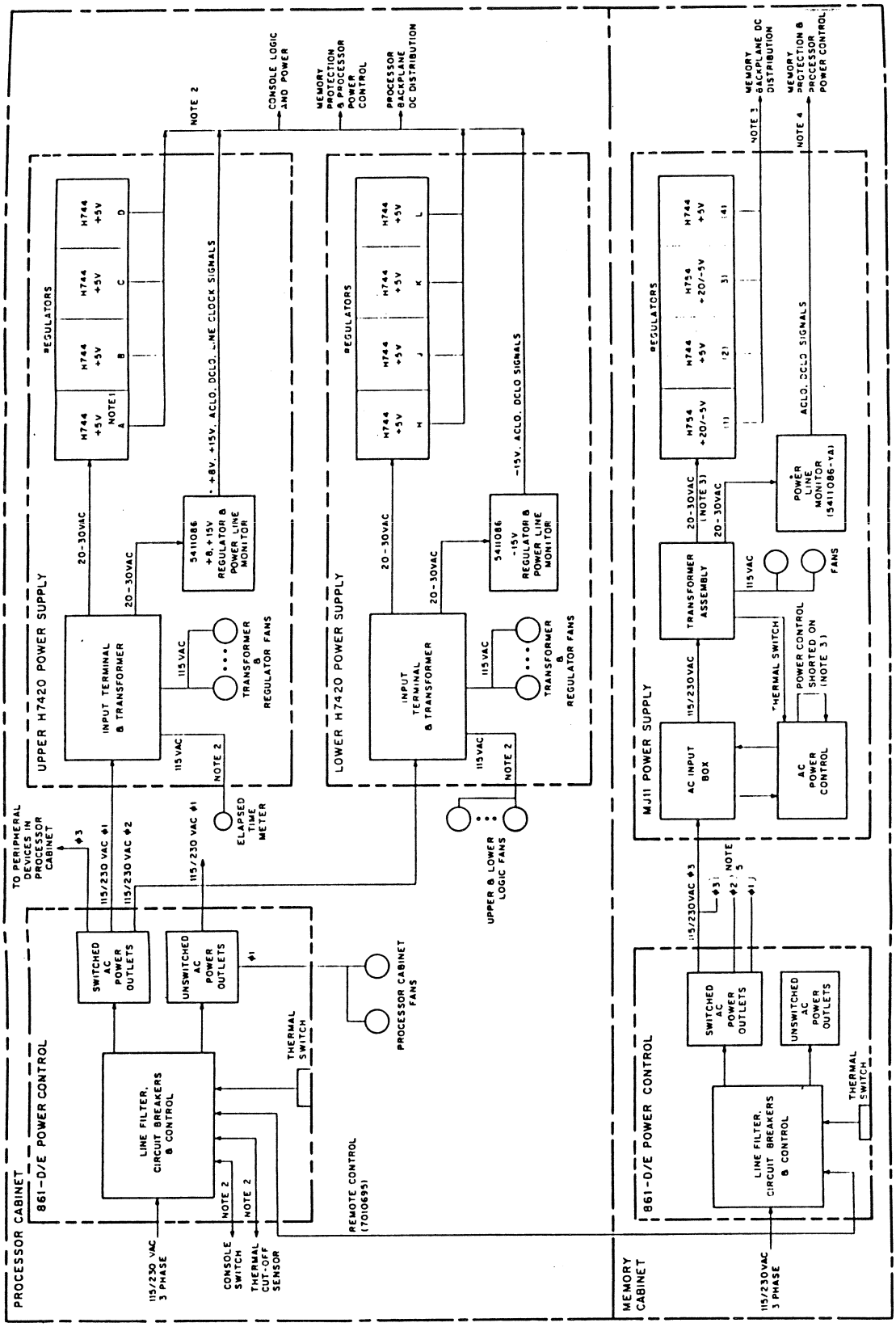
Copy DK1: = DK0:/New <cr>	Copies DECPACK Drive 0 (Master) Onto Drive 1 (Scratch and verifies.
Verify DK1: = DK0: <cr>	Verifies that DECPACK Drive 0 is identical to Drive 1.

Sometimes copy may not work and you may have to run Peripheral interchange program.

PIP Dev1:filename.Ext=Dev2:filename.Ext <cr>	Moves the file from device 1 to device 2

Some other useful commands under Update.

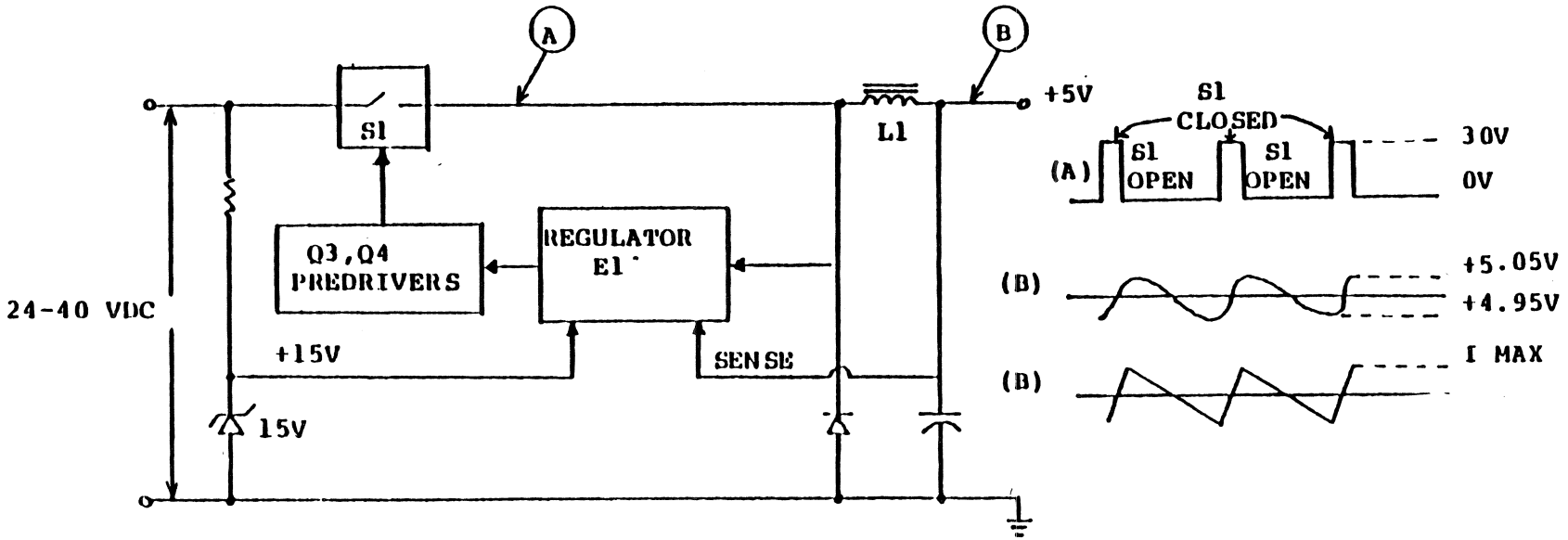
Boot <cr>	Boot up the operating system.
Dir <cr>	Directory of the Load media.
Dir/LP <cr>	Directory of the load media on a line printer.



NOTES
1. Included with floating point processor (optional)
2. Part of power distribution cable harness 7011051
3. Part of power cable harness 7010580.
4. AC/DC low wire harness 7010581 and main memory bus cable
5. To optional memory frames.

MR-4084

Figure 4-2 Typical PDP-11/70 Power System



NOTE:
a. E1 precision voltage regulator
b. S1 pass transistor Q2

SIMPLIFIED VOLTAGE REGULATOR

PDP-11/70
Hardware Introduction
Study Assignment 5

1. In reference to the 861 Power Control, the switched power is applied by energizing K1. The circuit that performs this function is the _____ board.

2. If the control console key is to control the switched power in the system the switch on the 861 Power Controller must be in the _____ position.

3. The thermal switch T1 on the 861 Power Control is normally _____ (open, closed), and _____ (opens, closes) when an overtemperature condition is detected.

4. The 861 Power Control outputs AC power to the two _____ power supplies located in the processor cabinet.

5. Located in the E7420 is the _____ 15V regulator and power line monitor. (Indicate part number)

6. The 5411086 regulator has a maximum peak to peak ripple of _____V.

7. The signal AC LO 1 can be found on power harness plug _____, pin _____.

8. The 5411086 cannot assert the signals DC LO 1 and DC LO 2. A possible cause for this failure is:

   A. Q 17 source and drain shorted together (zone B 1,2)
   B. Q 10 emitter and collector shorted together (zone B3)
   C. D 21 always conducting (zone B7)
   D. None of the above

9. If the CPU backplane slots 6-9 do not have +6V power, the H744 regulator in which regulator slot could be at fault?
   _____.

10. The PDP-11/70 processor has a basic timing cycle comprised of _____ time states.

11. It is possible to stop the CPU timing in the time states _____ and _____.

12. The basic free running clock signal which is used to produce the time state signals has a period of _____ns.

13. The TS1 L timing signal is asserted for _____ns during one CP cycle.

PDP-11/70
Hardware Introduction
Study Assignment 5
Answer Sheet

1. In reference to the 861 Power Control, the switched power is applied by energizing K1. The circuit that performs this function is the **PILOT CONTROL** board.

2. If the control console key is to control the switched power in the system the switch on the 861 Power Controller must be in the **REMOTE ON** position.

3. The thermal switch T1 on the 861 Power Control is normally **OPEN** (open, closed), and **CLOSES** (opens, closes) when an overtemperature condition is detected.

4. The 861 Power Control outputs AC power to the two **H7420** power supplies located in the processor cabinet.

5. Located in the E7420 is the **5411086** 15V regulator and power line monitor. (Indicate part number)

6. The 5411086 regulator has a maximum peak to peak ripple of **0.45** V.

7. The signal AC LO 1 can be found on power harness plug **P/J 15**, pin **8**.

8. The 5411086 cannot assert the signals DC LO 1 and DC LO 2. A possible cause for this failure is:

   A. Q 17 source and drain shorted together (zone B 1,2)
   B. Q 10 emitter and collector shorted together (zone B3)
   C. D 21 always conducting (zone B7)
   D. None of the above

9. If the CPU backplane slots 6-9 do not have +6V power, the H744 regulator in which regulator slot could be at fault?
   B.

10. The PDP-11/70 processor has a basic timing cycle comprised of **5** time states.

11. It is possible to stop the CPU timing in the time states 2 and 5.

12. The basic free running clock signal which is used to produce the time state signals has a period of 30 ns.

13. The TS1 L timing signal is asserted for 60 ns during one CP cycle.

POWER LAB

I. THE OBJECTIVE OF THIS LAB IS TO FAMILIARIZE THE STUDENT WITH THE VARIOUS COMPONENTS OF THE POWER SYSTEM OF THE 11/70.

THE STUDENT WILL ALSO USE THE MAINTENANCE MANUAL AS A GUIDE WHILE MAKING ADJUSTMENTS ON THE VARIOUS REGULATORS IN THE SYSTEM.

II. OVERVIEW

1. Using the block diagram (pg.4-2) and the illustrations(pgs.4-25,4-27) identify the major components of the power system in the 1170.Make sure that the connewctions from the power controller are correct.

FIND THE FOLLOWING

1. POWER CONTROLLER
2. UPPER/LOWER 7420 SUPPLIES
3. FIVE VOLT REGULATORS

III. CPU CABINET

The following adjustments will be made in the CPU cabinet.

A. VOLTAGE CHECKS

1. First check the voltages at the MATE-N-LOC connectors on the backplane. REF. 4-29,4-30,4-101

2. Next we will check the voltages at the backplane pin. REF. 4-101

B. VOLTAGE ADJUSTMEMTS

Using either of the measurement points from PART A adjust any regulator as needed. Follow the procedure described on page 4-104.

C. REGULATOR REMOVAL

Using the procedure described on page 4-107 remove and reinstall a voltage regulator from the cpu cabinet.

IV.MEMORY CABINET

A.CORE MEMORY

There are two locations in which the MJ11 regulators can be measured.

1.TEST POINTS- using the procedure on page 4-79 measure the voltages
2.MATE-N-LOC -Put cabinet into maintenane position. REF 4-82
Using page 4-34 as a reference measure the voltages at the connectors

2.CORE MEMORY ADJUSTMENTS
Adjust the regulators as needed using the procedure on page 4-82.

B.MOS MEMORY

1.Mos regulator measurement is made at the MATE-N-LOC connectors.
Put the cabinet into the maintenance position. REF. 4-82
and using page 4-70 as a guide measure the voltage outputs.

2.Adjust the regulators as needed . REF. 4-83

4.0 11/70 CPU AND MEMORY REGULATOR POWER SUPPLY VOLTAGE CHECKS AND ADJUSTMENTS

Random, intermittent system failures have often been caused by misadjusted or marginal CPU, MJ11, or MK11 D.C. voltage regulators. This section deals with the correct measurement test points, adjustment techniques, and D.C. voltage tolerances to avoid these problems. Particular attention should be given in the following areas:

1. H754 MJ11 +20/-5 VDC ADJUSTMENTS

The +20 VDC potentiometer effects voltage between +20 and -5 volts: actually 25 VDC. The -5VDC potentiometer controls the balance of +20 and -5 VDC with respect to ground, and has some effect on the +20VDC adjustment.

2. 70-14251 MK11 BATTERY BACKUP REGULATOR ADJUSTMENT

Always adjust +5 VB first and recheck after +/-12 VB adjustments. There is only one potentiometer for the +12 VB and -12 VB. Ensure that +12VB is within a tolerance of +5%/-0%, and -12VB is within +/-10% of nominal value.

3. H744/H7441 REGULATORS

The CPU H744/H7441 regulators in slots "C,H,J" are particularly critical as stated in Tech Tip 11/70-TT-2. They effect CACHE, Memory Management, UNIBUS MAp, and the Timing Generator.

4. SYSTEM ACTIVITY DURING ADJUSTMENTS

Always ensure all voltages are measured while there is a load on the system, such as DEC/X, EQKC??, or other TEST software.

5. USE DIGITAL VOLT METER

All voltage measurements must be made with a Digital Volt Meter. Scopes and VOM's are not sufficiently accurate to make voltage measurements. A scope may be used to check the various D.C. voltages for excess ripple (generally any ripple greater than 200 mv is considered excessive).

6. USE CORRECT TEST POINTS

Ensure that the correct test points are used. Do not measure voltages at the regulator Mate-N-Lock connector since there is a voltage drop in the power harness and connectors. Make measurements in Row A and Row F of the backplane and adjust the lower of the two measurements to the nominal voltage.

7. USE LOGIC GROUND

Always use Logic Ground within the box being measured. Do not use cabinet or chassis which occasionally will measure at Ø.2 - Ø.4 VDC difference with Logic Ground.

8) NOISY POTENTIOMETERS

Monitor voltage regulator response during adjustment to detect noisy (dirty contact) adjustment potentiometers or unstable regulators.

9. RDC CONSOLE INSTALLATIONS OR OPTION ADD-ONS

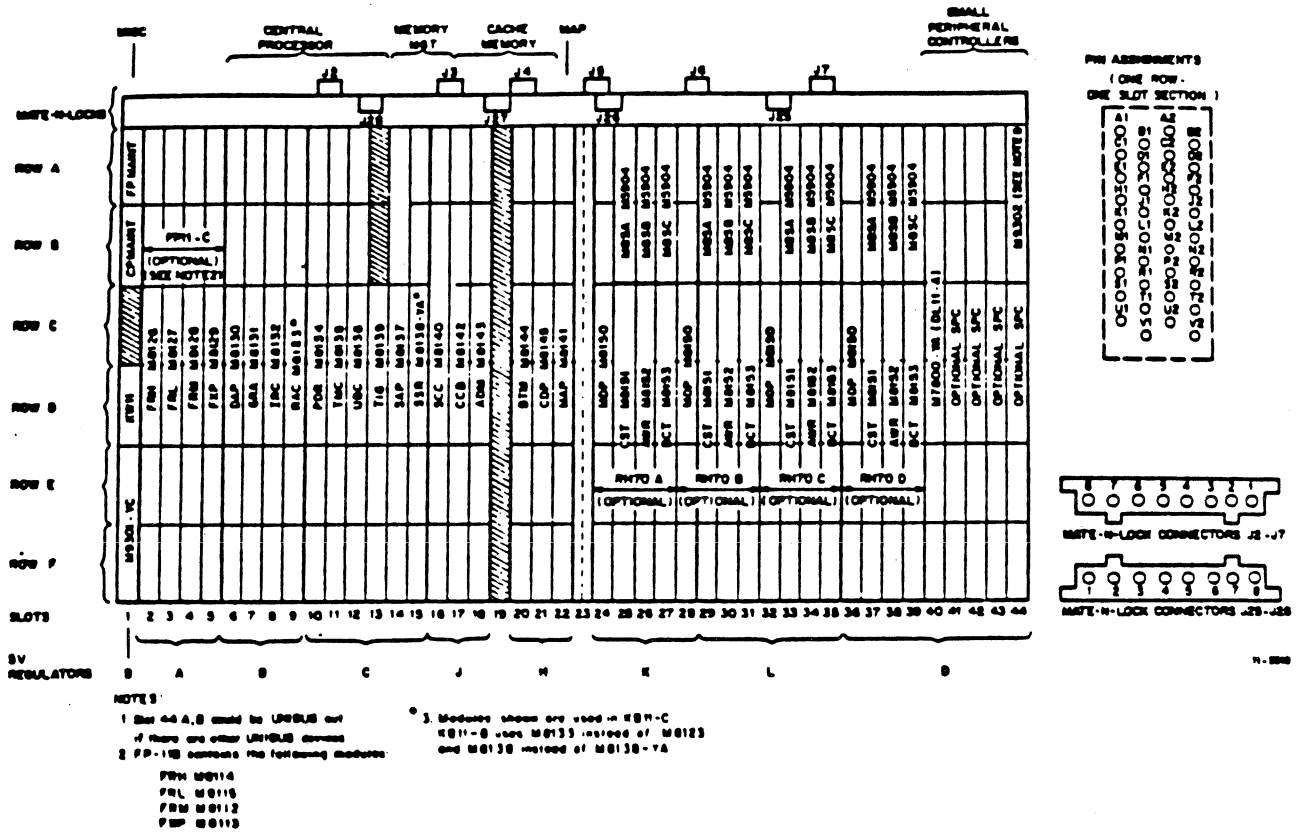
Always check CPU voltages after RDC Console installations (especially Regulator Slot J - See DDC-TT-2). After any option add-ons, check all the voltages in that box.

11/70 CPU VOLTAGES, TOLERANCES, AND TEST POINTS

OUTPUT	SLOTS	CPU BACKPLANE PIN	V.D.C.	MAX RIPPLE
H744 +5V Regulator A	2-5	F02A2	+5V ±0.1	0.2V p-p
H744 +5V Regulator B	1,6-9	F09A2	+5V ±0.1	0.2V p-p
H744 +5V Regulator C	10-15	F15A2	+5V ±0.1	0.2V p-p
H744 +5V Regulator D	36-44	F44A2	+5V ±0.1	0.2V p-p
H744 +5V Regulator H	20-22	F22A2	+5V ±0.1	0.2V p-p
H744 +5V Regulator J	16-18 Console	F18A2	+5V ±0.1	0.2V p-p
H744 +5V Regulator K	24-28	F28A2	+5V ±0.1	0.2V p-p
H744 +5V Regulator L	29-35	F35A2	+5V ±0.1	0.2V p-p
Upper H7420 5411086	1	B01B1	+8V ±1.2	0.24V p-p
Upper H7420 5411086	40-44	E13A1	+15V±1.5	0.45V p-p
Lower H7420 5411086	2,17,25, 27,29-31 33-35, 37-44	E13B2	-15V±1.5	0.45V p-p

All measurements should be made with a DVM. Ripple must be measured with an oscilloscope. A VOM is useful for making continuity and resistance checks in the power supplies.

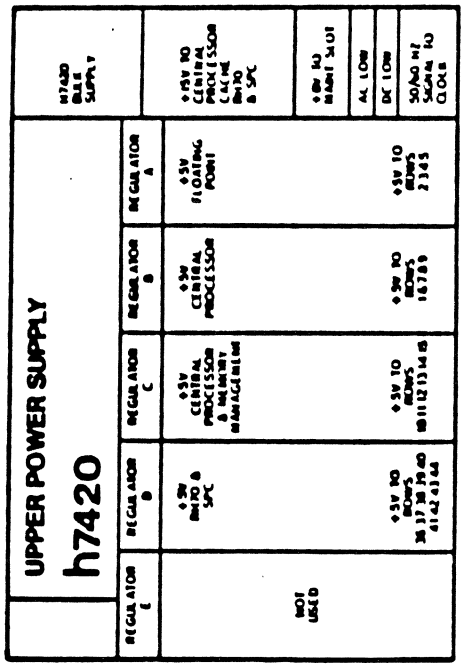
11/70 CPU VOLTAGE TEST POINTS AND REGULATOR MAP

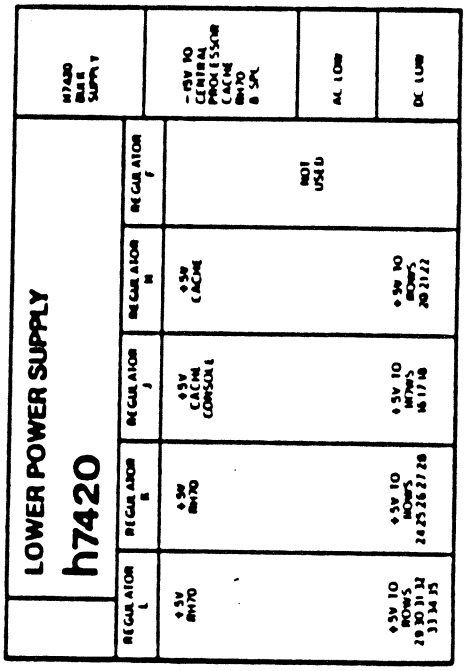


NOTES:

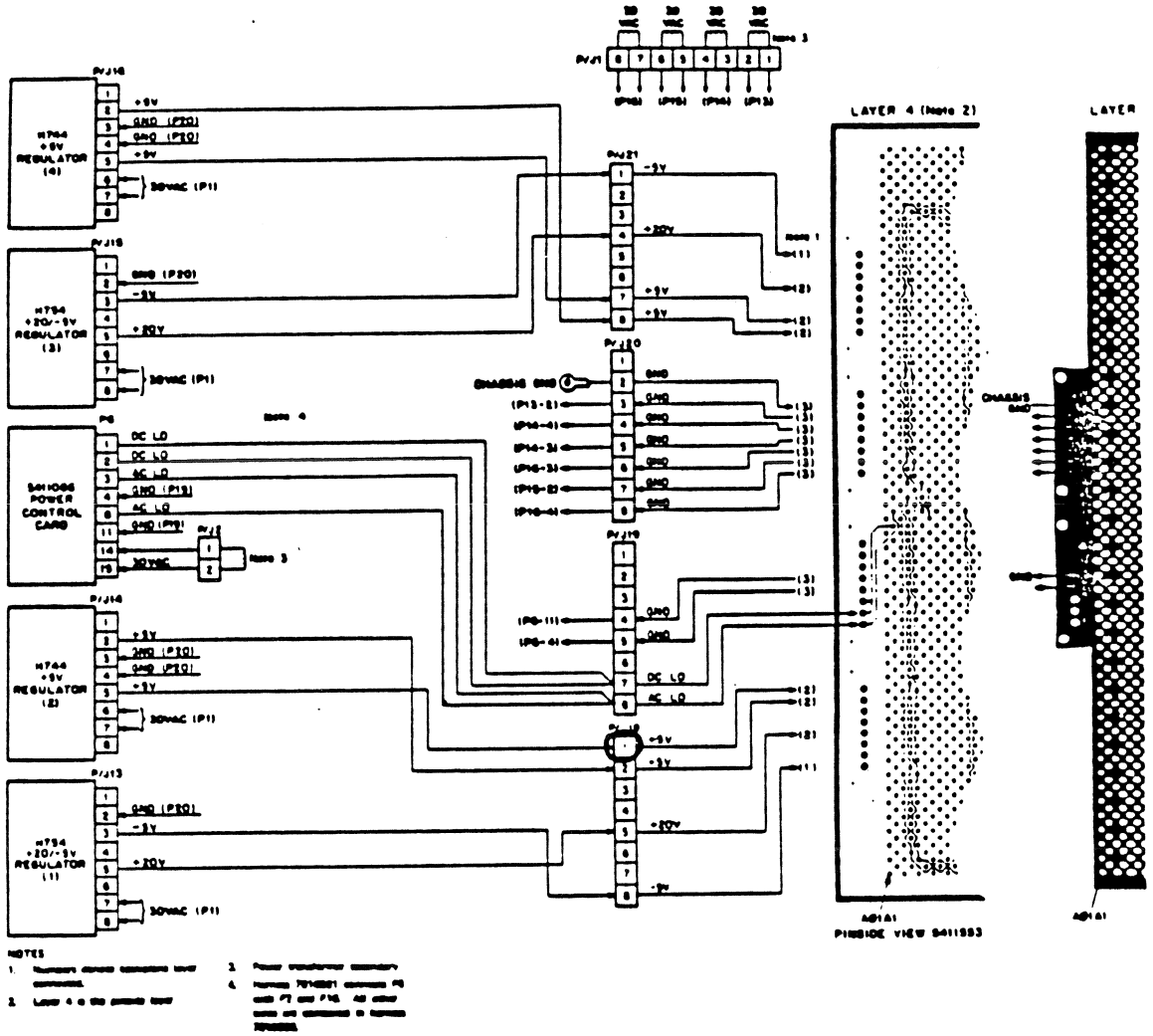
1. Slot 44 A,B should be UNIBUS out
   - if there are other UNIBUS devices
2. FP-11B contains the following modules:
   - FPH M9114
   - FPL M9115
   - FPM M9112
   - FPP M9113

* 3. Modules shown are used in KBH-C
   KBH-B uses M8133 instead of M8123
   and M8138 instead of M8138-YA



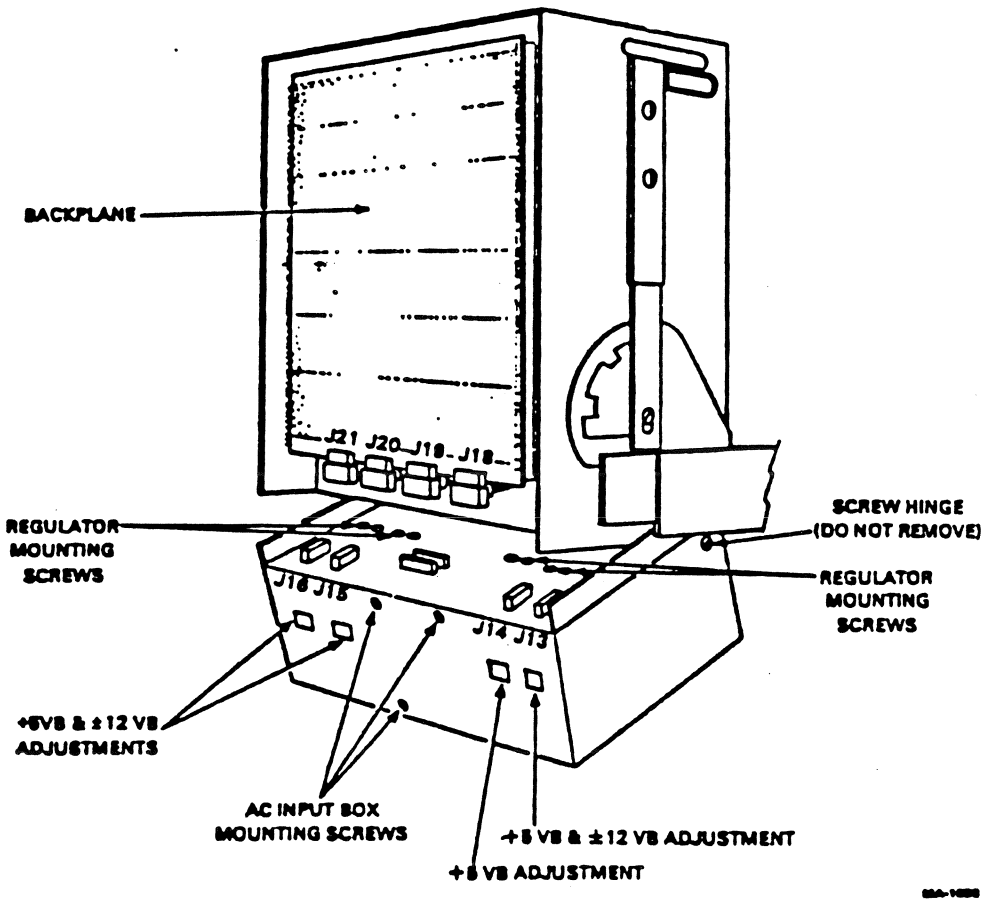


MJ11 VOLTAGE MEASUREMENTS



All voltages measured with a DVM, plus or minus 5%.

Voltage	Regulator	Pin	Backplane Connector Pins
+5V	H744 (#4)	P16-2, 5	P21-5,6,7,8
	H744 (#2)	P14-2, 5	P18-1,2,3,4
+20V	H754 (#3)	P15-5	P21-3, 4
	H754 (#1)	P13-5	P18-5, 6
-5V	H754 (#3)	P15-3	P21-1, 2
	H754 (#1)	P13-3	P18-7, 8



Power Supply in Maintenance Position

H7441 Voltage Regulator Adjustment.

