EK-VAXV3-HB-001

VAX Maintenance Handbook

VAX-11/750

1983 Edition

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, March 1983

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Printed in U.S.A.

The manuscript for this book was created on a Word Processing System. Book production was done by Educational Services Development and Publishing in North Lawrence, MA.

CONTENTS

CHAPTER 3 MODULES AND GATE ARRAYS

CHAPTER 4 CONFIGURATION AND CABLING

CHAPTER 5 BOOTSTRAPPING AND OPERATION

CHAPTER 6 RDM AND MICRODIAGNOSTICS

CHAPTER 7 BLOCK DIAGRAMS

CHAPTER 8 CMI, UNIBUS, AND MASSBUS

CHAPTER 9 TROUBLESHOOTING AIDS

CHAPTER 1
INTRODUCTION

INTRODUCTION

This handbook is a summary of VAX 11/780 system maintenance information. It is intended to serve as a single source reference for initial field service, and for engineering, manufacturing, and training personnel.

The material complements detailed information available in hardware and service manuals and in reference print sets. It contains tables, diagrams, and procedures not easily found in manuals with the VAX-11/780 system. It includes illustrations and information not available in any printed form.

For further information, refer to the preface to VAX-11/780 and VAX-11/750 Architecture Library, and to labels included in the computer that list related hardware manuals.

Hard copy documents may be ordered through the nearest Digital sales office. The Accessories and Supplies Group outlines documentation products primarily in the following references.

Customers and DECs:

```
Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532

Attn:  Publishing and Circulation Services, NJO
       Customer Services Section
```

Digital personnel:

```
Digital Equipment Corporation
12 Crosby Road
Bedford, MA 01810

Attn:  Publishing and Circulation Services, MR3
```

Technical descriptions and service manuals are also available on microfiche. For information on microfiche services (including performance print sets and diagnostic listings), contact:

```
Digital Equipment Corporation
Micropublishing Systems, BR/B42
12 Crosby Drive
Bedford, MA 01730
```

VAX-11/750 SYSTEM HARDWARE MANUALS

Title	Document Number
VAX-11 /750 Central Processor Unit Technical Description	EK-KA750-TD
VAX-11 /750 Mainframe Interconnect Technical Description	EK-SBI11-TD
VAX 11/750 Memory System Technical Description	EK-M8637-TD
DR750 Unibus Adapter Technical Description	EK-DR750-TD
KA750 General Purpose Interface Technical Description	EK-KA750A-TD
11/750 Floating Point Accelerator Technical Field Guide	EK-KE11A-TF
VAX 11/750 CPU/ Power System Technical Information	EK-KA750-TI
VAX Architecture Handbook	EK-VAXAR-HB
VAX Hardware Handbook	EK-VAXHB-HB
VAX Software Handbook	EK-VAXSH-HB
VAX Unibus Bit Assignments in Hex	AA-2021F-TF
VAX-11/750 Installation Guide Preplanning User Manual	EK-SVA50-IG
VAX-11/750 Owner Guide	EK-[illegible]-OG
VAX Maintenance Handbook, VAX systems	EK-VAXMH-HB
VAX Diagnostic System Owner's Guide	EK-VDS[illegible]-OG
VAX-11/xx Diagnostic System Software Manual	EK-VAX[illegible]-OM
VAX-11/750 Diagnostic Kit Information Guide	EK-[illegible]-MP
VAX 11/750 Data Error Cell Reference Manual	EK-[illegible]-RM*

*Available only on microfiche

Title	Document No.
Series Introduction to Technical Manuals	EK-KC750-TY
Series Options User Guide	EK-KC750-UG
KC11-A Options Installation Guide	EK-KC750-IT
KC750 Technical Changes Series	EK-KC750-TC
VAX Diagnostic Design Guide	EK-VAXDD-DG
VAX-11/750 Programmable Microprocessor Manual	EK-750VA-PM
VAX-11/750 Writable Microprocessor Diagnostic Guide	EK-750VA-UG
VAX-11/750 Change Notice	EK-750VA-CN
VAX-11/750 Console Map	EK-KC750-TP
VAX-11/750 PA Multiport	EK-[illegible]
VAX-11/750 Technical Data Sheets	EK-KC750-TD
Bit Disposition Data Sheets	EK-[illegible]

Title	Order Number
LA11 Operator's Manual	EK-KLA11-OP
LA12 RT/CX Terminal TU Kit	EK-LA12S-TP
LA120 DECwriter IV Technical Manual	EK-LA120-TM
Peripherals Interconnect (PI/O)	EK-LA11S-IG
LA34 DECscope TU User Guide	EK-VLA34-UG
LA36 DECTEC Guide Addendum	EK-VLA36-AG
LA36 DECwriter II Maintenance Guide	EK-DW11-MM
LA 30 Technical Manual	EK-LA130-TM
LA120 User's Guide	EK-LA120-UG
LA120 User's Guide Addendum	EK-LA12A-UG
LA 78 Operators Maintenance Card	EK LA 78-OC
LA 78 Parts Service Guide	EK LA 78-SV
LA120 Parts Service Guide Addendum	EK-LA12-SV
LA 78 Series Parts Levels Guide	EK-LA 78-PS
LA120 Thirteen Section A Data	EK-LA120-SD
LA120 Diagnostic ROM Card	EK-LA120-CP
VT50 DECscope II Technical Manual	EK-VT50-TM
VT50 DECscope II User's Guide	EK-VT50-UG
VT55 DECscope II Parts Service Guide	EK-VT55-PS
VT50 Serial Communications Equipment Data	EK-VT50-SC
VT50 Card Age Page Size ITU	EK-VT50-TP
VT55-YA DECscope II Installation	EK-VT55-IN
VT55 DECscope Installation Guide	EK-VT55-IG

VAX-11/750 PERIPHERAL MANUALS (CONT)

Title	Document Number
RA60/81 Moving-Head Subsystem Owner's Manual	EK-RA60/81-OM
RA60/81 Technical Description Manual	EK-RA60/81-TD
RA60/81 Quick Reference Card	EK-RA60/81-QR
TK50/70 Site Adaptation Guide	EK-TK50/70-SA
RA60/81 FCS Operating/Service Addition	EK-RA60/81-UP
RA60/81 Customer Care Guide	EK-RA60/81-CC
RX50 Disk Drive FRU	EK-RX50/70-FR
RX50 Subsystem Technical Description Manual	EK-RX50-TD
RX50 Controller FRU	EK-RX50-FR
RA81 Owner's Manual	EK-RA81-OM
RA81 Disk Drive Maintenance Manual	EK-RA81-MM
RX33 Disk Drive FRU	EK-RX33-FR
RF30/31 Fixed Disks	EK-RF31-IN
RA71/72 Disk Subsystem Service Manual	EK-RA72-SV
RF31/72 Customer Configuration Card	EK-RF31-CC
RF31/72 PN Worksheets	EK-RF31-WS
TU81 Subsystem Technical Manual	EK-TU81-TM
TU81 Subsystem Owner's Guide	EK-TU81-OG
TU81 Subsystem Customer Diagnostic Guide	EK-TU81-CG
TU81 Pocket Service Guide	EK-TU81-PS
TU81+E Magnetic Tape Subsystem FRU	EK-TU81E-FR
TU81 E Subsystem Tape Subsystem IPB	EK-TU81E-IP
TU81+E Magnetic Tape Subsystem IPD	EK-TU81E-IP

Title	Document Number
RP07 Disk Drive Service Manual	EK-RP07-SV
RP,S Disk Subsystem Owner's Guide	EK-RPOSS-OG
RS03 Disk Drive ICD	EK-RS03-IC
RM05 ICD Maintenance	EK-RM05-MG
TU77 Magnetic Tape Transport Manual	EK-TU77-OM
TU78 Maintenance Information Manual V2	EK-TU78-IM
TU77 Installation Update	EK-TU77-IN
TU77 Maintenance Information Update	EK-TU77-MI
TU78 Magnetic Tape Subsystem	EK-TU78-OM
TU78 Magnetic Tape Transport TCU	EK-TU78-TP
TU-77/TU78 Magnetic Tape Transport Exploded Views	EK-TU77B-EP
TU-77/78 PM Maintenance	EK-TU77B-MP
RM02 Disk Drive Technical Description	EK-RM02-TD
RM02 Disk Drive Service Manual	EK-RM02-SM
RM05 Disk Drive Field Guide	EK-RM05-FG
RM05 Disk Drive Parts Guide	EK-RM05-PG
RM80 Disk Drive & Shared Data Package	EK-RM80-IP
RP06 Disk Drive TPS	EK-RP06-TP
RP07/S Maintenance Addendum	EK-RP07-MA
RP07/S Illustrated Parts	EK-RP07-IP
RJ11 J&S Cable Kits	EK-RJ11-IN

VAX-11/780 MAINTENANCE PHILOSOPHY

The VAX-11/780 system maintenance philosophy is the repair of replacement of modules, components, or complete hardware units depending on the type of hardware under test and the level of repair that applies to each unit.

Quick fault isolation and location of replacing units during on-line execution of diagnostics can be accomplished by replacing the unit suspected and by the diagnostic's ability to fix the problem, the module must be replaced.

Corrective maintenance for internal CPU options such as the floating point processor (FPP) consists of removing and replacing diagnostic (FPP) modules of proven operation if the original diagnostic indicates a failure.

The massbus adapter (MBA) and memory (MEM) [sic] must be diagnosed and repaired to the module level. Allowed external devices are currently limited to the (RM) [sic] disk drives and tape controllers (TM) [sic] that are supported according to their respective maintenance procedures.

The following tables provide an overview of the corrective action prescribed for modules in the VAX-11/780 system cabinet.

CPU MODULES

Module Name (Mnem)	Module Number (Hex)	Level of Replacement Component Gate Array	Replacement Policy	Actions/Comments
Data Path (DP)	23B0 12	Primary (Interim diagnostic testing)	Secondary	Replace module if the gate arrays do not fix the problem.
Memory Interface Control (MIC)	1B00 10	Primary	Secondary	Replace module if the gate arrays do not fix the problem.
Writable Control Store (WCS)	1B20 16	Primary	Secondary	Tested by config (and T Diagnostic), R/O'd, R/P, or Tbus C.
I/O Control Logic (IOC)	1B00 15		Primary	As diagnostic is test this module.

WCS:
A way to determine the condition of the WCS is to run a write check as follows. The 1B00 should display during this test.

```
      >>> TEST 0          Performs this test which writes a
      BUMP 1/0            series of address patterns.
      BOOT 0X8 0
            .             If BUMP = 1/0 (hex),
            .             the WCS is bad.
            .
      BUMP = 1070
```

The WCS is also known as the Writable Control Store.
(WCS) to customers.

CPU OPTIONS

Option Name (Mnem)	Model Number (Unit)	Level of Replacement: Component/ Gate Array	Replace Module	Notes/Comments
Remote diag. Ports (RDP)	11/86? (5)		Primary	Utilize comm. option card to run diag. Downline locally or remotely. Yes, by dialing 8[3]6F,8xF
Floating Point Accelerator (FPA)	11/86? (1)		Primary	Tested by diag. "FPAACC.EXE".
VAX8600 Adapter (VA8A)	11/86? (1,1,4)		Primary	Ports three VAX bus exist.
Second UNIBUS (Inter-face Card)	11/86? (1,1,4)		Primary	Has logically resides on main U. Microcode driver. It occupies IP, CSR, VEC, ILEVEL, UBA [either 8K and 8 station for first and second UNIBUS and MASSBUS expansion cabinet].
CMI/Memory Control (CMC)		Primary (minimum diagnostic + memory)	Secondary	Microcode driver, the CMI controls memory re-location, error checking, data buffering, BR and control logic. Test by diag "CMAB.EXE".
CMI Ic. (CI) (1/8 MBs)	11/86? (1)			CMI = 256 kilobyte ic memory array module
CMI Ic. (CI) 64K Ibit (1)	11/86? (1)			CMI = 1 megabyte per memory card w/ECC
CUSP Control/ Cable			Replace each of drive unit	Use, with versions 8 and 9 of diagnostic "CUMAX.EXE".

THE REMOTE DIAGNOSTIC FACILITY

The customer is required to provide a voice grade telephone line and connector for DIGITAL Diagnostic Center (DDC) communication.

The RDF option will be included in all VAX-11/750 systems during installation to provide its value to the customer during the warranty period. It will be left in the background for all commands with the exception of maintenance mode.

CONTACTING THE DDC

Basic flow:

1. The customer calls the DDC toll free number when there is a problem.

1-800-638-6578	For DDC consultation (All Areas)
1-(617)-894-7825	For Field Service Use Only
1-(301)-493-7894	(U.S.A. Telex) (DDC Collocate Only)

   The library will stay in (617)? (Colorado Springs).

2. The DDC verifies, create customer fault isolation and the files the Field Service Branch Office.

3. The Branch office sends a Field Service Rep (FSR) with parts to correct the problem.

For CPU Problems:

1. The customer calls the DIGITAL Field Service Hot Line.

2. The customer runs the microdiagnostic console tests.

   For customers with remote capability, the RDF test is included in the VAX 11/750 background test to customer when work is complete.

   If CPU fault is, Field Service Rep uses specific module kit & go to a store or main inventory of the quad circuit.

3. The FSR then performs customer level replacement (CLR) by replacing the malfunction with a spare.

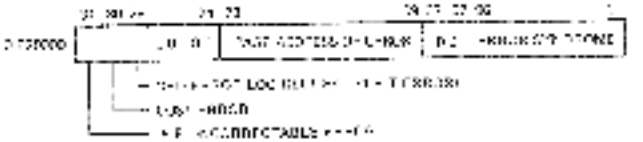
   Most DIGITAL customers (which the DDC or Field Service Rep will determine) have the capability to perform CLR.

   IMPLEMENTATION: The CLR should be explained with the RDF to assist them in building a base library of Critical Spares for the VAX-11/750.

CHAPTER 2
SYSTEM REGISTERS

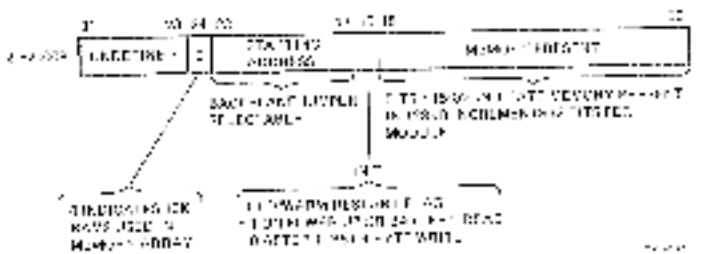
CMC (MS750) REGISTERS

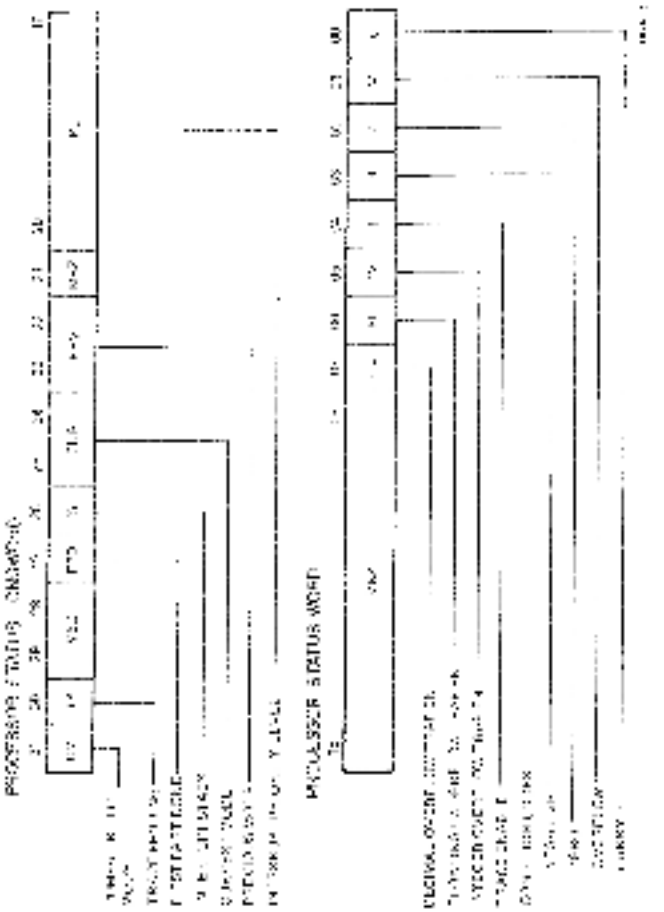
CONTROL/STATUS REGISTER 0 (CSR 0)



CONTROL/STATUS REGISTER 1 (CSR 1)

CONTROL/STATUS REGISTER 2 (CSR 2)





INTERNAL PROCESSOR REGISTER (IPR) SUMMARY

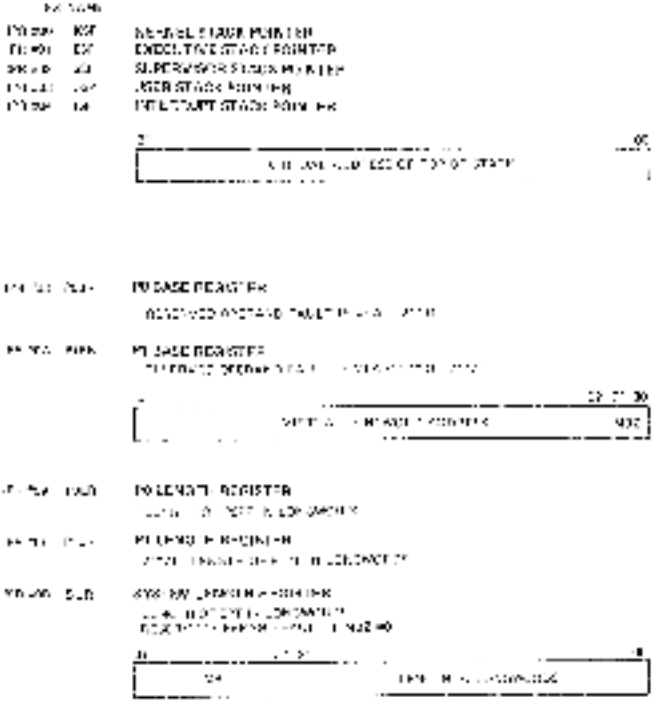
Address	Mnemonic	Reads*	Name
00	KSP		Kernel Stack Pointer
01	ESP		Executive Stack Pointer
02	SSP		Supervisor Stack Pointer
03	USP		User Stack Pointer
04	ISP		Interrupt Stack Pointer
05-07	Reserved		
08	P0BR		P0 Base Register
09	P0LR		P0 Length Register
0A	P1BR		P1 Base Register
0B	P1LR		P1 Length Register
0C	SBR		System Base Register
0D	SLR		System Length Register
0E-0F	Reserved		
10	PCBB		Process Control Block Base
11	SCBB		System Control Block Base
12	IPL		Hardware Priority Level
13	ASTLVL		AST Level Register
14	SIRR	W/O	Software Interrupt Request Register
15	SISR		Software Interrupt Summary Register
16	Reserved		
17	MAPEN	R/W	CPU Mapen Register
18	TBIA		Invalidate Translation Buffer All
19	TBIS	W/O	Invalidate Translation Buffer Entry
1A	TBIS	R/O	Translation Buffer Check Register
1B	Reserved		
1C	SID		Processor Identification
1D	TBCHK		Translation Buffer Check
1E	RXCS	R/W	Console Receive Control/Status
1F	RXDB	R/O	Console Receive Data Buffer
20	TXCS	R/W	Console Transmit Control/Status
21	TXDB	W/O	Console Transmit Data Buffer

*Registers are read/write unless otherwise specified.
W/O writes disregarded; R/O reads write only to zeroes
the specified bit(s).