1. Using a digital voltmeter, measure the voltage at J18 pin 1 or J21 pin 7 under normal load conditions.

2. Adjust the voltage to +5V ±5%. If the voltage cannot be adjusted to +5V, check for faulty regulator or harness.

7014251 Voltage Regulator Adjustment. +5VB, ±12VB

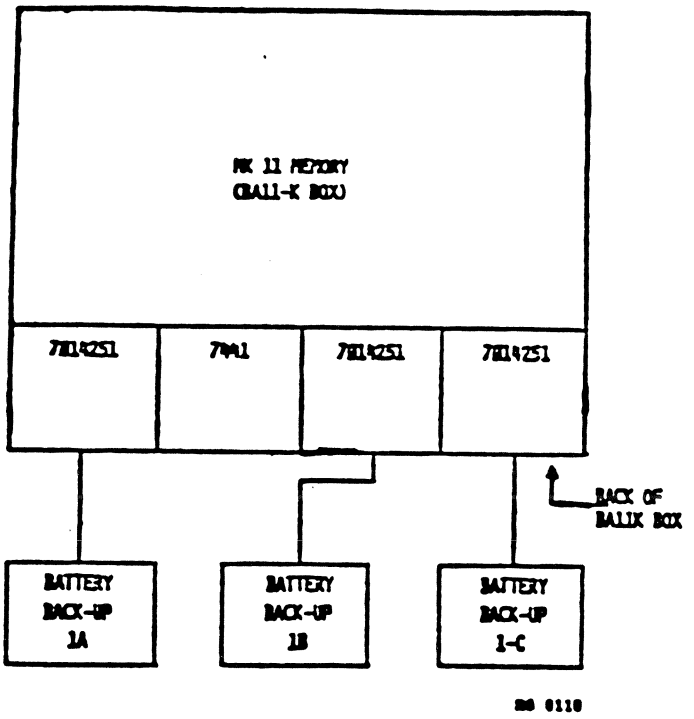
1. Using a DVM, measure the +5VB(A) at the backplane.

2. Adjust the voltage to +5V. (Test points and locations on next page.)

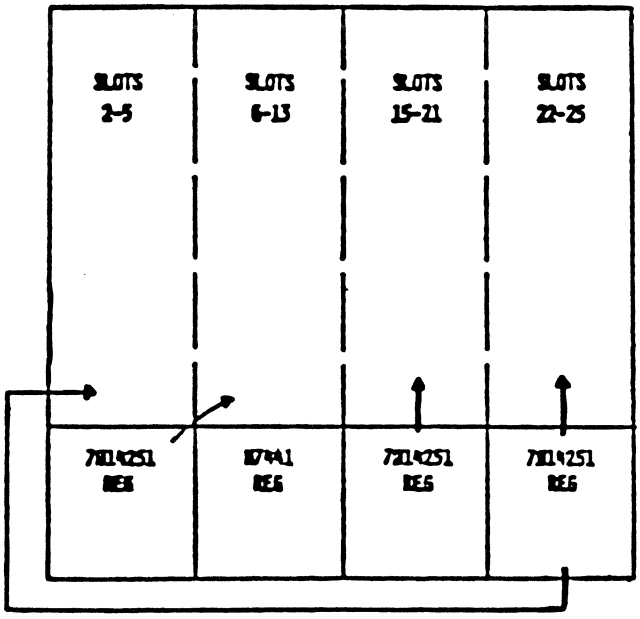
3. Measure the voltage of +12 VB(A) and -12VB(A).

4. Repeat steps 1-3 for the +5VB(B) and ±12VB(B) and the + 5VB(C) and ±12VB(C) voltages.

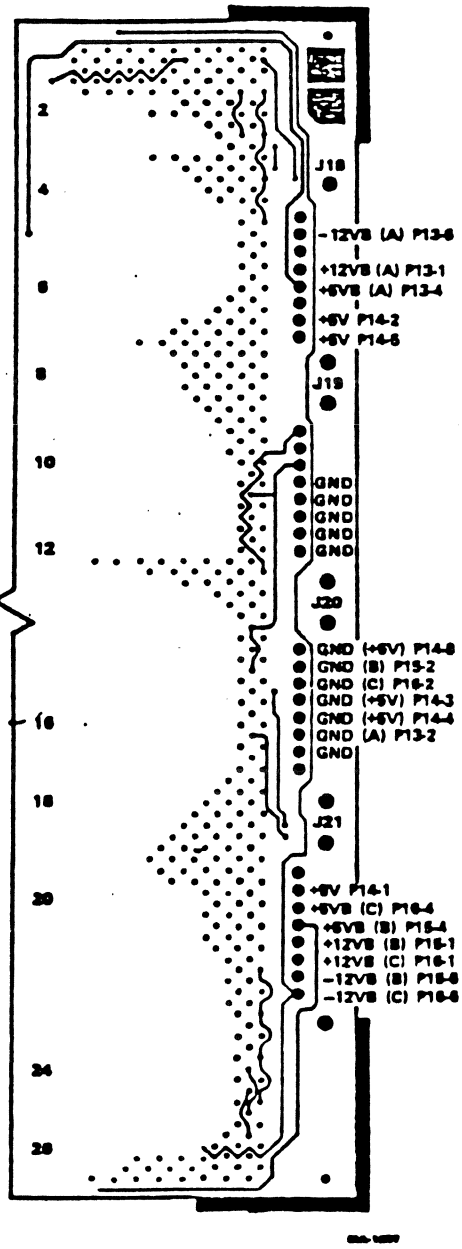
MK11 BACKPLANE AND POWER CONFIGUPATION



Battery Backup with Regulators



Regulator versus Array Slots



Backplane Power Connections

11/70 TIG Board Adjustment Procedure

digital

INTEROFFICE MEMORANDUM

TO:
WA

DATE: 28 NOV 79
FROM: John Jones
DEPT: NED Support
DTN : 8-221-5151
LOC : Burlington, MA. (BN)

SUBJ: SYMMETRY ADJUSTMENTS FOR 11/70 TIG BOARD

Do you have an 11/70 account with inexplicable failures such as traps to 0 or 10, intermittent cache errors, or system hangs with no apparent reasons.

Obviously these problems can be tracked to many causes, one of which could be a non-symmentrical clock signal which may allow the various synchronous chunks of the system to lose synchronization. This normally manifests itself as once/day or week type failure with almost no tracability.

All TIG boards are set up in a special manufacturing jig which allows not only the basic adjustments to be made, but a host of other operational checks to be perfomed as well.

Since your 11/70 has been functioning perfectly for some time and just started with an intermittent problem, it may be advisable to just check the clock signals to see if age and temperature has affected the manufacturing set-up of the module, and to assume other functionality is still good. (since most of these checks deal with the RC clock this should be valid)

There is also the case of installing a new TIG in an existing machine whose characteristics may be different enough to warrant adjustment.

This is not meant to be a check which everyone should go do immediately, just something to keep in mind.

Procedure For adjusting 11/70 TIG Boards.

1. Install M8139/09 on Extender

2. Monitor E32-8 with an oscilloscope. (be sure to use a ground lead on probe.)

3. Adjust L1 tank circuit in XTAL oscillator for oscillation by turning tuning slug all the way in (don't over-tweak, it is a ferrite core and will break) then turning counter clockwise until oscillation just starts then continue one and one-half turns more.

   (note: refer to figure 1 for the following symmetry adjustments. Oscillator Symmetry.

4. While monitoring E32-8 adjust R130 (R211 if M8109) for a symmetrical 15ns for each half cycle.

5. Phase Splitter Symmetry
   a. Monitor signal TIGC TF L (backpanel pin ER2 or emitter lead of Q57 or Q58.) Adjust R129 (R210 for M8109) for a symmetrical 15 for each half cycle.

   B. Monitor signal TIG C TF H (backpanel pin DM2 or emitter lead of Q51 or Q52) Adjust R128 (R209 for M8109) for a symmetical 15 ns for each half cycle.

   C. Recheck steps A & B, as they may interact, and readjust if necessary.

6. Reinstall TIG Board in machine and monitor backpanel pins in 5A and 5B to insure extender board did not cause a gross loading/mismatch problem.

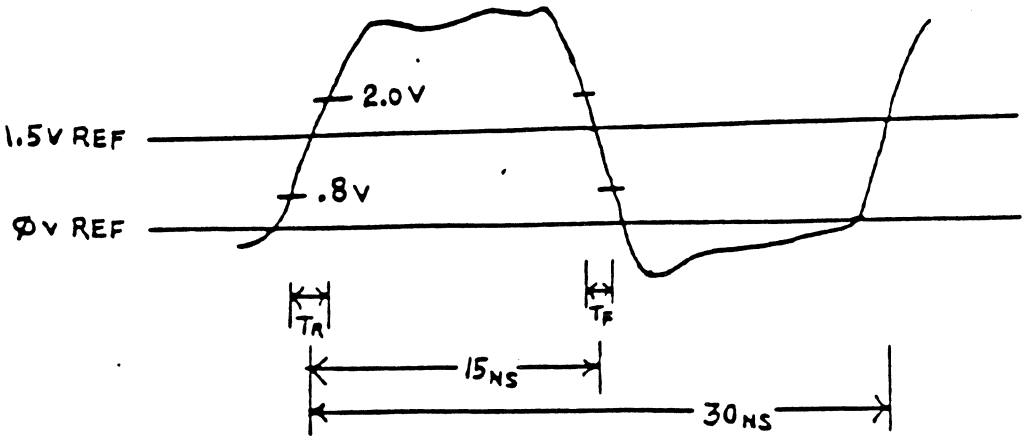
FIGURE 1 REPRESENTATIVE WAVE FORM

Frequency = 33.333 MHZ (period = 30 NS ± 0)

Rise time 2 NS
Fall time 2 NS

Rise and fall times are measured between .8V & 2V

Symmetry measurements are easiest made when ground reference is set 1.5 V below center line and centerline is used as reference during adjustment.



11/70 CPU Timing Margins Procedure

Perform the timing margin procedure as follows:

a.) Connect a jumper from F13J1 to GND (E21T1), or install the PDP-11/70 CPU maintenance card and switch to RC clock.

b.) Set up oscilloscope as follows:

1 V/cm, 0.1uSec time base
Sync: Channel 1, Internal

c.) Halt the processor, and, while monitoring pin D13T2 with Channel 1, adjust R162 on the M8139 module (Slot 13) so that a slow margin of 250 nS from leading edge to leading edge is set.

d.) Run one pass of the PDP-11/70 Instruction Exerciser (MAINDEC-11-DEQKC).

1.) Start at address 200; select applicable devices.
2.) Run one pass.
3.) If errors occur, correct the malfunction and rerun the test.

e.) Halt the processor, and, while monitoring pin D13T2 with Channel 1, adjust potentiometer R162 on the M8139 module (Slot 13) so that a fast margin of 140 nS from leading edge to leading edge is set.

f.) Repeat step d.) from above.

g.) Halt the processor, and, while monitoring pin D13T2 with Channel 1, adjust R162 on the M8139 module (Slot 13) for normal timing of 150 nS from leading edge to leading edge.

h.) Disconnect the jumper from F13J1 and E21T1 (GND).

Procede with normal acceptance testing.

II. RUNNING XXDP DIAGNOSTICS

1.To Run Individual Diagnostics

R NAME??

This type of command will run the individual diagnostic over and over untill the halt key is hit.

EXAMPLE .R EKBA?? wil run the CPU PART 1 diagnostic untill you HALT

2.Running a Chain File

A CHAIN file may be set up under XXDP to run a series of Diagnostics in a chained sequence. The number of repetitions of the diagnostic is specified when the chain is built.

C NAME.CCC

(NOTE: 1170 is the name of the chain file on this floppy the user creates the name when he builds the chain)

EXAMPLE .C 1170.CCC

3.QUICK VERIFY

In the quick verify mode each diagnostic in the chain will be executed only once regardless of the number specified in the chain file. This is a good commandd to use to isolate a problem to a subsystem level.

C 1170/QV

PDP-11/70 Diagnostics and Module Level Repair

DAY FIVE

BLOCK DIAGRAM DESCRIPTION

This chapter introduces the KB11-B, C Central Processor Unit architecture by describing the block diagrams, which show all major logic elements and interconnections in the processor. The description of the processor is divided into two major sections: data paths and control. The data paths section includes all logic elements that operate on data that is used external to the processors. The data paths block diagram is shown on the following page. The control section, which includes all logic elements that operate on data used entirely within the processor (control information), is shown on the control section block diagram. A drawing prefix, which indicates where each element is shown in the block schematic, is included within each block on the diagrams.

DATA PATHS BLOCK DIAGRAM

The data paths block diagram includes data storage elements, data manipulation elements, and data routing elements.

The data storage elements are divided into three groups:

a. general storage registers
b. temporary storage registers
c. special purpose registers

The data manipulation logic elements include:

a. the ALU
b. shifter logic
c. constant multiplexers
d. destination register
e. shift counter

The data routing logic elements consist of:

a. ALU interface multiplexers
b. temporary storage register input multiplexers
c. external interface multiplexers

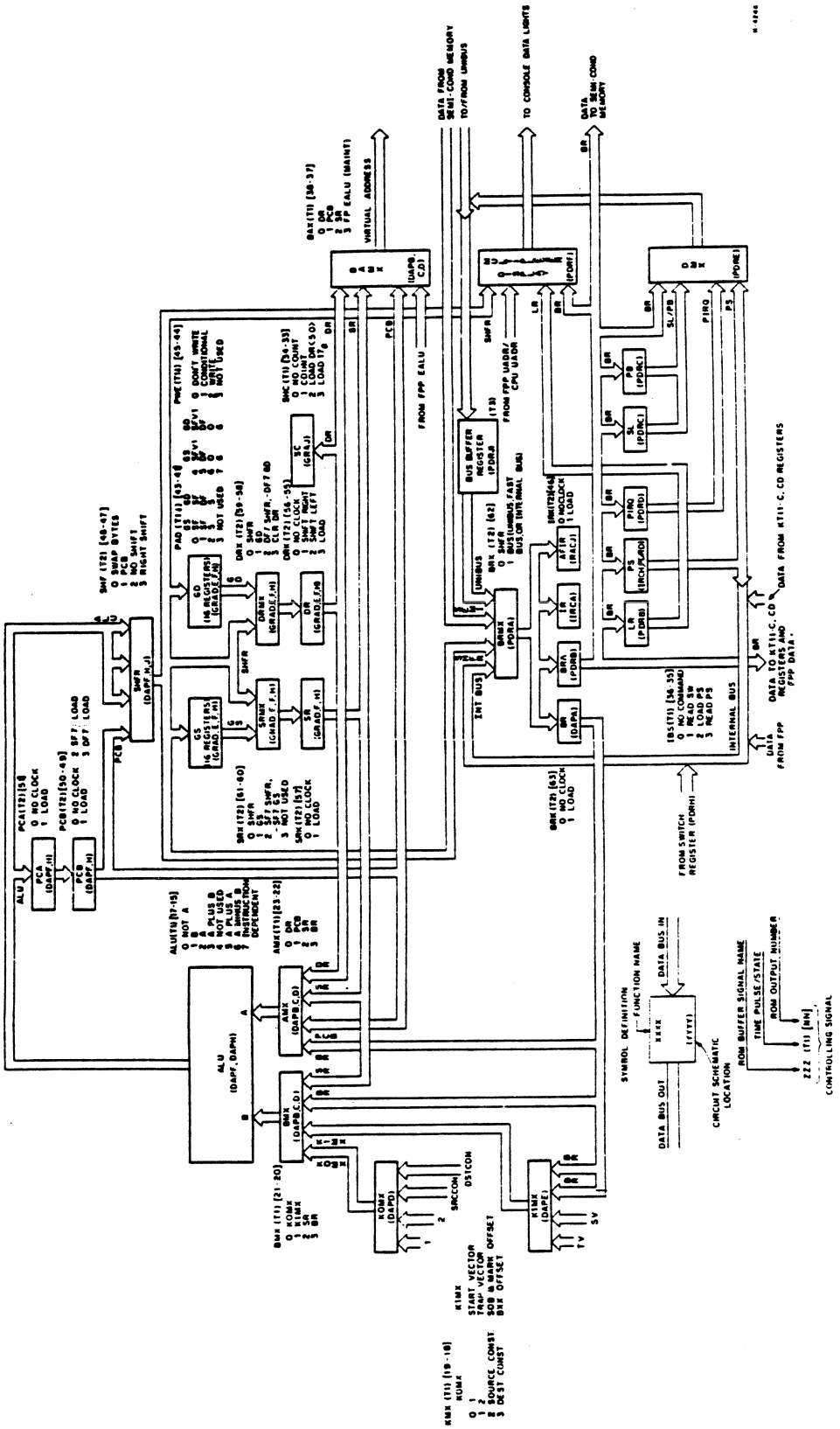


Figure 1-1 KB11-B, C Central Processor Data Paths, Block Diagram

GENERAL STORAGE REGISTERS

This group of registers includes the program counter (PC), three stack pointer registers (SP), and two sets of general registers (RØ through R5).

Program Counter (PC)

The PC provides the address of the next instruction to be fetched. For some address modes, instructions that transfer data can consist of more than one word. In these cases, the PC points to each word of the instruction in the order that the words are needed. When the PC is used as an address source, the contents of the PC are gated to the virtual address lines by the bus address multiplexer (BAMX). The PC can be updated while it is being used as an address source. To accomplish this, the PC is implemented by a buffered pair of registers, so that the PCA can be loaded with a new value, while the PCB maintains the old value; the PCB can then be loaded from the PCA when the old value is no longer needed.

The processor can transfer data to the PC from any source that can supply data to the other general registers, and can transfer data from the PC to the same destinations that can be loaded from the other general registers. Specifically, all data loaded into a general register must come through the ALU, which also supplies the inputs to the PCA. The only exception to this rule is for right shifts and byte swaps. If the processor attempts to right-shift the contents of the PC, the PCA is loaded from the ALU outputs, not the shifter (SHFR) outputs, so the data in the PC is unchanged.

During the interrupt and trap service sequences, when new PC and PS values are read from locations specified by a vector address, the old PC and PS are temporarily stored in the PCB and PCA (during internal machine cycles). This is the only time that any data, other than the contents of general register 7, is stored in the PCB. However, the PCA is often loaded in parallel with the general registers so that the PCB can be loaded if the specific register used is number 7.

Stack Pointers (SP)

The KB11-B, C has three stack pointer (SP) registers. Each SP is used as the hardware stack pointer during one of the processor operating modes. The kernel, supervisor, or user mode is selected by two bits in the processor status (PS) register. All the SP registers are also addressed as general register 6. The selection of a particular SP register is performed by the general register

control logic, depending on the current or previous processor state. The previous state, which is used during certain cycles of MTPI, MTPD, MFPI, and MFPD (move to/from previous instruction/data space) instructions, is determined from bits 13 and 12 of the PS, through logic in the general register control.

The SP registers are implemented in the two general register storage elements. These storage elements are the general source (GS) registers and the general destination (GD) registers. The two sets of storage elements contain duplicate copies of all the general registers except the PC. The use of the duplicate copies, and the specific addresses of the different SP registers within the storage elements, are described in the following paragraph.

General Register Sets

In all instructions that transfer data, each address reference specifies one of eight general registers. The specific register (of the 16 in the KB11-B, C processor) used for each reference depends both on the value of the 3-bit register specification and on the processor state, as represented by the contents of the processor status (PS) word.

Two of the eight general registers that can be specified in the instruction code are also used by the KB11-B, C as special purpose registers. If the register specification has a value of 7, it specifies the program counter (PC) register. This always refers to the hardware PC register. If the specification has the value 6, it specifies the hardware stack pointer (SP) register. One of three hardware registers, within the general register data storage elements, is selected depending on the processor mode. If the register specification has the value Ø through 5, one of two registers is selected, depending on the register set selection bit (bit 11 in the PS word). The Figure 1-2 illustrates the general selection in the KB11-B, C processor.

Each of the 16 general registers is duplicated. The duplication allows the processor to access more than one register at a time. Each general register, with the exception of register 7, is implemented by two copies in the two general register storage elements. The general source (GS) registers include 16 registers allocated as shown. The general destination (GD) registers contain 16 registers used in an identical manner. When data must be written into a general register, it is written into both copies to ensure that all attempts to read the data will read the same value. However, by specifying different register addresses to the GS and GD storage elements, it is possible to read the contents of a different register from each. This feature is used primarily in reading the contents of the two registers specified by double operand instructions.

Whenever the general registers, as a group, serve as a data source, the PC (register 7) can be selected as one of the general registers. This is accomplished by selecting the PCB input to the SHFR, and allowing the source or destination multiplexer to select the SHFR input, if register 7 is selected, and the GS or GD input if any other register is selected.

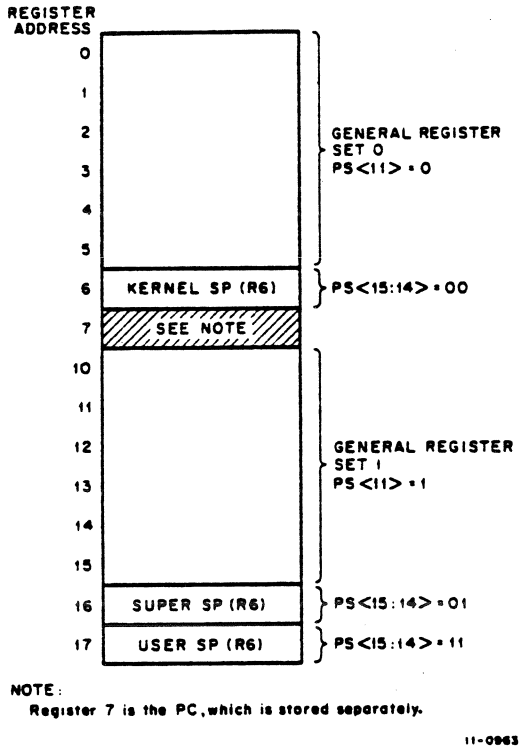


Figure 1-2 General Register Storage in GS and GD Storage Elements

TEMPORARY STORAGE REGISTERS

Temporary storage registers include the source register, the destination register, and the bus register. The source and destination registers are used primarily with the general register sets. The bus register is used primarily to communicate with external data handlers.

Source Register (SR)

The source register (SR) performs two major functions. It is the output buffer for all the general registers when addressed as the source register in an instruction, and it provides temporary storage during the source data-fetch operations.

All output from the GS registers must be transferred through the SR. When the PC is selected as a source register, the data from the PCB is routed through the SHFR to the SR. From the SR, data can be routed anywhere in the processor through the ALU inputs, or the contents of the SR can be used as an address for external data transfers through the BAMX. The SR is also used as a temporary storage register during transfers of data within the processor; e.g., when the old PC and PS are being stacked during an interrupt or trap service sequence, the SR holds the vector address.

The SR is used as a data storage element for intermediate results during instruction execution. The register and operand group instructions, such as multiply, divide, and the arithmetic shifts, use the SR to hold both operands and results.

Destination Register (DR)

In addition to performing two functions similar to the major functions of the SR, the destination register (DR) also operates as a data manipulation element; specifically, the DR is used as a shift register during register and operand instructions such as ASH, ASHC, MUL, and DIV.

All output from the GD registers (and from the PC, when it is selected as a destination register) must be through the DR. Data from the DR can be routed anywhere in the processor through the arithmetic and logic unit (ALU), or used as an address in external data transfers through the BAMX. To transfer the contents of either the SR or the DR to an external data storage location, the data must first be transferred from the SR or DR through the ALU to the BR, and then from the BR to the Fastbus or the UNIBUS.

The DR differs from the SR in its ability to act as a 16-bit, left or right shift register. This is shown by the values of the DRK microprogram field. The DR is used as a control register and to accumulate the less-significant part of the result during register and operand instructions such as multiply, divide, or the arithmetic shifts. The DR is also the source for data to be loaded into the shift counter (SC) register.

Bus Register (BR and BRA)

The bus register is the data interface between the KB11-B, C processor and all external devices. All data entering the processor data paths, and almost all data transmitted from the processor, is transferred through the BR. The BR provides many of the inputs to the ALU and is the source of data input to all the special processor control registers.

Because of the wide utilization of the BR outputs, the BR is duplicated to reduce the electrical loading on the register outputs. The second copy of the BR is called BRA. In addition, two registers (IR and AFIR), which share the same inputs as the BR (but are clocked separately), serve to hold instructions and provide inputs to the instruction decoding circuits.

Data inputs to the processor enter the processor on one of three data buses:

a. The UNIBUS, which connects the processor to a variety of UNIBUS devices, including memories, mass storage devices, and input/output peripherals.

b. The memory bus, which connects processor to main memory via cache.

c. The internal bus, which connects the processor to the FP11 Floating-Point Processor, the KT11-C, CD Memory Management Unit, and some of the special purpose registers.

Any of these buses can be selected as the input to the BR by the bus register multiplexer (BRMX). The bus selected is dependent only on the physical address used in the external data transfer.

The BR can also be loaded from the processor data paths. In data transfers from the processor to an external device, or to any of the processor control registers, the data is loaded into the BR from the SHFR after passing through the ALU. The BR is used as a temporary register in the same way as the SR or DR during the execution of instructions. In particular, the BR accumulates the more-significant half of the result during multiply and arithmetic shift instructions.

The BR can provide outputs to any of the devices on any of the three data buses. Devices on the UNIBUS use bidirectional data lines. There are separate data lines on the Fastbus for each direction of transfer. The internal bus, which is used only for transfers into the BR, is paralleled by data lines for transfers out of the BR.

SPECIAL PURPOSE REGISTERS

The data section includes a number of special purpose registers that provide control information for use by the control section, or provide communication between the console and the processor. The majority of these registers are loaded from, or in conjunction with, the bus register (BR), and can be read into the BR via the internal bus. These registers include the instruction register, the shift counter, the processor status register, the programmed interrupt request register, the stack limit register, and the microprogram break register.

Instruction Register (IR)

When an instruction is fetched from an external data storage location, the data word enters the processor through the bus register multiplexer (BRMX), and is loaded into the BR. To retain the instruction word for decoding during the execution of the instruction, while releasing the BR for other data transfers that may be required during the execution of the instruction, the outputs of the BRMX are simultaneously loaded into the instruction register (IR). The IR is clocked only during data transfers that fetch instructions. The BR is clocked during every external data transfer that brings data into the processor.

To reduce the electrical loading on the outputs of each register, the IR is duplicated. The second copy of the IR is used only by the fork A logic, which has particularly stringent timing requirements, and is therefore called the A fork instruction register (AFIR). The primary instruction register (IR) is used with decoding circuits which operate the subsidiary ROMs, the B and C forks, and a variety of instruction class selectors. All the instruction decoding logic is shown on the next figure, the control section block diagram.

Shift Counter (SC)

The shift counter (SC) is a register that performs a data manipulation function. However, the data loaded into the SC is used only for processor control information, and can not be transferred out of the SC.

The SC function is to count towards Ø. The direction of counting depends upon the current sign of the SC contents. The control data loaded into the SC is considered a repetition count, which indicates the number of cycles required to execute a complex data manipulation, such as an arithmetic shift or a multiplication. The only indication that the processor receives of the contents of the SC is an indication that the SC does, or does not, contain Ø; the counting function is completely defined once the initial count has been loaded.

Processor Status Register (PS)

The processor status (PS) register contains a number of individual bits. Some of these bits control the operation of the processor while others indicate the value of the result of the last data manipulation operation.

In addition to accepting inputs from the BR, the PS receives inputs from the condition-code generation logic. In certain circumstances (the current mode field replaces the previous mode field), some bits of the PS also receive inputs from other bits of the PS. The outputs from the PS during data transfers can be directed to the processor data paths through the BR (by selecting the PS inputs to the internal bus (IBS) and the IBS inputs to the BRMX), or directed to the UNIBUS through the PS inputs to the UNIBUS A data multiplexer (DMX). The IBS path is used only for data transfers that implicitly select the PS, such as the stacking operations during interrupt and trap service sequences. When the PS is addressed specifically, the data is transferred on the UNIBUS, even if the transfer is to the processor data paths (through the BR).

The specific bit utilization in the PS is detailed in Table 1-1.

Table 1-1

Processor Status Word Bit Assignments

Bit	Name	Utilization
15-14	Current Mode	Specifies the current processor mode as follows: a. When PS <15:14> = ØØ, the processor is in kernel mode; all operations are legal. b. When PS <15:14> = Ø1, the processor is in supervisor mode; HALT, RESET, and SPL instructions are illegal, and the SUPER address space is used. c. When PS <15:14> = 11, the processor is in user mode; HALT, RESET, and SPL instructions are illegal and the USER address space is used.
13-12	Previous Mode	Specifies the processor mode prior to the last trap, interrupt, or loading of the PS; the values are the same as for the current mode.
11	Register Set	Specifies which general register set is used; if PS11=Ø, register set Ø is selected; if PS11=1, register set 1 is used.
1Ø-Ø8	Unused	Unused
Ø7-Ø5	Priority	Set the processor priority; this priority determines which levels of programmed and external device interrupt requests are honored.
Ø4	Trace	When PSØ4=1, the processor traps the trace trap vector address; after each instruction fetch; this facility is used for debugging programs.
Ø3	N	This bit is set whenever the result of the last data manipulation is negative.

(continued)

Table 1-1 (continued)

Processor Status Word Bit Assignments

Bit	Name	Utilization
Ø2	Z	This bit is set whenever the result of the last data manipulation is Ø.
Ø1	V	This bit is set whenever the result of the last data manipulation is incorrect because of an arithmetic overflow.
ØØ	C	This bit is set whenever a carry (generally out of the most-significant bit) occurs during a data manipulation.

Programmed Interrupt Request Register (PIRQ)

The programmed interrupt request register (PIRQ) allows a program to schedule the execution of various subprograms according to a priority scheme, at the same time allowing various levels of hardware interrupt priority to interact with the software priority levels. The register stores interrupt requests set by transferring request data to the PIRQ, and provides information about the requests through encoded data transferred from the PIRQ.

Data is transferred to the PIRQ through the BR whenever the processor recognizes that the physical address is the address assigned to the PIRQ (address 777772). The transfer is entirely internal to the KB11-B, C processor. The contents of the PIRQ are then output into the priority arbitration logic of the processor, which uses the information from the PIRQ with information from the UNIBUS and the PS priority level to determine when requests should be honored.

The data in the PIRQ can be transferred to other devices or to other registers in the processor by generating the physical address of the PIRQ during an external data transfer. Because the only outputs from the PIRQ are to the DMX (UNIBUS A data multiplexer), all transfers which read the PIRQ must be UNIBUS A data transfers.

Stack Limit Register (SL)

The KB11-B, C processor performs hardware stack operations. Because the number of locations occupied by a stack is unpredictable, some form of protection against the stack expanding

into locations containing other information must be provided. The basic form of protection is the address relocation provided by the KT11-C CD Memory Management Unit; however, if the processor is operating in kernel mode with the address relocation inhibited, the processor provides for stack overflow detection through the use of the stack limit register (SL).