Address	Mnemonic	Type*	Name
32	RXCS		Console Receive Control/Status
33	RXDB	R/O	Console Receive Data Buffer
34	TXCS		Console Transmit Control/Status
35	TXDB	W/O	Console Transmit Data Buffer
36	TBDR		Translation Buffer Disable Register
37	CADR		Cache Disable Register
38	MCESR		Machine Check Error Summary Register
39	CAER		Cache Error Register
3A	ACCS	R/W	Accelerator Control/Status
3B-3D	Reserved		
3E	TB_RESET	W/O	Initialize TB(IBUF)
3F			Memory Management Trap
40	TBIA	W/O	Translation Buffer Invalidate All
41	TBIS	W/O	Translation Buffer Invalidate Single
42	TB_TEST		Translation Buffer Data
43	Reserved		
44	PMR		Performance Monitor Register
45	SID	R/O	System Identification
46	TBUF	W/O	System Translation Buffer for TB Mi.

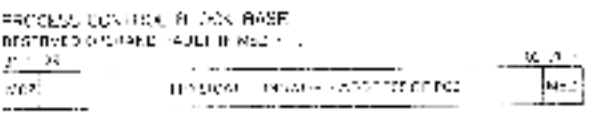
*Key: where are read/write values (Operating Condition)
R/O means read-only; W/O means Write-only to Affect
the specified Functions).

INTERNAL PROCESSOR REGISTERS

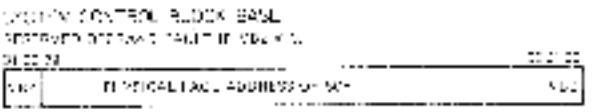


HEX NAME

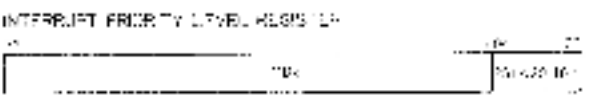
IPR #18 PCBB



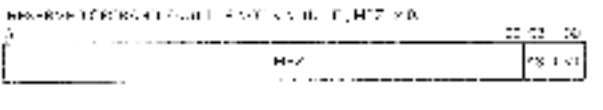
IPR #19 SCBB



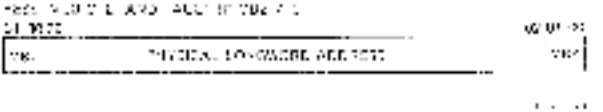
IPR #20 PRBR



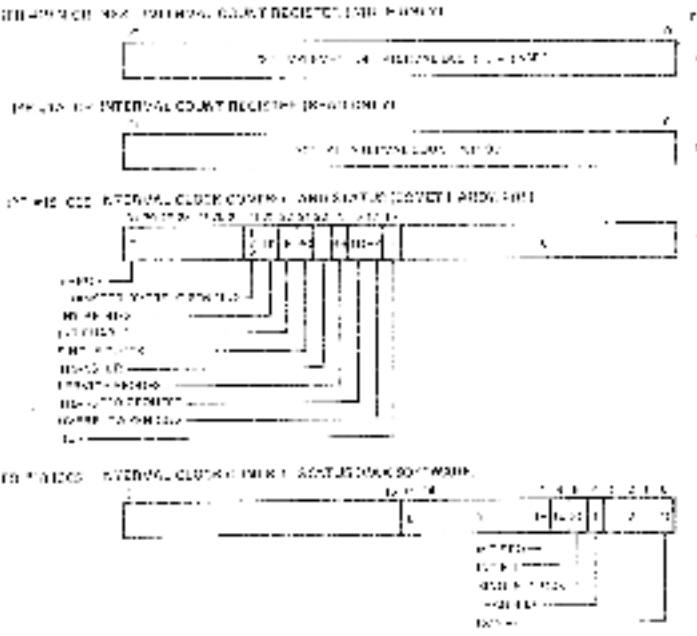
IPR #21 SISR



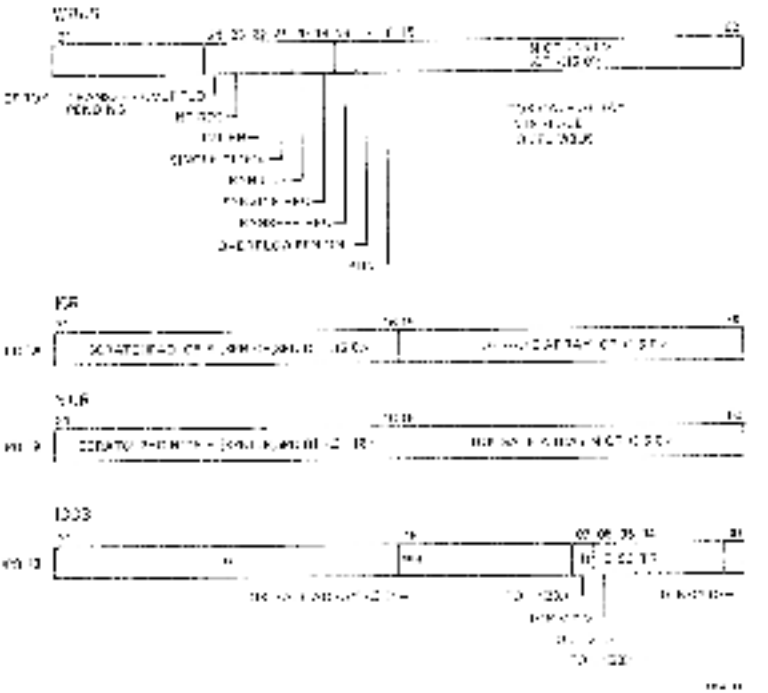
IPR #22 ASTLVL

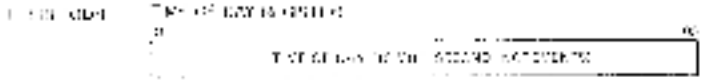


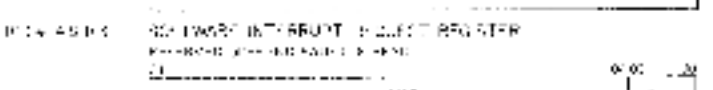
IPR #23 TBCHK

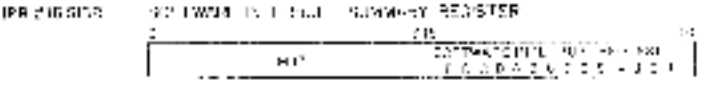


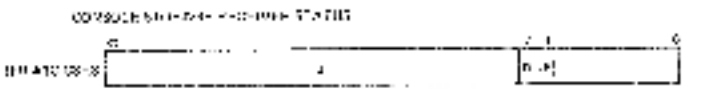
INTERVAL REGISTER HARDWARE USAGE

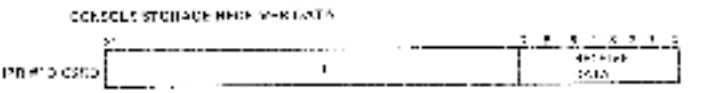


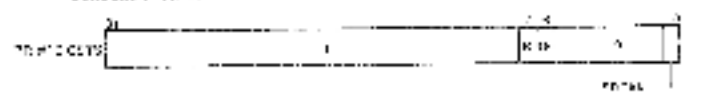


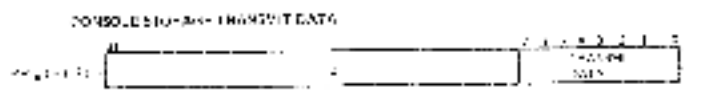








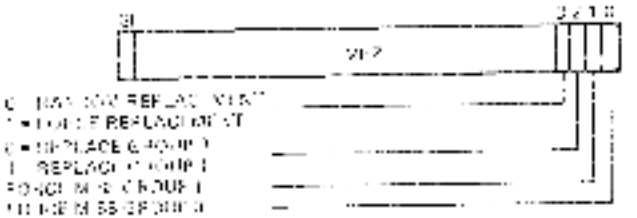




INTERNAL PROCESSOR REGISTERS (CONT)

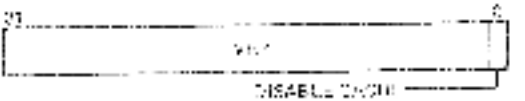
PR #27 VECTOR — VIRTUAL I/O BUFFER / GROUP DISABLE REGISTER — PR #27

THIS REGISTER HAS TWO 1-BIT FIELDS



PR #28 CACF — CACHE DISABLE REGISTER — PR #28

THIS REGISTER HAS ONE FIELD

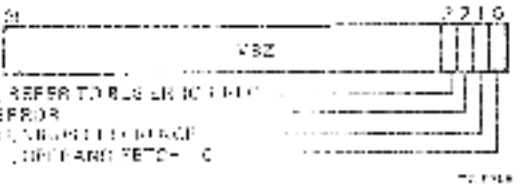


PR #27 CAER — CACHE ERROR REGISTER — PR #27

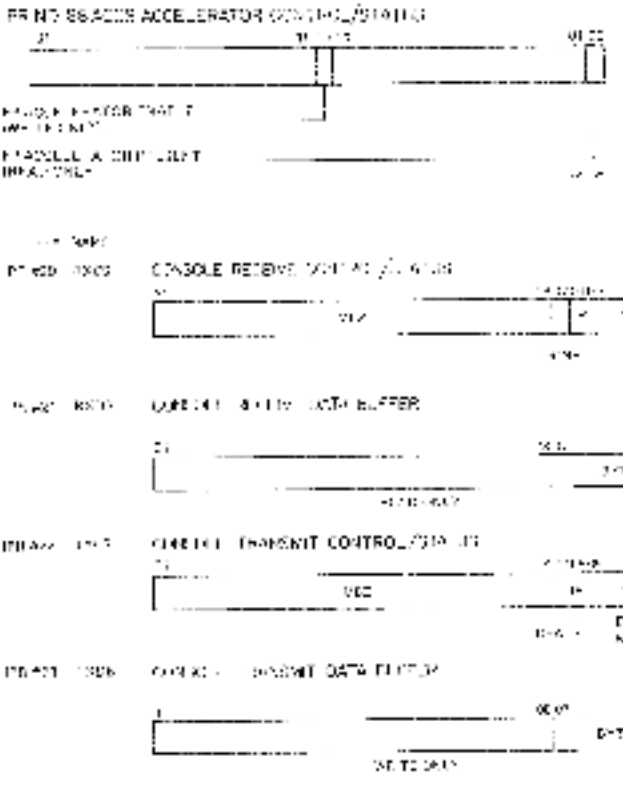
THIS REGISTER HAS

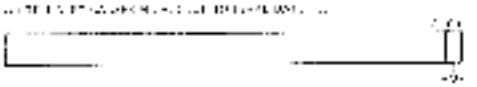
PR #26 VCR — MACHINE CHECK ERROR SUMMARY REGISTER — PR #26

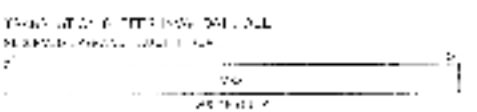
THIS REGISTER IS A MACHINE CHECK ARCHITECTURAL ATTRIBUTE. IT CONTAINS THE BUS ERROR REGISTER, WRITE ERROR REGISTER, AND PARITY ERROR GROUP PARITY ERROR BITS.

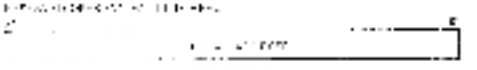


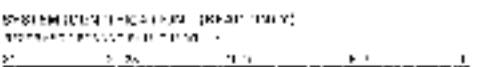
INTERNAL PROCESSOR REGISTERS (CONT)