The SL is an 8-bit register that is loaded from the eight most-significant bits of the BR whenever the SL is selected by the physical address generated in an external data transfer. This requires the bus address 777775_8 during a byte transfer, or the address 777774_8 during a word transfer. The data is transferred directly from the BR to the SL; no bus operations are required. To read the contents of the SL, however, the SL must be selected by the DMX and the data transferred from the UNIBUS to the BR. This requires a UNIBUS data transfer operation. Although the SL and PB registers share a common DMX input, each register uses a different set of eight data lines, and only one set is selected at a time. Therefore, when the SL is transmitted on the eight most-significant data lines, all Øs are transmitted on the eight least-significant data lines.

Microprogram Break Register (PB)

The microprogram break register (PB) is intended for use as a maintenance tool. When the processor is being operated under the control of the maintenance card, the processor can be halted during any specific microprogram state by setting the address of that state in the PB and setting the switches on the card to the proper positions. During normal operation of the processor, any value can be loaded into the PB without effect on the operation of the processor. The specific procedures are detailed in Chapter 5 of the PDP-11/70 Maintenance Manual.

The PB is loaded directly from the BR whenever the PB address is generated during an external data transfer. The PB is an 8-bit register that is loaded from the eight least-significant bits of the BR. When the PB is read, the data must be transferred through the DMX to the BR by a UNIBUS A data transfer operation. The PB is selected by physical address 777770.

Console Switches (SW) and Light Register (LR)

The light register (LR) and the console switches (SW) are not, strictly speaking, data storage elements, but are included in this paragraph because they act as a data sink and a data source, respectively.

The console switches are a form of input to the processor. When an external data transfer with the physical address 777570 attempts to transfer data into the processor, the value set in the SW is transferred to the BR on the internal bus. The LR is a form of output from the processor. Any attempt to output to the same physical address transfers the contents of the BR to the LR, which can be displayed in the console lights. There are no connections between the LR and the SW, so data stored in one can not be retrieved from the other. Although both input and output to the physical address is successful, there is no correspondence between the values output and the subsequent input data.

DATA MANIPULATION

The major data manipulation elements in the KB11-B, C processor are the arithmetic and logic unit, with the accompanying constant multiplexers, and the shifter. In addition, two registers perform specific data manipulation operations.

Arithmetic and Logic Unit (ALU)

The primary data processing element in the KB11-B, C (in fact, the only element that can combine two operands to form a result) is the arithmetic and logic unit (ALU). The ALU can perform a variety of arithmetic operations on two variables, such as addition or subtraction, and can perform a variety of logical operations on one or two variables, such as complementing or ANDing. The specific operation performed at any time is selected by the processor control on the basis of the microprogram word and the current instruction. The manipulated operands are selected by two multiplexers, one for each of the ALU inputs. The operands can be the contents of the SR, the DR, the BR, the PCB, or one of a variety of numbers generated by the constant multiplexers.

The output of the ALU passes through the shifter, and can then be routed to any of the general registers, or to the SR, the DR, or the BR (and the IR, although this is not used). All of these destinations for manipulated data are internal to the processor; when data is transferred out of the processor, it must go through the BRA. Note that when the ALU outputs are routed to the program counter (PC), the signal paths do not pass through the shifter; this means that when certain shift or byte-swap operations are attempted with register 7 as the destination, the data that enters the PCA is unchanged. For example, an ASR PC instruction is executed as a TST PC instruction.

Shifter (SHFR)

In general, the data operand formed by the ALU is routed through the shifter (SHFR) to its ultimate destination. The SHFR can perform right-shift or byte-swap operations on the data, or substitute the contents of the PC for the ALU outputs. In many cases, where an instruction is performed for an odd-byte destination operand, the data manipulation required by the instruction is completed in the ALU and the transfer of the result to the odd-byte data lines is performed in the SHFR, all during one machine cycle.

In addition to its data manipulation (shifting and byte swapping) activity, the SHFR is used as a routing element. When a general register is transferred to the SR or DR, if that register is register 7 (the PC), the PCB is routed through the SHFR to the SRMX and DRMX.

Constant Multiplexers (KØMX, KlMX)

The constant multiplexers (KØMX, KlMX) are primarily routing elements, but they can perform certain limited data manipulation operations. The source and destination constants which can be selected by the KØMX are numbers generated by the processor on the basis of the instruction type. These numbers are used to add or subtract from addresses during the data fetch sequences. The offsets generated by the KlMX are formed from the contents of the BR by shifting and sign-extending the least-significant bits of the data word.

Destination Register (DR)

The destination register (DR) is primarily a temporary storage register; however, it is also used to manipulate the less-significant half of a 2-word operand by performing shifts on the operand. A word of data that is stored in the register can be shifted one bit to the left or right. The bit that is shifted into the register to fill the vacated bit position is generated by special logic in the processor, based on the data in the more-significant word being manipulated and on the instruction type.

Shift Counter (SC)

The shift counter (SC) performs incrementing and decrementing operations on data loaded into it during the execution of certain instructions. This register is primarily a processor loop counter

register; its data manipulation capability is a function of its utilization and can not be used for data operands because the SC can not be read.

DATA ROUTING ELEMENTS

When the processor performs an operation on data operands, the operation is defined by the selection of the data operands, the storing of the result, and the manipulation of the operands. While the last function is performed by the data manipulation elements, the first two functions are performed by the data routing elements.

Data routing is performed in two ways. First, the selection of inputs to storage and manipulation elements is performed by a variety of multiplexers. Second, the loading of data storage elements is controlled to select which elements are loaded at any time. Therefore, all operand selection is performed by multiplexers, and all result storage is performed by generating load signals only for the desired storage elements.

This paragraph describes the multiplexers, which are the data routing elements in the KB11-B, C processor. The loading of data storage elements was described previously. The multiplexers are organized in the following three groups:

a. ALU interface multiplexers
b. temporary register input multiplexers
c. external interface multiplexers

ALU Interface Multiplexers

The ALU has two sets of inputs and one set of outputs. Each input is connected to a number of data storage (or manipulation) elements by a multiplexer, and the output is passed through a data manipulation element that acts as a multiplexer. One of the input multiplexers can select inputs from two other multiplexers. The following table lists the inputs and outputs for each of the five multiplexing elements that control the flow of data through the ALU.

Temporary Storage Register Input Multiplexers

Each of the three temporary storage registers (SR, DR, and BR) receives inputs through a multiplexer which selects one of two or four inputs.

Table 1-2

ALU Interface Multiplexers

Multiplexer	Output To	Input From	Type of Input
AMX	A input of ALU	source register	variable operand
		destination register	variable operand
		bus register	variable operand
		program counter	variable operand
BMX	B input of ALU	source register	variable operand
		bus register	variable operand
		KØMX	constants
		K1MX	constants and sign-extended operands
KØMX	BMX	1	fixed constant
		2	fixed constant
		source constant	generated constant
		destination constant	generated constant
K1MX	BMX	trap vector	generated constant
		start vector	fixed constant
		BR (SOB & MARK)	shifted and sign-extended operand
		BR (branch)	shifted and sign-extended operand
SHFR	general registers, SR, DR, BR, Disp.	ALU	variable operand (can swap bytes or perform right shift)
		PC	variable operand

Table 1-3

Temporary Storage Register Input Multiplexers

Multiplexer	Output To	Input From
SRMX	source register	general source (GS) registers shifter (SHFR)
DRMX	destination register	general destination (GD) registers shifter (SHFR)
BRMX	bus register	shifter (SHFR) Unibus (via PDRJ Bus Buffer register, clocked at each T3) Fastbus (SEMI) internal bus (IBS)

External Interface Multiplexers

The KB11-B, C Central Processor Unit external interface is divided into three parts:

a. the explicitly addressed interface
b. the implicitly addressed interface
c. the display interface

The explicitly addressed interface is used in all data transfers where the address is specified by the processor. The address is supplied to the interface through the bus address multiplexer (BAMX), from one of three sources. These sources are the PC and the two temporary registers, SR and DR, that are used as buffers for the general registers. In addition to these inputs, the BAMX can select an input from the exponent arithmetic and logic unit (EALU) of the FP11. This input is used only to allow this data to be displayed in the console lights during specific machine states when executing floating-point instructions. Data is supplied to the interface through the data multiplexer (DMX) for UNIBUS transfers, and directly from the BR for Fastbus transfers or internal bus transfers. On data transfers into the processor, one of the three buses is selected by the BRMX.

Implicitly addressed transfers (e.g., to the FP11) do not require sending an address. The data is transmitted by sending a load signal to the appropriate device, or to a register in the processor if the transfer is into the processor. Data is transferred on the internal bus. The internal bus is, therefore, a form of data routing element; selection is accomplished by gating specific data onto the bus from a device, and by loading

only specific registers. The display interface selects the data that is to be displayed in the console lights. There are two sets of lights which display program-dependent data; the DATA lights and the ADDRESS lights. The DATA lights display one of four data words selected by the display multiplexer; the ADDRESS lights display addresses based on the outputs of the BAMX. Both displays are controlled by switches on the console.

CONTROL SECTION

The control section of the KB11-B, C processor determines the sequence of operations performed by the processor, and controls the interaction of the processor with other devices in the PDP-11 System. Control of the processor is based on the signals generated by the microprogram read-only memory (ROM), while the control of processor-system interaction is primarily by asynchronous circuits on three control modules.

The control elements shown on the control section block diagram are divided into three groups:

a. the microprogram ROM, together with the ROM address generation logic and the ROM output buffer logic

b. the external interface, which comprises the UBC, TMC, and TIG modules shown at the bottom of the drawing

c. the combinational logic circuits that interact with the microprogram outputs and with data from the data paths section of the processor to generate some of the processor control signals

Each of these three groups is described in the following paragraphs.

ROM Microprogram Control

The microprogram ROM contains 256 stored processor control words. For each processor machine cycle, one of these stored words is output to the data paths section and to the other processor control circuits. The ROM word is divided into fields, and each field controls a different (but always the same for a given field) part of the processor. In the following figure, each control field is listed by a mnemonic name and by bits of the microprogram word occupied by the control field. The control selection that is made, or the action that takes place for each value that can be stored in the field, is listed under the field name. Where possible, the field name and description are placed next to the logical element controlled by that field.

The microprogram ROM outputs that control other parts of the processor must be stored in a buffer register, so that the next microprogram word can be selected while the current word is being used. Therefore, a ROM buffer register (RBR) is provided for these outputs. The three output fields that are used to select the next microprogram word (FEN, BEF, and ADR) are not buffered because they are used immediately and the resulting address is buffered.

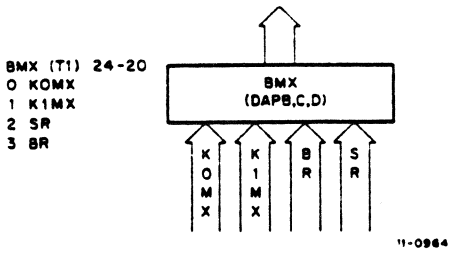


Figure 1-4 Control Field Description Example

Immediately after the beginning of a machine cycle, when a new microprogram word is available, the ROM-address generation circuits begin the calculation of the next ROM address to be selected. This corresponds to selecting the next machine state. The generated address is assembled by the address gating logic and loaded into the ROM address register (RAR). There are three copies of the RAR to accommodate the output loading required for 16 ROM elements.

The address gating logic assembles the address from five sets of inputs. The basic input, which is always present, is the address (ADR) field of the current microprogram word. The ADR is ORed with the outputs of the branch logic, which is controlled by the BEF field of the microprogram word. The branch control logic selects a set of condition inputs from signals received from the processor data paths, the condition codes, and from the processor interface modules (specifically, the TMC module). Depending on the state of the selected inputs, the branch control generates one or two signals that are used to modify the ADR.

The three other inputs to the address gating circuits are from the fork logic. The three forks are similar in implementation and purpose. Each fork uses combinational logic to decode the instruction type and a variety of processor conditions, and generates one of a large number of addresses that is combined with the ADR input by masking. Each fork can be enabled by one bit in the fork-enable (FEN) microprogram field; normally all forks are disabled. No more than one fork is ever enabled at a time.

The fork A logic, used to select the machine state to follow an instruction fetch, requires a separate instruction register (AFIR) because this fork must operate rapidly and therefore puts a heavy load on the IR outputs. The B and C forks decode inputs from the primary IR and use the outputs of a subsidiary ROM, which decodes some classes of instructions. These forks are used after a destination operand fetch and a source operand fetch, respectively.

To summarize the operation of the microprogram ROM control logic, during each machine cycle, an address is assembled from any enabled fork combined with the address field of the microprogram word and any enabled branches. This address is loaded into the ROM address register to select a new microprogram word. At the beginning of the next machine cycle, the new microprogram word is loaded into the ROM buffer register and the sequence is continued.

External Interface Control

The interaction between the KB11-B, C processor and the other parts of the PDP-11/45 or 11/70 System is controlled by three modules in the processor. These modules include the asynchronous circuits that perform timing adjustments, the circuits that generate and receive interlocking bus control signals, and the basic processor timing circuits. The functions of each module are discussed in one of the following paragraphs.

UNIBUS and Console Control (UBC) Module

The UNIBUS and console control (UBC) module includes the circuits that control transfers with external devices (and some processor special registers), and the circuits that allow the processor to be controlled by the console. The data transfer control circuits perform the necessary operations to gain control of the required data buses, select the address that is to participate in the transfer, and complete the transfer. The console control circuits provide information to the branch control circuits so that the microprogram control can be used to execute various console operations.

Traps and Miscellaneous Control (TMC) Module

This module is used to recognize a variety of asynchronous conditions and change the sequence of processor operations in response to these conditions. The TMC module detects various abnormal conditions within the processor, such as power failure, odd address on word transfers, stack overflow, or reserved instructions. When any of these conditions occur, the processor enters a trap service sequence of microprogram states, and the TMC module generates a trap vector that is used to transfer system control to a specific trap service program. The TMC module can also handle a variety of trap-type instructions, which are legal in programs that use them in a defined manner and that have set up the trap vectors for those instructions.

The TMC module also performs priority arbitration for UNIBUS A, which is controlled by the KB11-B, C processor. The priority arbitration determines which device shall be bus master, based on the priority level of the bus or non-processor request, and the priority level of the processor. The processor normally assumes the role of bus master when no other device is requesting the bus; the processor must be bus master in order to perform any data transfer on UNIBUS A. Fastbus transfers can be performed even though the processor is not UNIBUS A bus master. One of the devices that can request bus mastership, but only to perform an interrupt operation, is the processor's programmed interrupt request (PIRQ) register.

The Timing Generator (TIG) Module

The timing generator (TIG) module controls all timing of operations within the processor. All register loading, all data path transfers, and all microprogram word selection is controlled by timing signals from the TIG module which gate the control signals to the respective processor elements. The TIG module contains the processor clock, the time pulse generators that produce timing signals from the basic clock output, and a variety of control circuits that can stop and restart the clock based on asynchronous conditions detected by the UBC and TMC modules. The timing of the processor operations thus interacts with the timing of data transfers in the PDP-11/45 or 11/70 System, and with the console control operations.

SPECIAL CONTROL LOGIC

There are three special control circuits in the processor which use combinational logic to increase the flexibility of the processor control. Two of these circuits use subsidiary ROMs to define specific operations for individual instructions, and the third performs the additional decoding necessary to control the general register sets. Each of these circuits is described in one of the following paragraphs.

Arithmetic and Logic Unit (ALU) Control

The arithmetic and logic unit (ALU) used in the KB11-B, C processor can perform 16 different arithmetic operations and 16 different logical operations. Only a subset of these operations are used in the KB11-B, C The ALU control circuit transforms the ALU microprogram field (which is compressed into three bits, and can only express eight different operations) into the six control signals necessary to select the appropriate ALU operations. The ALU control circuit can also substitute control signals derived

from a subsidiary ROM (whose output is selected by the individual instruction being executed) for the signals derived from the ALU field. This allows the same microprogram word to be used for the execution machine state of a large number (32) of instructions.

The subsidiary ROM is one of two used for a group of data manipulation instructions. When these instructions are being executed, the subsidiary ROM control converts the instruction type to a 5-bit address that selects one world in each of the subsidiary ROMs. This word contains the control signals that correspond to the value required for that instruction. Through the use of the control signals in the subsidiary ROM, any ALU function can be performed.

NOTE

The SHFR operation is also affected when the output of the ALU subsidiary ROM is used.

Condition Code Control

The KB11-B, C processor condition codes are used to store information about the results of each instruction, so that this information can be used in following instructions. The conditions recorded in the condition-code bits differ for each instruction type, and often for the part of the instruction being executed. In addition, the sources of the information to be recorded in the condition codes can vary for different types of instructions.

The condition-code control circuit uses the CCL microprogram field and a subsidiary ROM to determine what data shall cause each condition-code bit to be set or cleared. For most machine cycles, only the CCL field is required to determine what function the condition-code control logic performs. When the CCL field contains a value that specifies that the operation is instruction-dependent, the outputs of the subsidiary ROM determine the exact operation.

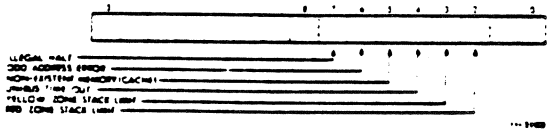
General Register Control

The KB11-B, C general registers include two register sets that are duplicated for extra speed in reading the registers. The selection of registers within each implementation is controlled by the PAD microprogram field, and all input in the registers is controlled by the PWE microprogram field.

Because the specific register to be selected can depend on the contents of the instruction register (IR), the contents of the switch register (during a console operation), or directly on the

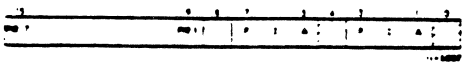
PAD field, combinational logic is used to combine all the different sources according to the requirements of the current machine state. The combinational logic also determines, for conditional write operations, whether the register that is selected is in the general register set or is register 7 (the program counter). In the latter case, no write operation is done within the general register storage area.

PROCESSOR CONTROL REGISTERS / ADDRESSES



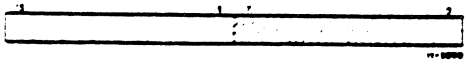
(TMCD) 17 777 766

CPU Error Register



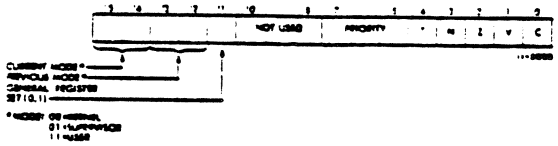
(PDRD) 17 777 772

Program Interrupt Register



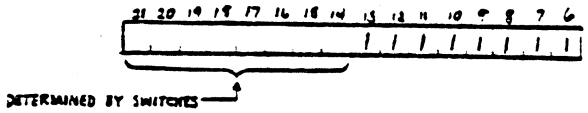
(PDRC) 17 777 774

Stack Limit Register



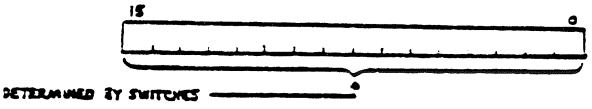
(IRCH,PDRD)
17 777 776

Processor Status Word



Lower Size Register

(SCCN) 17 777 760



System ID Register

(SCCN) 17 777 764

Figure 1-6

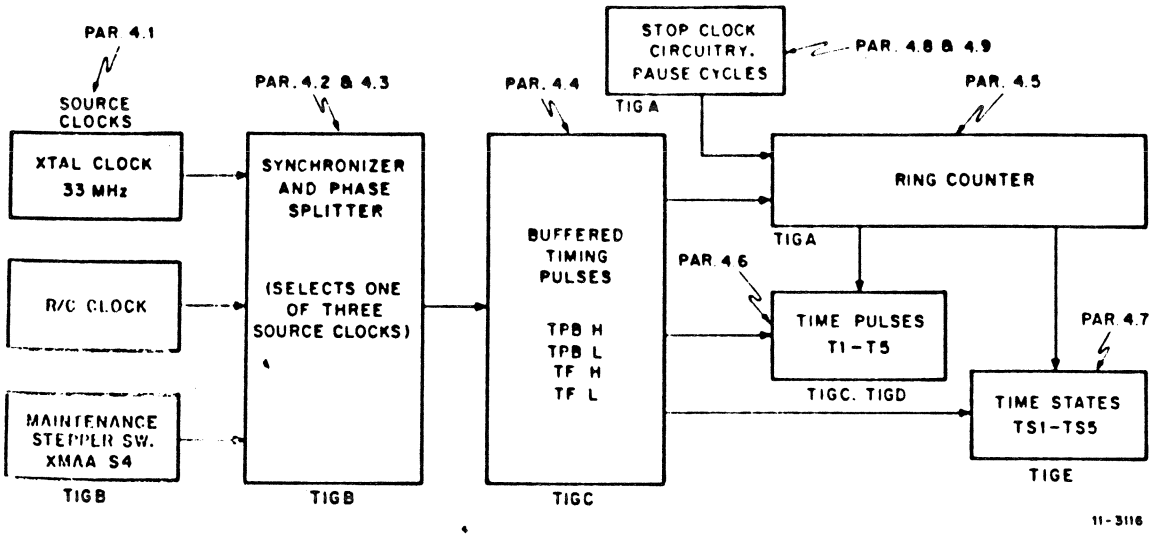


Figure 4-1 Timing Generator Block Diagram

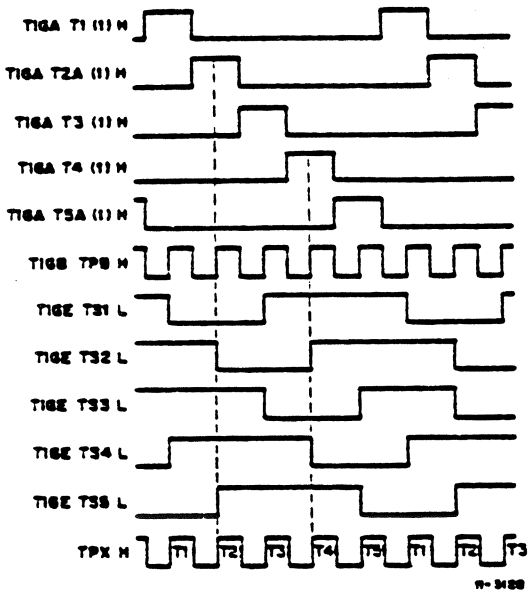


Figure 1-8 Time States

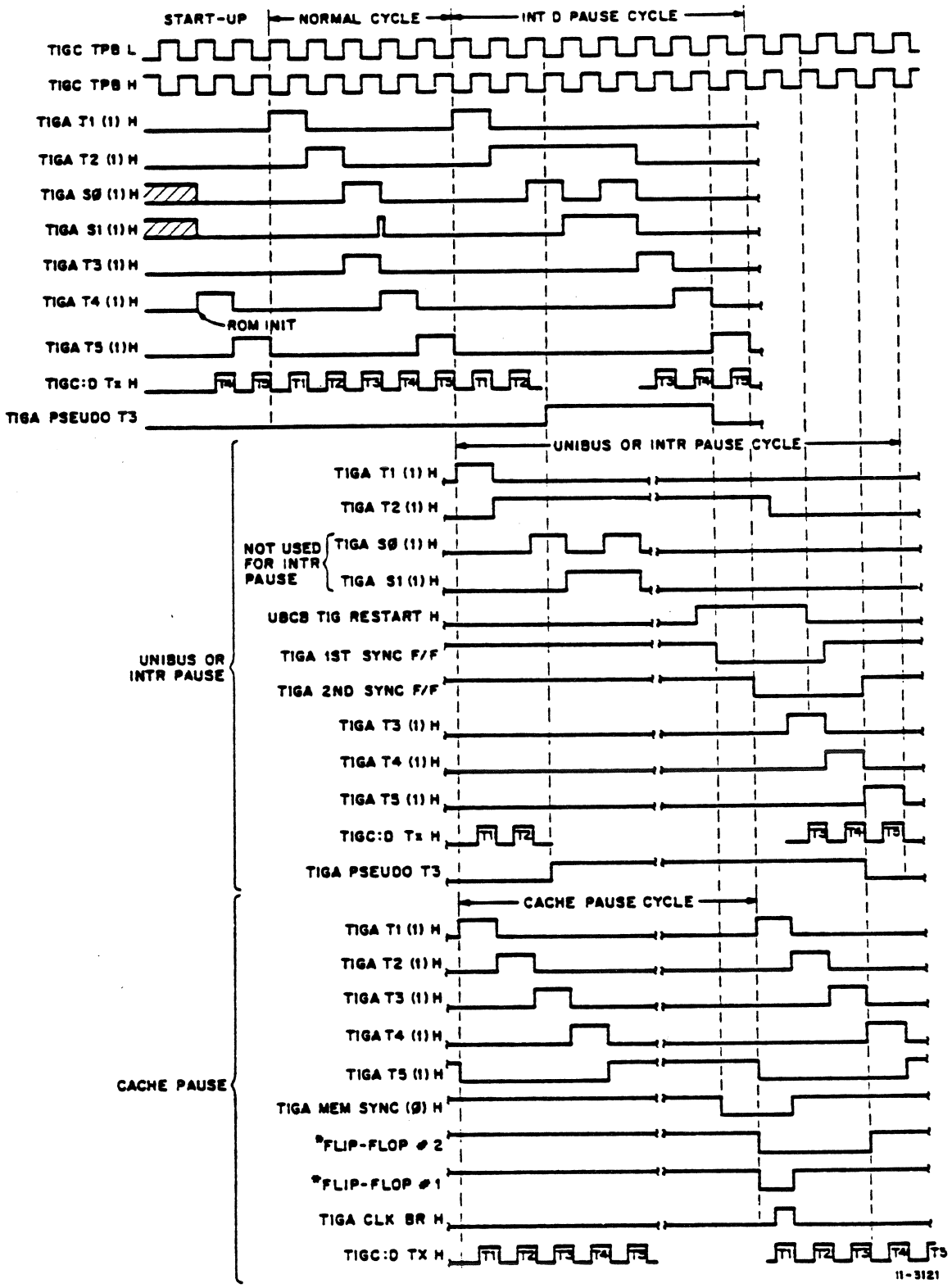


Figure 1-9 Timing Generator & Pauses

Table 1-4
Ring Counter Stop and Pause Conditions

STOP IN T2		
Internal Bus Pause	Stop:	SAPN NOT CACHE ADRS H TIGA PAUSE H (UBSD = 2 or 3) TIGA SØ (Ø) H or TIGA S1 (Ø) H
	Restart:	SØ and S1 count to 3 (9Ø) ns).
Unibus Pause CPU Control Registers	Stop:	Same as Internal Bus Pause
	Restart:	Same as Internal Bus AND UBCB TIG RESTART H (BUS SSYN)
Interrupt Pause	Stop:	UBSD =1 (INTR Pause) UBCD EXT BRQ H
	Restart:	UBCB TIG RESTART H (Passive Release or BUS INTR)
Single ROM Cycle	Stop:	TIGB ROM+UPB (1) H
	Restart:	CONTINUE or MAINTENANCE (XMAAS4) switches
STOP IN T5		
Cache Pause	Stop:	TMCF CACHE ADRS H TIGA PAUSE H (UBSD = 2 or 3) No Aborts (not TMCC ABORT H)
	Restart:	TIGA MEMSYNC (1) H
Single Bus Cycle	Stop:	TIGB SINGLE CY L TIGA PAUSE H
	Restart:	CONTINUE or MAINTENANCE (XMAA S4) switches

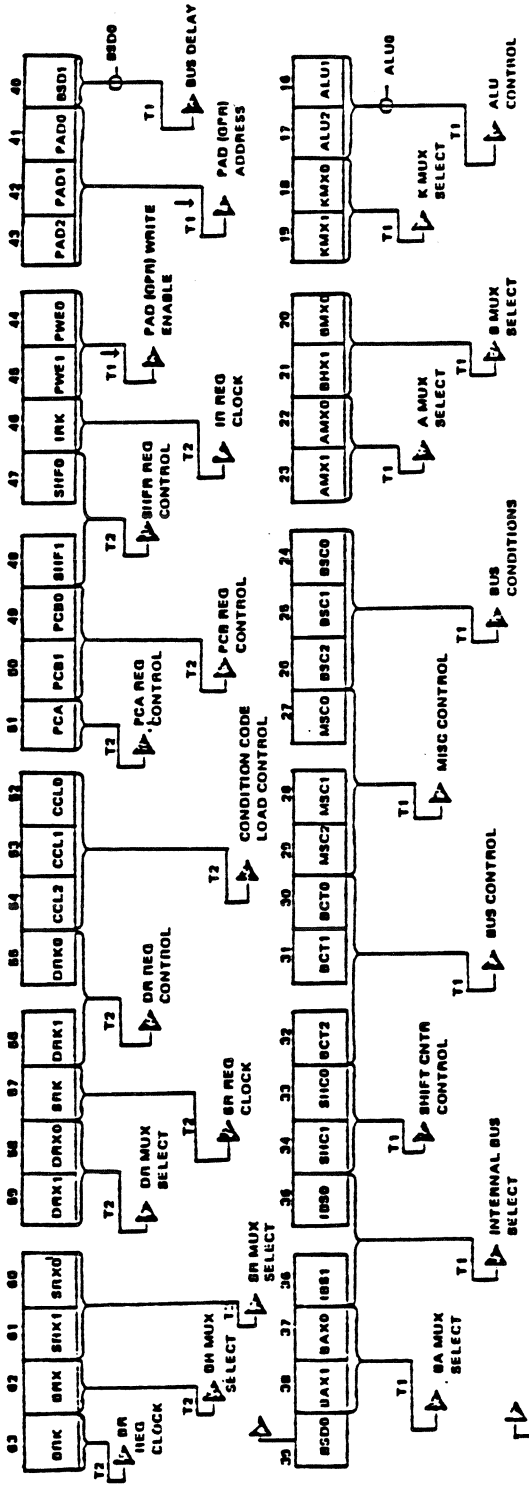


Figure 1-10 PDP-11/45, 11/70 Microinstruction Format

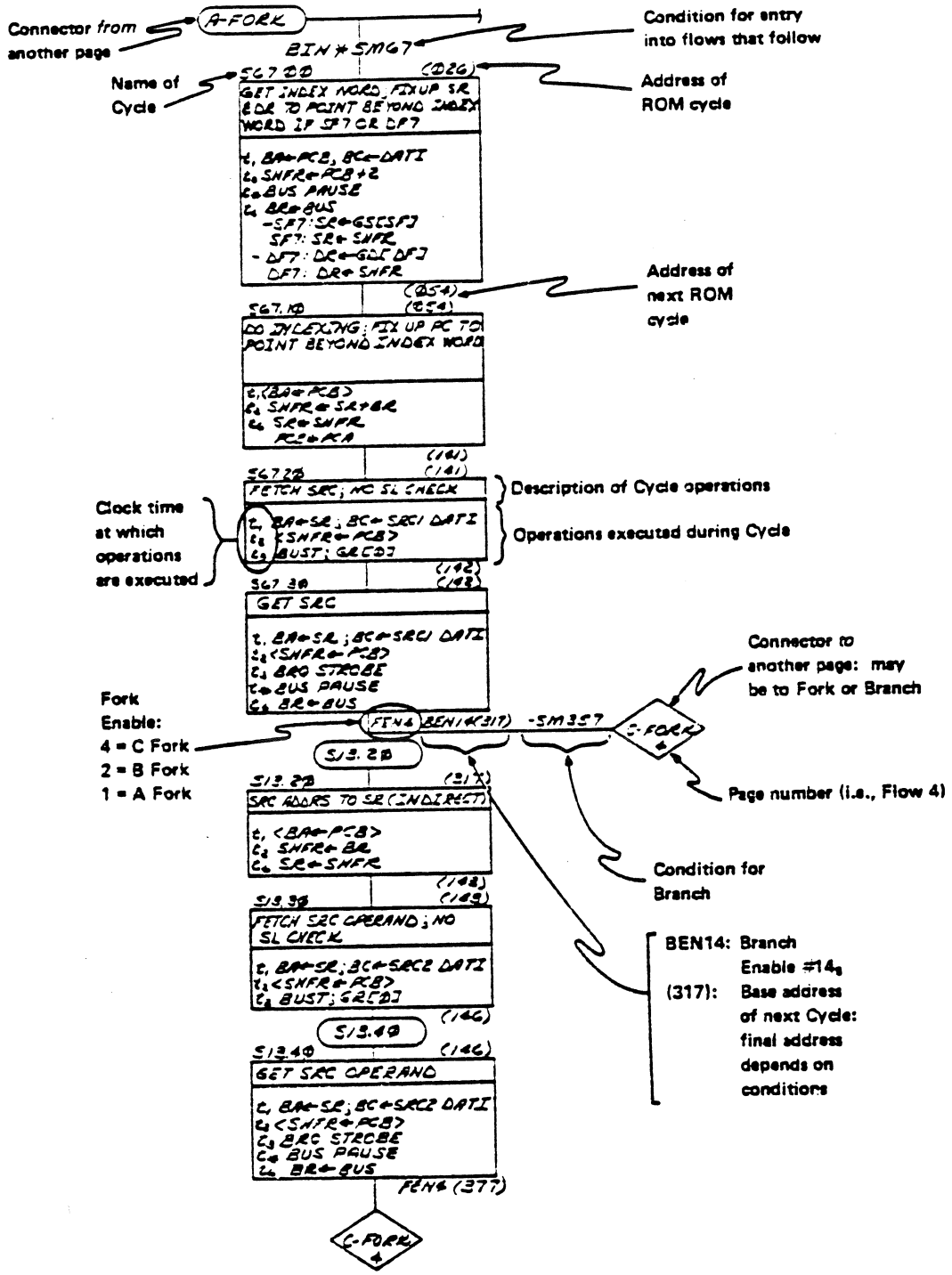


Figure 1-11 Flow Chart Symbols (P/O Flows 2)

Table 1-5 ON CONSOLE OPERATIONS

ROM STATE	PCA	PCB	ALU	SR	DR	BR	IR	SHFR	BAMX	NOTES
CON.00 170	------	------	------	------	------	------	------	------	------	Console Idle: Loop on CON.00 Until key is pressed
CON.00 170	------	------	------	------	------	1000	------	------	------	Value Placed in SWR = 1000 Press Load Adr.
ADR.00 270	1000	------	1000	1000	0	1000	------	1000	------	Store Adr. in PCA and SR.
CON.20 230	1000	------	0	1000	0	1000	------	0	1000	CONS ACKN (clears respective Function F/F)
CON.00 170	1000	------	0	1000	0	257	------	0	1000	SWR = 257 Press Deposit
DEP.10 073	1000	------	257	1000	0	257	------	257	1000	Start Dato Bus Cycle Send Adr to M.M For Relocation
DEP.20 303	1000	------	257	1000	257	257	------	257	1000	Finish Dato cycle Data in BRA sent to Mem or Bus; Wait-Resp
CON.20 230	1000	------	257	1000	257	257	------	257	1000	CONS ACKN
CON.00 170	1000	------	257	1000	257	401	------	257	1000	SWR = 401 Press Deposit
DEP.00 074	1000	------	1002	1002	257	401	------	1002	1000	Consecutive Dep. Update Address in SR
DEP.10 073	1000	------	401	1002	257	401	------	401	1002	Start Dato Cycle
DEP.20 303	1000	------	401	1002	401	401	------	401	1002	Finish Dato
CON.20 230	1000	------	401	1002	401	401	------	401	1002	CONS ACKN

FOR INSTRUCTION: CONSOLE OPERATION FLOW 14

LOAD ADDRESS
DEPOSIT

Table 1-6

ROM STATE	PCA	PCB	ALU	SR	DR	BR	IR	SHFR	BAMX	NOTES
CON.00 170	1000	------	401	1002	401	1000	------	401	1002	SWR = 1000 Load Adr
ADR.00 270	1000	------	1000	1000	0	1000	------	1000	1002	Store Address in PCA & SR
CON.20 230	1000	------	0	1000	0	1000	------	0	1000	CONS ACKN
CON.00 170	1000	------	0	1000	0	X	------	0	1000	SWR = Irrelevent EXAM
EXM.10 070	1000	------	0	1000	0	X	------	ls	1000	Start Dati Cycle Send Adr to M.M.
EXM.20 174	1000	------	0	1000	0	X	------	ls	1000	Finish Dati Data into BR & BRA
EXM.30 134	1000	------	257	1000	257	257	------	257	1000	Store Data in DR, Display in SHFR
CON.20 230	1000	------	257	1000	257	257	------	257	1000	CONS ACKN
CON.00 170	1000	------	257	1000	257	X	------	257	1000	SWR = X EXAM
EXM.00 071	1000	------	1002	1002	257	X	------	1002	1000	Successive Exam Update Address Store in SR
EXM.10 070	1000	------	1002	1002	257	X	------	50377	1002	Start Dati
EXM.20 174	1000	------	1002	1002	257	X	------	50377	1002	Finish Dati Get Data
EXM.30 134	1000	------	401	1002	401	401	------	401	1002	Store Data in DR, Display in SHFR

CON.20
CON.00
For Instruction: CONSOLE OPERATION FLOW 14

EXAM
STEP EXAM

MAINT CARD-FLOWS 14

I.SINGLE ROM CYCLE
Set the maintenance card to SINGLE ROM CYCLE S2-1 S1-0

A.LOAD ADDRESS
1.Set 1000 in switch register and hit LOAD ADDRESS key
2.Advance the micoprogram one Rom state at a time using the STEPPER S4 on the Maintenance card.
3.What ROM PATH was executed to do the this console operation?

_____

_____

_____ VERIFY USING FLOWS 14

B.DEPOSIT
1.Set 77 in switch register and hit the DEPOSIT key
2.Advance the micoprogram one Rom state at a time using the STEPPER S4 on the Maintenance card.
3.What ROM PATH was executed to do the this console operation?

_____

_____

_____ VERIFY USING FLOWS 14

_____

C. DEPOSIT STEP
1.Set 44 in the switches and hit DEPOSIT key.
2.Advance the micoprogram one Rom state at a time using the STEPPER S4 on the Maintenance card.
3.What ROM PATH was executed to do the this console operation?

_____

_____

_____ VERIFY USING FLOWS 14

_____

D.EXAMINE
SET the maintenance card to NORMAL operation S2=0 S1=0
Load address 1000
Now set the Maintenance card to SINGLE ROM CYCLE again

1.Hit the EXAMINE key
2.Advance the micoprogram one Rom state at a time using the STEPPER S4 on the Maintenance card.
3.What ROM PATH was executed to do the this console operation?

_____

_____

_____ VERIFY USING FLOWS 14

_____

_____

E.EXAMINE STEP
1.Hit the examine key again
2.Advance the micoprogram one Rom state at a time using the STEPPER S4
on the Maintenance card.
3.What ROM PATH was executed to do the this console operation?

_____

_____

_____

_____

VERIFY USING FLOWS 14

II.This part of the Lab will help you become familiar with those registers in the CPU block diagram that are visable from the fromt panel by manipulating the rotational data switch.These will be executed in SINGLE ROM CYCLE mode.

A. LOAD ADDRESS

1.Set 1000 in the switches and hit LOAD ADDRESS KEY
2.Fill in the table below using the rotational data display switch to examine the various registers.
3.Execute the program one rom state at a time using the STEPPER SWITCH.

UADD	ADDRESS LIGHTS	SHIFTER	BR

B.DEPOSIT

1.Set 77 in the switches and hit DEPOSIT.
2.Fill in the table below using the rotational data display switch to examine the various registers.
3.Execute the program one rom state at a time using the STEPPER SWITCH.

UADD	ADDRESS LIGHTS	SHIFTER	BR

C.DEPOSIT STEP
 1.Set 44 in the switches and hit DEPOSIT again.
  2.Fill in the table below using the rotational data display switch to
    examine the various registers.
  3.Execute the program one rom state at a time using the STEPPER SWITCH.

UADD	ADDRESS LIGHTS	SHIFTER	BR

D.EXAMINE
 Return the MAINTENANCE CARD to NORMAL
 LOAD ADDRESS 1000
 RESET THE MAINTENANCE CARD TO SINGLE ROM CYCLE .

 1.Hit the EXAMINE KEY
  2.Fill in the table below using the rotational data display switch to
    examine the various registers.
  3.Execute the program one rom state at a time using the STEPPER SWITCH.

UADD	ADDRESS LIGHTS	SHIFTER	BR

E.EXAMINE STEP
1.Hit EXAMINE again
2.Fill in the table below using the rotational data display switch to examine the various registers.
3.Execute the program one rom state at a time using the STEPPER SWITCH.

UADD	ADDRESS LIGHTS	SHIFTER	BR

III. This part of the Lab will help you become familiar with those registers in the CPU block diagram that are visable from the fromt panel by manipulating the rotational data switch. These will be executed in SINGLE TIME PULSE mode

A. LOAD ADDRESS

1. Set 1000 in the switches and hit LOAD ADDRESS KEY
2. Fill in the table below using the rotational data display switch to examine the various registers.
3. Execute the program one TIMING state at a time using the STEPPER SWITCH.

ROM STATE 170

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 270

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 230

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

B.DEPOSIT
1.Set 77 in the switches and hit DEPOSIT.
2.Fill in the table below using the rotational data display switch to examine the various registers.
3.Execute the program one TIMING state at a time using the STEPPER SWITCH.

ROM STATE 170

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 73

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 303

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 230

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

C.DEPOSIT STEP
1.Set 44 in the switches and hit DEPOSIT again.
2.Fill in the table below using the rotational data display switch to examine the various registers.
3.Execute the program one TIMING state at a time using the STEPPER SWITCH.

ROM STATE 170

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 74

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 73

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 303

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 230

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

D.EXAMINE
Return the MAINTENANCE CARD to NORMAL
LOAD ADDRESS 1000
RESET THE MAINTENANCE CARD TO SINGLE TIME PULSE.

1.Hit the EXAMINE KEY
2.Fill in the table below using the rotational data display switch to examine the various registers.
3.Execute the program one TIMING state at a time using the STEPPER SWITCH.

ROM STATE 170

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 70

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 174

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 134

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 230

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

E.EXAMINE STEP
1.Hit EXAMINE again
2.Fill in the table below using the rotational data display switch to examine the various registers.
3.Execute the program one TIMING state at a time using the STEPPER SWITCH.

ROM STATE 170

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 71

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 70

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 174

TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 134

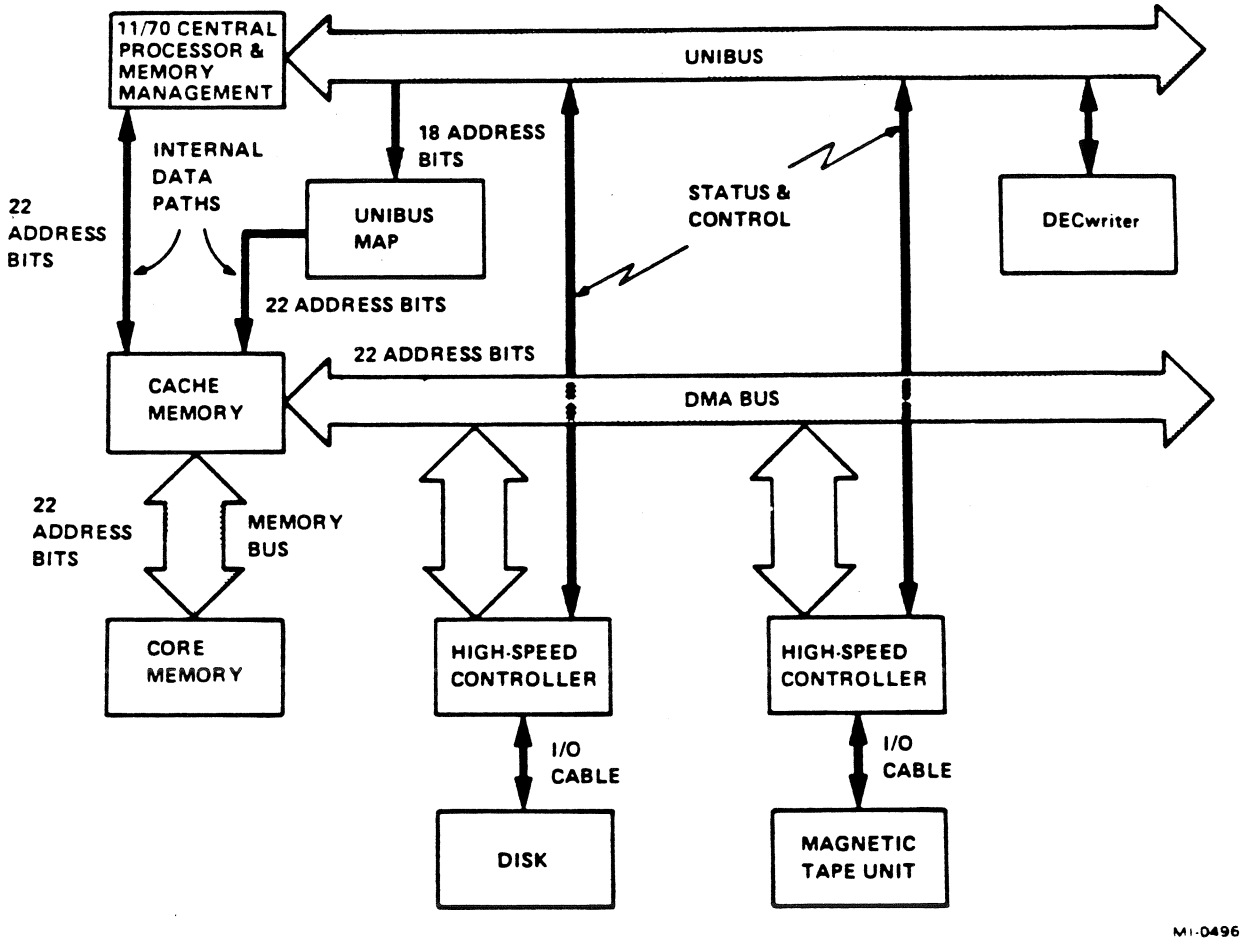
TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

ROM STATE 230

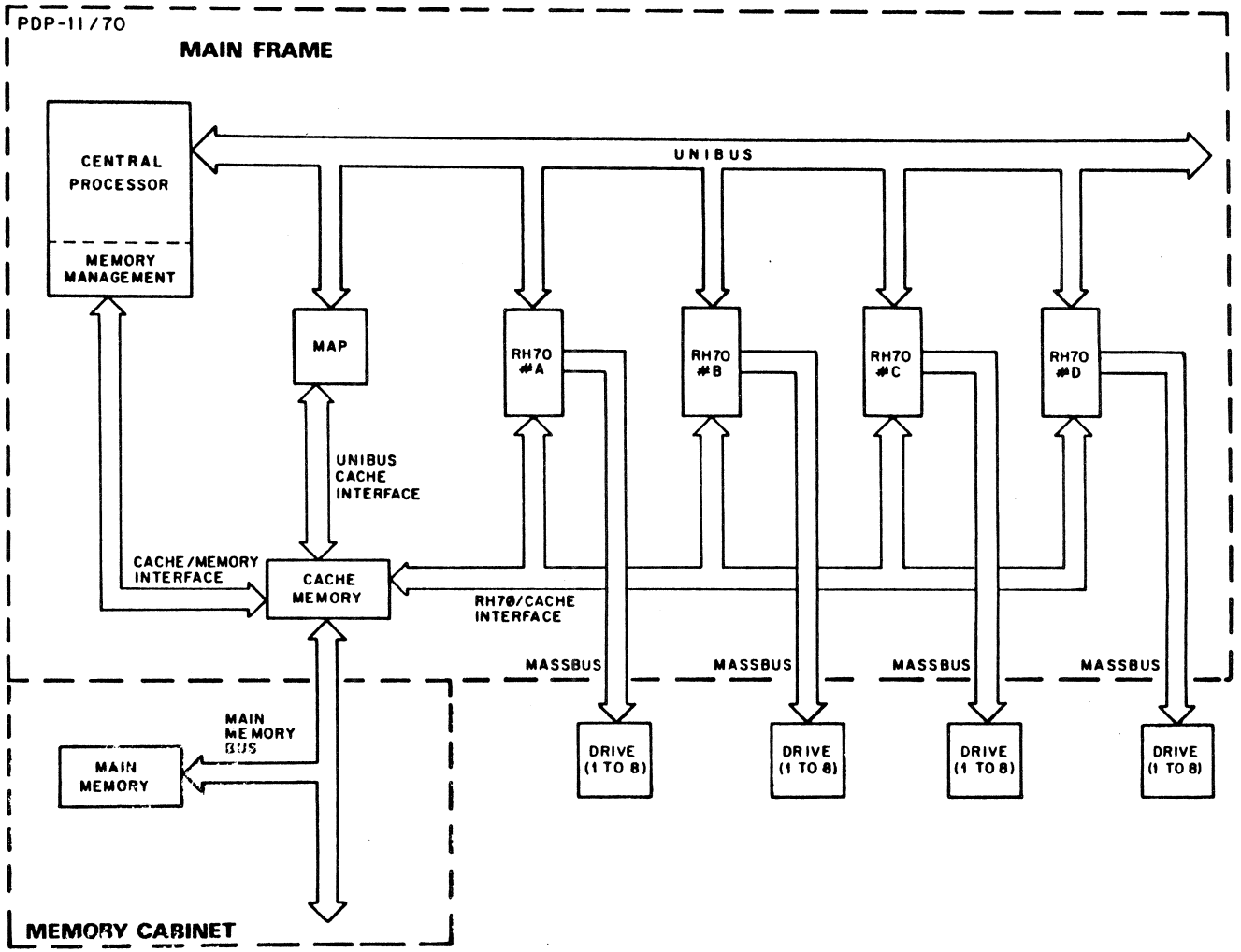
TIME STATE	ADDRESS LIGHTS	SHIFTER	BR	UADD

PDP-11/70 Diagnostics and Module Level Repair

DAY NINE



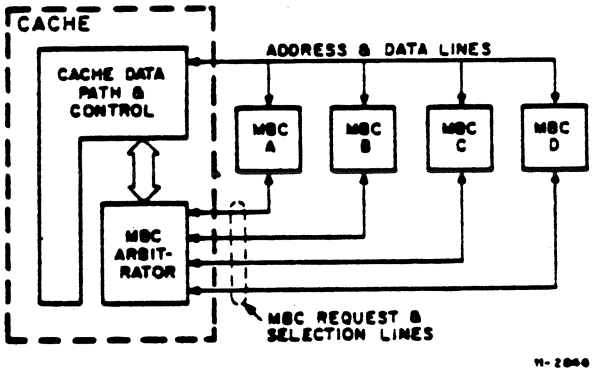
Typical PDP-11/70 Computer System



RH70 Simplified System Block Diagram

MBC Selection Priorities

Jumper Configuration			Priority Structure *
W1	W2	W3	
OUT	OUT	OUT	(A ↔ B) ↔ (C ↔ D)
OUT	OUT	IN	(A → B) ↔ (C ↔ D)
OUT	IN	OUT	(A ↔ B) ↔ (C → D)
OUT	IN	IN	(A → B) ↔ (C → D)
IN	OUT	OUT	(A ↔ B) → (C ↔ D)
IN	OUT	IN	(A → B) → (C ↔ D)
IN	IN	OUT	(A ↔ B) → (C → D)
IN	IN	IN	(A → B) → (C → D)



Relationship of the MBC Arbitrator to the Cache

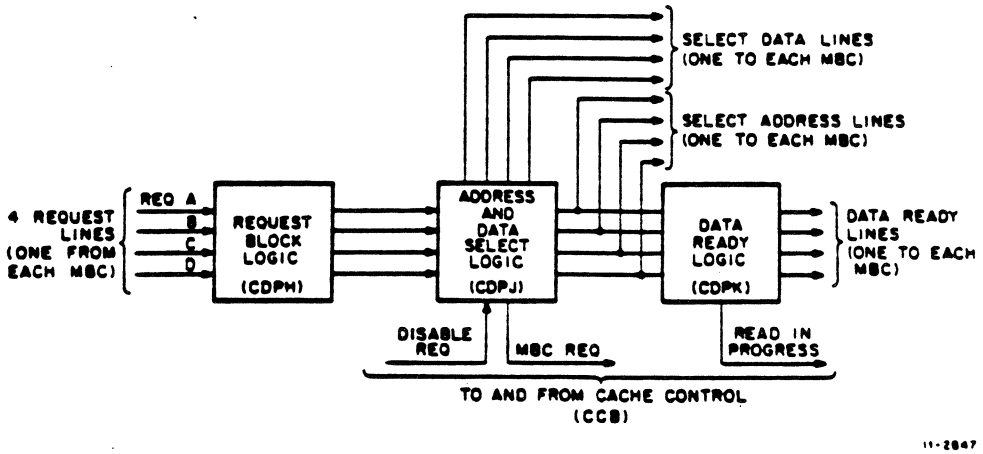


Figure 4-5 MBC Arbitrator Block Diagram

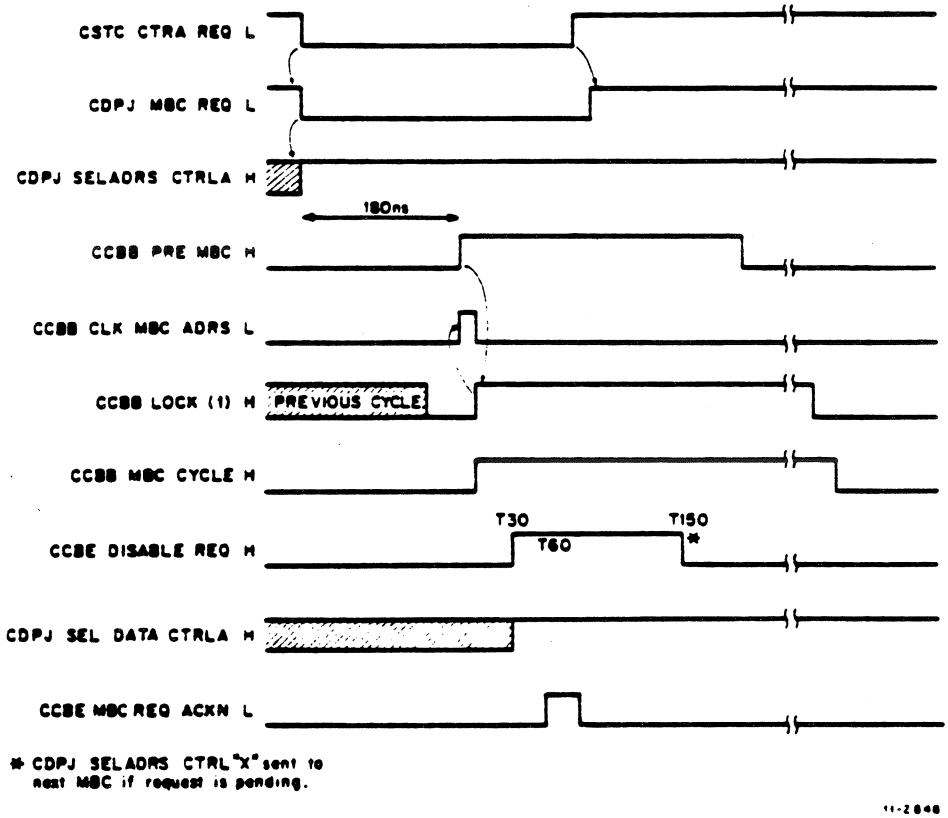
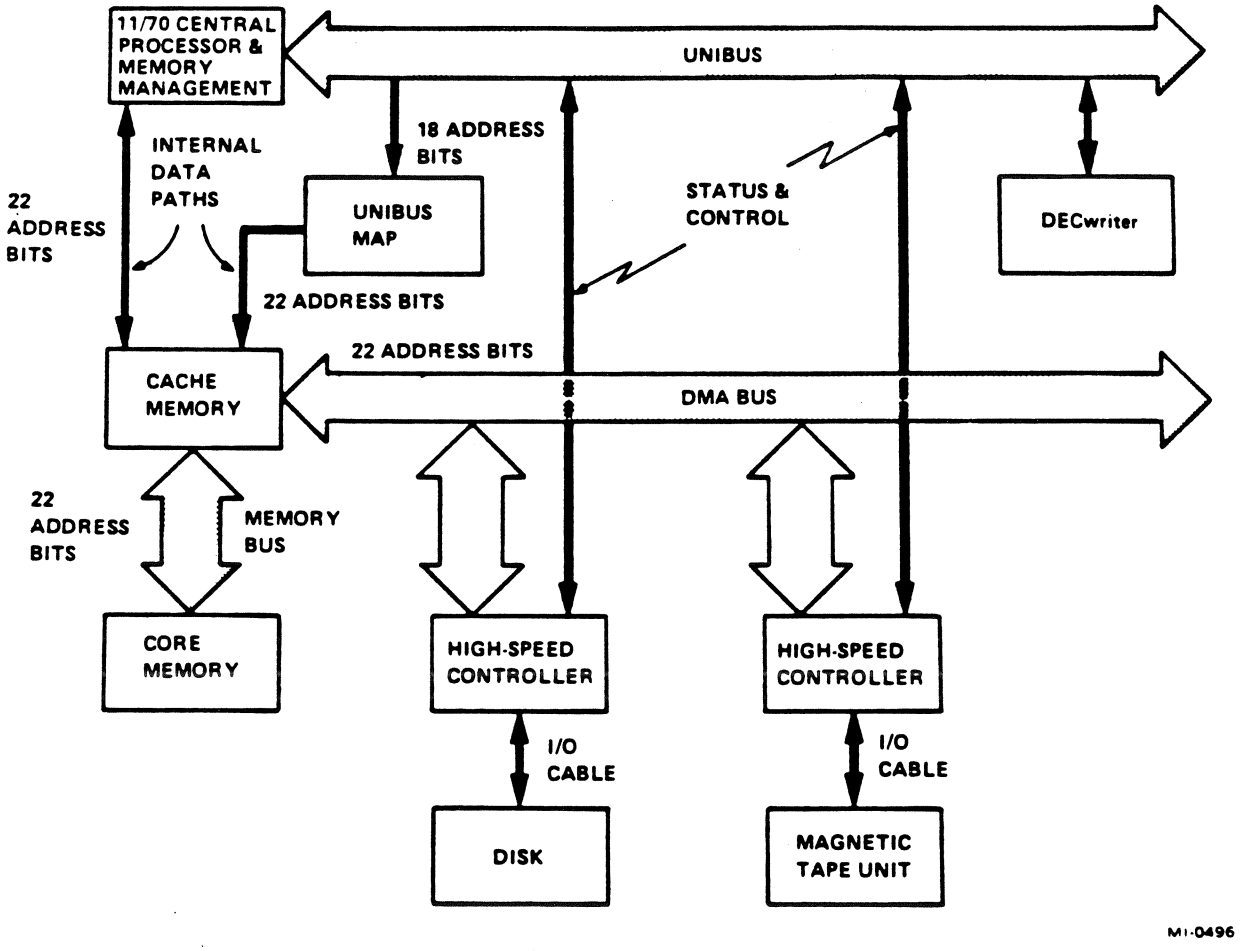


Figure 4-6 MBC Request Timing (MBC A Requesting)

PDP-11/70 Diagnostics and Module Level Repair

DAY TEN



Typical PDP-11/70 Computer System

MJ11 MEMORY OPTIONS

MJ11-AE 32K words

MJ11-AA: Memory Box with Controller and Transceiver, 32K words, 110 V.

AB: Memory Box with Controller and Transceiver, 32K words, 220 V.

MJ11-AG: Memory Box with Controller and Transceiver, 128K words, 110 V.

AH: Memory Box with Controller and Transceiver, 128K words, 220 V.

MJ11-AC: Memory Cabinet and one Memory Box with Controller and Transceiver, 128K words, 110 V.

AD: Memory Cabinet and one Memory Box with Controller and Transceiver, 128K words, 220 V.