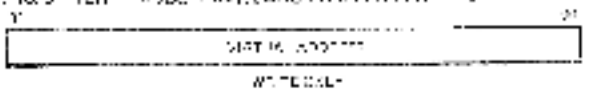


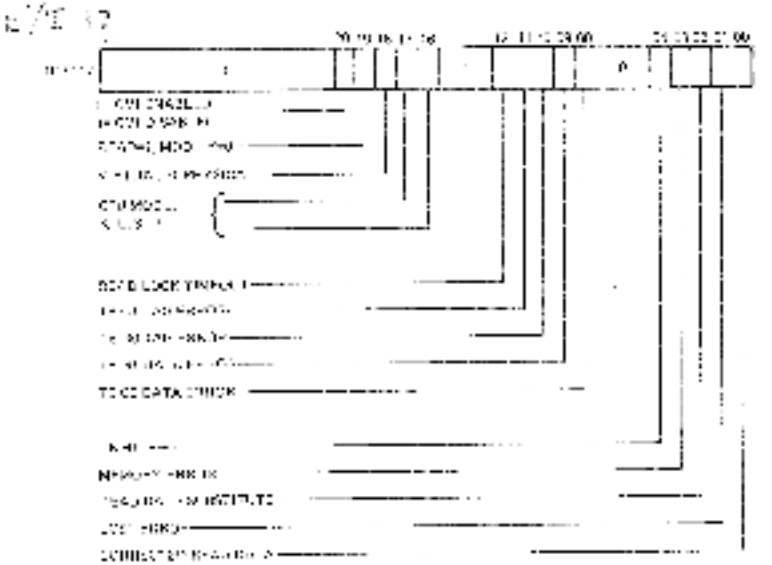


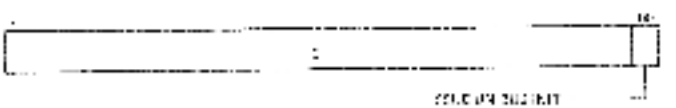


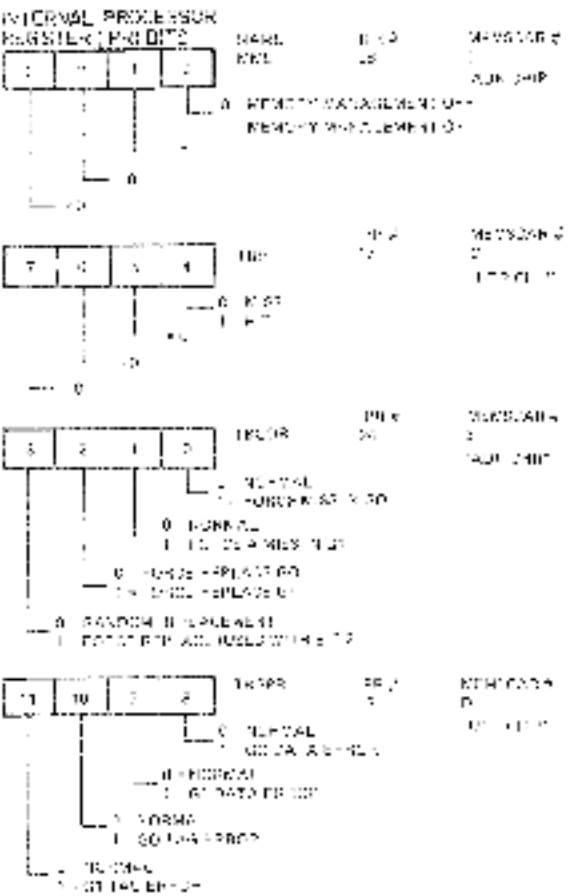




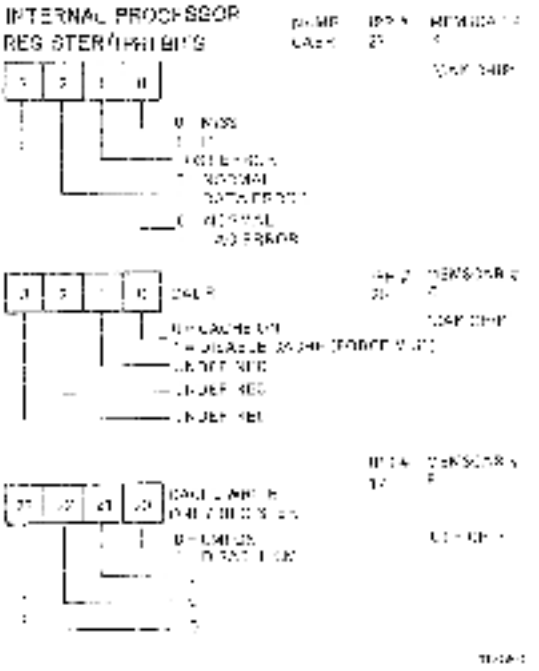




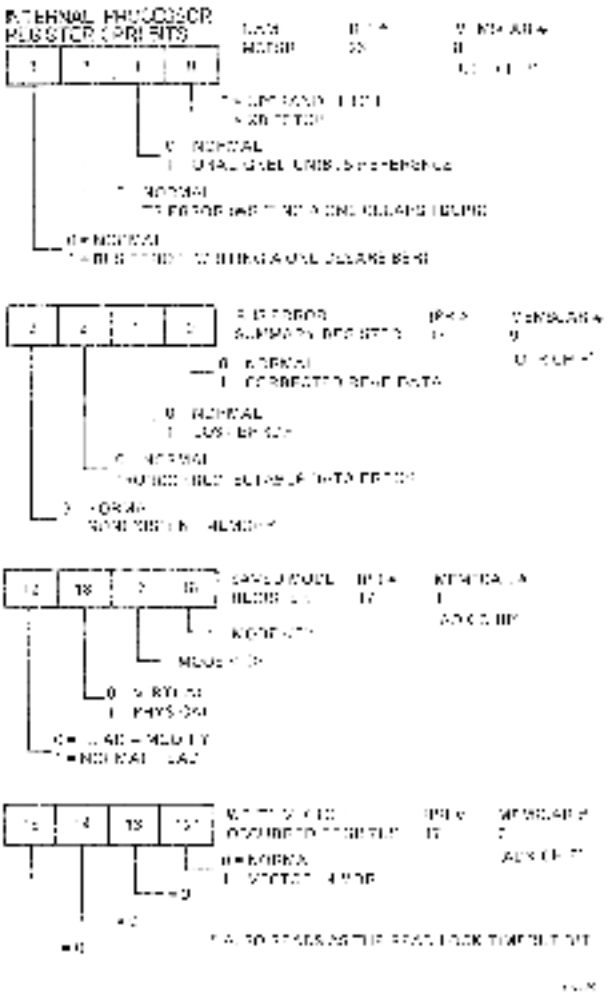




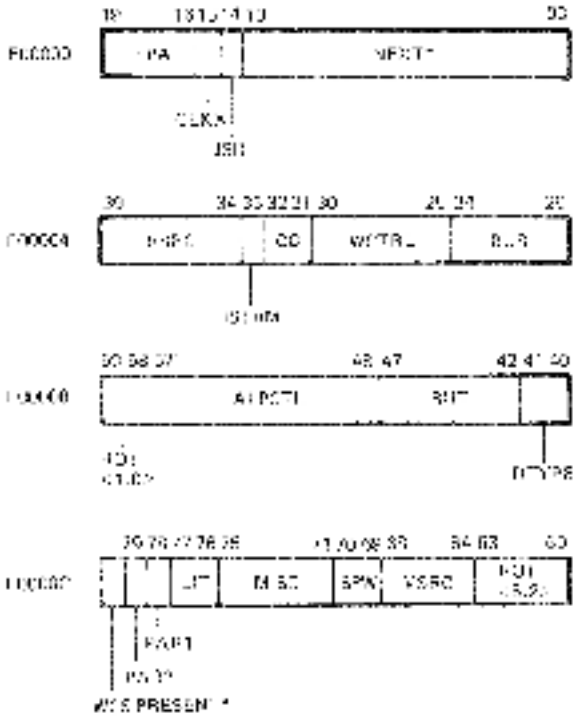
CACHE REGISTER BIT FIELDS



TB AND CACHE CONTROL/STATUS REGISTER BIT FIELDS



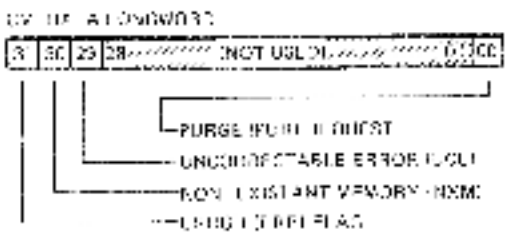
WCS (KU750) REGISTER DATA WRITE FORMAT



*NOTE: WCS ADDRESS SPACE IS FOCCUT THROUGH 3FFFC

BIT <20> IS WRITTEN AS 1 ON THE LAST WORD WRITTEN TO THE WCS. WCS MICROSEQUENCER ENABLES THE WCS MICROCODE TO BE EXECUTED.

UBI OR SUB (UN750) CONTROL/STATUS REGISTER (CSR)



DEB Addresses

[illegible] Register	UBI Address	SUB Address	Buffered Data Path
DEB 1	[illegible]	[illegible]	BUP 1
DEB 2	[illegible]	[illegible]	BUP 2
DEB 3	[illegible]	[illegible]	BUP 3

UBI OR SUB MAP REGISTER DATA

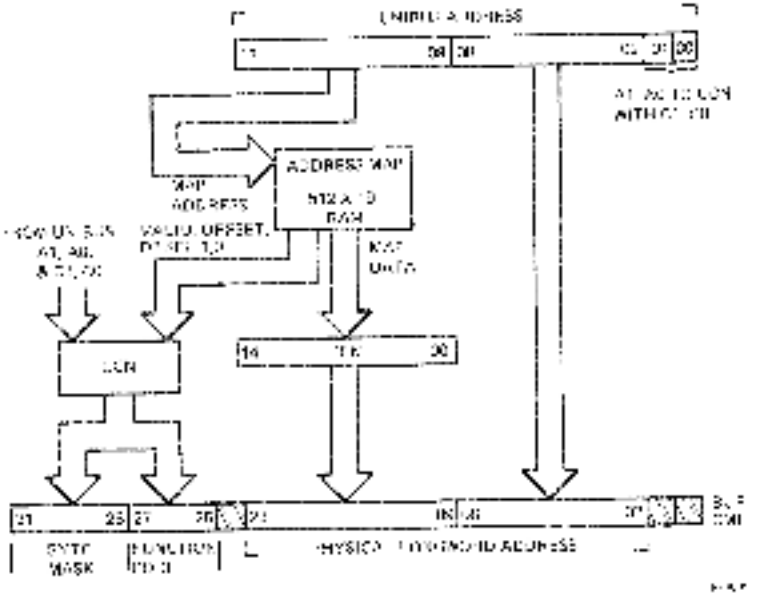
BIT	NAME	DESCRIPTION
31	V	Data Path Valid (DPV)
30	C	Corrected Data Path (CDP)
29	2	Uncorrected Data Path 2 (UDP 2)
28	1	Uncorrected Data Path 1 (UDP 1)

MAP Register Addresses

MSU Registers	UBI Address	SUB Address
MAP 000	F30800	F32000
MAP 001	F30804	F32004
.	.	.
.	.	.
MAP 3FE	F30FF8	F33FF8
MAP 3FF	F30FFC	F33FFC

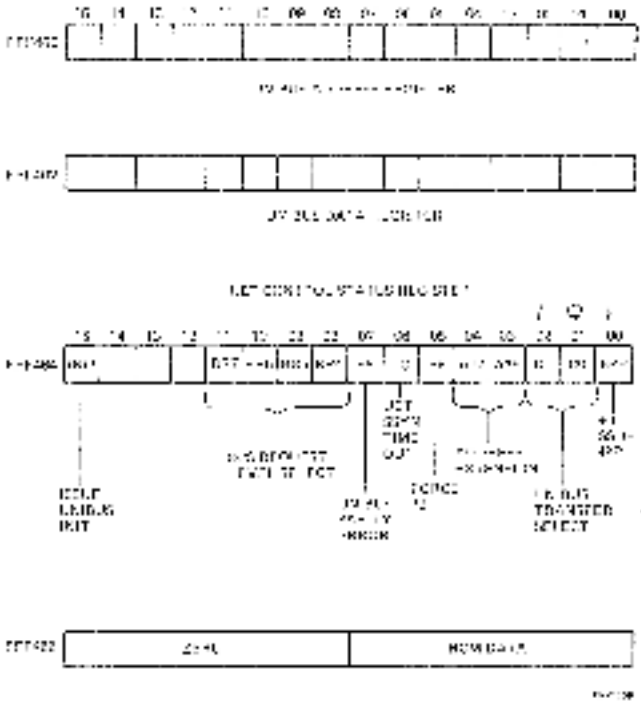
UBI AND SUB REGISTERS (CONT)

UNIBUS TO CMI MAP ADDRESS TRANSLATION



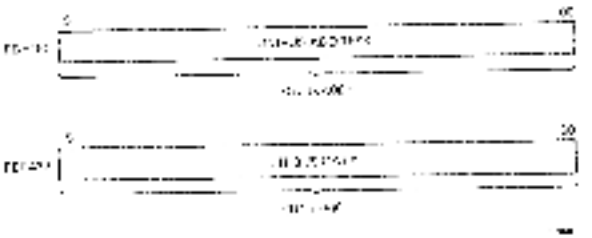
UBI AND SUB REGISTERS (CONT)

UBT REGISTERS

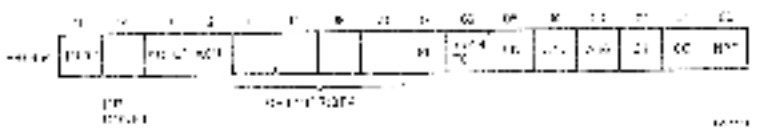


IPEC REGISTERS

UNIBUS Address and Data Registers



Control Register 1 (CR1)

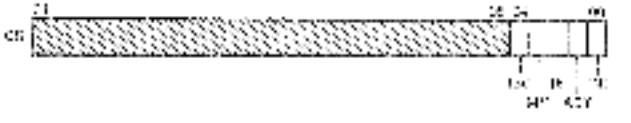


Control Register 2 (CR2)

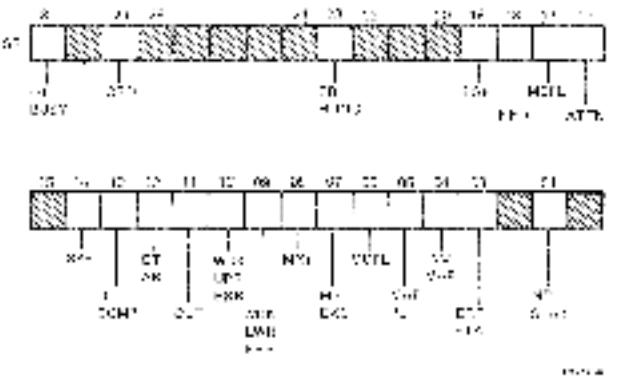
MBA (RH750) REGISTERS

MBA INTERNAL REGISTERS

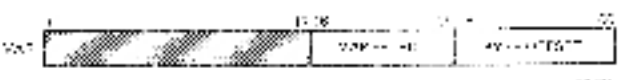
Control Register (CR) — F25004



Status Register (SR) — F25008

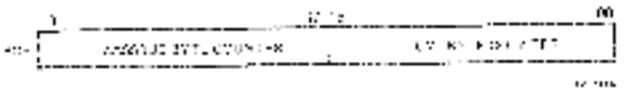


Virtual Address Register (VAR) — F2800C

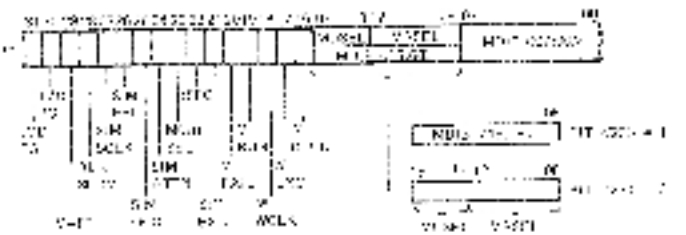


MBA INTERNAL REGISTERS (CONT)

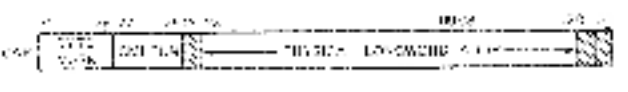
Byte Count Register (BCR) – F28010



Diagnostic Register (DR) – F28014

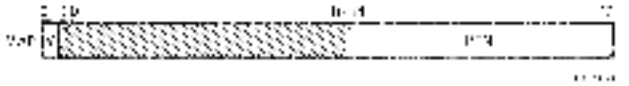


Command Address Register (CAR) – F28018

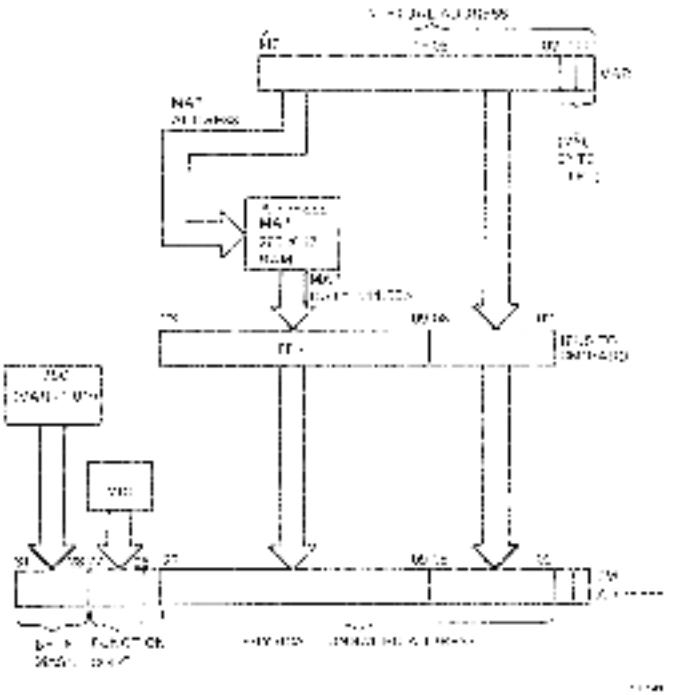


MBA (RH/SBI) REGISTERS (CONT)

MBA MAP REGISTER DATA — F28SDC - F28BFC

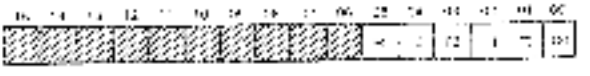


MBA TO CMI MAP ADDRESS TRANSLATION

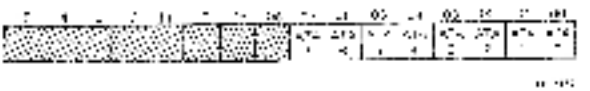


MASSBUS EXTERNAL DEVICE REGISTERS

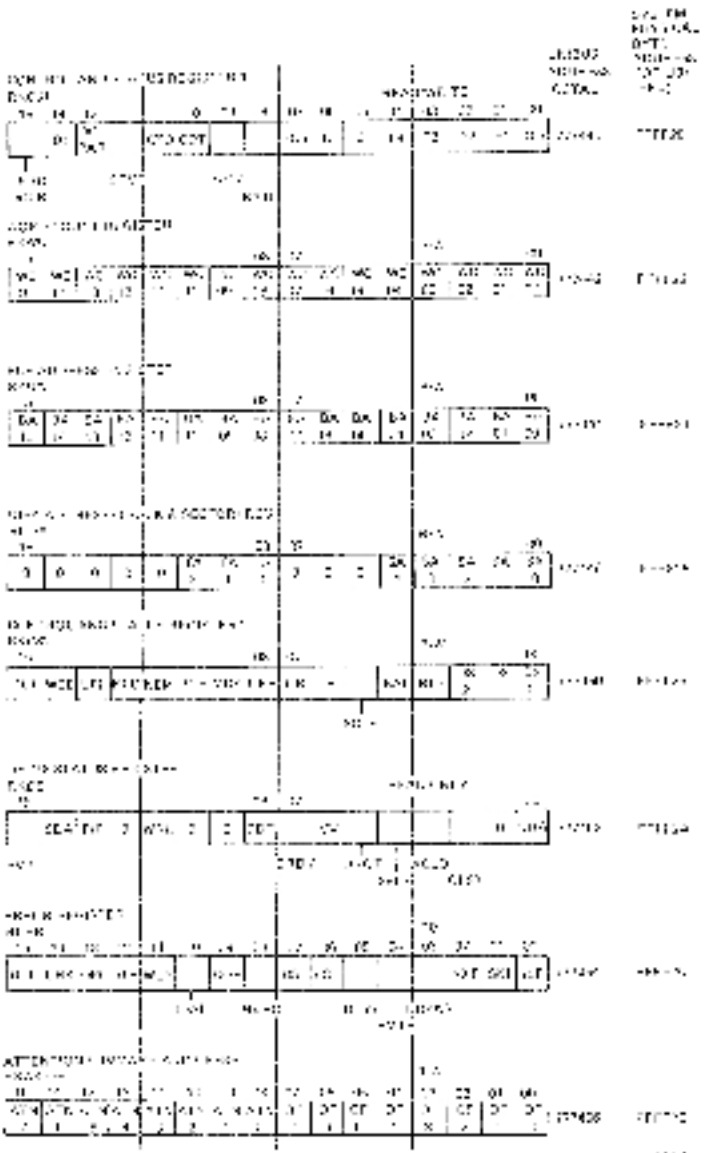
Drive Control Register (R3) – F28480



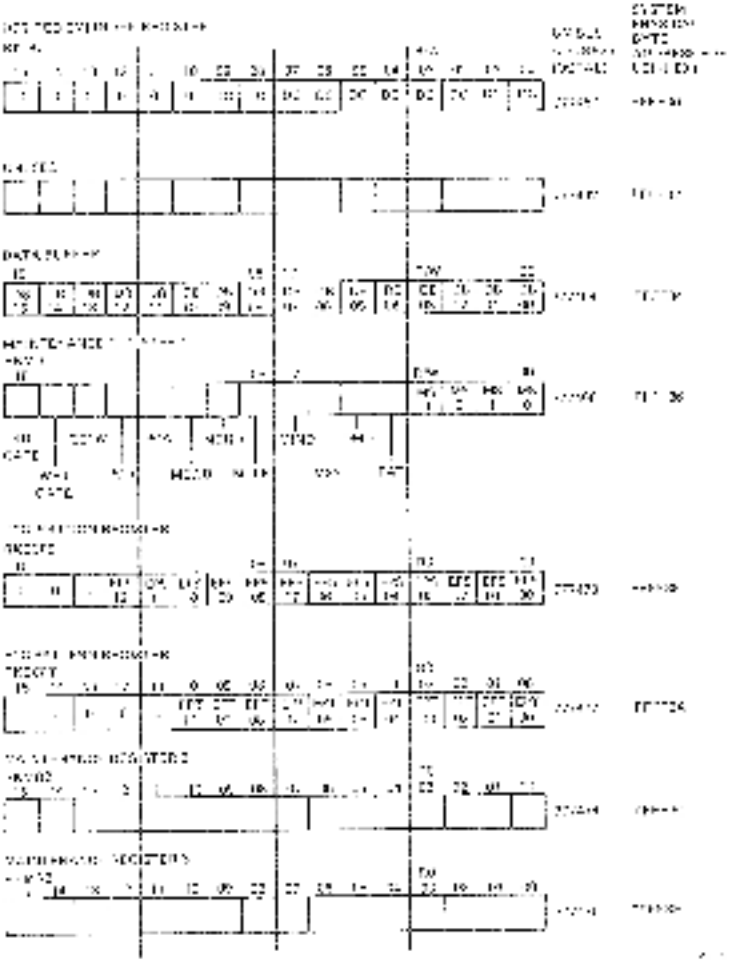
Attention Summary Register (R4) – F28410



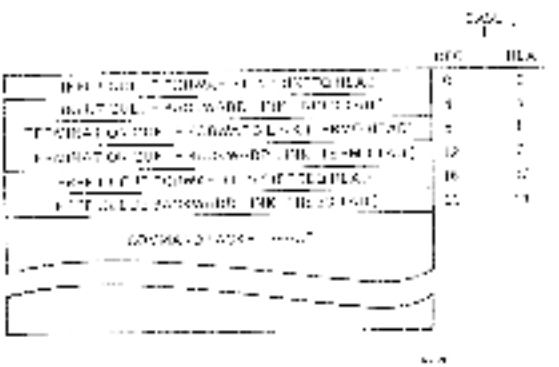
0211 REGISTERS



RK611 REGISTERS (CONT)

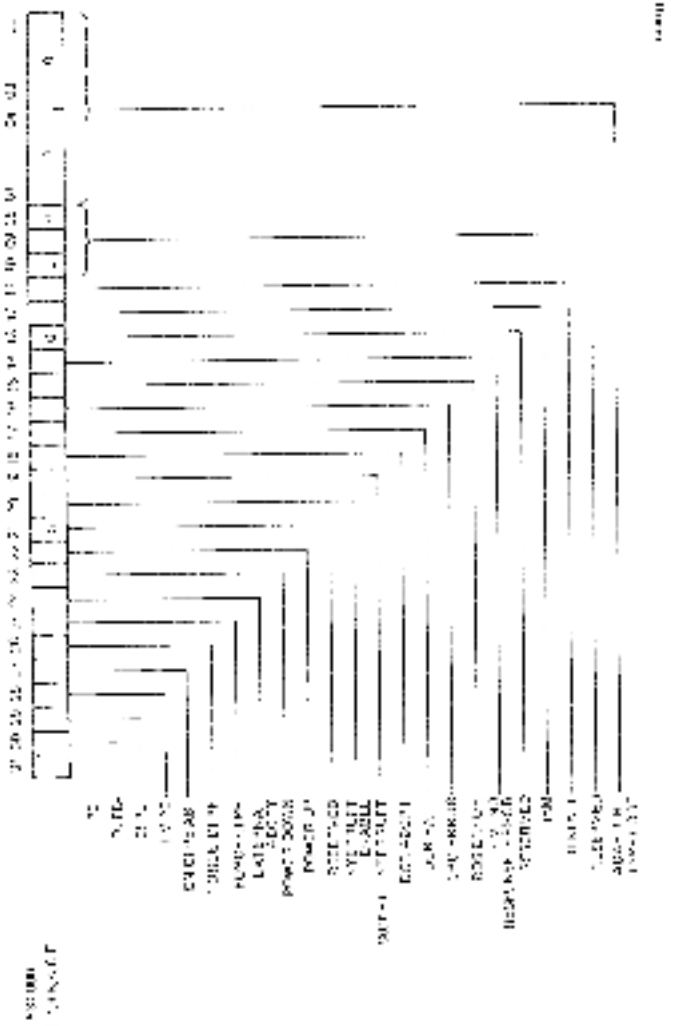


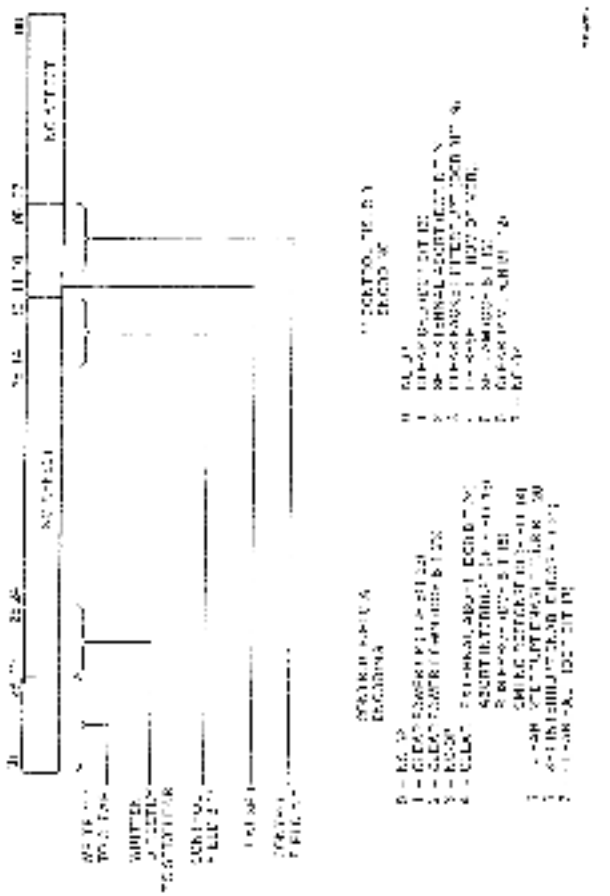
DR750 COMMAND BLOCK



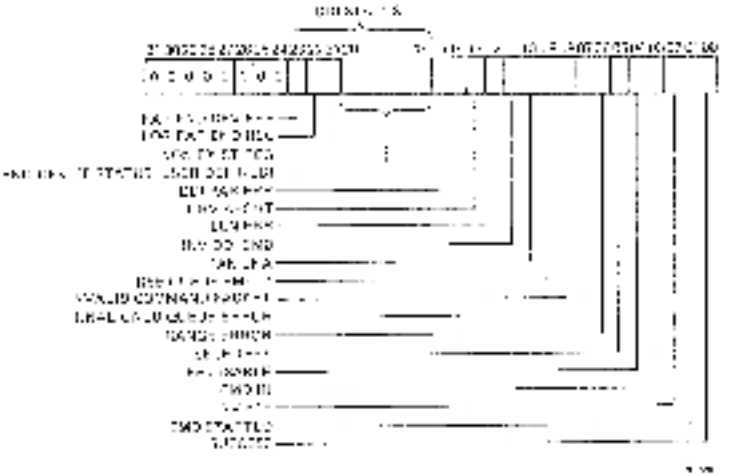
DR750 COMMAND PACKET

DR750 DCR REGISTER — READ FORMAT





DR750 STATUS LONGWORD



DR750 UTILITY REGISTER

DI CLOCK DATA RATE SELECTION

Utility Register	DI CLOCK RATE
Value	(Mbits/sec)
DI CLOCK PS*	--
P0	1.12
P1	2.50
P2	3.69
P3	1.76
P4	.56
P5	.30
P6	.17*
P7	1.14
P8	1.04
P9	0.91
PA	0.81
PB	0.71*
:	:
:	:
FF	0.27

*Results of clock registers by the Intel 8274 on prototype.

DR750 REGISTERS (CONT)

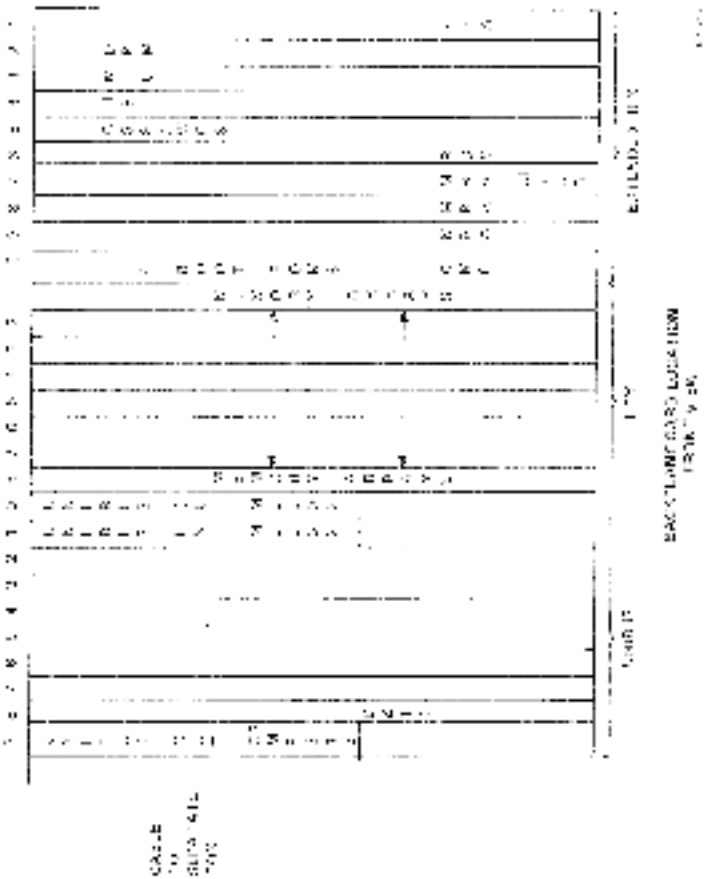
DR350 DEVICE VECTORS

Base Address	Vector Number	RH0	RH1	RH2	RH3
[illegible]	0	200	1F8	1A0	1E8
[illegible]	1	1D0	1B0	1A0	1B0
[illegible]	2	90	178	1A8	1A8
[illegible]	3	150	170	180	190
[illegible]	4	130	198	188	198
[illegible]	5	110	100	1B0	1B0

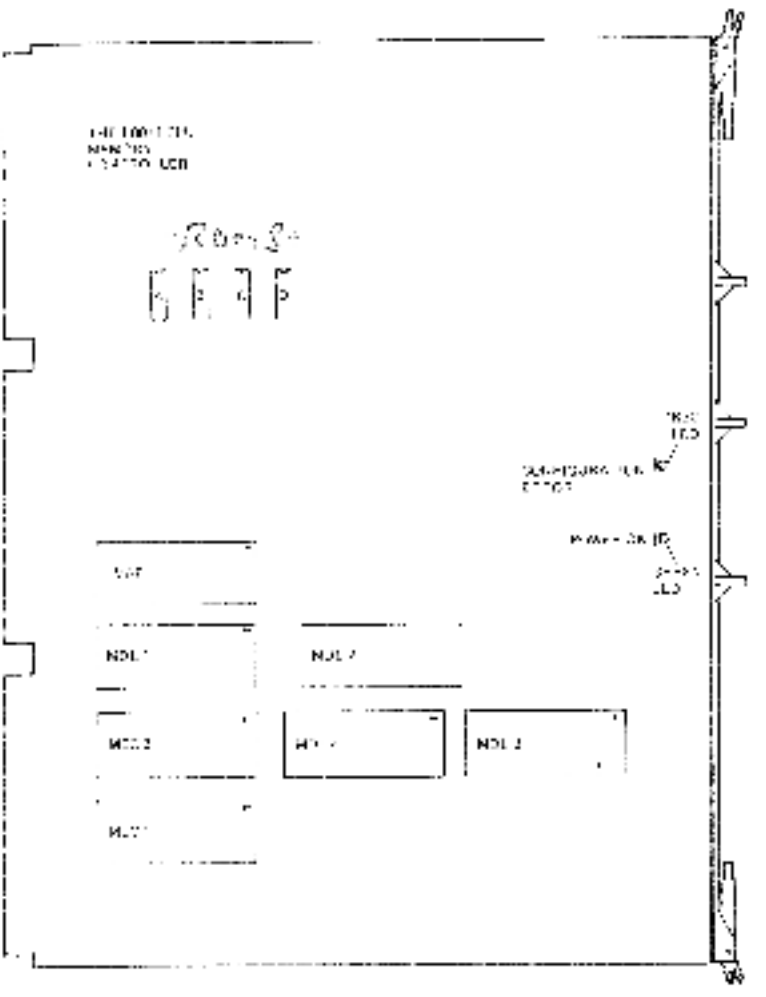
*Standard Interrupt Level 5 Used.

CHAPTER 3
MODULES AND GATE ARRAYS

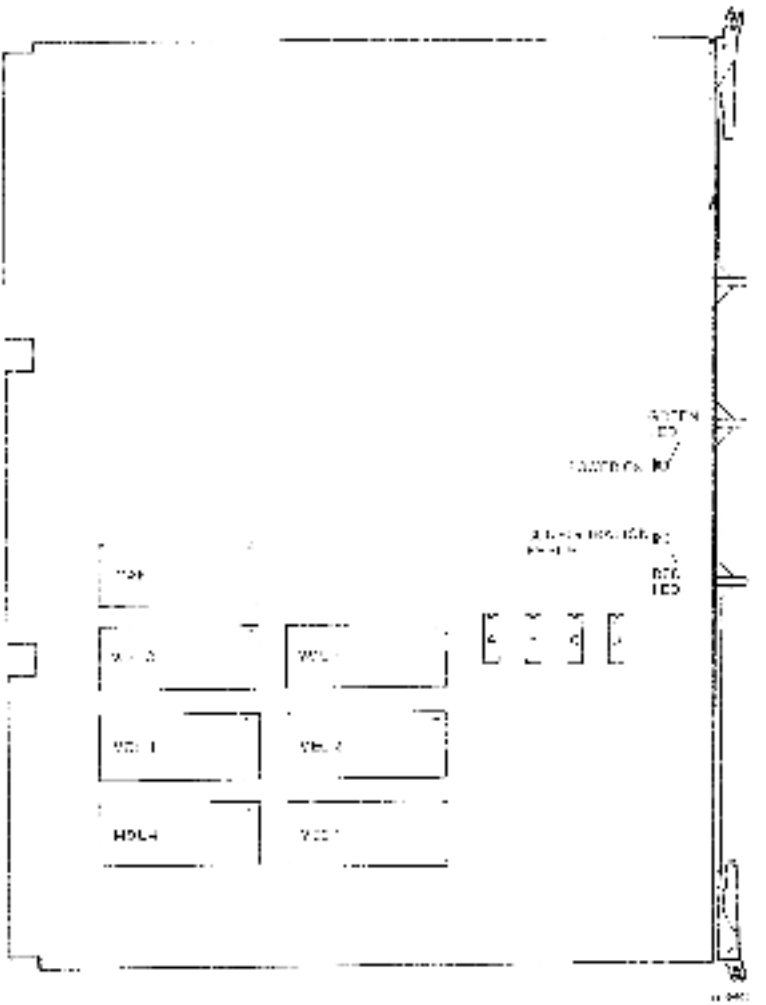
VAX-11/780 MODULE UTILIZATION



CMC GATE ARRAY LAYOUT (L0011)



CMC GATE ARRAY LAYOUT (L0016)



```
MRW   - Read / Write Data Bus (BDB). High asks up the data
        path between the CMT and memory, the CMI and the
        control/status registers, and from the bootstrap
        ROMs to memory

MEC   - Two memory error detection (MED) error output bits
        I.E. parity, and detect and correct single bit errors.

MA(19:0) - One memory address processor (MAP) chip is
           used. The memory address decoder (MAD) chip is
           not used. Thirteen bits are decoded to enable
           memory arrays and selection starting addresses
           thereby. Address verification errors are detected
           and memory error board population is determined
```

BOOTSTRAP DEVICE ROMS

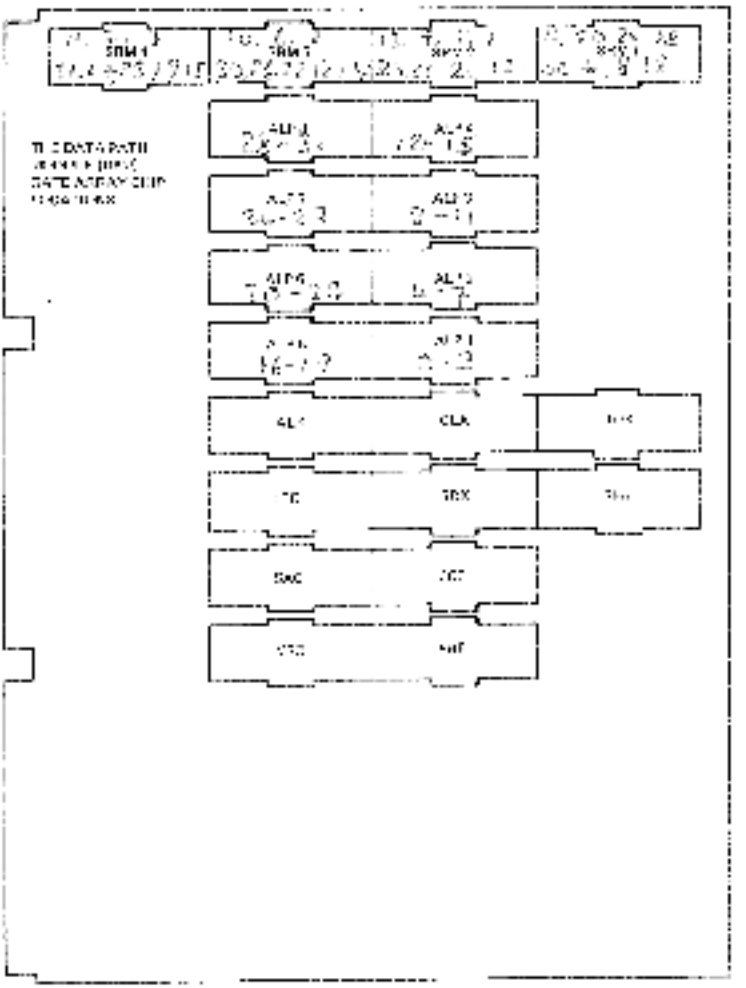
ROM Unit Number	Device Code	Device Type	Control Bar	CSR
23-08000-01	RL	2048	RF50	
23-08000-02		2048	RL02	
23-09000-02	RX	2048 - Single		

*Modified serial serial processor (MSP) is required for MFM

ROM Starting Microaddresses

Device Code	Starting Microaddress
A	RX02
B	RX01
C	RL02
D	RK0[illegible]

DMI GATE ARRAY LAYOUT (LD002)



MPY - Four mutually exclusive multiply (MPY) chips perform 64 functions under the control of the SAM.

EXP - The exponent control (EXP) chip controls MPY operations.

ALU - Eight cascadable 1-bit processor (ALU) chips make up the arithmetic logic unit (ALU) of the DPM. The ALU performs most of the data manipulation during the execution of instructions.

ACS - One arithmetic logic control (ALC) chip cascades and coordinates by decoding microcode inputs and generating the necessary signals.