Maintenance Spares (not for expansion purposes)

MJ11-AM: 16K words

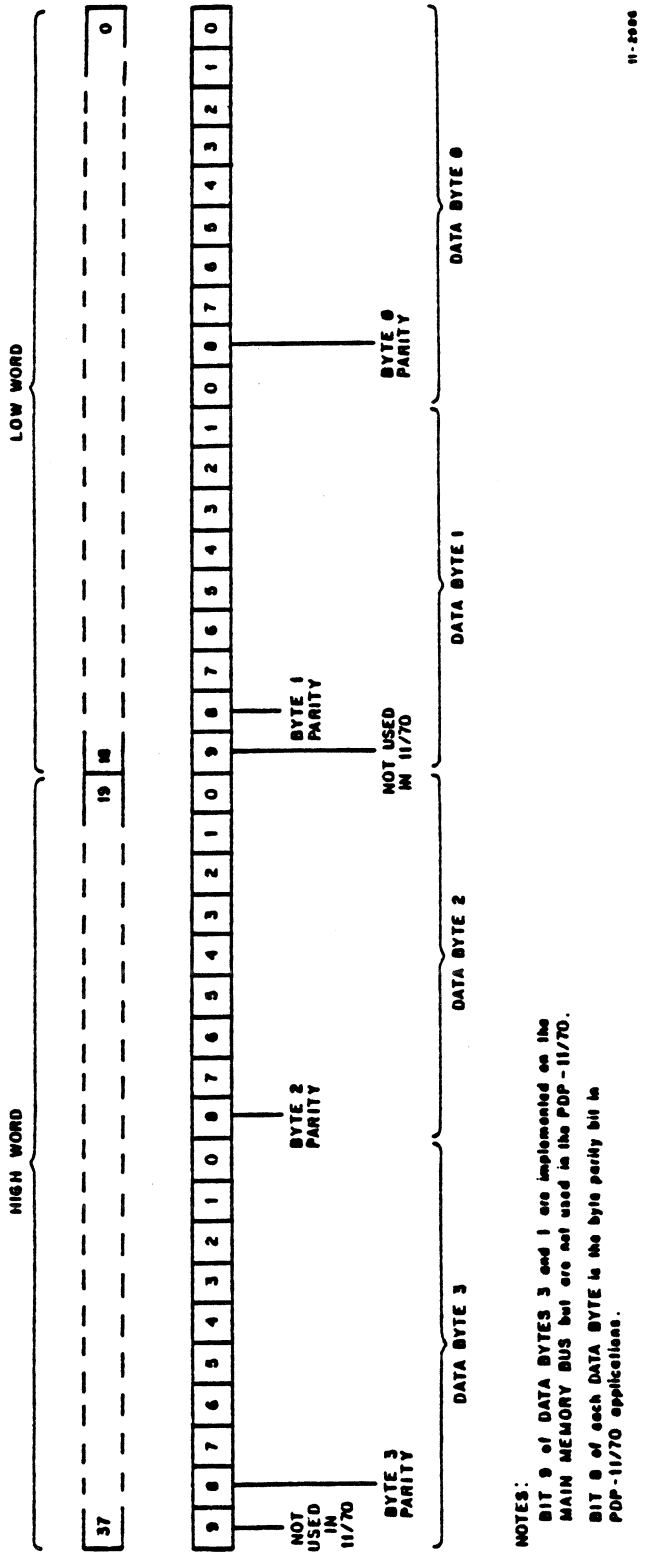


Figure 5-4 Data Word Organization

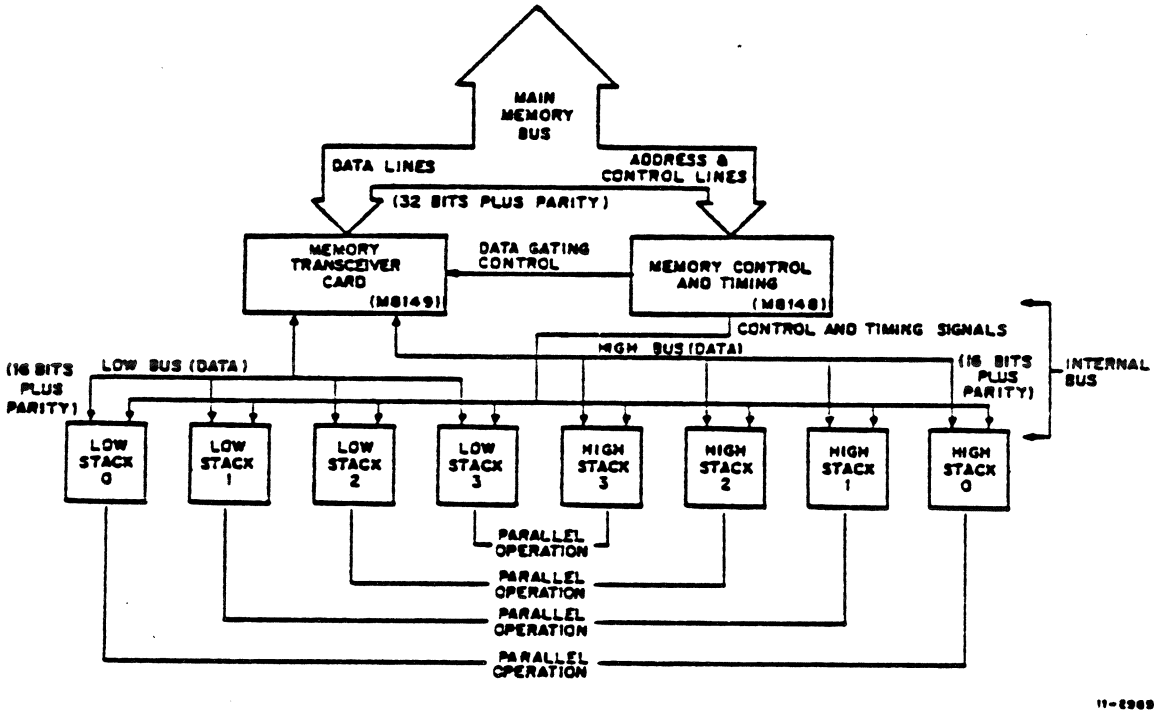


Figure 5-5 MJ11 Simplified Block Diagram

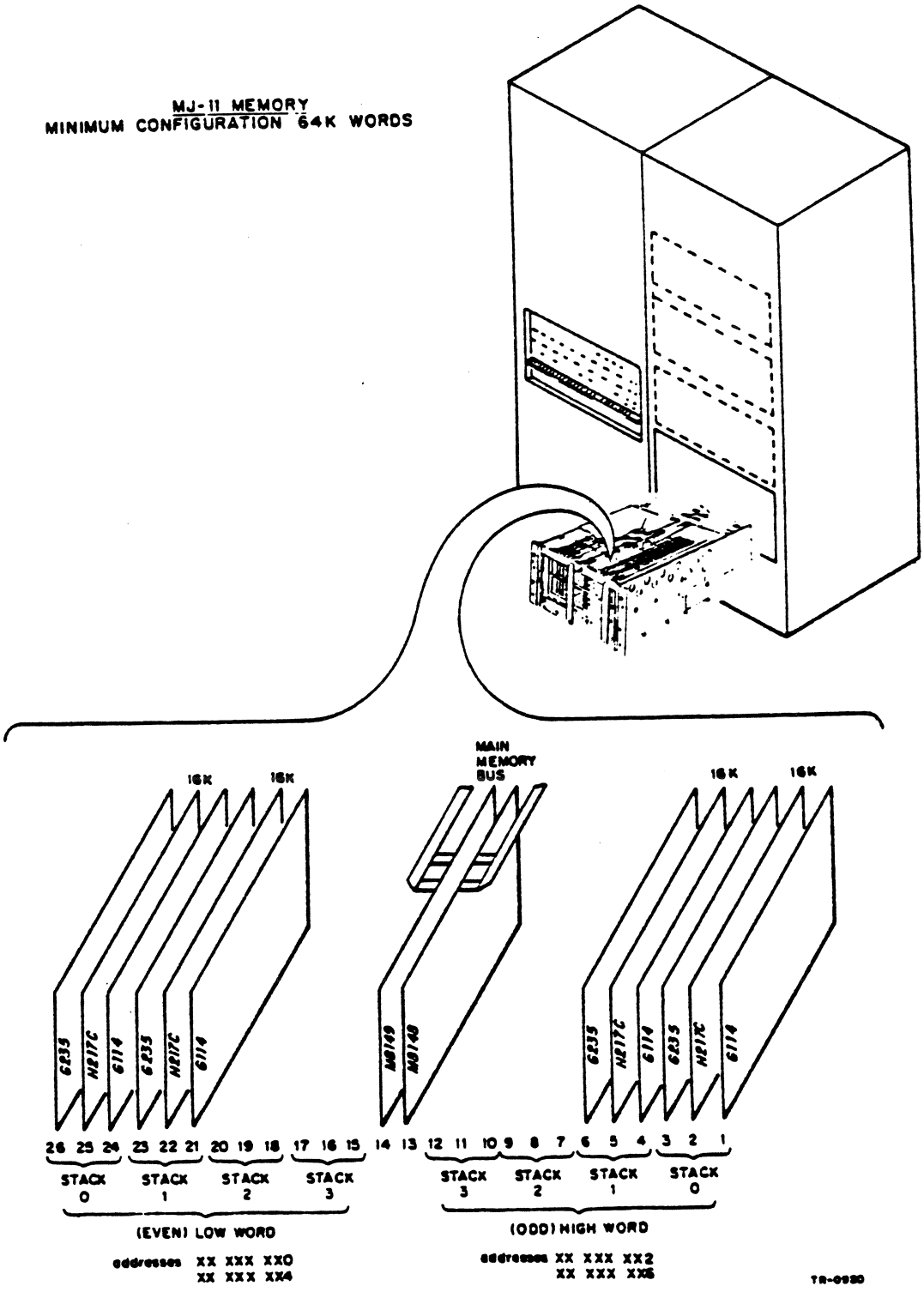

MJ-11 MEMORY
MINIMUM CONFIGURATION 64K WORDS
MAIN MEMORY BUS
16K
16K
16K
16K
26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
STACK 0
STACK 1
STACK 2
STACK 3
STACK 3
STACK 2
STACK 1
STACK 0
(EVEN) LOW WORD
addresses XX XXX XX0
XX XXX XX4
(ODD) HIGH WORD
addresses XX XXX XX2
XX XXX XX6

Figure 5-6

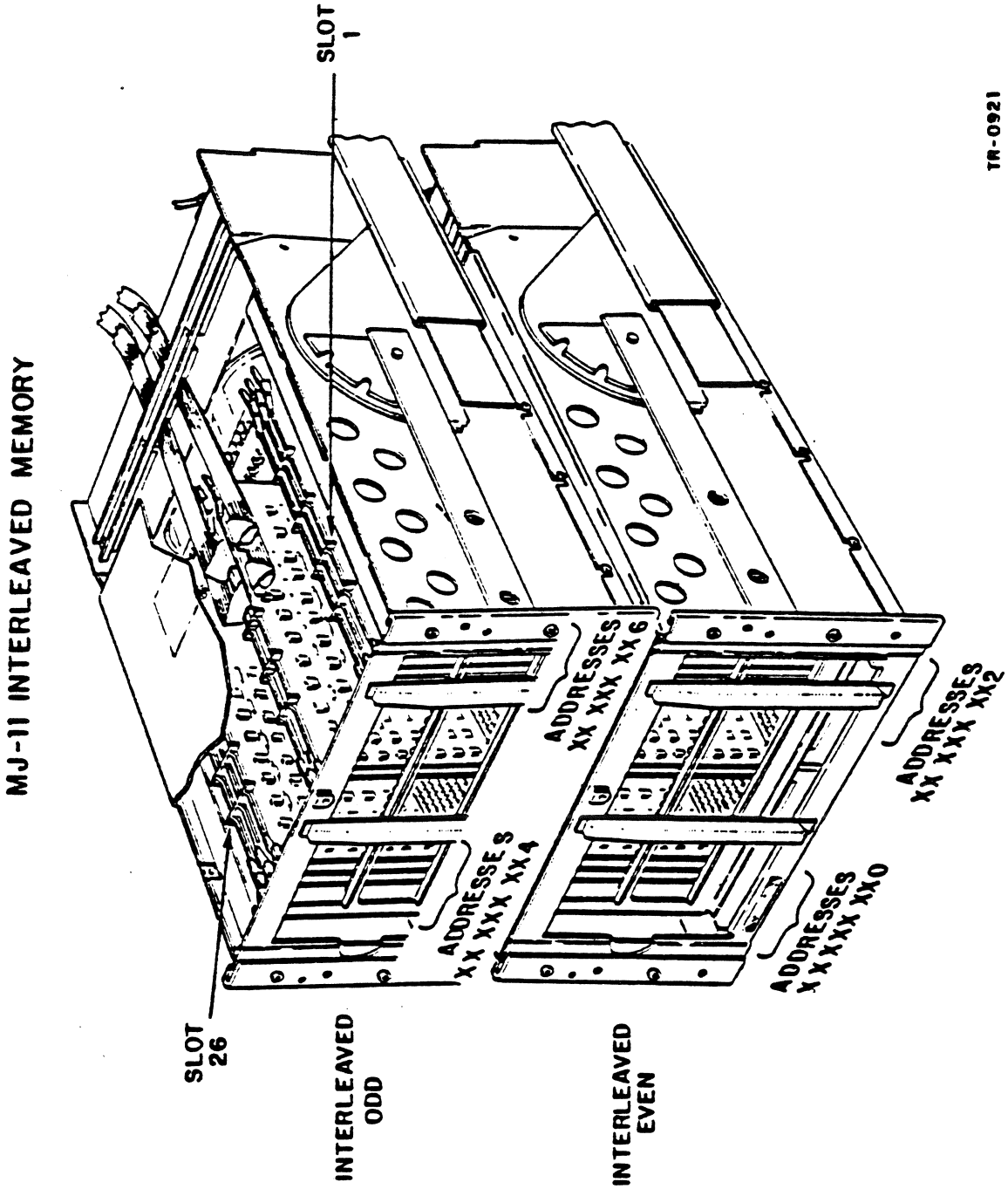

MJ-11 INTERLEAVED MEMORY
SLOT 1
SLOT 26
INTERLEAVED ODD
INTERLEAVED EVEN
ADDRESSES XX XXX XX4
ADDRESSES XX XXX XX6
ADDRESSES XX XXX XX0
ADDRESSES XX XXX XX2
TR-0921

Figure 5-7

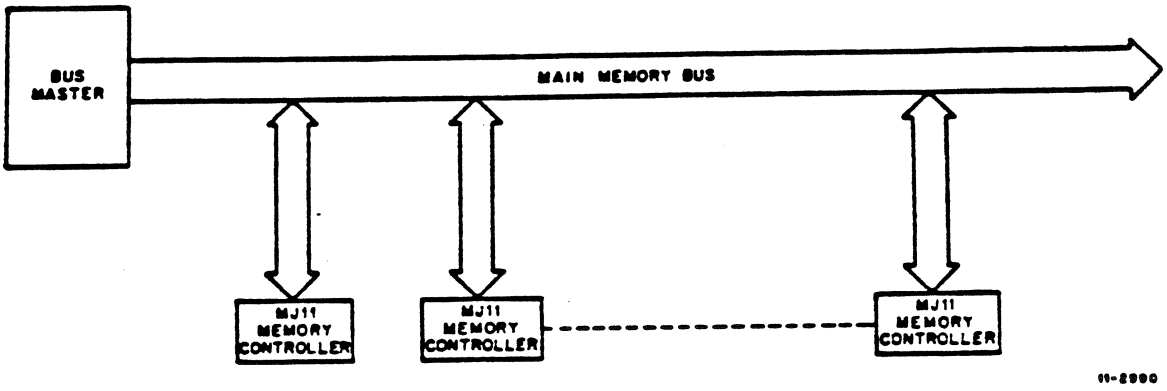


Figure 5-8 Relationship of Bus Master to Memory Controllers

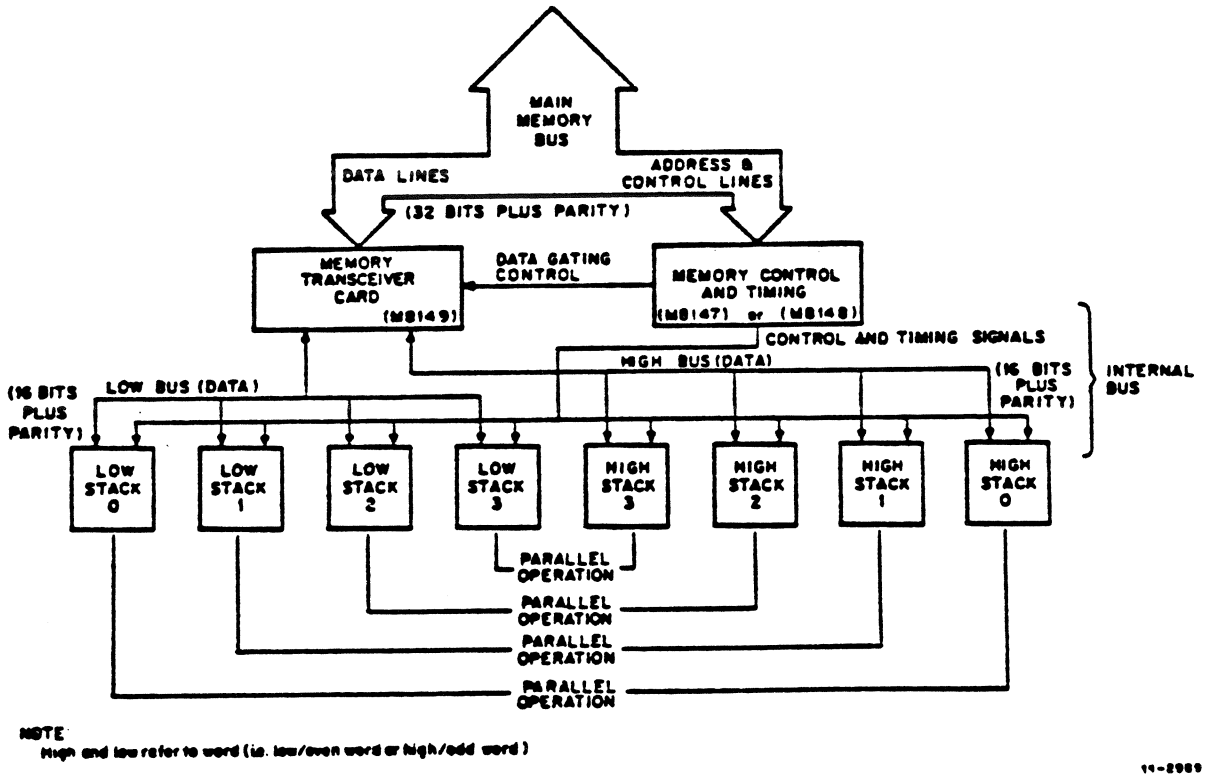


Figure 5-9 MJ11 Simplified Block Diagram

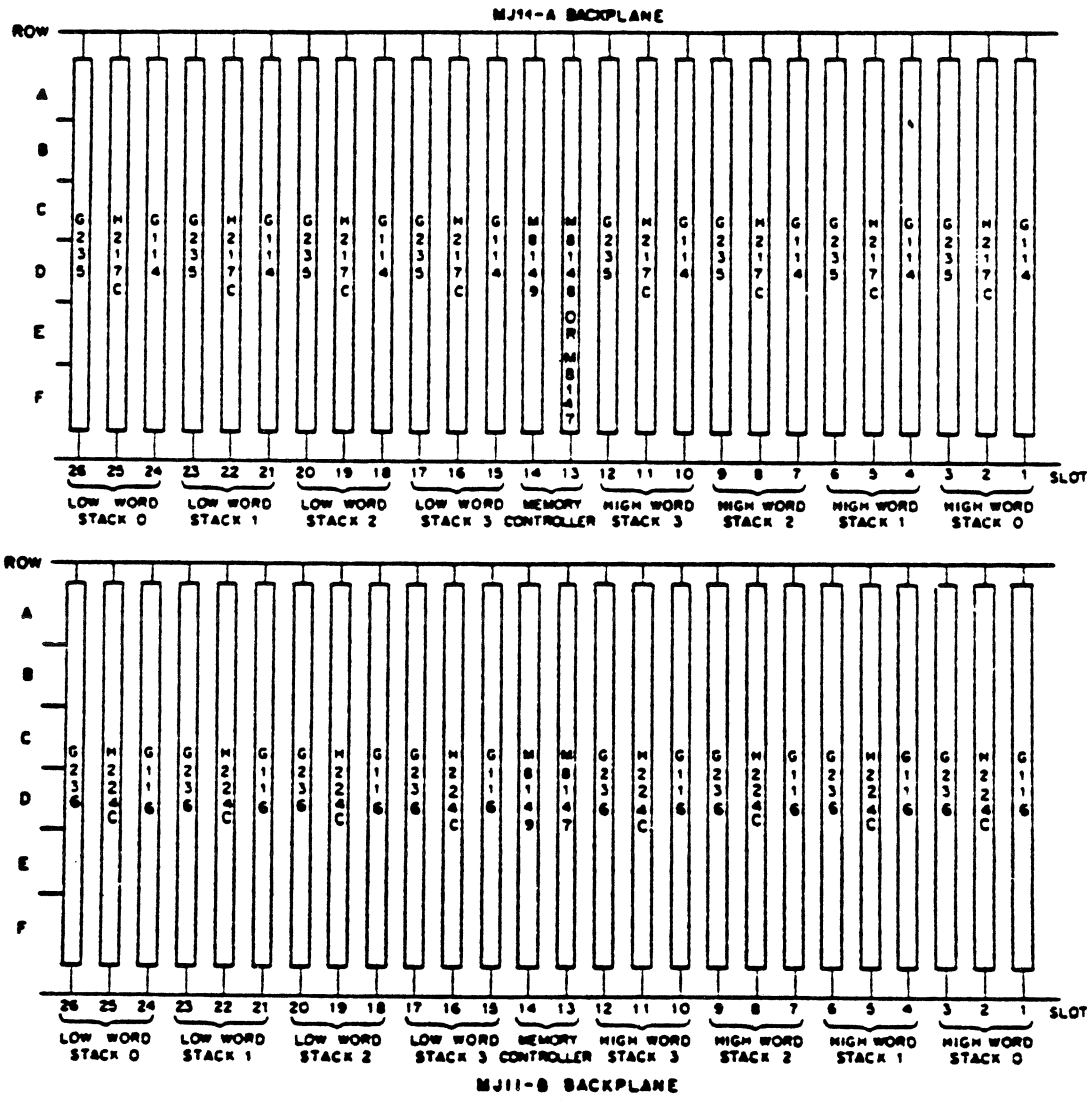

MJ11-A BACKPLANE
ROW
A
B
C
D
E
F
G235
M2117C
G114
M8149
M8147
ORM8147
26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
SLOT
LOW WORD STACK 0
LOW WORD STACK 1
LOW WORD STACK 2
LOW WORD STACK 3
MEMORY CONTROLLER
HIGH WORD STACK 3
HIGH WORD STACK 2
HIGH WORD STACK 1
HIGH WORD STACK 0
ROW
A
B
C
D
E
F
G236
M2240C
G116
M8149
M8147
26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
SLOT
LOW WORD STACK 0
LOW WORD STACK 1
LOW WORD STACK 2
LOW WORD STACK 3
MEMORY CONTROLLER
HIGH WORD STACK 3
HIGH WORD STACK 2
HIGH WORD STACK 1
HIGH WORD STACK 0
MJ11-B BACKPLANE

Figure 5-10

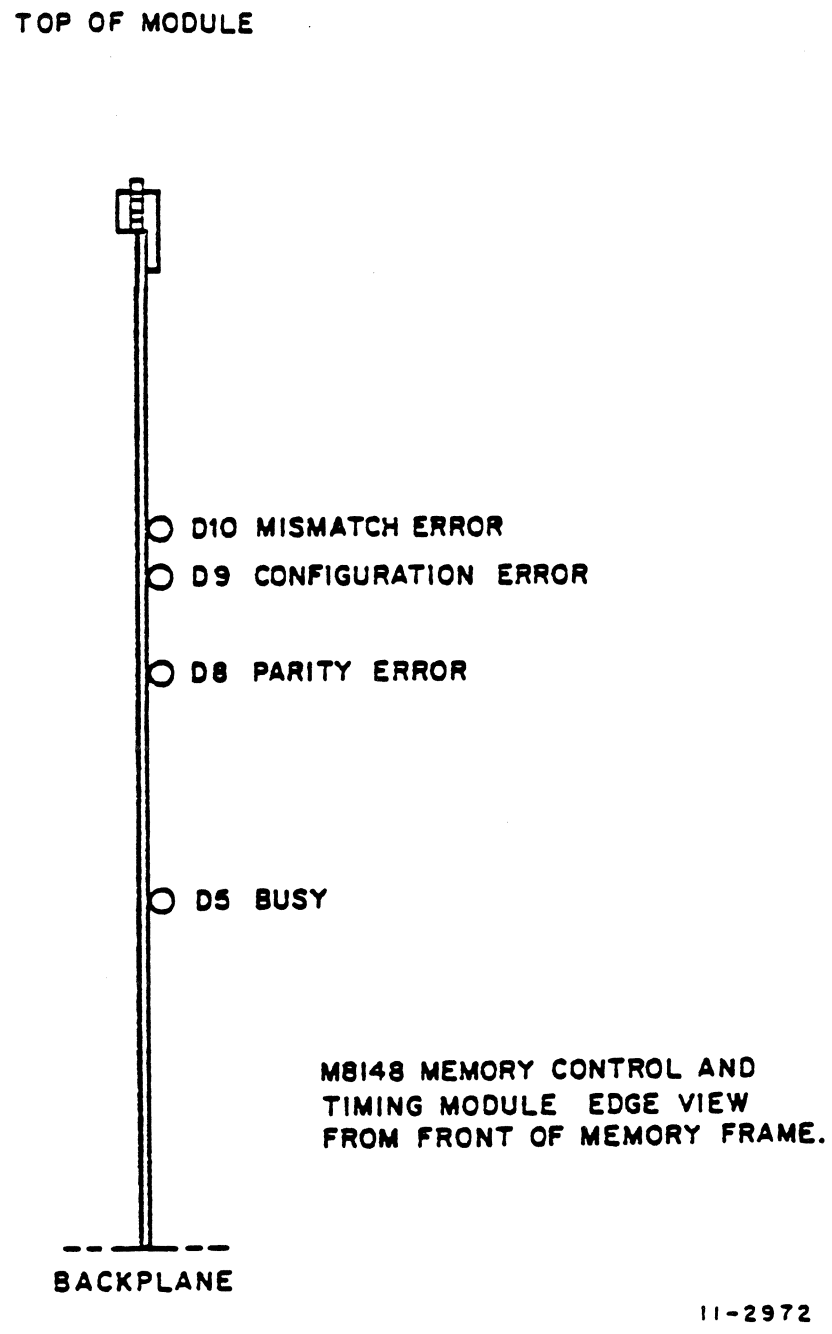


Figure 5-11 Location of Memory Controller Status LED Indicators

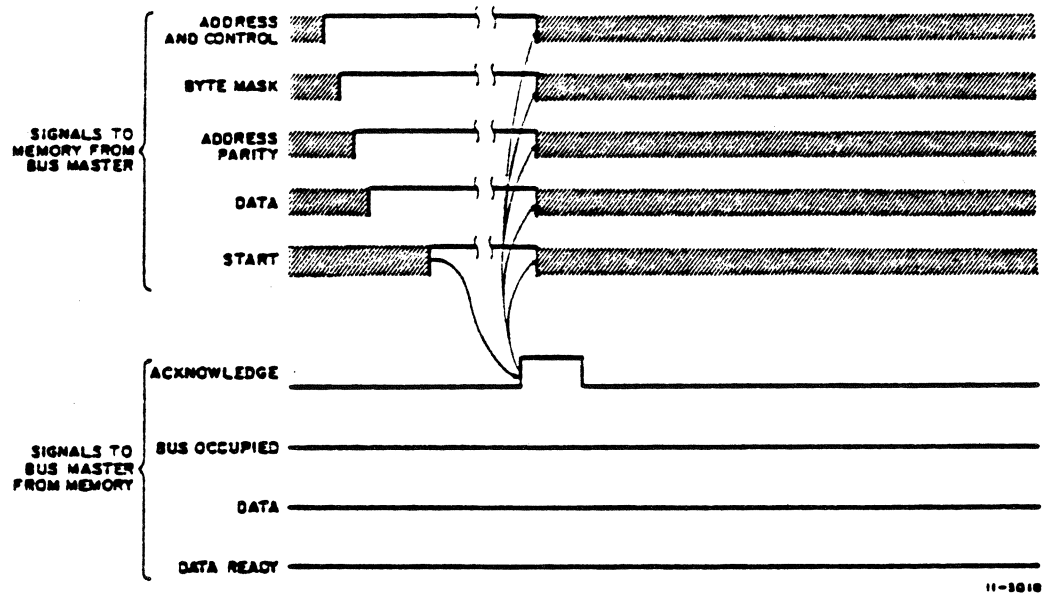


Figure 5-12 Write Operation - Main Memory Bus Protocol

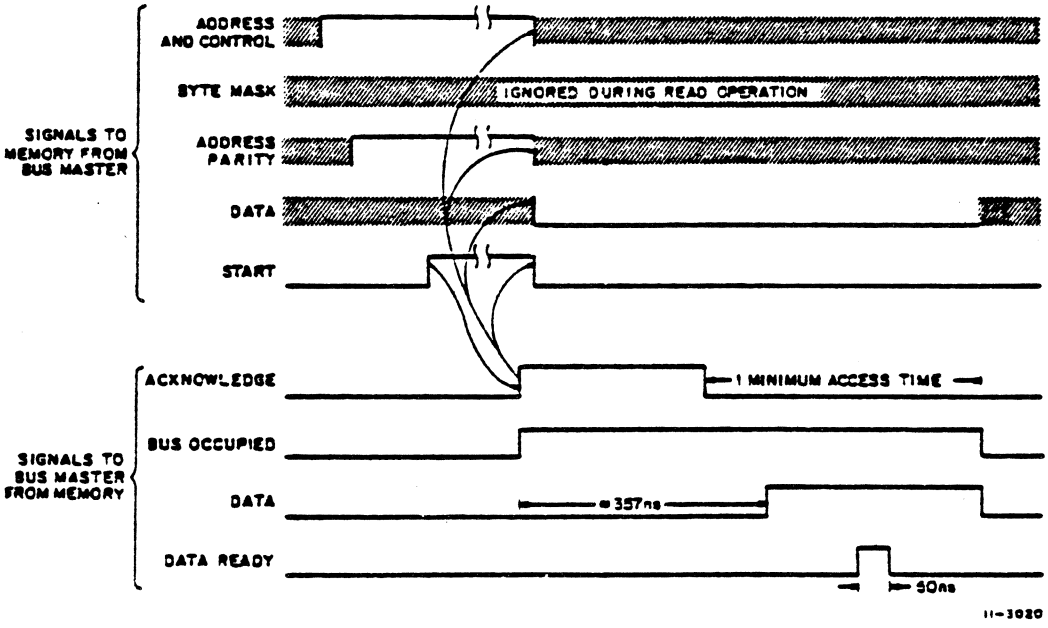


Figure 5-13 Read Operation - Main Memory Bus Protocol

TABLE 5-1 SYSTEM SIZE TABLE (JUMPER SETTINGS ON M8140)