LKA - One micro-lookahead (LKA) chip contains the logic that the ALU uses to generate the propagate carries.

TC2 - One three operation control (TOC) chip controls the programmable interval timers.

IID - One interrupt, opcode decode (IID) chip processes the IR sequence. Both the opcode and operand specifier for the current instruction (IR), decodes it, and generates signals to select the correct microcode routine.

SAM - One scratchpad addressing (SAM) chip controls the 64 word scratchpad register addressing. Contains logic that keeps track of current program register (CPR) selections and autoselections.

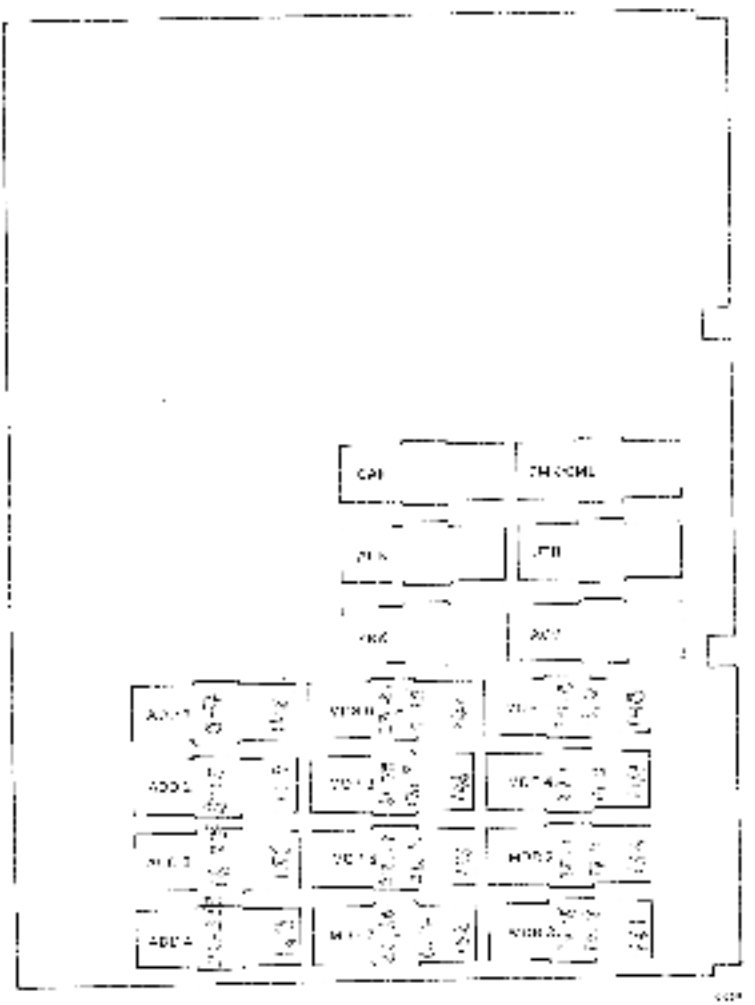
SAC - One service arbitration and clock (SAC) chip is associated with the DMA request, service arbitration, and system clocks.

CCC - One operation code (CCC) chip is associated with the condition codes for both shift and cryptability mode instructions. It stores PSL bits 16, 17, 20, 21, 22, and 23, and makes the bits at the request of the microcode.

MSQ - One microsequencer (MSQ) chip, together with the CCC device, forms the DPT microsequencer that addresses the CPU microcode.

PTH - One previously sold the SMUX (PTH) chip contains RAM bits, status flags, the DMA and DAT, and a four parallel input bit of the DUT micro-drivers.

MIC GATE ARRAY LAYOUT (LB003)



RAM - Right memory data register (RDR) chips contain logic that cycles the data to or from the CPU in 1, 2, 4 bit, 8-bit, or nibble.

PBV - Bus buffer control (BBC) chip - provides bi-directional control of 8 (SYSAD) bytes from memory multiplexed with low 32 address. It also replaces most TTLMSI or a like during prefetch operation.

ADC - One address control (ADC) chip which controls the MIC addressing logic.

ADD - Four address (ADD) chips used by the 32 x 32 64 registers and their load ports.

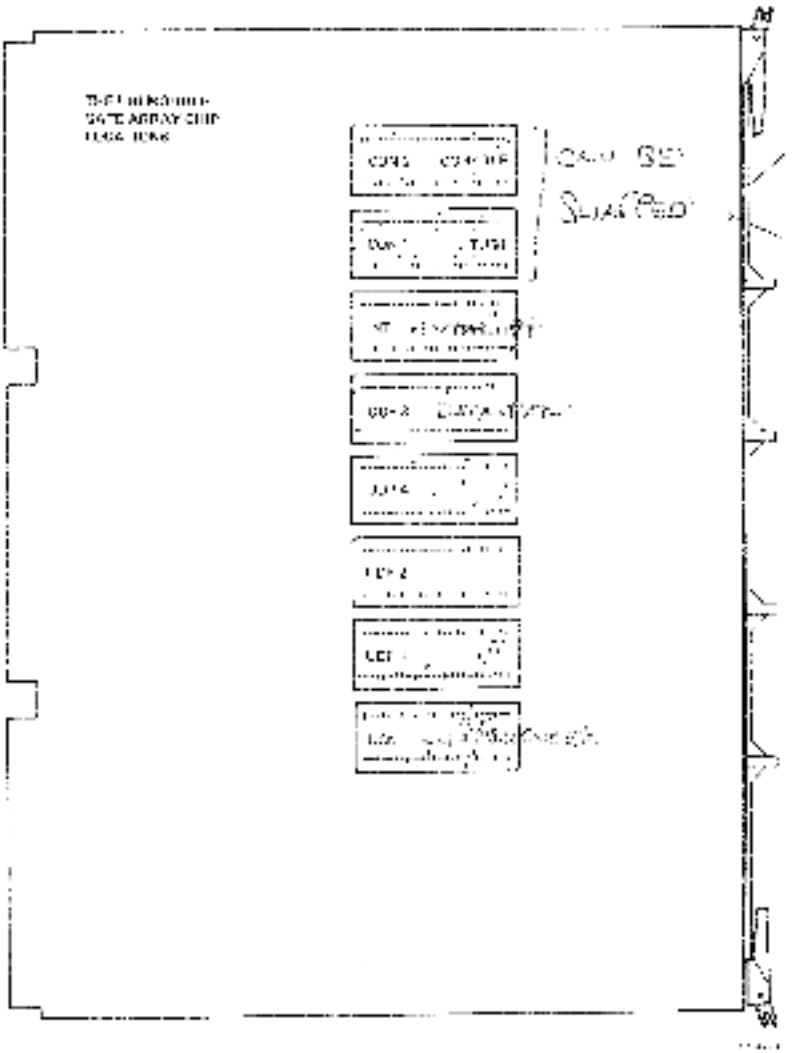
CMC - One CMT control (CMC) chip sequences and transmits CMI serial data. That is for diagnostic mode certain conditions, A CRC chip is located at the place for the CPTSC.

DAC - One cache control (CCT) chip enables or disables and writes. Controls the transfer of data to or from the MIC chips.

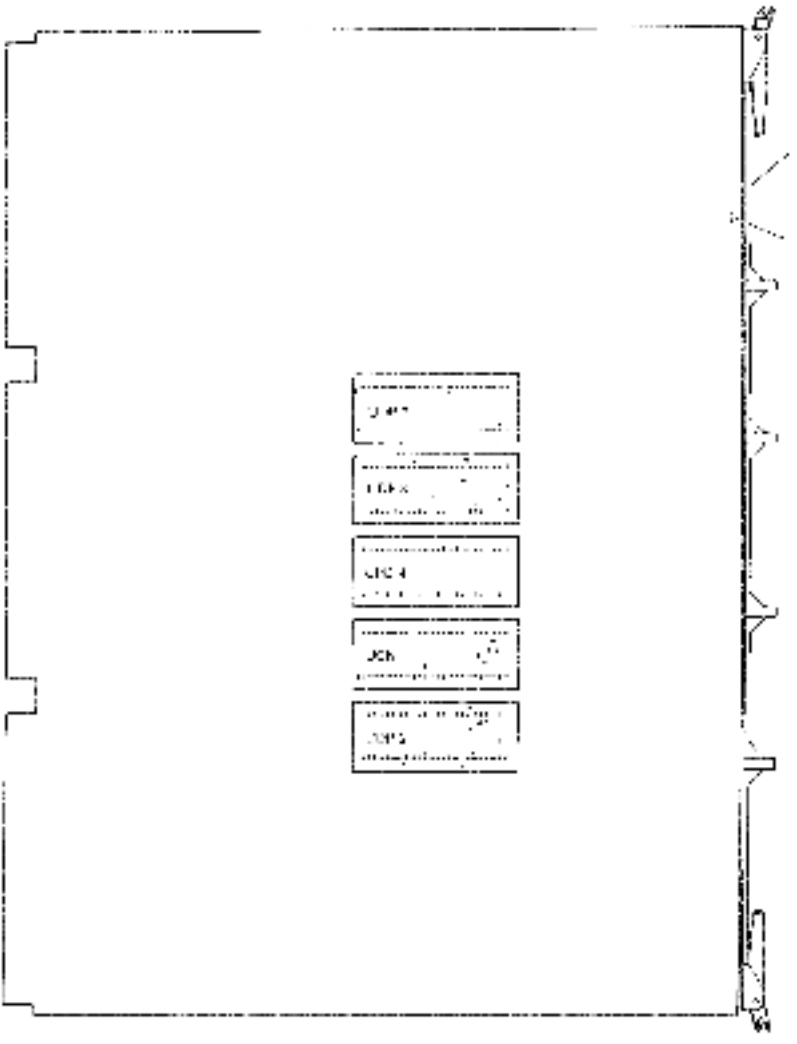
UCP - Six microprog (UCP) chip maintains machine conditions that cause a microtrap.

MPV - Six access violation (ACV) chip detects access violations, control store parity errors, the reserved addresses, illegal data, and page boundary violations.

UBI GATE ARRAY LAYOUT (L0004)



SUB GATE ARRAY LAYOUT (L0010)



UBI AND SUB GATE ARRAY DESCRIPTION

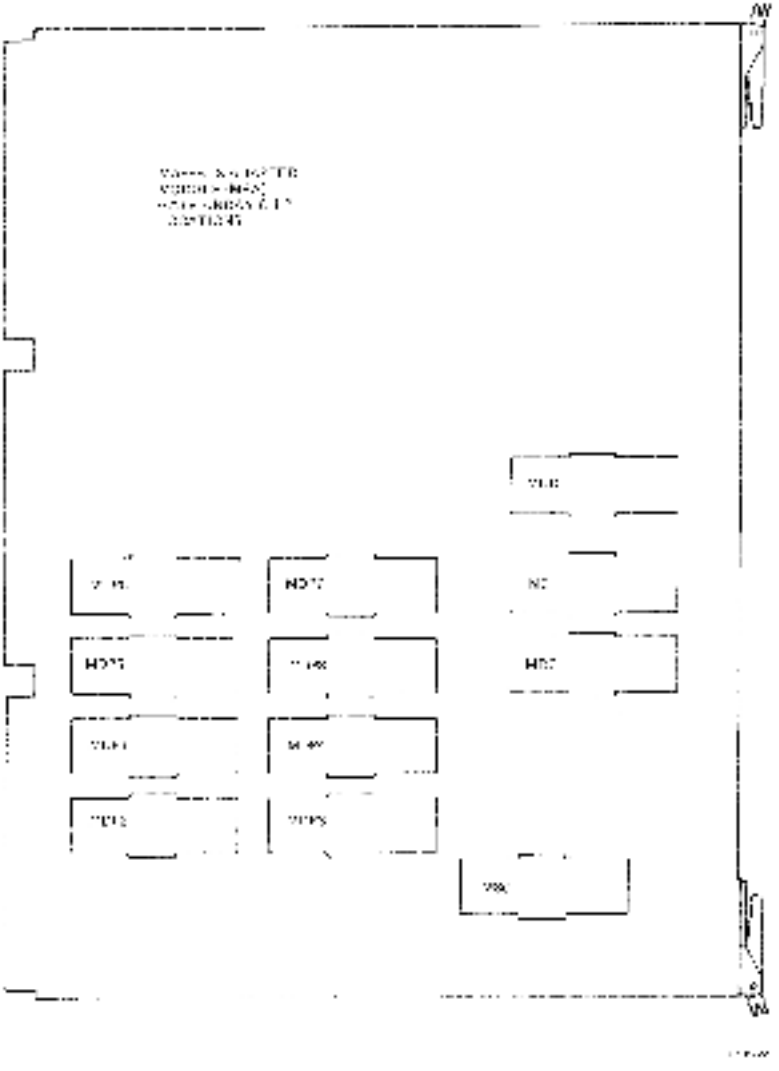
UDP - Four UNIBUS data path (UDP) chips make up the data path logic for the UBI and SUB. All address and data information passes through the UDP between both private ports (CPU, P.I.P CPU, µB data, µB address), between buffers and between both ports and buffers. Internal address latch and parity logic for the buffered data paths, perform interpretation between the CPU and UNIBUS address formats and address operations.

UBI - One UNIBUS control (UBC) chip works in conjunction with the CPU and SUB control chips to provide UNIBUS/memory access, or a I operation. Contains the byte and error flags for the buffered data paths, performs interpretation between the CPU and UNIBUS address formats and defines operations.

UBT - One interval timer (UBT) chip in the µB performs arbitration to provide all interrupt requests and vectors on the arbitration lines to start the CPU. Contains all of the µB's architecture for translations from the CPU, initiates bus requests (BR4) from the parts and, and issues bus grants (BG).

CCA - Two console (CON) chips on the SUB each convert serial/parallel data for communication between the CPU and the user and console terminal via the M-bus. Contain. Initial console keyboard character recognition.

MEA GATE ARRAY LAYOUT (LG007)



MBA GATE ARRAY DESCRIPTION

MDP - Eight (8/9) data path (MDP) chips route data between disk controllers ports (DPT, TAPE, CRUD, PIOD, MBUS). All address and data information is routed through the MDPs. Control address latch and compare log a for DMA access to VAX or MASSBUS buffers.

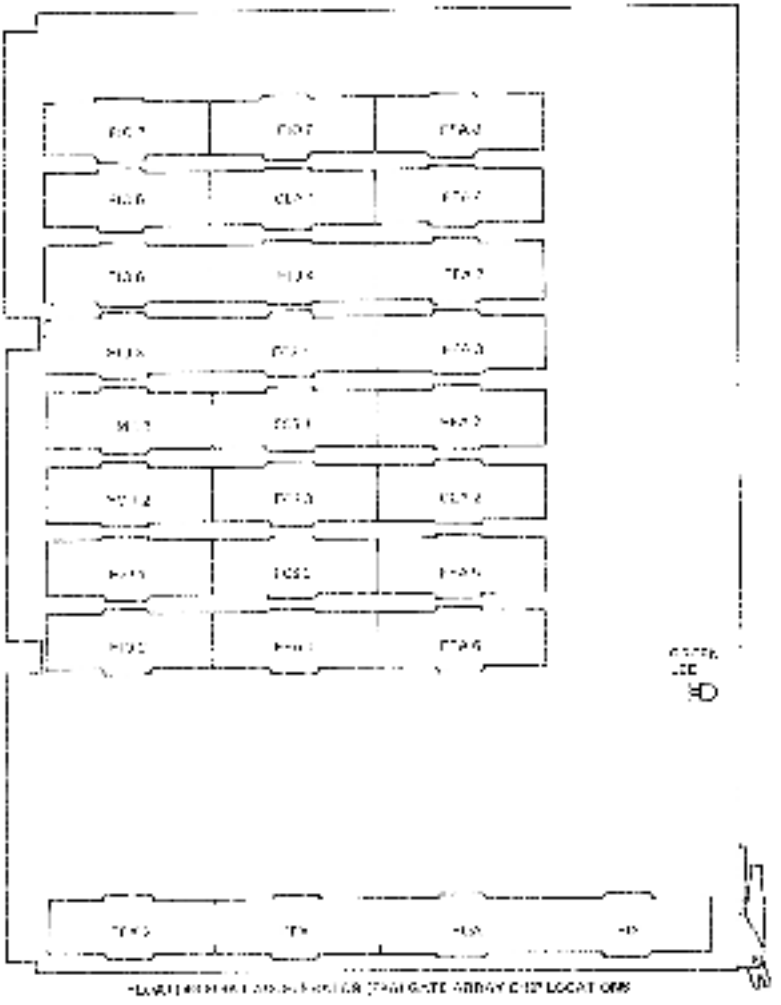
MAP - One MASSBUS data bus control (MAP) chip controls and maintains status on control bus parity and VAX memory and parity. Also checks and controls bit flags on the VAX and initiates MBP parity/control clip checks.

DCA - One MBA CMI interface control (MCI) chip controls MBA and CMI interactions plus CMI acquisition, status generation and checking, and processor interrupts. Generates the CMI operation codes for MBA transfers to the CMI and controls MAP address translations. Plays basic clocks to translate with internal or external requests and directs the MAP to perform the transfer.

MCC - One MBA register control (MRC) chip responds to the MCI to CPU operations with internal or external registers. Detects the issue of a prior data transfer command and informs the MBA. Produces parity signals to the control bus and other boundary interface points that control clocking in the MBA registers.

MTC - One MBA bus control (MBC) also produces data transfer termination codes that control routing and alignment between the CMI and MBA registers. Produces the byte mark/ack, detects byte rot or CMI supply conditions, and generates and checks parity on data and MASSBUS data. Produces the CMI 3.5 byte mask for DMA transfers on the CMI.

FPA GATE ARRAY LAYOUT (LD001)



FPA GATE ARRAY DESCRIPTION

FPA - Eight floating point arithmetic (FPA) chips form the arithmetic logic unit (ALU) of the FPP. The ALU performs most of the data manipulations during the execution of floating-point instructions. It outputs the 64-bit ZU bus to the FTC.

FLA - Two carry look ahead (CLA) chips contain logic used by the ALU with [unreadable] [illegible] circuitry.

FIC - Eight floating input/output (FIO) chips interface operands from the Y bus and Y bus. Also includes an 8-word operand queue.

FMM - Four floating point multiplier (FPM) chips perform eight or both single or multiplies or less. Produces a 72-bit product from a 54-bit input.

FQA - One floating point quick aligner (FQA) chip positions the FPR for normal operations.

FMR - Two division multiplier (FMD) chips contain most of the logic that performs division multiplication.

FRQ - Two first in sequence (F I) chips form the sequence data path.

FCT - One floating condition code (FCC) chip processes all operands to create condition code status for traps, faults, and the PSL.

GATE ARRAY PART NUMBER/MODULE CROSS-REFERENCE

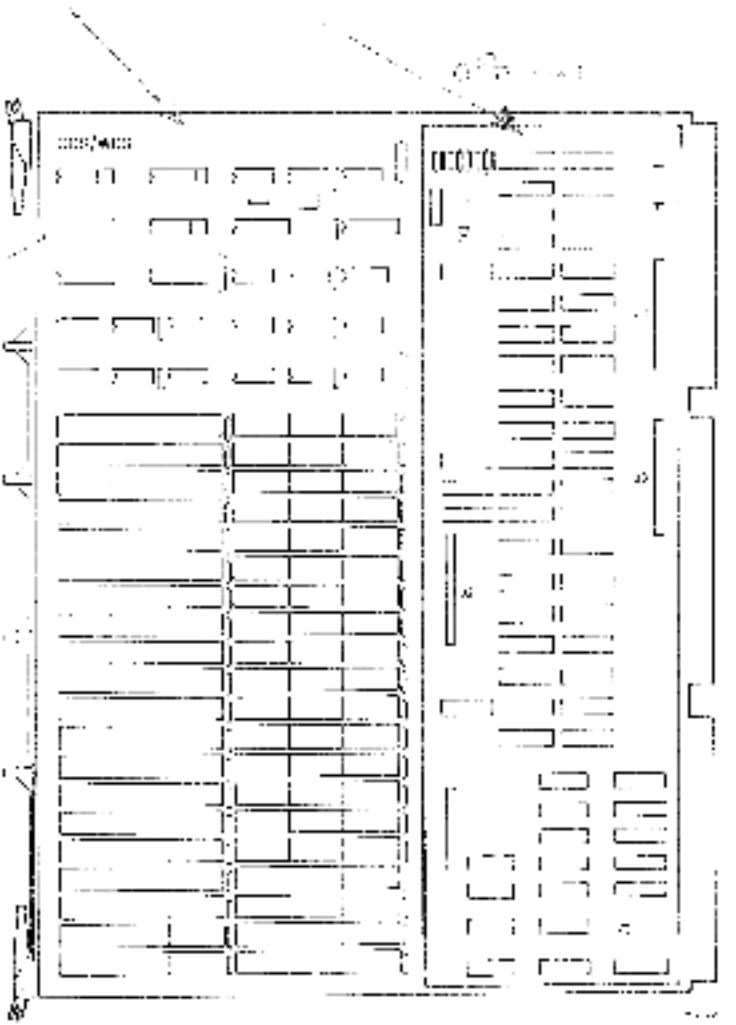
Chip Number	Chip Function	CUSTOM Part No.	Base Module	Pin Number	Pin Function
70033	LCD	19-14388	DPN	1	(PWR INT 2.2V)
70037	MPX	19-14381	MTP		<00,09,10,27>
				2	<01,26, 7, 30>
				3	<02,08,11,27>
				4	<03,21,29,27>
				5	<04,22,28,27>
				6	<05,23,31,27>
				7	<06,24,32,27>
				8	<07,25,33,27>
70046	ALL	19-14373	DPN	1	<13:08>
				2	<07:04>
				3	<11:06>
				4	<15:12>
				5	<19:16>
				6	<18:20>
				7	<17:17>
				8	<14:14>
70089	ATT	19-14[illegible]3	MJC	1	<07:03>
				2	<15:03>
				3	<23:15>
				4	<31:06>
70107	CTC	19-[illegible]6137	DPN	1	
70111	CCD	19-14035	CTT	1	(PWR)
				2	(GND,5V,12)
70112	CTC	19-14076	DPN	1	
	SIO			3	<TC/CTCB>
				4	<CT:CS>
70113	LCD	19-14084	DJC	1	<00,04,09,13, 16,20,27,30,32>
				2	<01,05,09,13, 17,21,25,29,33>
				3	<02, 6, 8, 14, 18,22,26, 28,34>
				4	<03,07,11,15, 19,23,27,31>
70114	SIO	19-14068	JPN	1	
70115	KIE	19-14069	JCT	1	
70117	TBC	19-14140	DPN	1	
70119	SIO	19-14091	DPN	1	

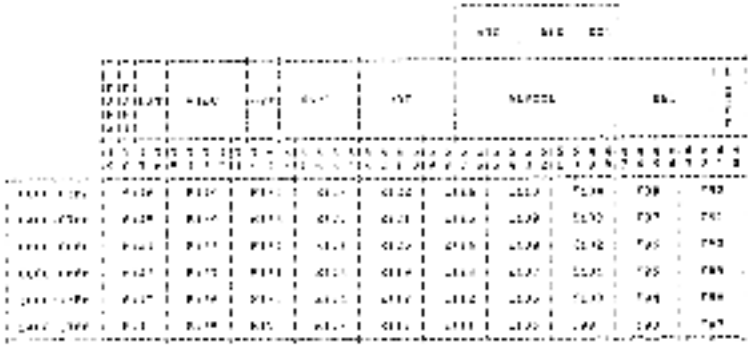
GATE ARRAY PART NUMBER/MODULE CROSS-REFERENCE (CONT)

Chip Number	Chip Mnemonic	DIGITAL Part No.	Host Module	Diagram Number	Data Ordering
DC6[illegible]	[illegible]	19-14592	[illegible]	1	<28,[illegible], [illegible]>
				2	<[illegible], [illegible]>
				3	<[illegible], [illegible]>
				4	<[illegible], [illegible]>
DC61[illegible]	[illegible]	19-14593	[illegible]		
DC620	TCS	19-14594	[illegible]	1	
DC621	[illegible]	19-14595	[illegible]	1	
DC622	[illegible]	19-14596	[illegible]	1	
DC6[illegible]	[illegible]	19-14597	[illegible]	1	
DC62[illegible]	[illegible]	19-14598	[illegible]	1	
DC625	[illegible]	19-14599	[illegible]	1	
DC626	[illegible]	19-14[illegible]	[illegible]	1	
DC627	[illegible]	19-14[illegible]	[illegible]	1	
DC628	[illegible]	19-14[illegible]	[illegible]	1	
DC629	[illegible]	19-14703	[illegible]	1	
DC630	[illegible]	19-14704	[illegible]	1	
DC631	[illegible]	19-14[illegible]	[illegible]	1 2	<[illegible]> <31:16>
DC632	[illegible]	19-14[illegible]	[illegible]	1	[illegible]
DC633	[illegible]	19-14707	[illegible]	1 2 3 4	<[illegible]> <[illegible]> <[illegible]> <[illegible]>
DC6[illegible]	[illegible]	19-[illegible]	[illegible]	1 2 3 4 5 6 7 8	<[illegible],[illegible]> <[illegible],[illegible]> <[illegible],11:[illegible]> <[illegible]> <[illegible]> <[illegible]> <[illegible]> <[illegible]>

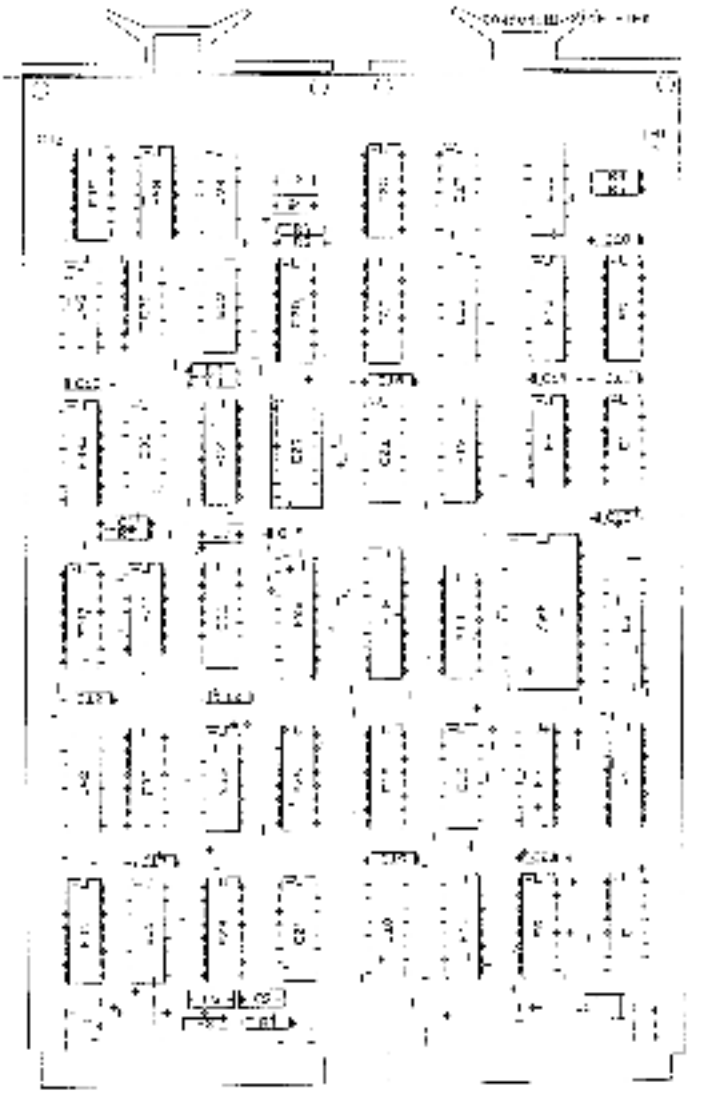
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	3 4 5 6	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	1 2 3 4 5 6 7 8	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible] 3	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	1 2 3 4 5 6 7 8	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	1	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	1	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	1	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	1	[illegible]

CCS MODULE WITH WCS (L0005)

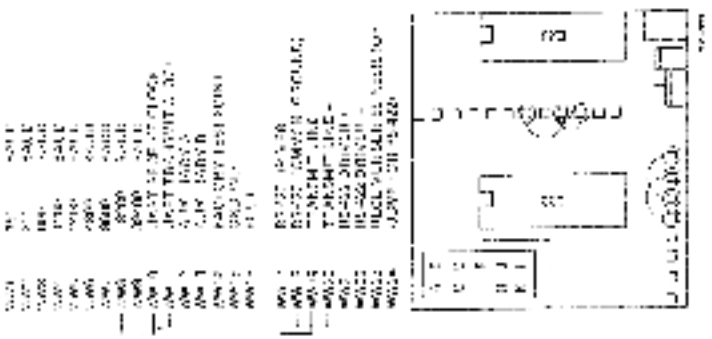


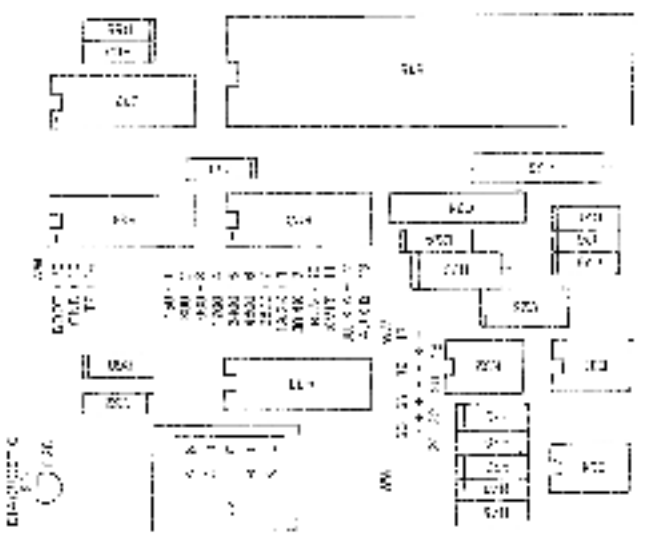


UET COMPONENT LAYOUT



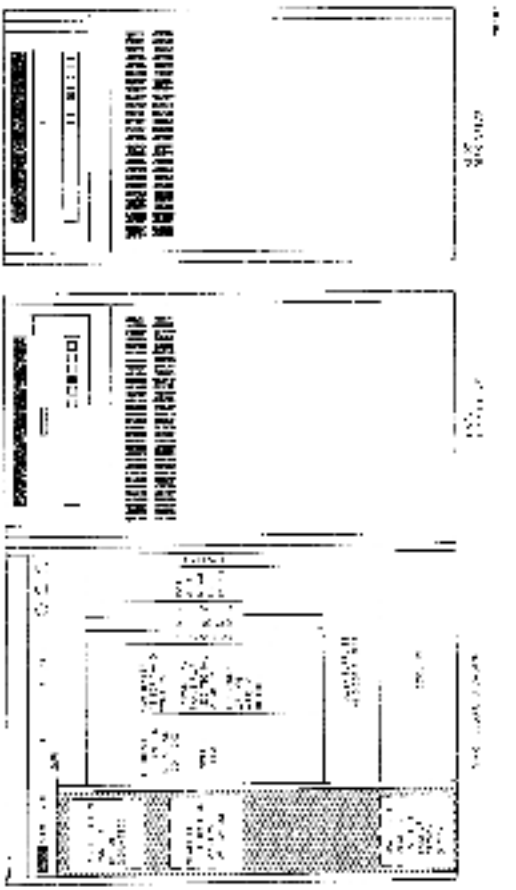
TU58 INTERFACE COMPONENT AND JUMPER LAYOUT



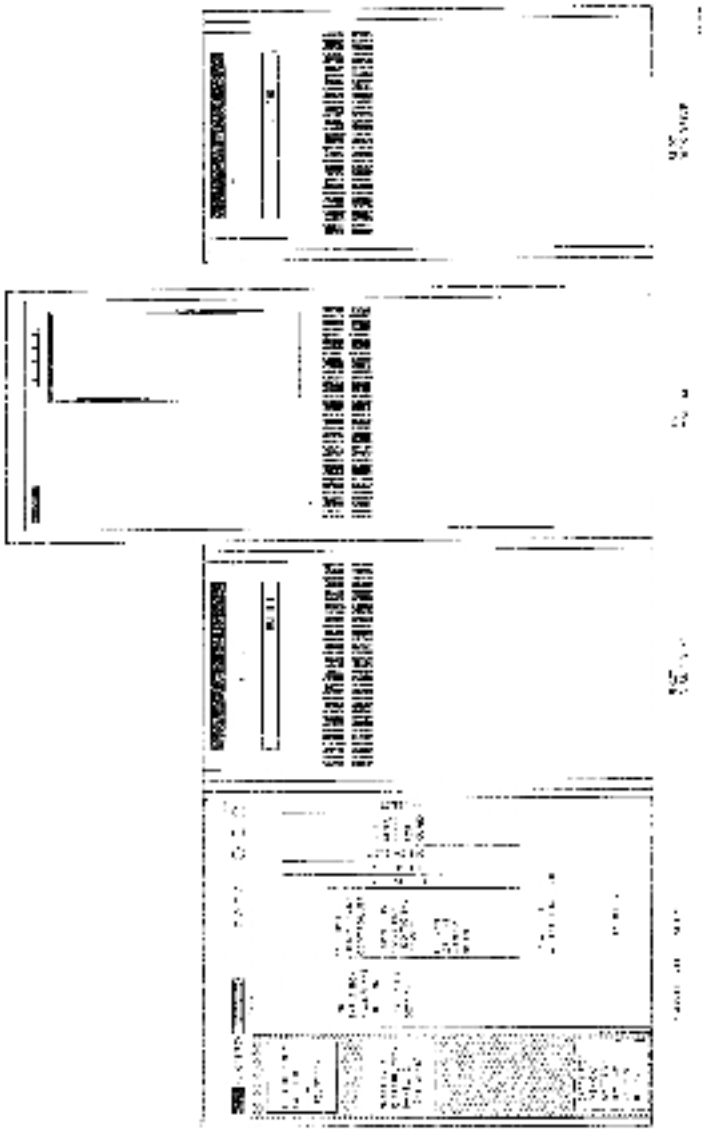


CHAPTER 4
CONFIGURATION AND CABLING

DUAL RK07 SYSTEM

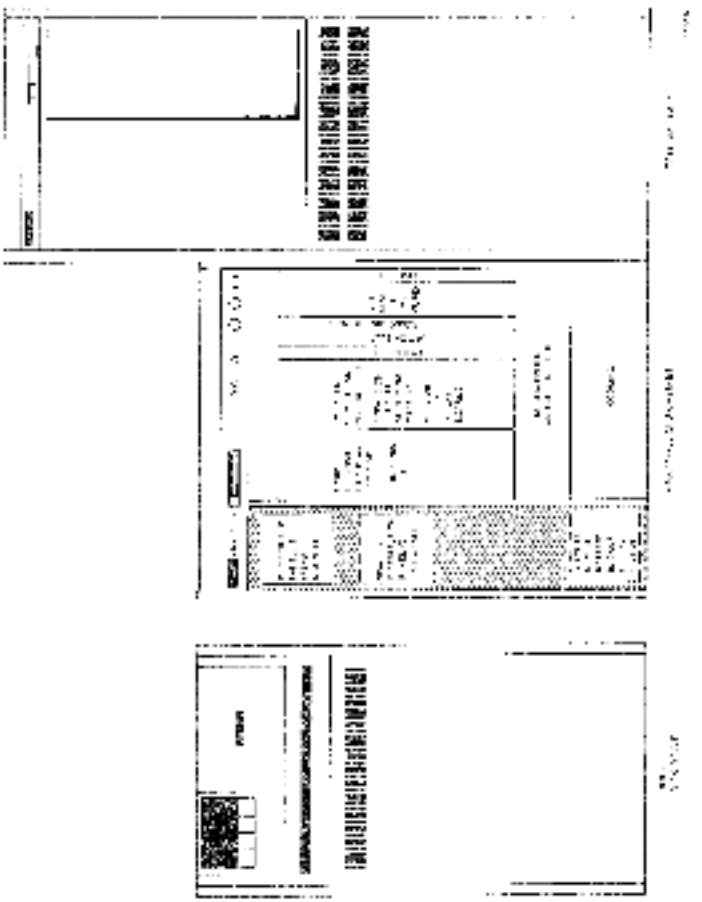


DUAL RK07 SYSTEM WITH TS11

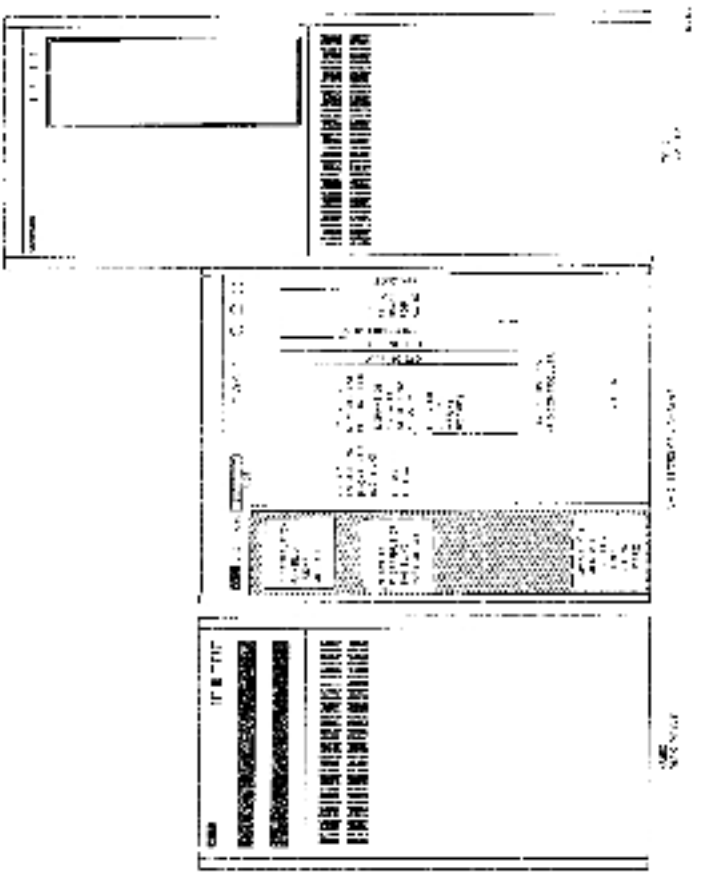


VAX-11/750 SYSTEM CONFIGURATIONS (CONT)