SYSTEM SIZE IN "K" WORDS (10)	LAST MEMORY ADDRESS (8)	1 = SW OFF 0 = SW ON SIZE JUMPER SETTINGS	CONTENTS OF SYSTEM SIZE REGISTER 17777760
32	00177776	00000101	001777
64	00377776	00010101	003777
96	00577776	10000101	005777
128	00777776	10010101	007777
160	01177776	00001101	011777
192	01377776	00011101	013777
224	01577776	10001101	015777
256	01777776	10011101	017777
288	02177776	00000111	021777
320	02377776	00010111	023777
352	02577776	10000111	025777
384	02777776	10010111	027777
416	03177776	00001111	031777
448	03377776	00011111	033777
480	03577776	10001111	035777
512	03777776	10011111	037777
544	04177776	00100101	041777
576	04377776	00110101	043777
608	04577776	10100101	045777
640	04777776	10110101	047777
672	05177776	00101101	051777
704	05377776	00111101	053777
736	05577776	10101101	055777
768	05777776	10111101	057777
800	06177776	00100111	061777
832	06377776	00110111	063777
864	06577776	10100111	065777
896	06777776	10110111	067777
928	07177776	00101111	071777
960	07377776	00111111	073777
992	07577776	10101111	075777
1024	07777776	10111111	077777

TABLE 5-1 SYSTEM SIZE TABLE (JUMPER SETTINGS ON M8140) (Cont'd)

SYSTEM SIZE IN "K" WORDS (10)	LAST MEMORY ADDRESS (8)	1 = SW OFF 0 = SW ON SIZE JUMPER SETTINGS	CONTENTS OF SYSTEM SIZE REGISTER 17777600
1056	10177776	01000101	101777
1088	10377776	01010101	103777
1120	10577776	11000101	105777
1152	10777776	11010101	107777
1184	11177776	01001101	111777
1216	11377776	01011101	113777
1248	11577776	11001101	115777
1280	11777776	11011101	117777
1312	12177776	01000111	121777
1344	12377776	01010111	123777
1376	12577776	11000111	125777
1408	12777776	11010111	127777
1440	13177776	01001111	131777
1472	13377776	01011111	133777
1504	13577776	11001111	135777
1536	13777776	11011111	137777
1568	14177776	01100101	141777
1600	14377776	01110101	143777
1632	14577776	11100101	145777
1664	14777776	11110101	147777
1696	15177776	01101101	151777
1728	15377776	01111101	153777
1760	15577776	11101101	155777
1792	15777776	11111101	157777
1824	16177776	01100111	161777
1856	16377776	01110111	163777
1888	16577776	11100111	165777
1920	16777776	11110111	167777
1952	17177776	01101111	171777
1984	17377776	01111111	173777
2016	17577776	11101111	175777
2048	17777776	11111111	177777

MEMORY SYSTEM ORGANIZATION

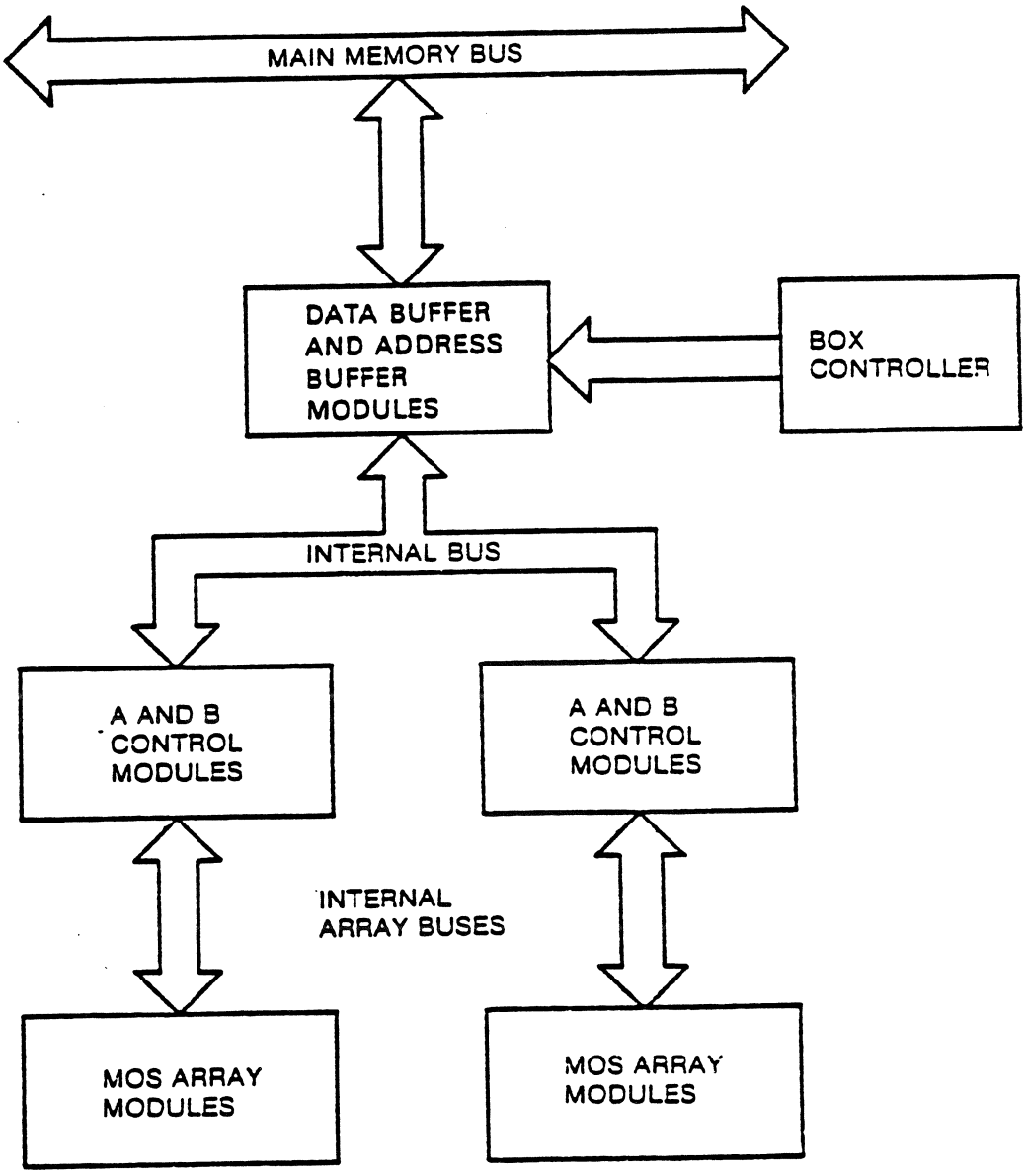


Figure 5-24 Internal Organization of Memory System

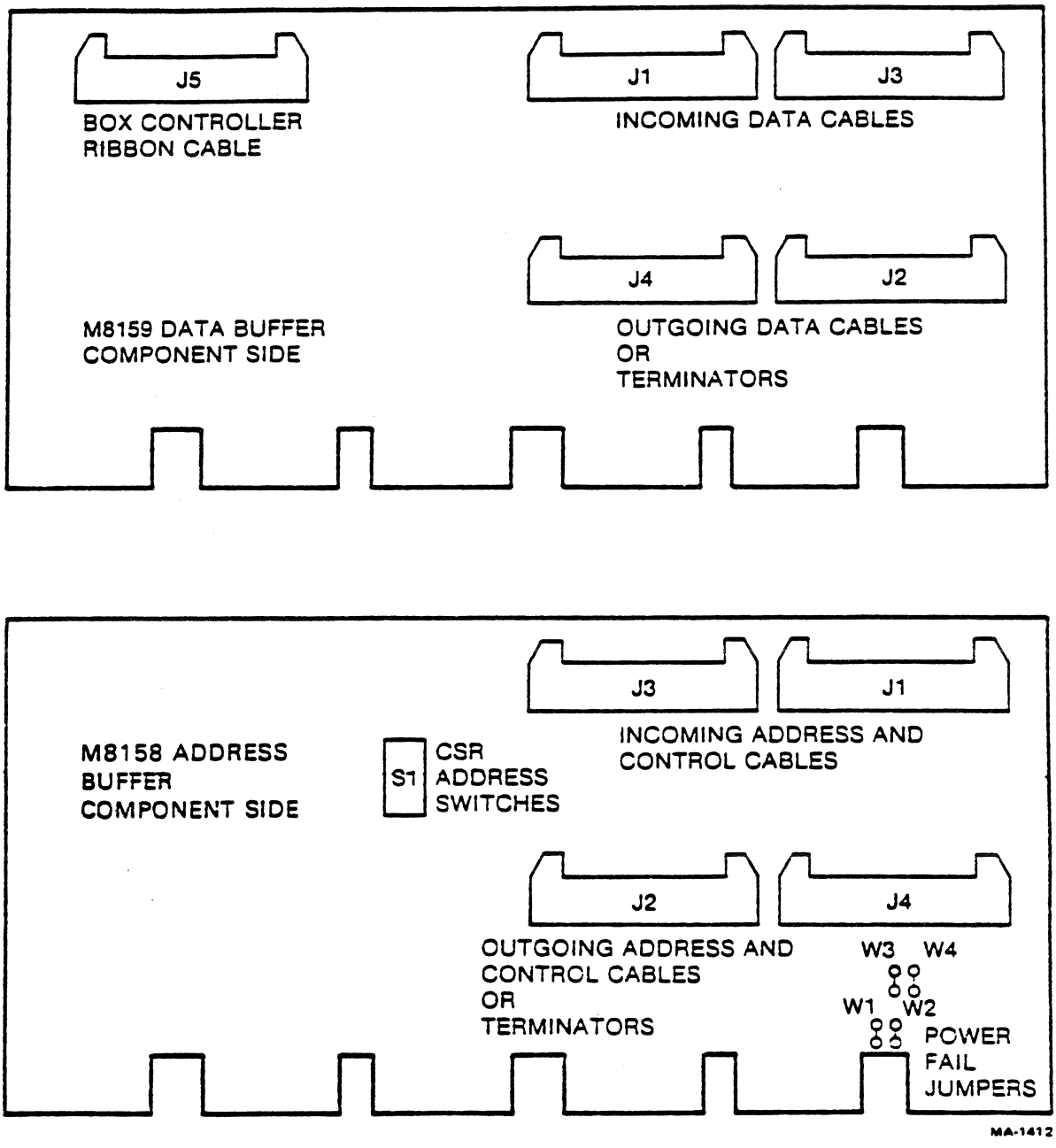
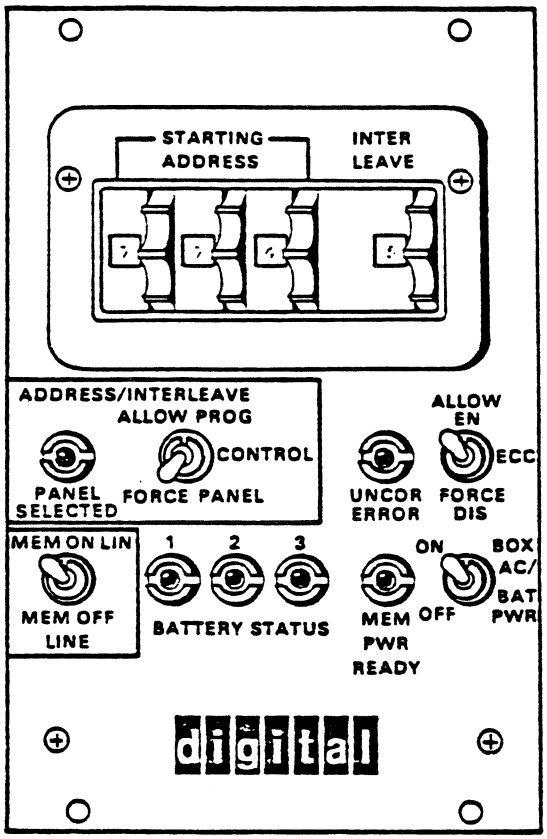


Figure 5-26 Address and Data Buffer Plug Locator, Plug In Cables with Ribbed Side Up and Red Stripe to the Left



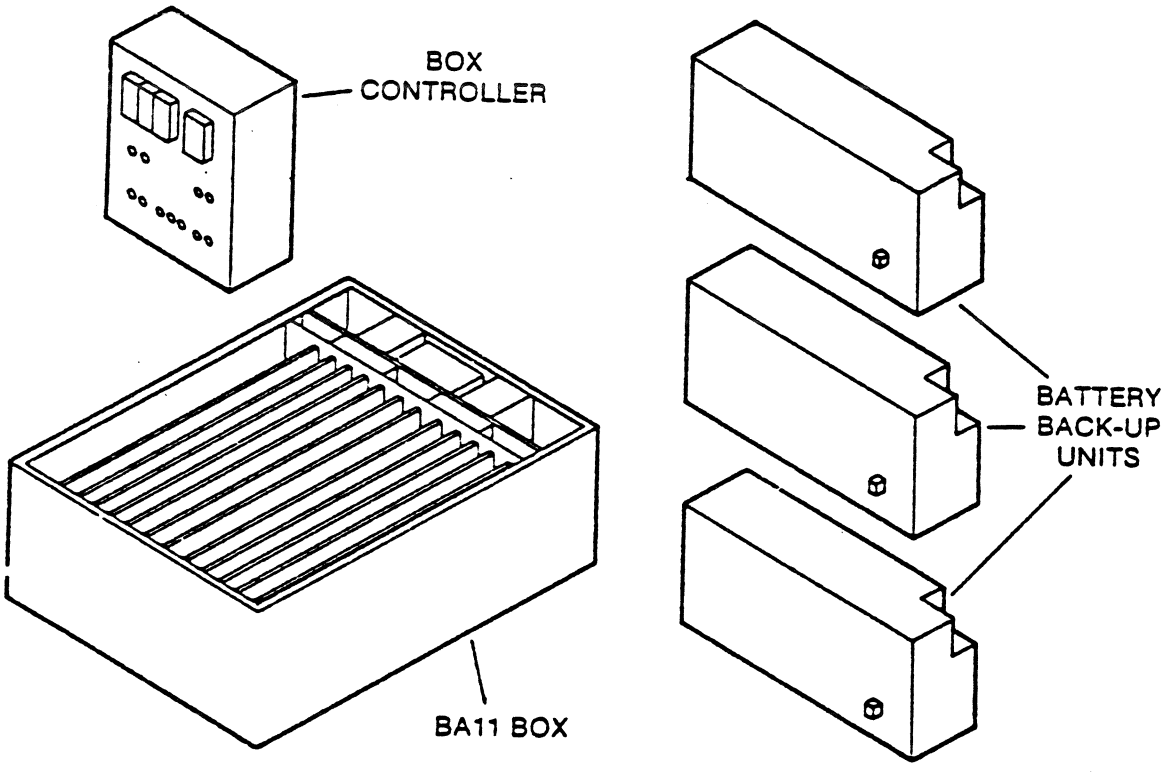


Figure 5-16 Major Assemblies

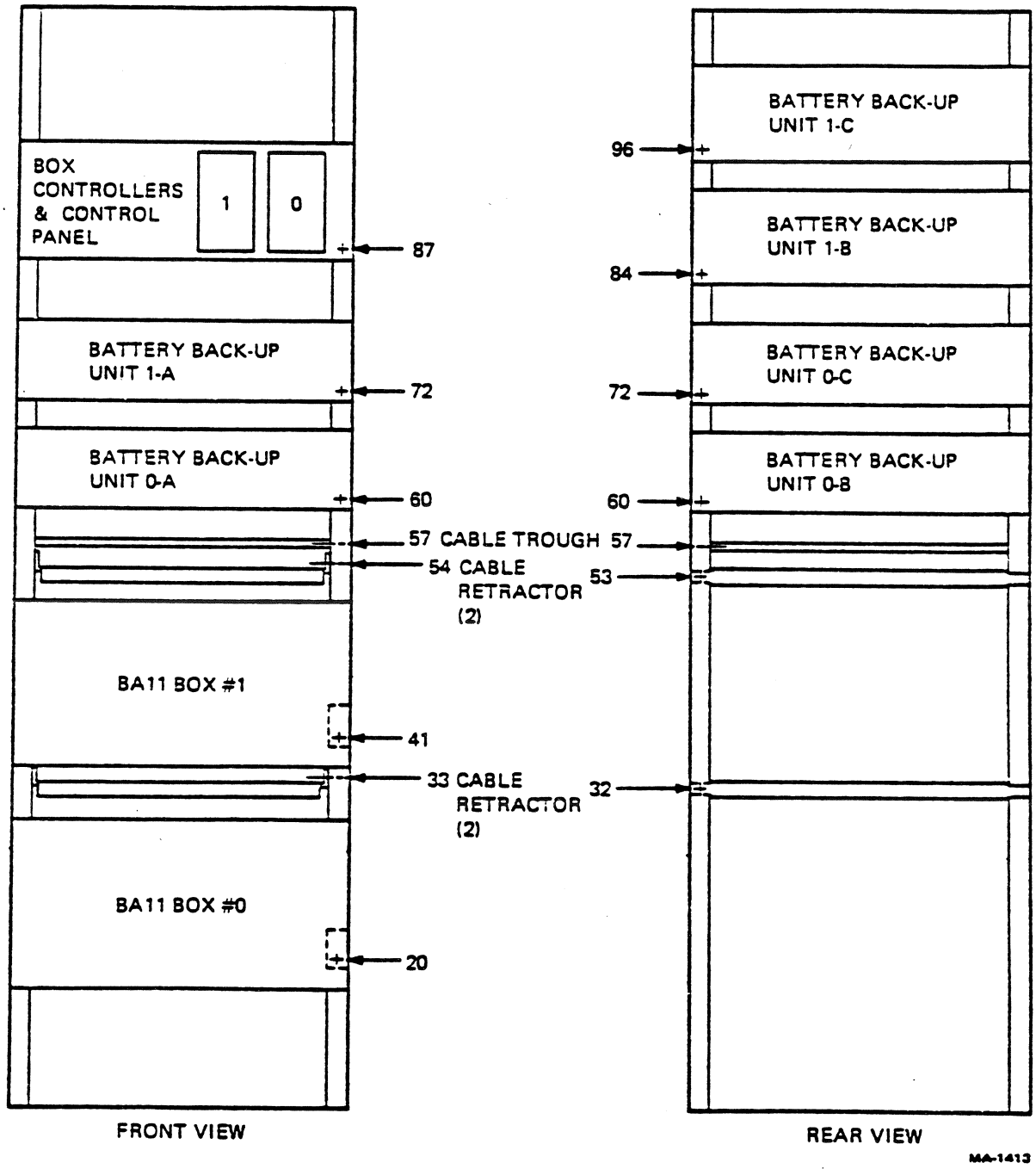


Figure 5-17 Memory Cabinet Configuration

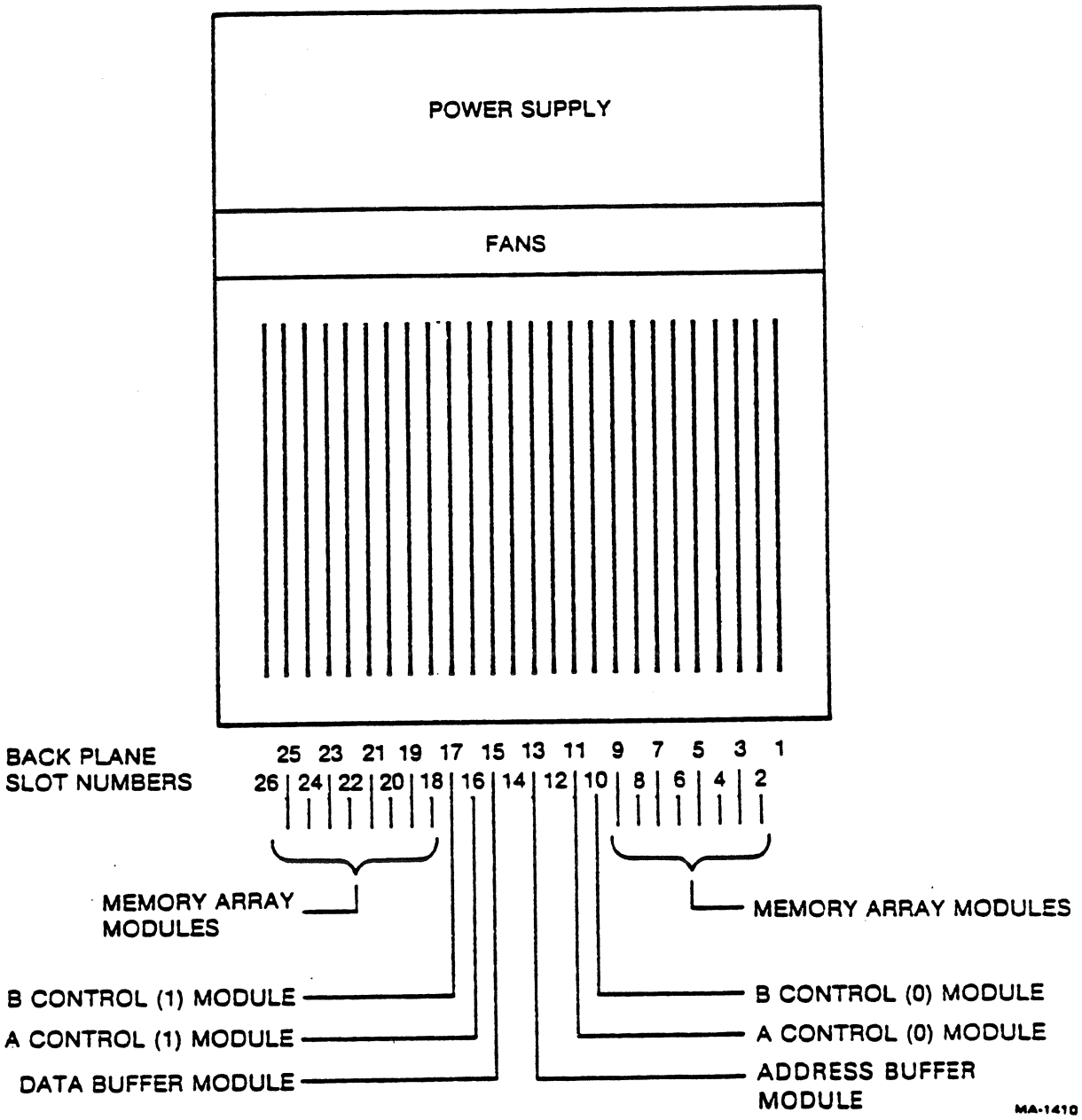


Figure 5-19 Module Locator

RESERVED FOR FUTURE USE

mk11 memory system

ACTUAL MEM. BUS ADDRESS

BOX STARTING ADDRESS (OCTAL) _____

CONTROL & STATUS REGISTER STARTING ADDRESS ☐ 1607721 ☐ 177721

CK FOR 11/70 CPU

INTERLEAVED ☐ EXT 2 WAY ☐ EXT 4 WAY | RESPONDS TO A=2: _____ A=3: _____

MEMORY SYSTEM SERIAL NO. _____ | DATE INSTALLED _____

SLOT	MODULE TYPE	MODULE GROUP SER. NO.	MODULE FUNCTION
01	NOT USED		
02	M7984 –		+14 STORAGE ARRAY
03	M7984 –		+12 STORAGE ARRAY
04	M7984 –		+10 STORAGE ARRAY
05	M7984 –		+8 STORAGE ARRAY
06	M7984 –		+6 STORAGE ARRAY
07	M7984 –		+4 STORAGE ARRAY
08	M7984 –		+2 STORAGE ARRAY
09	M7984 –		+0 STORAGE ARRAY
10	M8161		0 CONTROL B (DATA & ECC)
11	M8160		0 CONTROL A (TIMING & CONT.)
12	NOT USED		
13	M8158		ADDRESS BUFFER
14	NOT USED		
15	M8159		DATA BUFFER
16	M8160		1 CONTROL A (TIMING & CONT.)
17	M8161		1 CONTROL B (DATA & ECC)
18	M7984 –		+1 STORAGE ARRAY
19	M7984 –		+3 STORAGE ARRAY
20	M7984 –		+5 STORAGE ARRAY
21	M7984 –		+7 STORAGE ARRAY
22	M7984 –		+9 STORAGE ARRAY
23	M7984 –		+11 STORAGE ARRAY
24	M7984 –		+13 STORAGE ARRAY
25	M7984 –		+15 STORAGE ARRAY
26	NOT USED		

M8 0111

Figure 5-20 Memory Box Decal

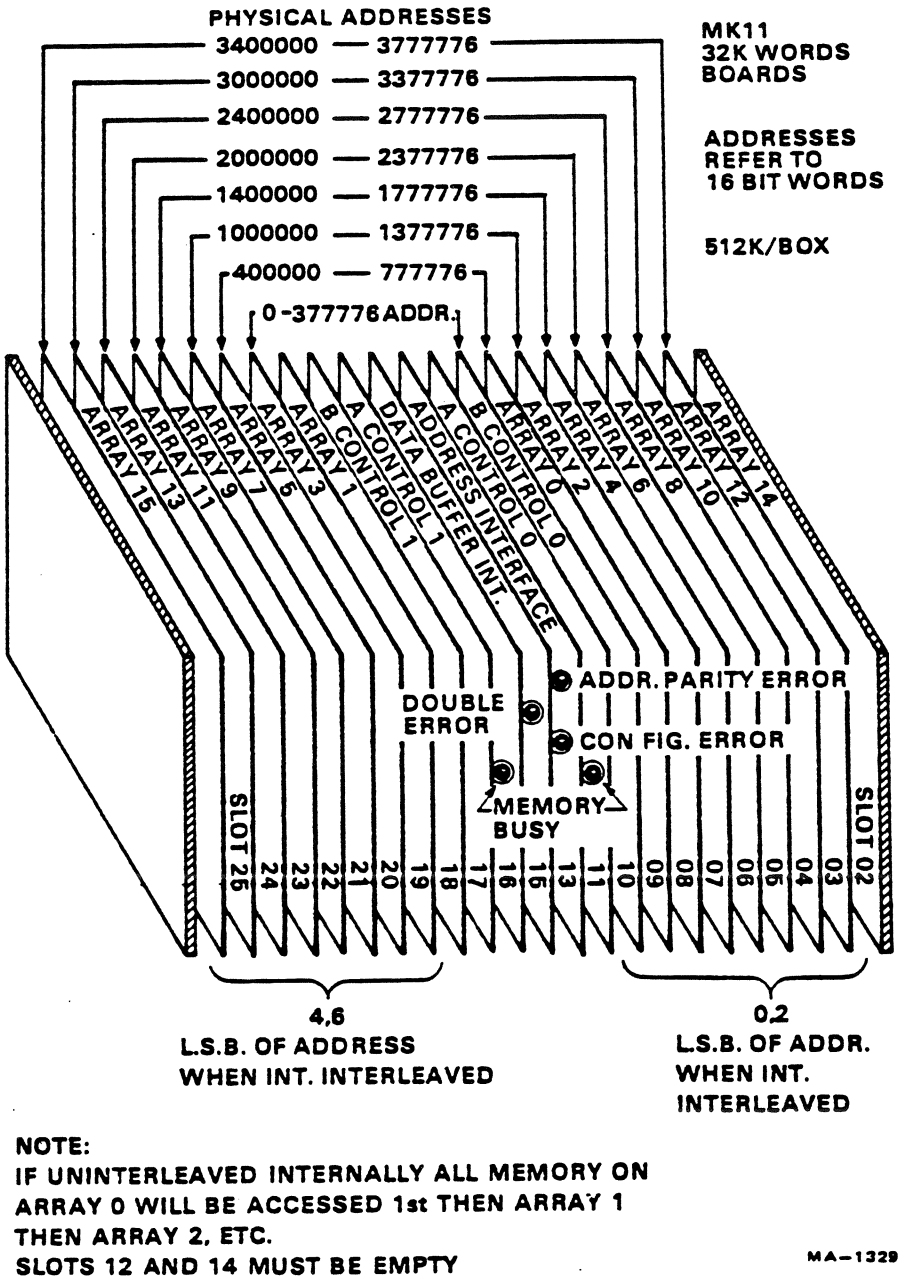
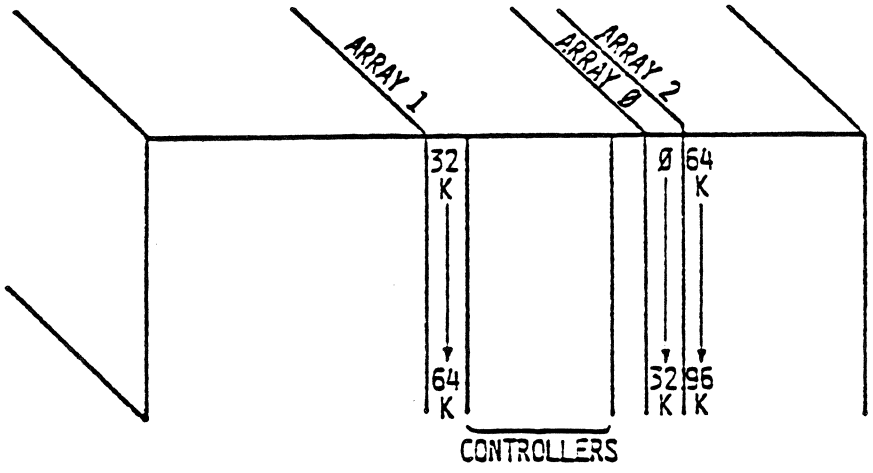


Figure 5-27 Relating an Array Number to a Physical Address (PADD)