RM03/TS11 SYSTEM

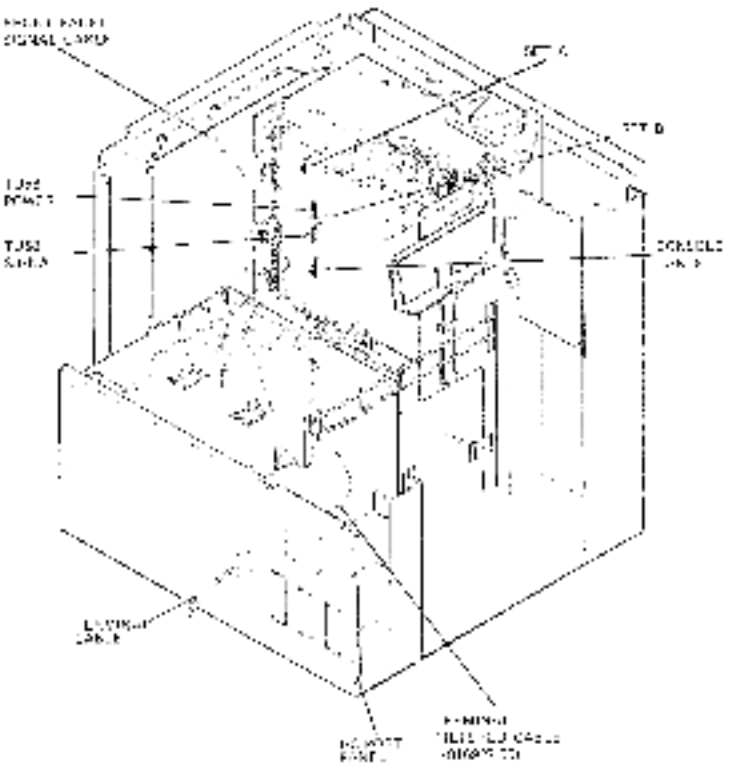


RM80/TS11 SYSTEM

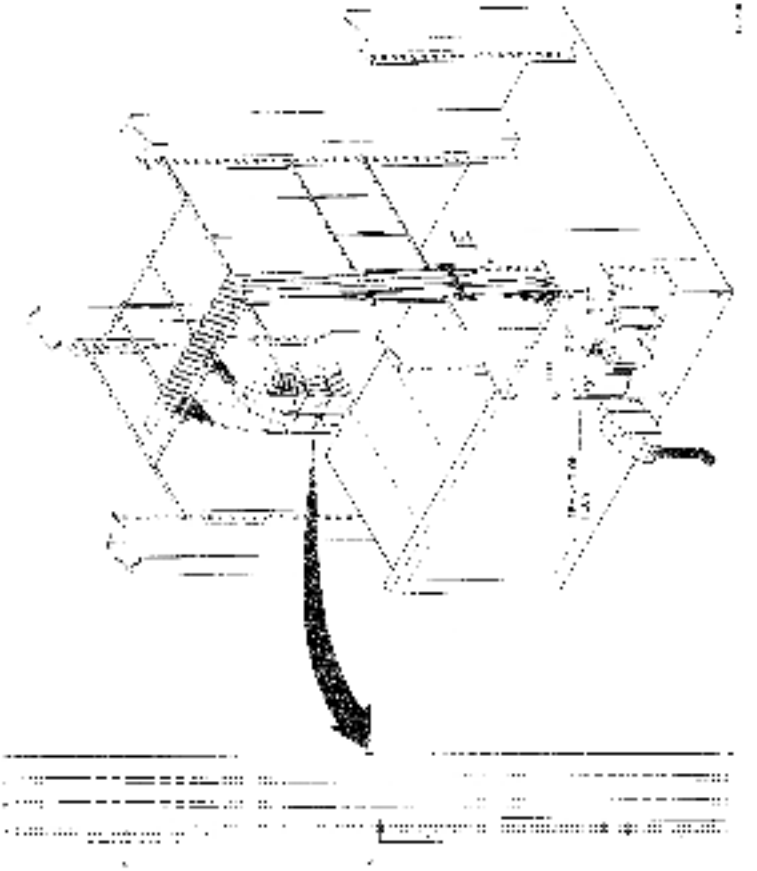


SYSTEM CABLING DIAGRAMS

TUBE AND FRONT PANEL CABLES

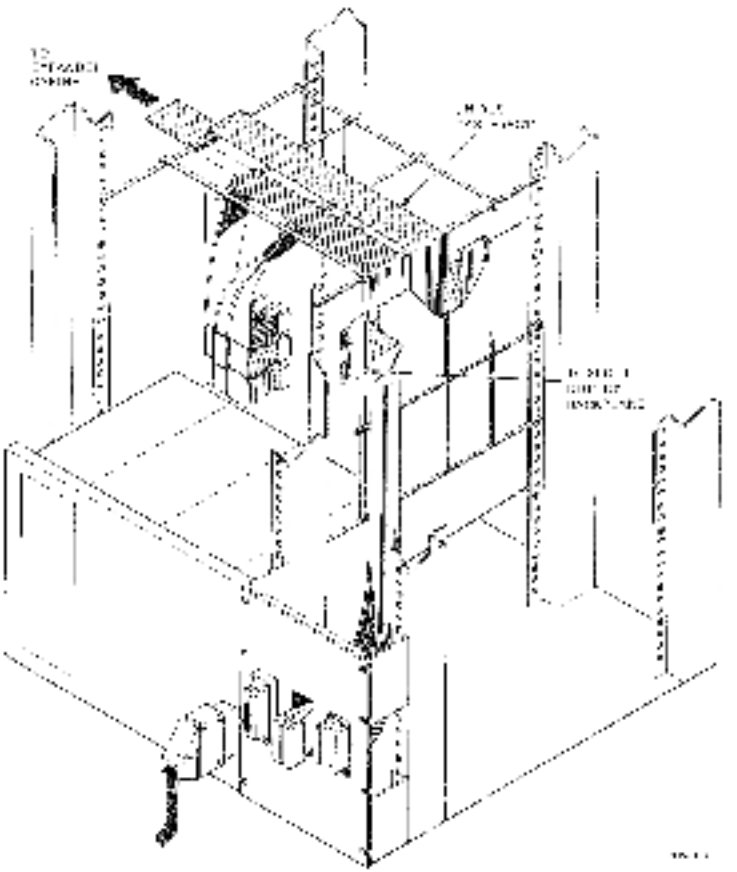


MASSBUS CABLES

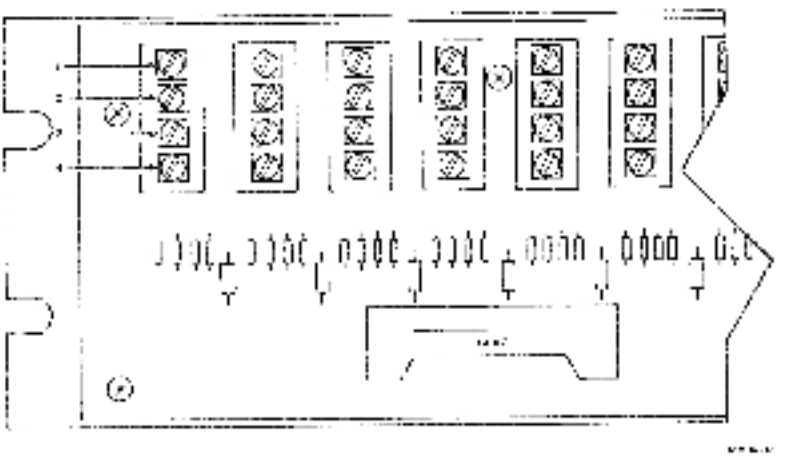


SYSTEM CABLING DIAGRAMS (CONT)

UNIBUS CABLES



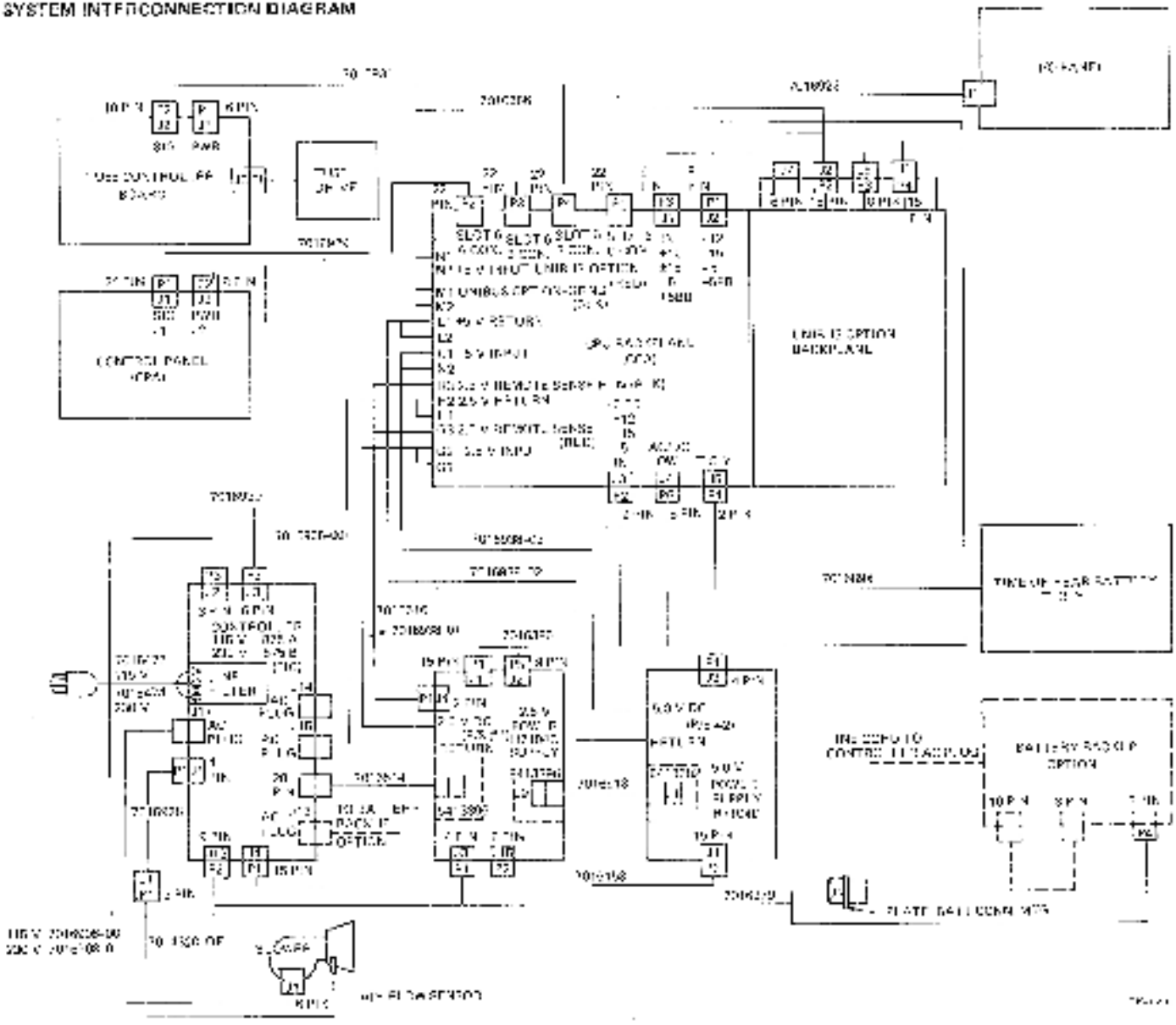
DZ11 DISTRIBUTION PANEL



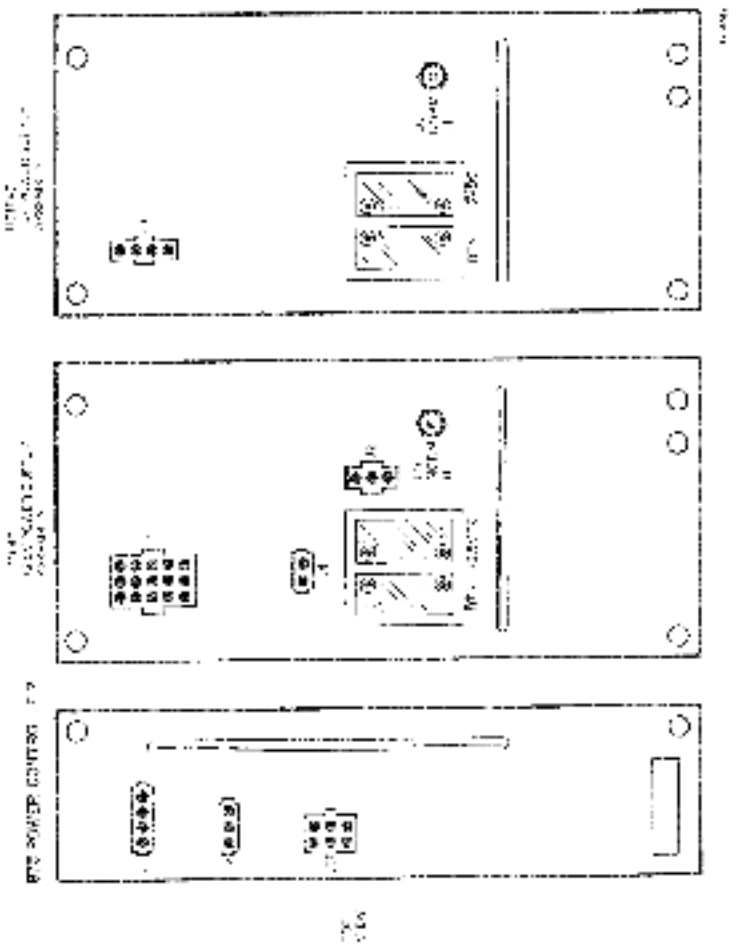
DZ11 DISTRIBUTION TERMINALS

Terminal Number	Signal Name
1	REC -
2	REC
3	XMIT -
4	XMIT +

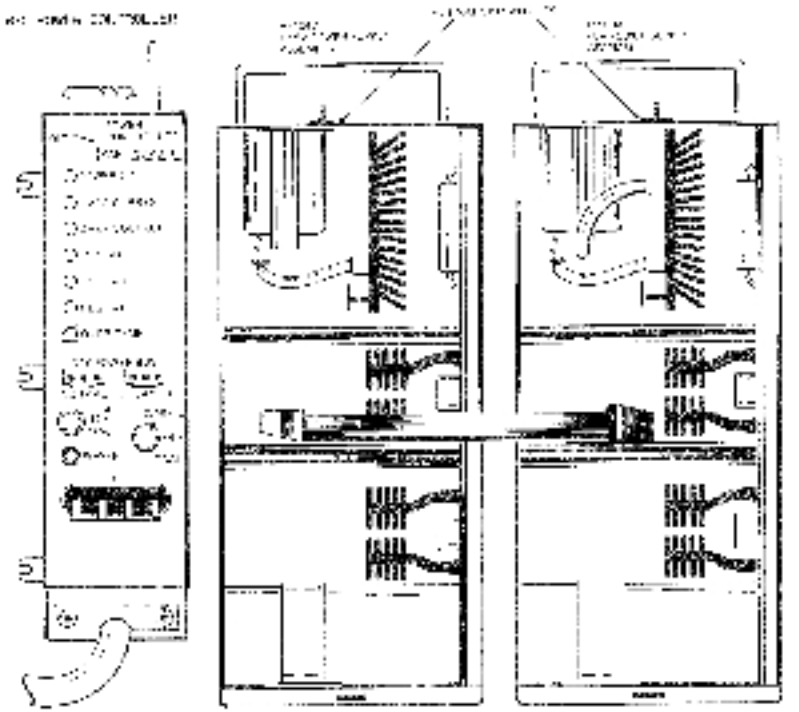
SYSTEM INTERCONNECTION DIAGRAM



VAX-11/750 POWER SUPPLY, PART 1



VAX-11/750 POWER SUPPLY, PART 2



The ac power controller panel consists of eight indicators, one circuit breaker, one REMOTE/LOCAL switch, one LED display, and the DIGITAL power bus connectors.