MK11 - UNINTERLEAVED



MK11 - INTERNALLY INTERLEAVED

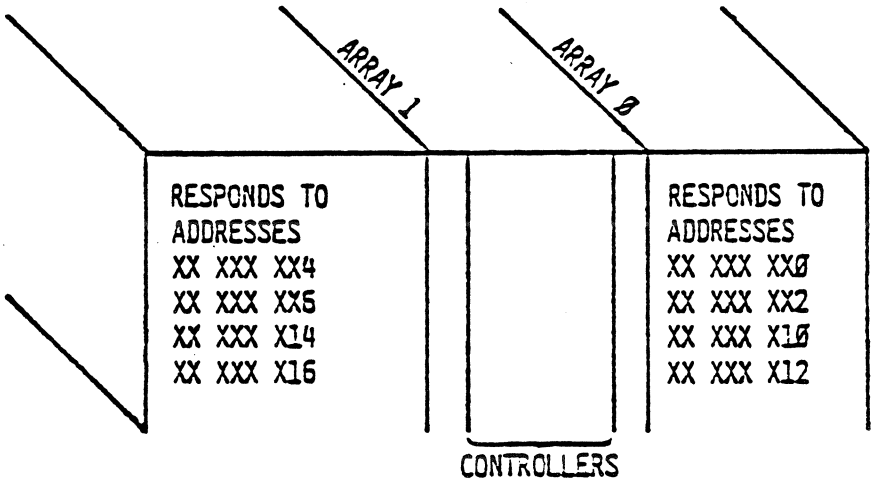


Figure 5-28 Address Allocations, Internally Interleaved

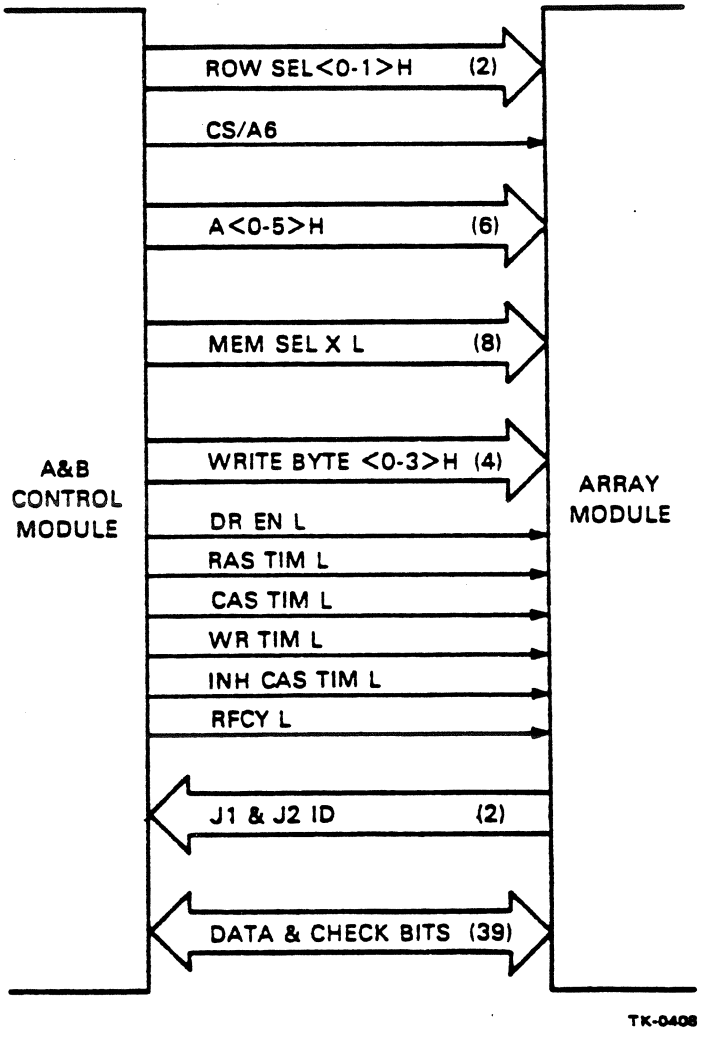


Figure 5-22 Internal Array Tri-State Bus

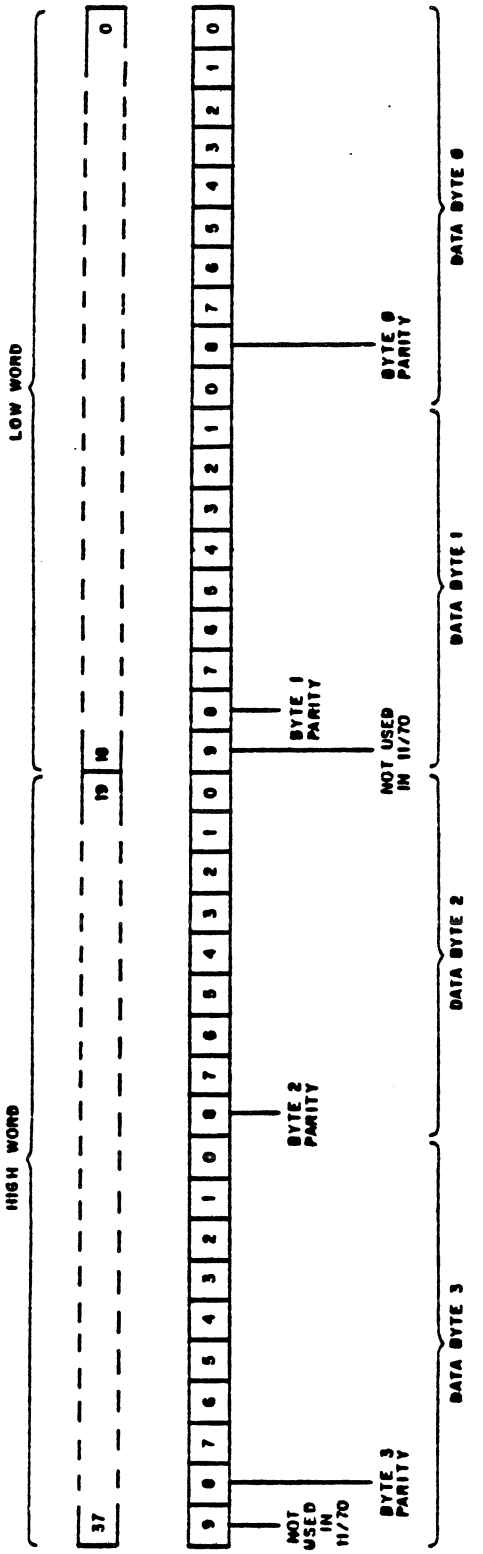


Figure 5-23 Data Format

Table 5-2

Numeral	Option Includes
MK11-BD (for 230 Vac)	Same as MK11-BC but for 230 Vac operation
MK11-BA (for 115 Vac)	Main memory bus cable B BA11 memory box* with 64K words MOS memory Box controller and cables Three battery backup units and cables
MK11-BB (for 230 Vac)	Same as MK11-BA but for 230 Vac operation
MK11-BE	64K word expansion (2 MS11-KE)
MK11-BF	256K word expansion (8 MS11-KE)
MK11-BG	MK11-BA with 512K words MOS memory
MK11-BH	Same as MK11-BG but for 230 Vac operation
MK11-BY	MK11-BA without cables, box controller, or battery backup units
MK11-BZ	MK11-BB without cables, box controller, or battery backup units
MS11-KE	32K words MOS memory array module (4K MOS chips)

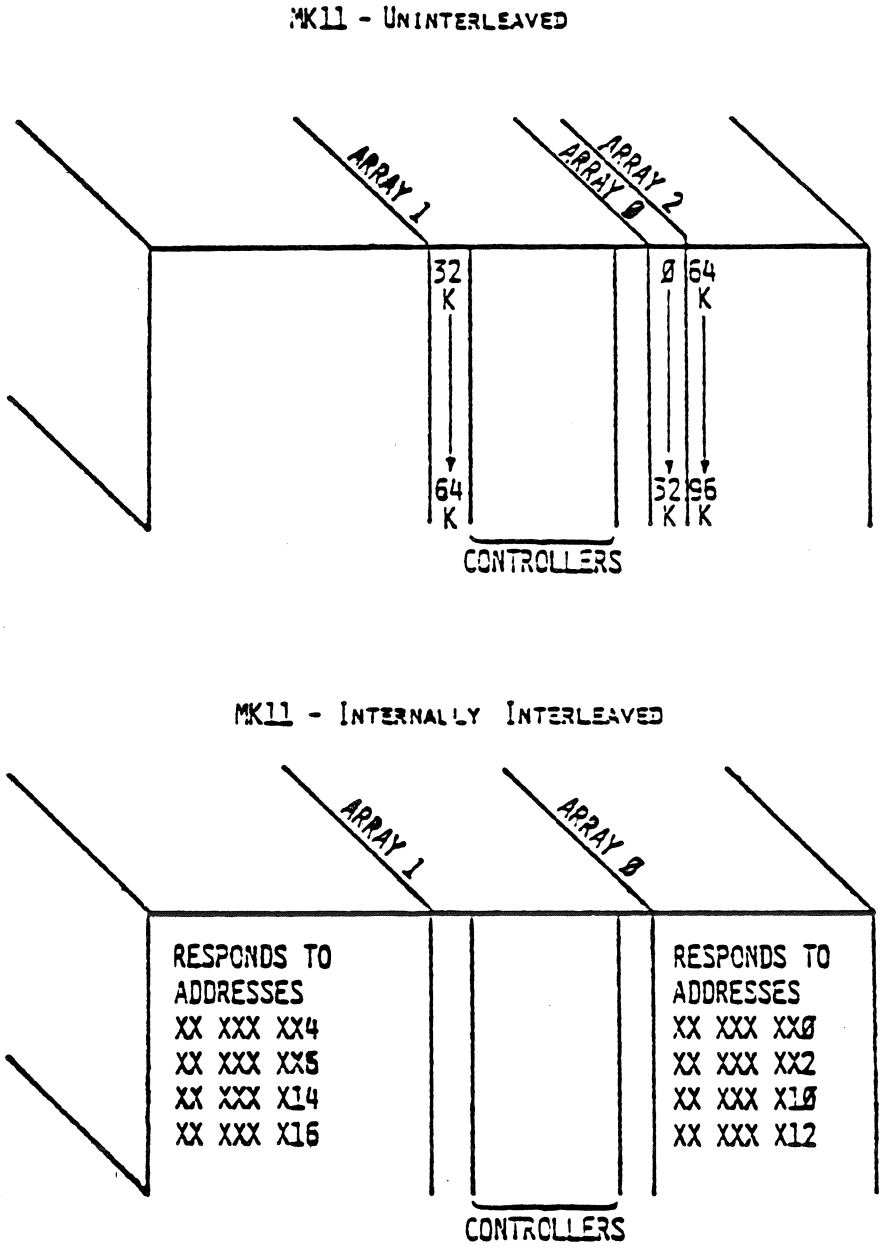


Figure 5-28 Address Allocations, Internally Interleaved

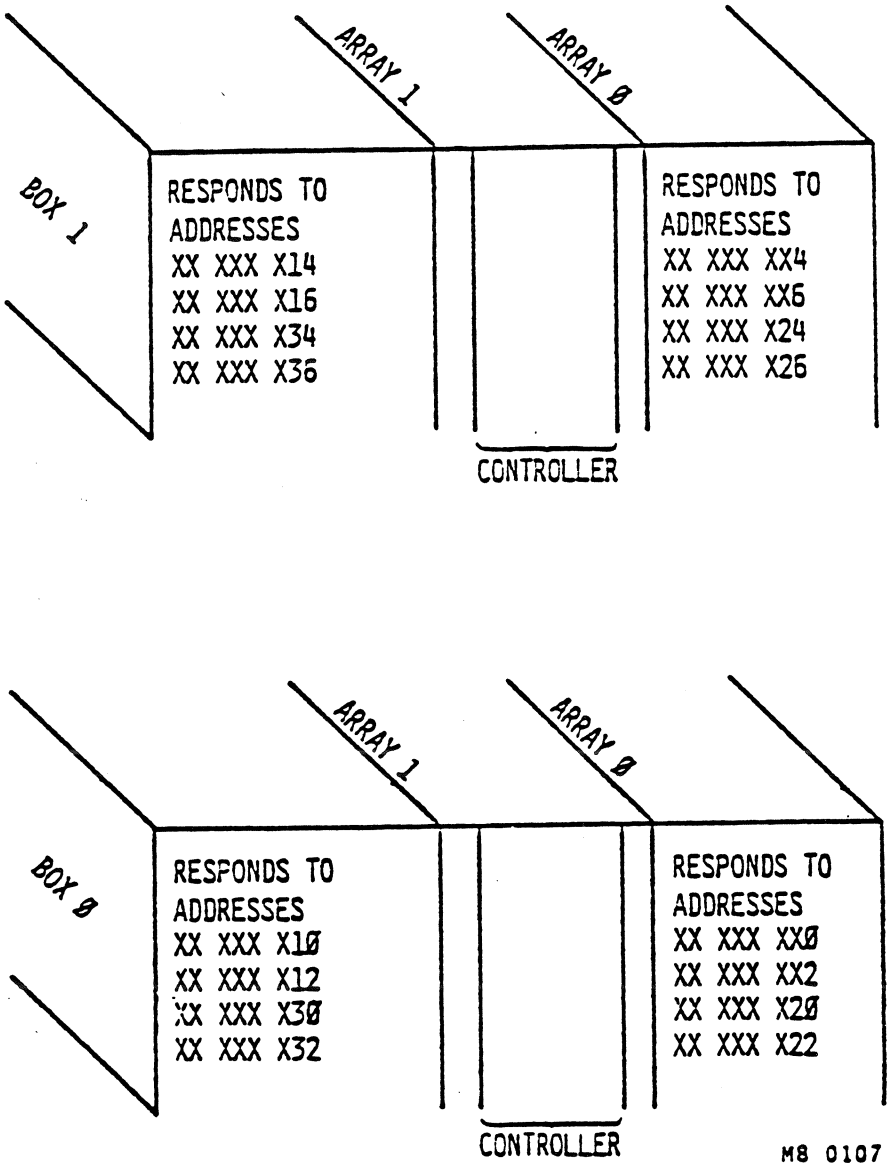


Figure 5-29 Internally and Externally Interleaved

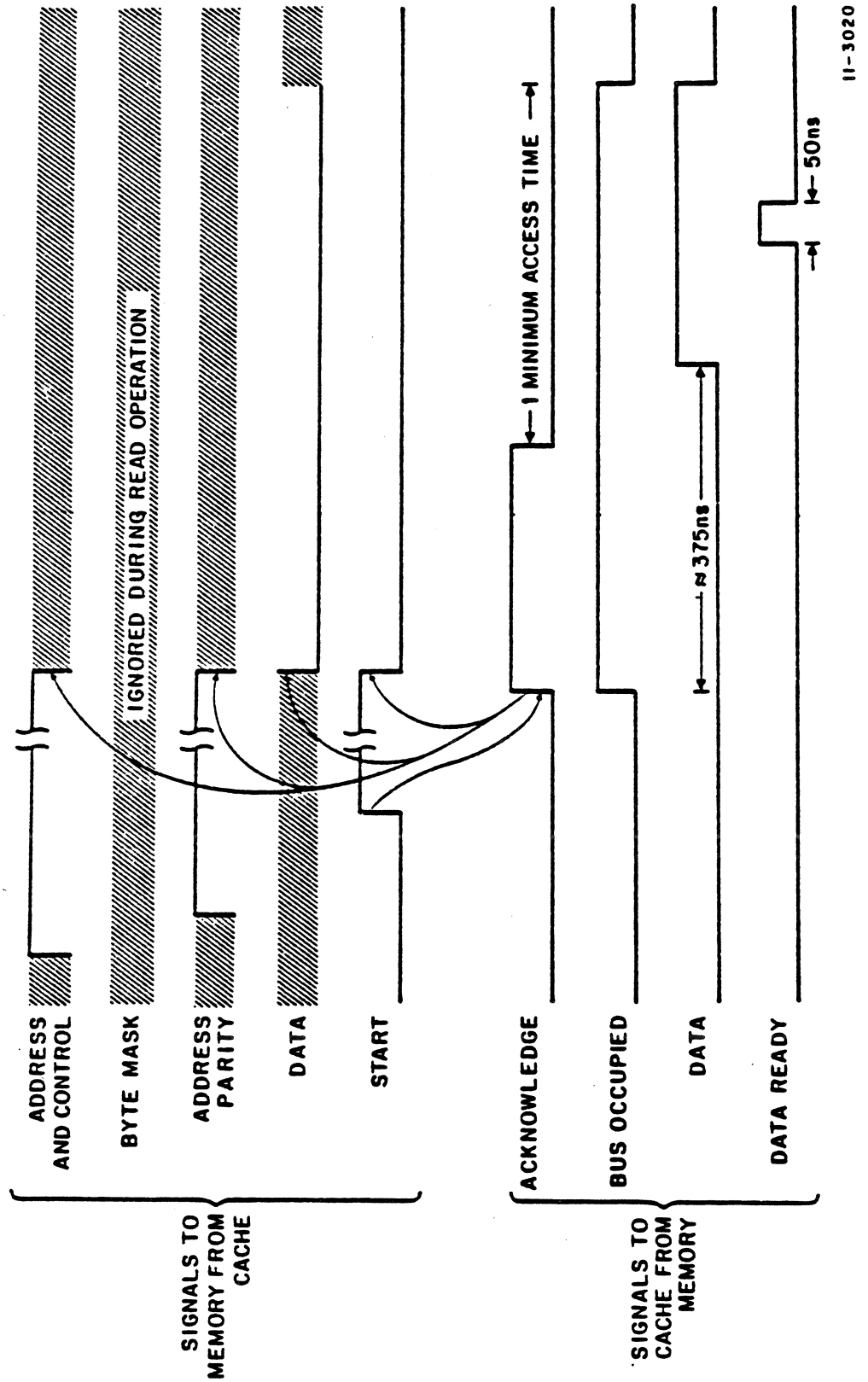


Figure 5-34 Read Operation - Main Memory Bus Protocol

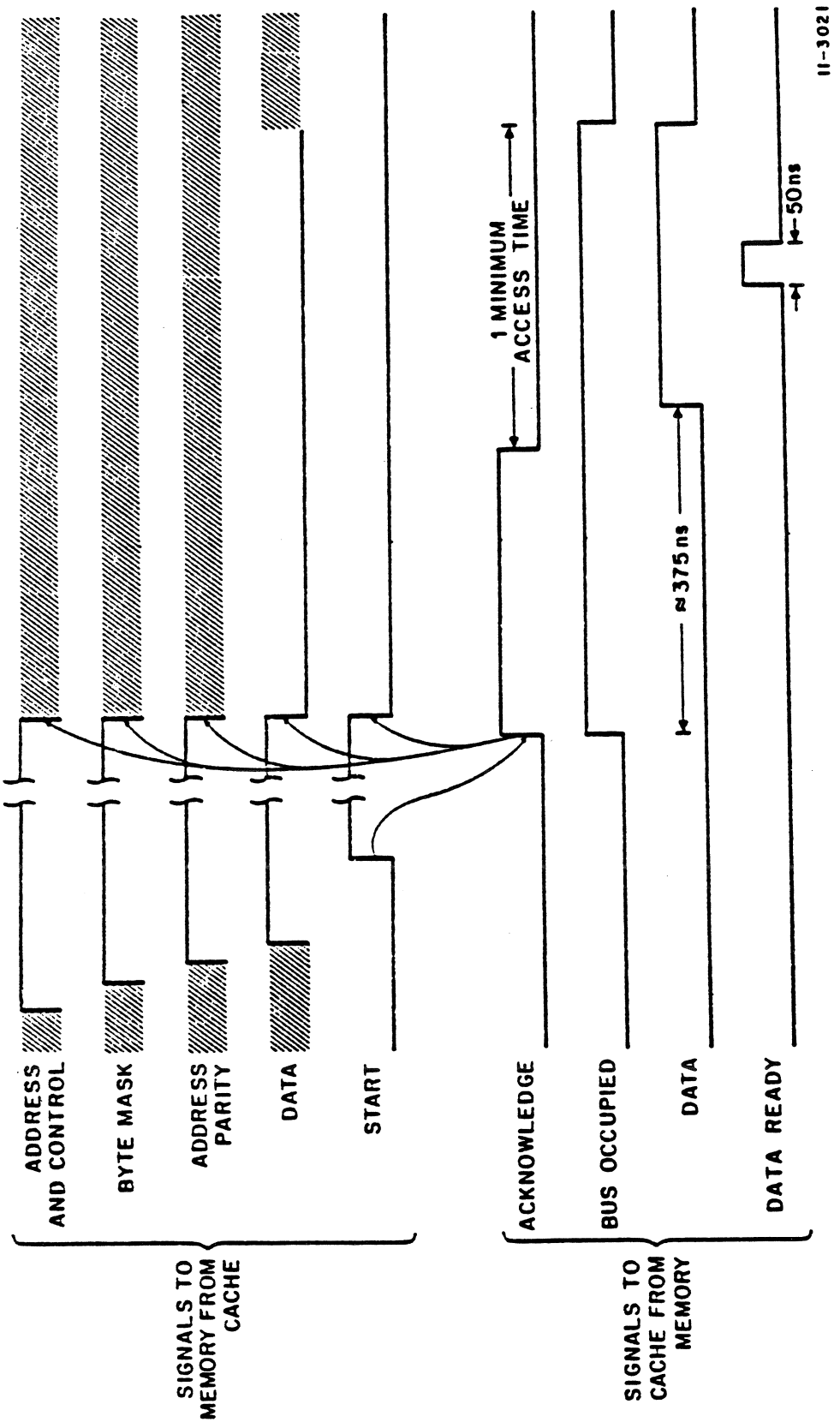


Figure 5-35 Exchange Operation Main Memory Bus Protocol

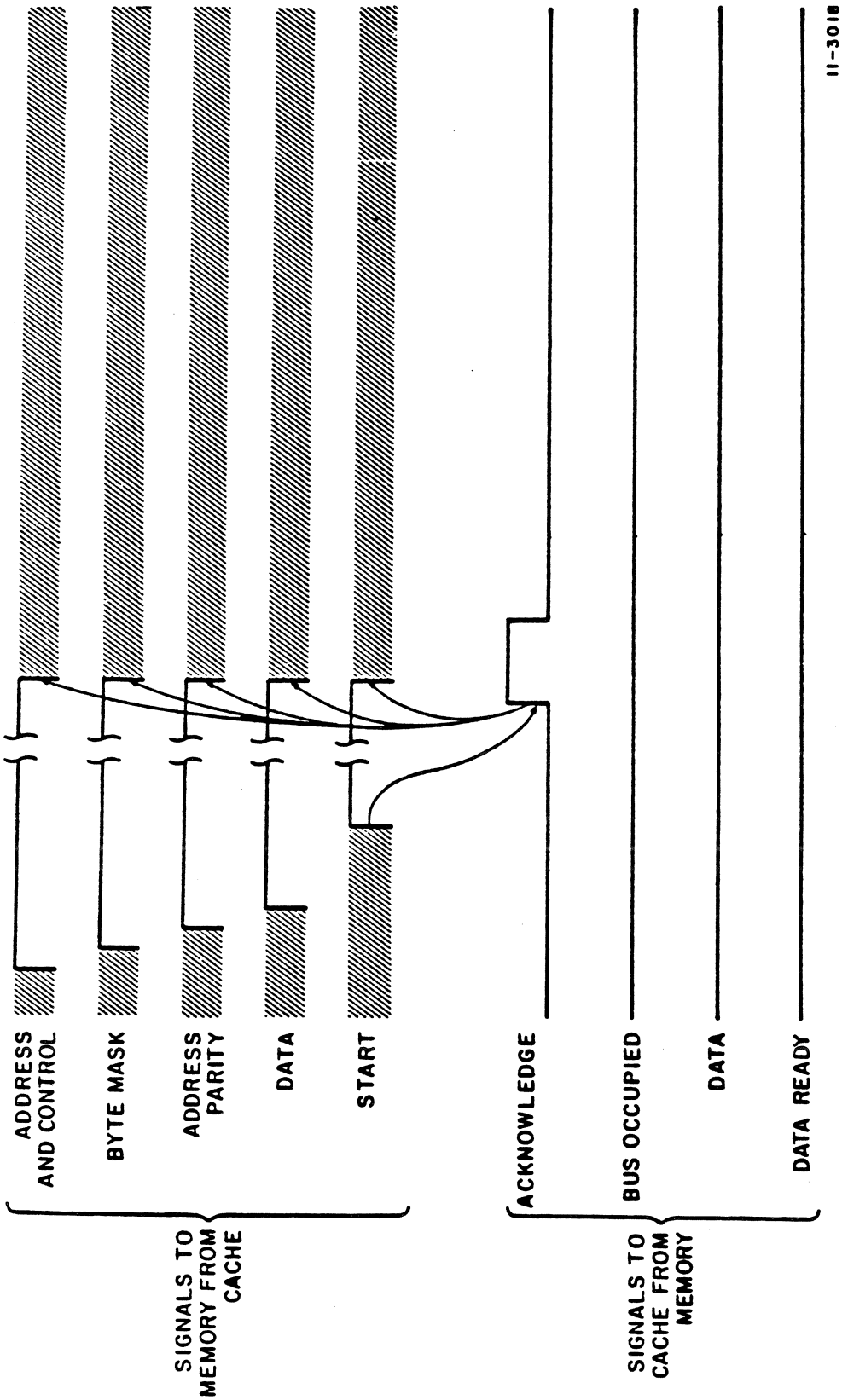


Figure 5-36 Write Operation Main Memory Bus Protocol

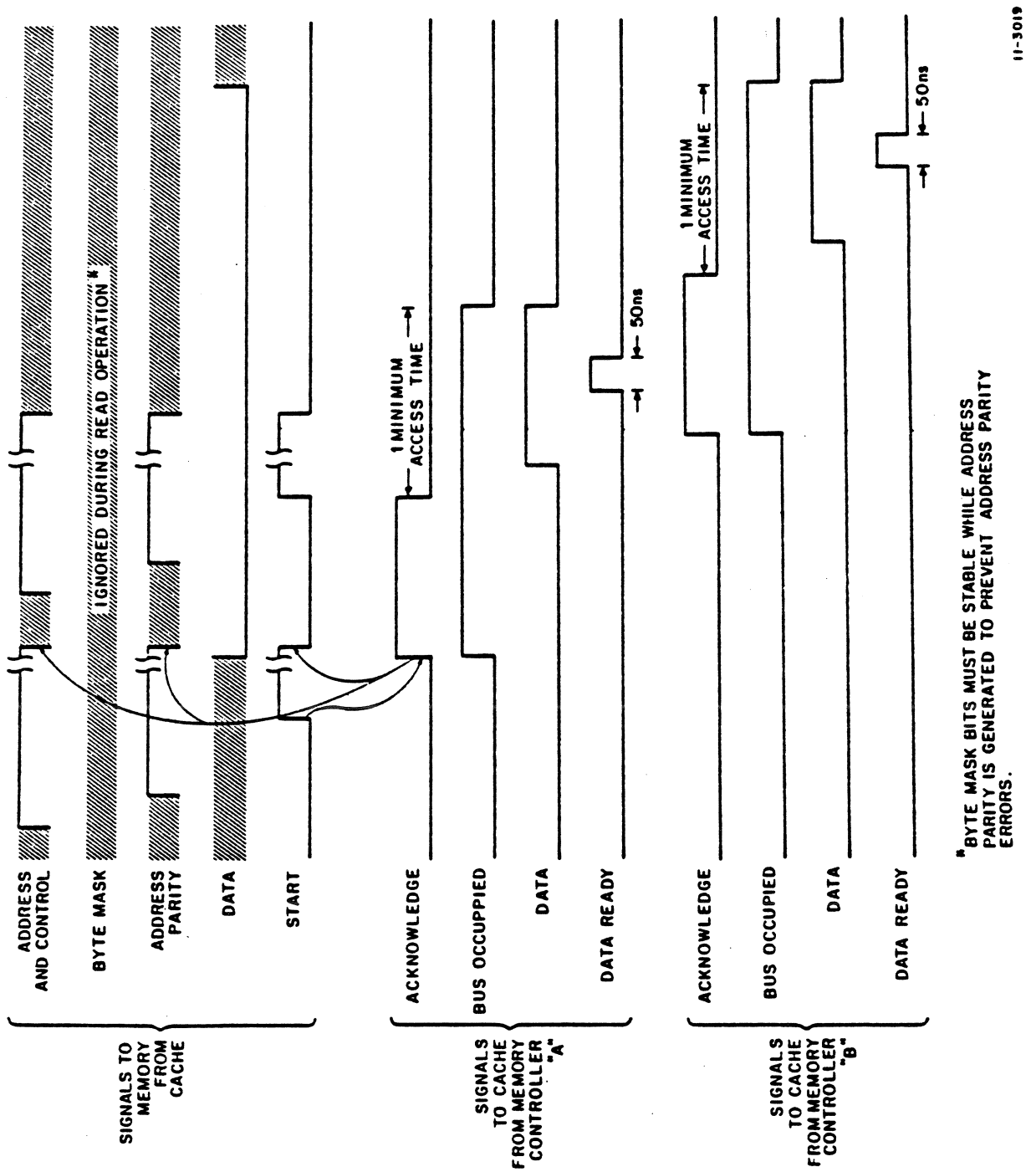


Figure 5-37 Read Overlapped by Read - Main Memory Bus Protocol

TROUBLESHOOTING

The most powerful tool available for troubleshooting the memory is the diagnostic program. However, there will be instances when faults in the memory will prevent running the diagnostic or booting the system. A troubleshooting flowchart is included to direct troubleshooting efforts in order to isolate the problem. If the diagnostic program cannot be run, follow the suggestions in the flowchart. Take full advantage of the information given by the various fault and indicator lights.

Sheet 1 of the flowchart seeks problems that prevent booting the system. Indicator lights and switch positions are checked to give clues to the source of the problem. The cabling in this memory is more complex than previous memory systems and should always be suspected as a probable cause of malfunctions. Memory problems also tend to be interactive. In memory systems with more than one memory box, concentrate troubleshooting efforts to one box at a time. Confusion can be minimized by uninterleaving the boxes to separate them. Turn all the external interleave switches to Ø and select the proper starting addresses as shown on the control panel decal. Boxes can be removed from the system simply by switching the box off-line at the box controller (except for the memory box which contains the bus terminators).

On sheet 3 of the flowchart, the fault has been isolated to one memory box. If the diagnostic still does not run at this point, the switch register can be used to test for the location of the error. When corrections to the system allow running the diagnostic program, run the program to locate any other errors, then reconnect all the boxes to the system and run the diagnostic again.

TEST PROCEDURE FOR ECC INITIALIZATION AND BOOT CHARACTERISTICS

The following procedure checks the memory's ECC initialization which writes all Ø data patterns into the arrays after a power-up boot operation. If power is lost to the memory, the data will remain in the memory for at least 5 minutes while the backup batteries supply power to refresh the MOS chips. Upon recovery from a power fail, the memory is not initialized, but retains the data stored previous to the failure. Perform this procedure if the memory fails the battery backup test in the field service mode (command 7) or if the batteries or power supply modules are installed or replaced.

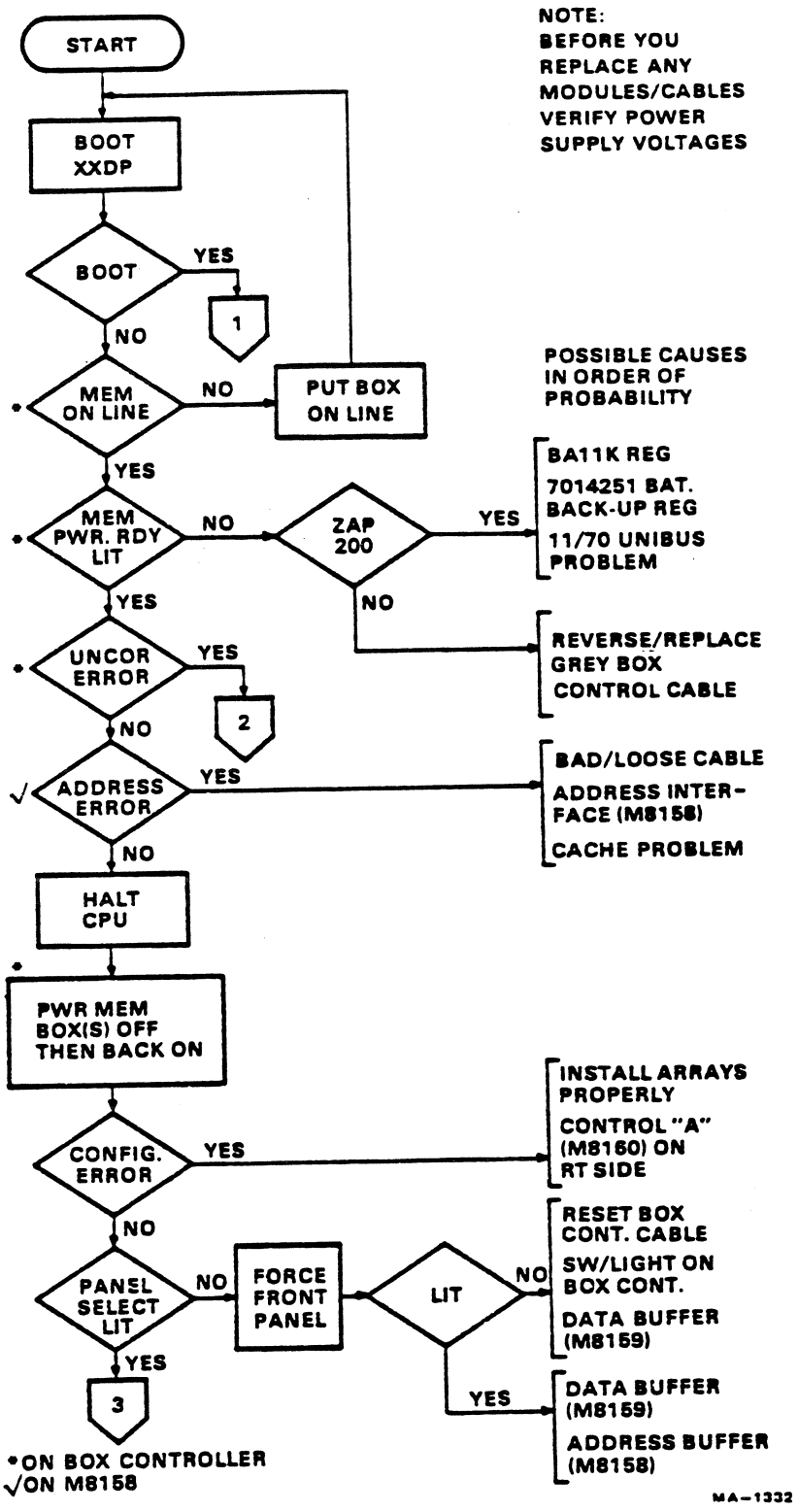


Figure 5-48 MK11 Troubleshooting Chart (Sheet 1 of 3)

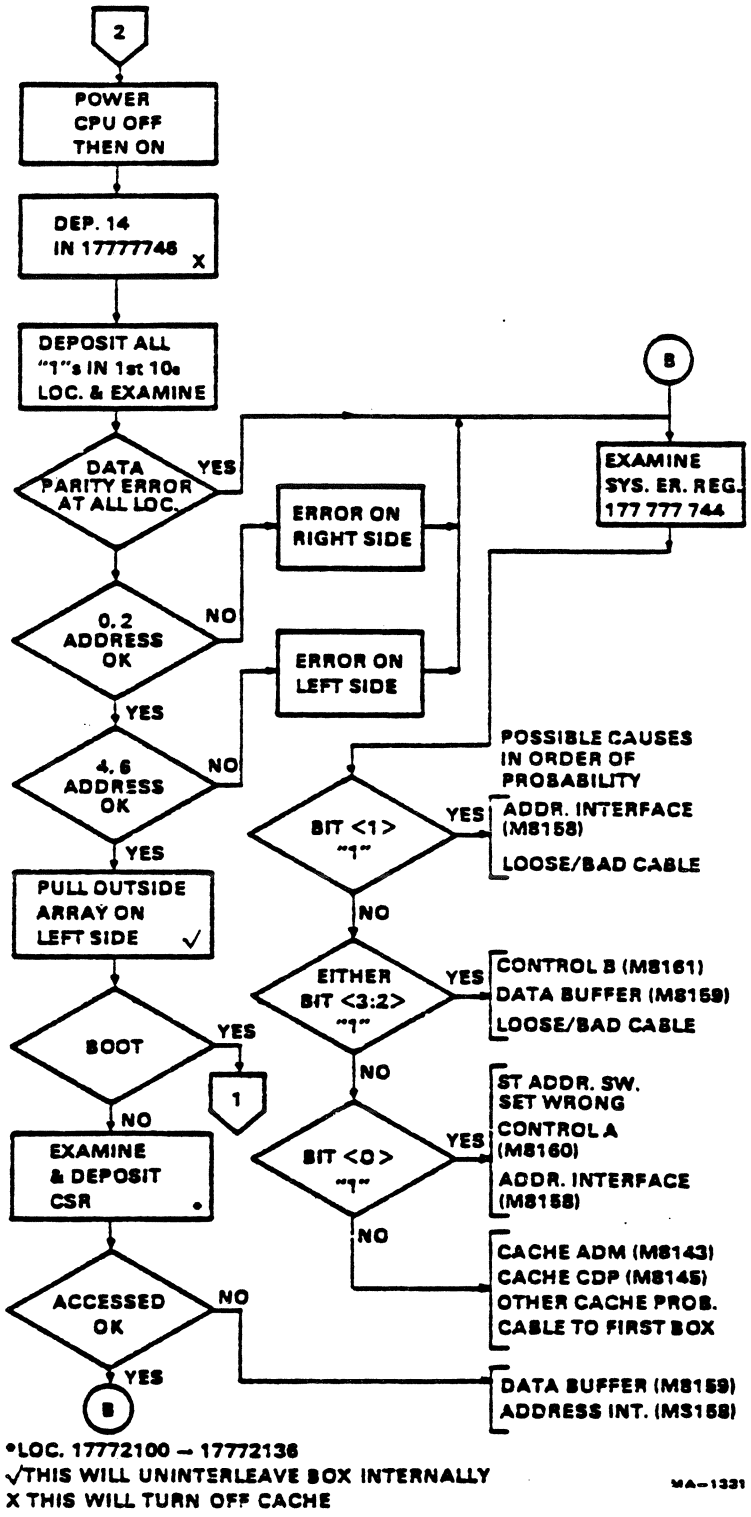


Figure 5-49 MK11 troubleshooting Chart (Sheet 3 of 3)

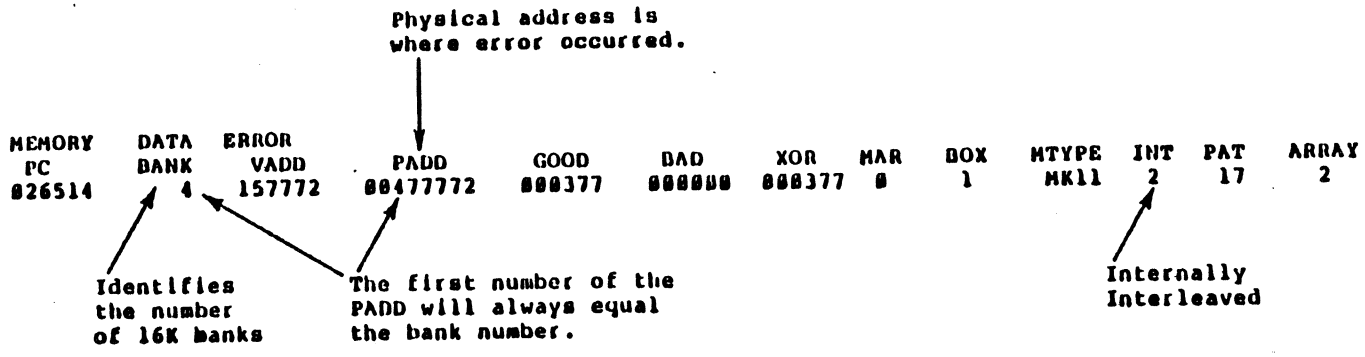


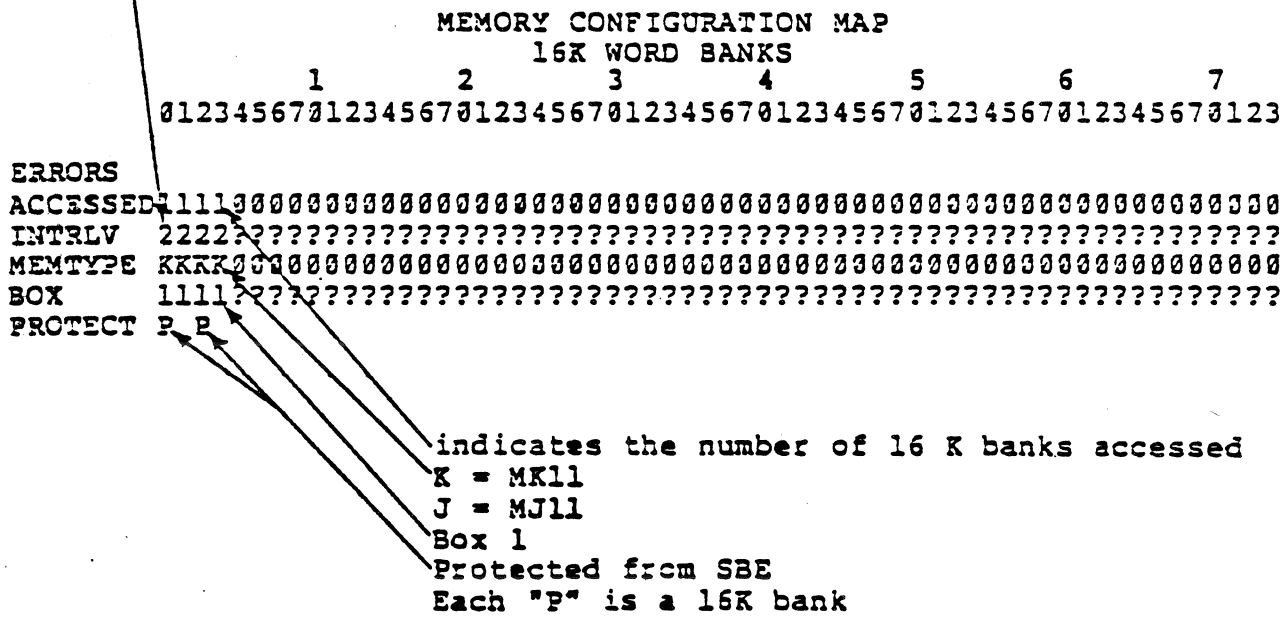
Figure 5-50 MK11 Error Printout

MK11 Diagnostic Configuration Map

Example 1

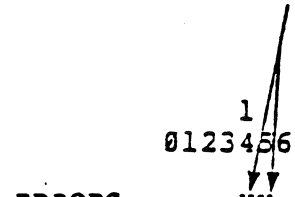
- 64K of memory
- internally interleaved
- no errors

2 = Internally Interleaved



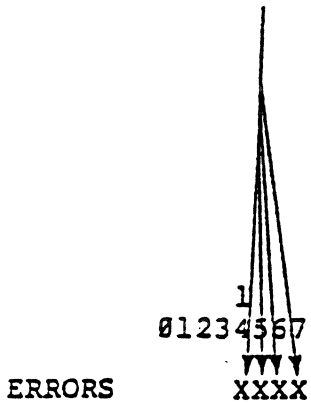
Example 2

- 96K of memory
- no internal interleaving
- with errors



```
                    MEMORY CONFIGURATION MAP
                         16K WORD BANKS
          1         2         3         4         5         6         7
0123456701234567012345670123456701234567012345670123456701234567012345670123
ERRORS       XX
ACCESSED1111110000000000000000000000C000000000000000000000000000000000C00
INTRLV  000000?????????????????????????????????????????????????????????????
MEMTYPE KKKKKK000000600000000000000000000000000000000000000000000000C0C00
BOX     111111?????????????????????????????????????????????????????????????
PROTECT P P
```

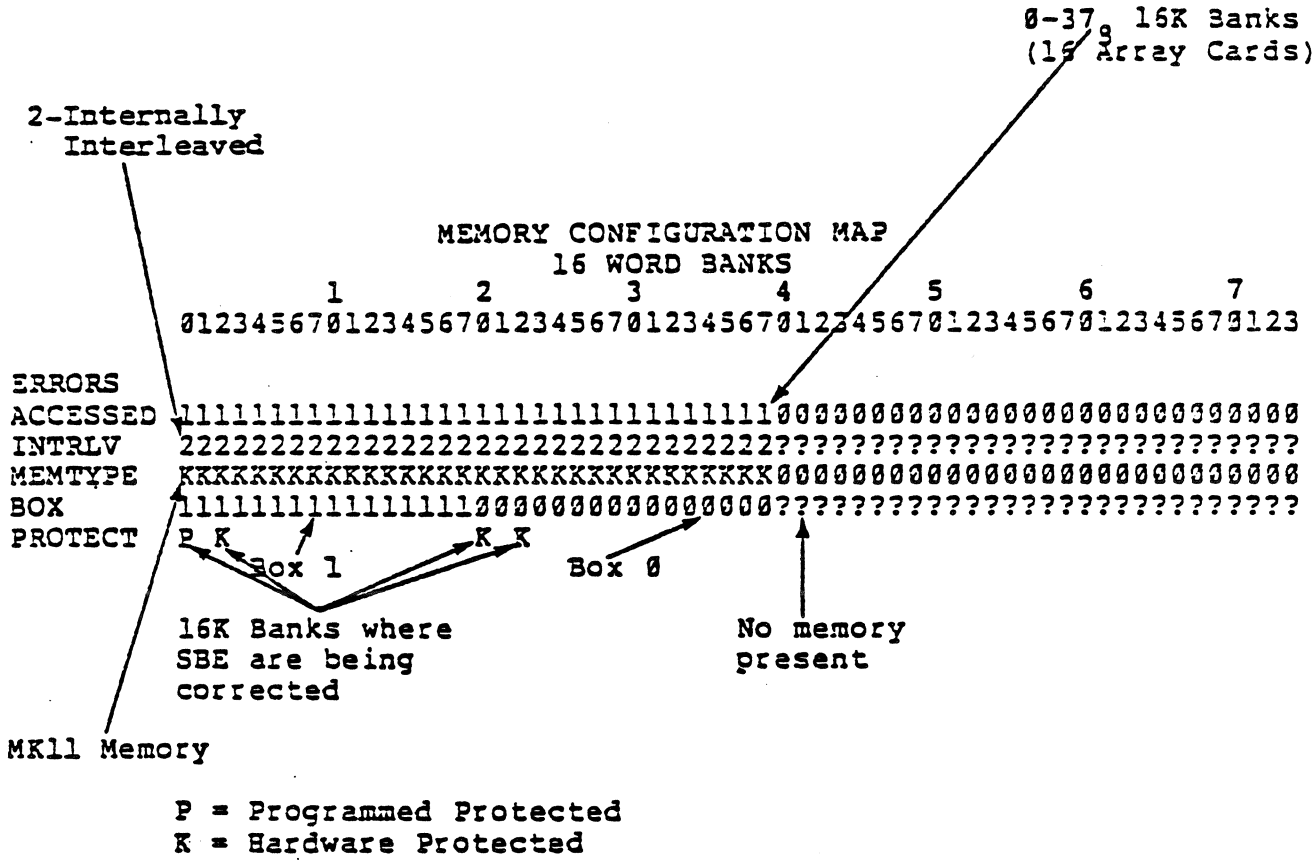
- 128 K of memory
- internal interleaving
- with errors



```
                    MEMORY CONFIGURATION MAP
                         16K WORD BANKS
          1         2         3         4         5         6         7
0123456701234567012345670123456701234567012345670123456701234567012345670123
ERRORS       XXXX
ACCESSED11111111000000000000000000000000000000000000000000000000000000000
INTRLV  22222222?????????????????????????????????????????????????????????????
MEMTYPE KKKKKKKK000000000000000000000000000000000000000000000000000000000000
BOX     11111111?????????????????????????????????????????????????????????????
PROTECT P P
```

Example 3

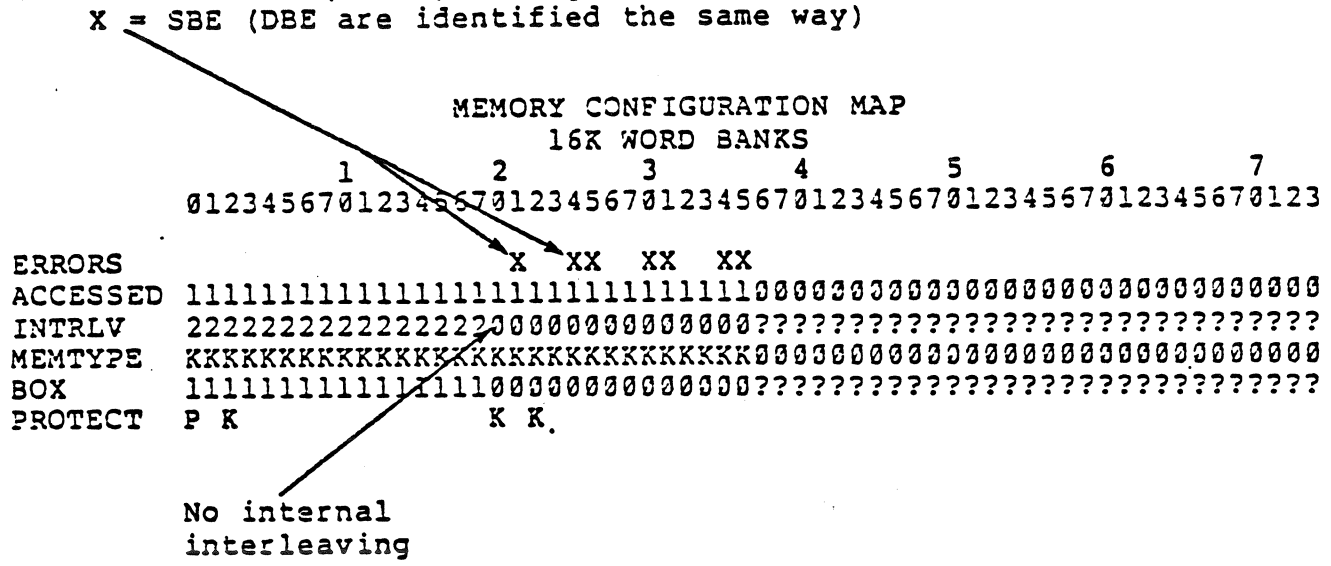
- 512K of memory in two boxes
- internal interleaved
- no errors



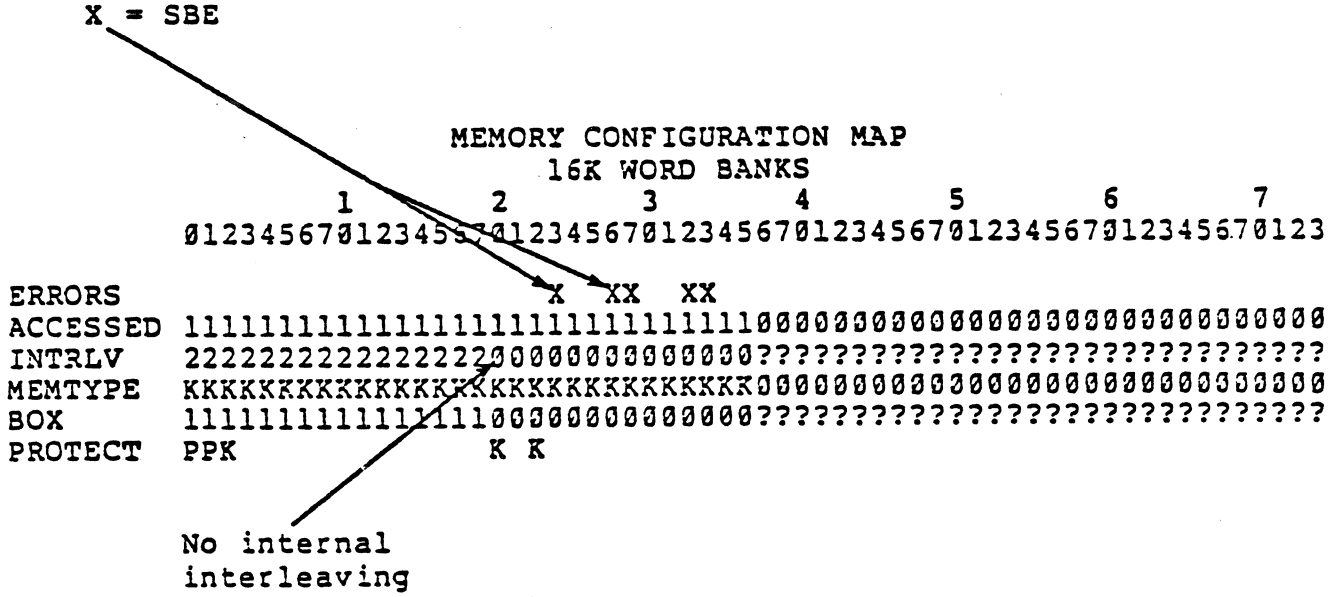
Example 4

- 512K of memory in two boxes
- internally interleaved in box 1
- with errors

Bad Control "B" (M8161) on right side
X = SBE (DBE are identified the same way)



Bad control "B" (M8161) on left side
X = SBE



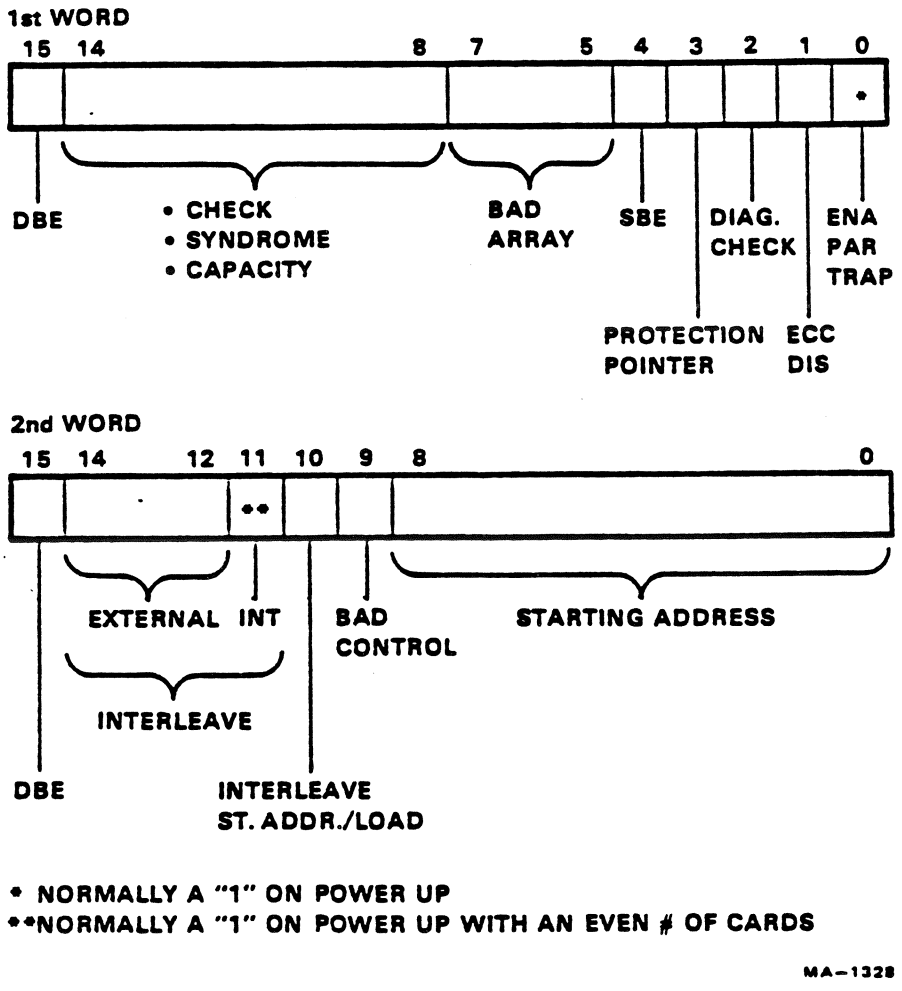


Figure 5-52 Control and Status Registers (CSRs)

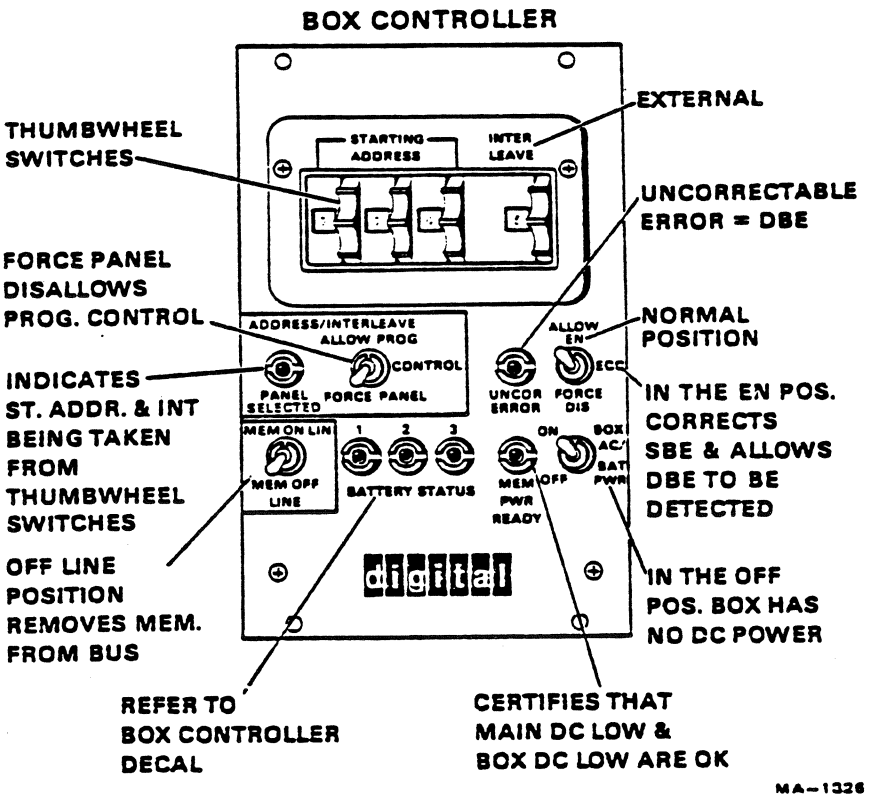


Figure 5-53 Box Controller

Checkout Procedure for Box ECC Initialization

1. Power box on, ensure battery backup (H775) units are on.
2. Examine locations $0\text{-}20_8$ for all 0s.
3. Deposit all 1s in locations $0\text{-}20_8$.
4. Examine locations $0\text{-}20_8$ for all 1s.
5. Place HALT ENABLE to HALT. (HALT switch will remain in HALT position for remainder of test.)
6. Turn memory box breaker off, then on.
7. Examine locations $0\text{-}20_8$ for all 1s.
8. Turn memory box breaker off; turn battery backup 1-A off.
9. Turn memory box breaker on; turn battery backup 1-A on.
10. Press START switch on PDP-11/70; load address 0.
11. Examine locations $0\text{-}20_8$ with data below:

Address	
	0 = 0s
	2 = 0s
	4 = 1s

6 = 1s
10 = 0s
12 = 0s
14 = 1s
16 = 1s

12. Deposit all 1s in locations $0-20_8$.

13. Turn memory box breaker to off position; turn battery backup 1-B to off position.

14. Turn memory box breaker to on position, turn battery backup 1-B to on position.

15. Press START switch on PDP-11/70; load address 0.

16. Examine locations $0-20_8$ with data below:

Address	
	0 = 1s
	2 = 1s
	4 = 0s
	6 = 0s
	10 = 1s
	12 = 1s
	14 = 0s
	16 = 0s

17. Deposit all 1s in locations $0-20_8$.

18. Turn memory box breaker to off position; turn battery backup 1-C to off position.

19. Turn memory box breaker to on position; turn battery backup 1-C to on position.

20. Press START switch on PDP-11/70; load address 0.

21. Examine locations $0-20_8$ and check that all locations are 0s.

Checkout Procedure for Box BOOT Operation

1. Power box on, ensure battery backup (H775) units are on.

2. Get a 50_8 in the PDP-11/70 switches.

3. Place the HALT ENABLE switch to the ENABLE position. (HALT switch will remain in the ENABLE position for the remainder of the test.)

4. Turn memory box breaker off, then on.

5. Check physical address of the PDP-11/70, and ensure that the processor stopped at address 2_8.

6. Turn memory box breaker off; turn battery backup 1-A off.

7. Turn memory box breaker on; turn battery backup 1-A on.

8. Ensure that the PDP-11/70 BOOTS, and that the processor stops at the correct physical address, (50_8) loaded in the switches).

9. Turn memory box breaker off; turn battery backup 1-B off.

10. Turn memory box breaker on; turn battery backup 1-B on.

11. Ensure that the same results happen as in step 8.

12. Turn memory box breaker off; turn battery backup 1-C off.

13. Turn memory box breaker on; turn battery backup 1-C on.

14. Ensure that the same results happen as in step 8.

digital™

Made in U.S.A.