AC POWER CONTROLLER PANEL

Indicator	Definition
AC ON	The green AC ON indicator is on when the power monitor is functioning correctly. It is off if any fault indicator is on in the power supply system.
OVERVOLTAGE	The red OVERVOLTAGE indicator is on when there is an overvoltage condition in the +5 V or +5 V power supply. The affected power supply is indicated by the fault indicator being on.
OVERCURRENT	The red OVERCURRENT indicator is on when there is an overcurrent condition in the +5 V or +5 V power supply. The affected power supply is indicated by the fault indicator being on.
+5 V FAIL	The red +5 V FAIL indicator is on when there is a malfunction in the +5 V power supply.
+12 V FAIL	The red +12 V FAIL indicator is on when there is a malfunction in the +12 V power supply.
REG FAIL	The red REG FAIL indicator is on when there is a regulator malfunction in the +/-5 V, -12 V, or +/-12 V power supply.
OVER TEMP	The OVER TEMP indicator is on when there is an overtemperature condition in the +5 V or +12 V power supply. The indicator is not lit, however, when the system is shut down by an overtemperature condition in the controller.
POWER	The POWER indicator is on when the ac power cable is plugged into a live ac power source. It is on whether dc is on or off.

REMOTE/LOCAL SWITCH

Position	Definition
REMOTE	This is the normal operating position. No switches or power is applied to the power bus. Switches or power is controlled by the key operated switch on the CPU front panel.
OFF	No switches or power can be applied to the system.
LOCAL	Switches or power is applied to the system regardless of the position of the CPU power key switch.

CIRCUIT BREAKER (CB)

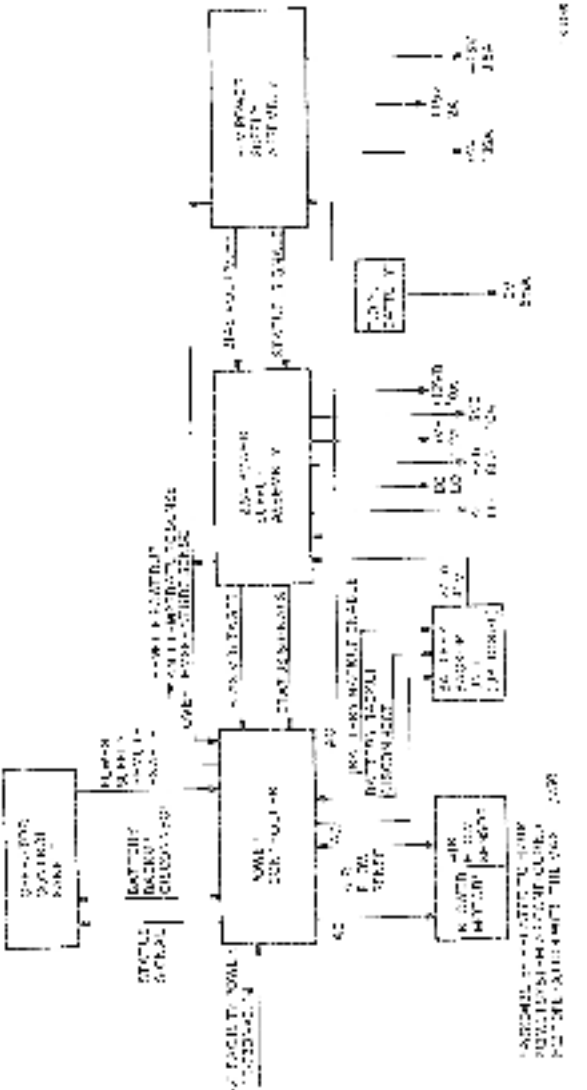
CB is the main circuit breaker to the power supply system. When in ON position, switches or power is applied to the power bus and switches or power is determined by the position of the REMOTE/LOCAL switch. When in the OFF position, no power cannot be applied to the power bus.

TWO POWER SYS

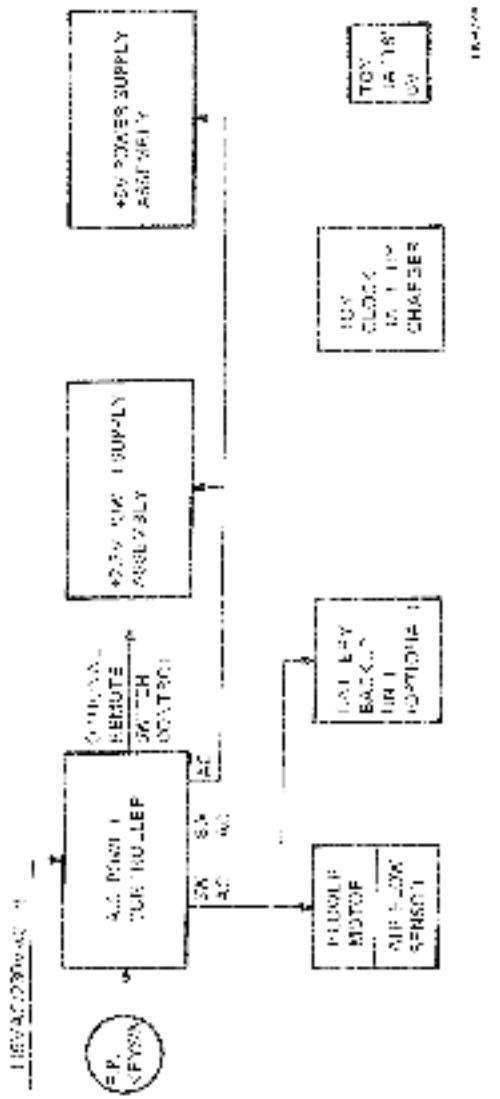
There are two BA11D(A) power box enclosures permit use between. Each enclosure provide a input to H9017F signal bus. An interconnection between the two power system are remote power systems.

There is a half-second delay on the over power bus shutdown circuit CB.

H7704 POWER SYSTEM BLOCK DIAGRAM

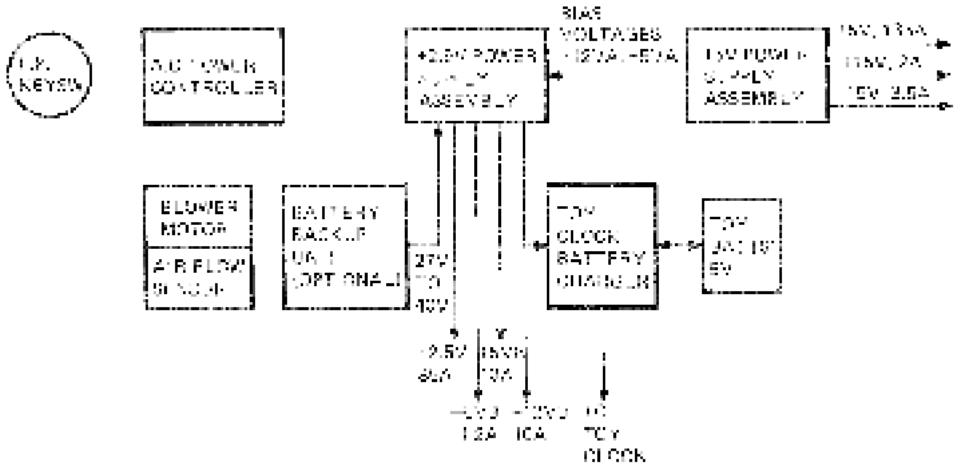


AC POWER DISTRIBUTION

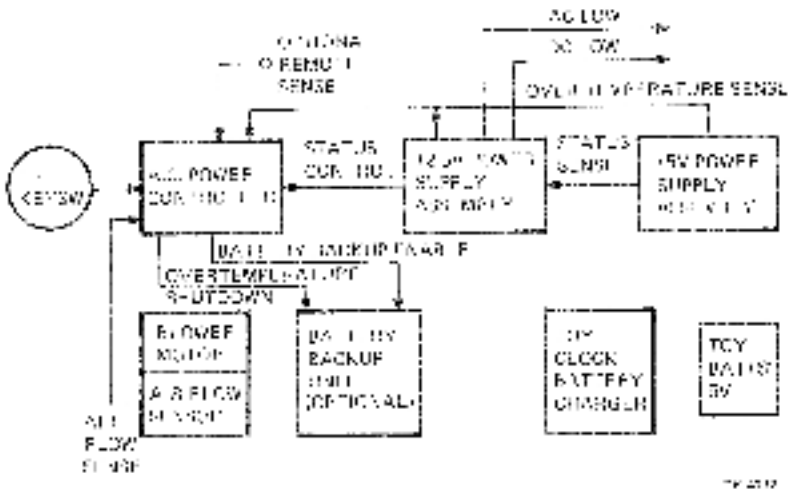


VAX-11/750 POWER SYSTEM (CONT)

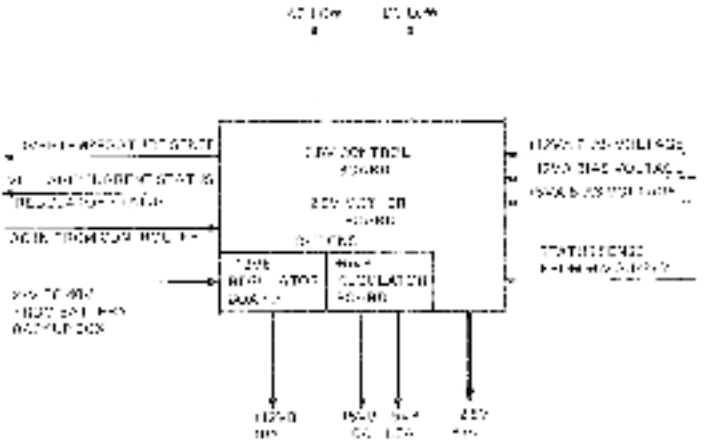
DC POWER DISTRIBUTION



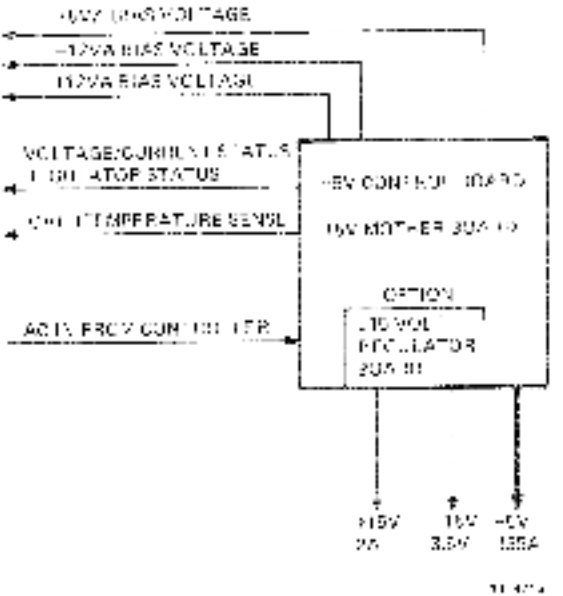
POWER SYSTEM SENSING



+2.6 V POWER SUPPLY BLOCK DIAGRAM



+5 V POWER SUPPLY BLOCK DIAGRAM



APPLYING SYSTEM POWER

Use the following procedure to correctly apply power to the system.

1. Ensure that the CPU's main circuit breaker (CB) on the AC power controller is off (down).

2. Verify that the two 3 position switches on the power supply are in the center position (up).

3. Place the REMOTE/LOCAL switch on the AC power control panel to the LOCAL position. The CPU power key lock switch on its front panel should be in the OFF position.

4. Connect the CPU cabinet AC power cable to the external AC power source. The power phase indicator on the AC power controller should now be on.

5. Move the CB1 circuit breaker to the ON position (up).

6. Power can now be applied to the CPU by the key lock switch on the CPU front panel.

CPU CABINET POWER REQUIREMENTS

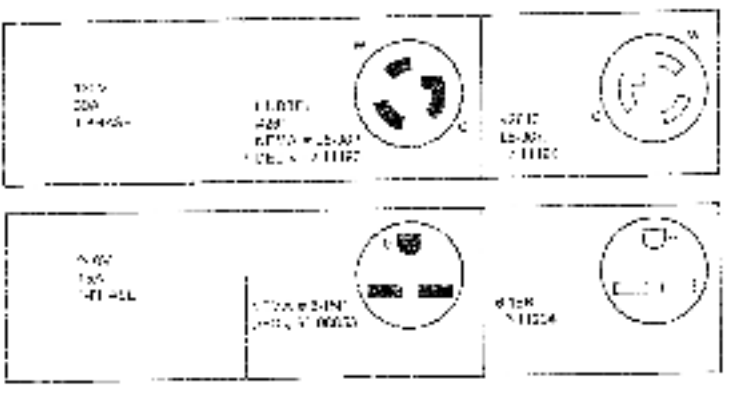
Single Phase	Nominal	Minimum	Maximum
VAC (RMS)	120	98	128
Phase to neutral	120	98	128
Phase to ground*	120	98	128
Neutral to ground	N/A	N/A	N/A
Hertz	60	47	63
Hertz	50	47	63
Current (amperes)			15.2 @ AC 60 V.a

Vac (RMS)	240	196	256
Phase to ground	240	196	256
Phase to phase	N/A	N/A	N/A
Neutral to ground	N/A	N/A	N/A
Hertz	60	47	63
Hertz	50	47	63
Current (amperes)			8.1 @ AC 196 Vac

CAUTION

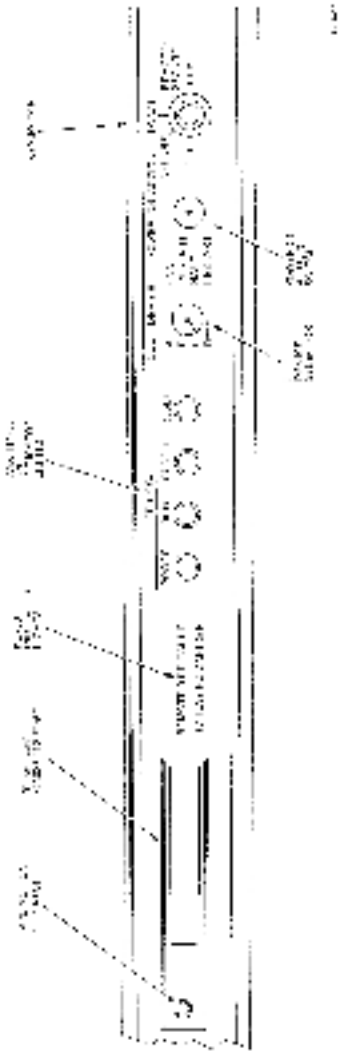
Expansion cabinets with separate power cables must be powered from the same AC phase as the CPU or damage may result to the equipment. Also, systems that are interconnected by cables and share the same logic and/or console grounding must share the same phase.

STANDARD POWER PLUGS AND RECEPTACLES



CHAPTER 5
BOOTSTRAPPING AND OPERATION

VAX-11/750 FRONT PANEL



CONSOLE PANEL INDICATORS

Indicator	Description
POWER	Indicates that the entire subsystem is supplied with correct voltages. If a VAX-11/750 processor has lost partial power and will not boot UMB and allows diagnostic testing.
RUN	The RUN is in program mode, running a user's memory program.
ERROR	Glows dimly to indicate a control store (CS) or MOS parity error. Fully lit indicates a double control store parity error and the clock stopped.
ADDR SELECT	Provides the ADDR switch value the system is waiting for value entered by the ADDR SW. ADDR SELECT light blinks the processor.
<NOTE 1>	Lower key switch in the SERVICE or SERVICE position.
20 MSEC	ON - on a 0.01 sec. Lights for one to two seconds after a console power-up as part of logic self-test.
RD LOCK	RDL lock command is processing RD cycle.
RD CANCEL	Parity signal detected from UBA.
Logic Module	Blinks on during a write to control store. Blinking indicates a load of micro code is in progress.

CONSOLE SWITCH FUNCTIONS

position	description
Break Key Detect	
OFF	Switched off power is removed from the system. Deactivation of power is still applied.
LOCK	Normal operations. All keys is applied and the console cannot be taken from the computer. Remote mode = The operator activities console commands. Program mode = The operator communicates with the system program. Ctrl/P and CTRL/B are not passed to the system program but are recognized by the console subsystem.
LOCAL SECURE	Normal operation as for LOCAL except as follows: Break = mode = The halt, baseload and the REBT switch are disabled. Program mode = CTRL/P and CTRL/B are ignored by the console subsystem and are passed to the system program.
REMOTE	System responds only to the remote terminal and to a remote CPU/P or CTRL/B to check processor status.
REMOTE SECURE	System responds only to the remote terminal and remote CTRL/P or CTRL/B are passed to the system program. The remote operator can disable the manual CPU4 state to remain only with the local operator.

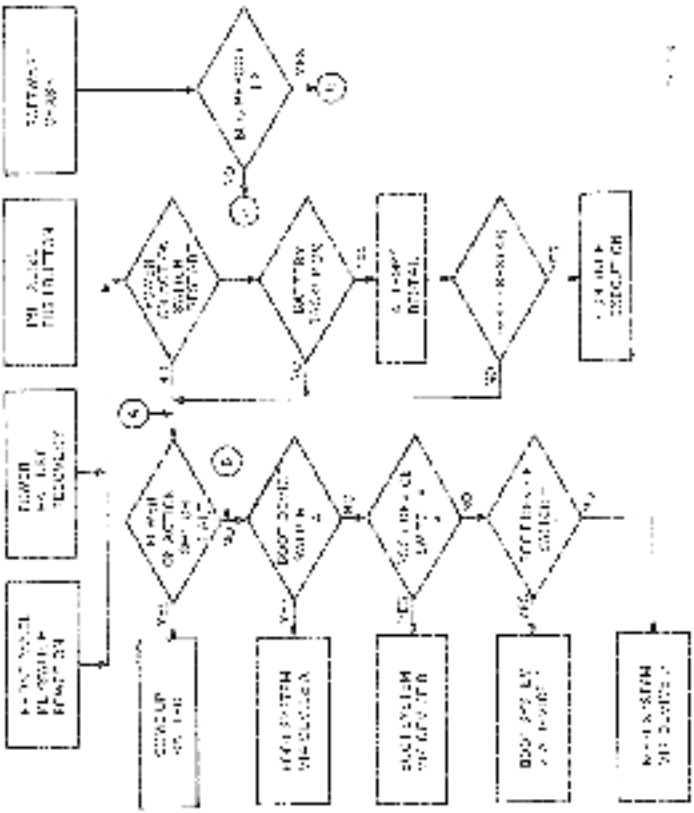
CONSOLE SWITCH FUNCTIONS (CONT)

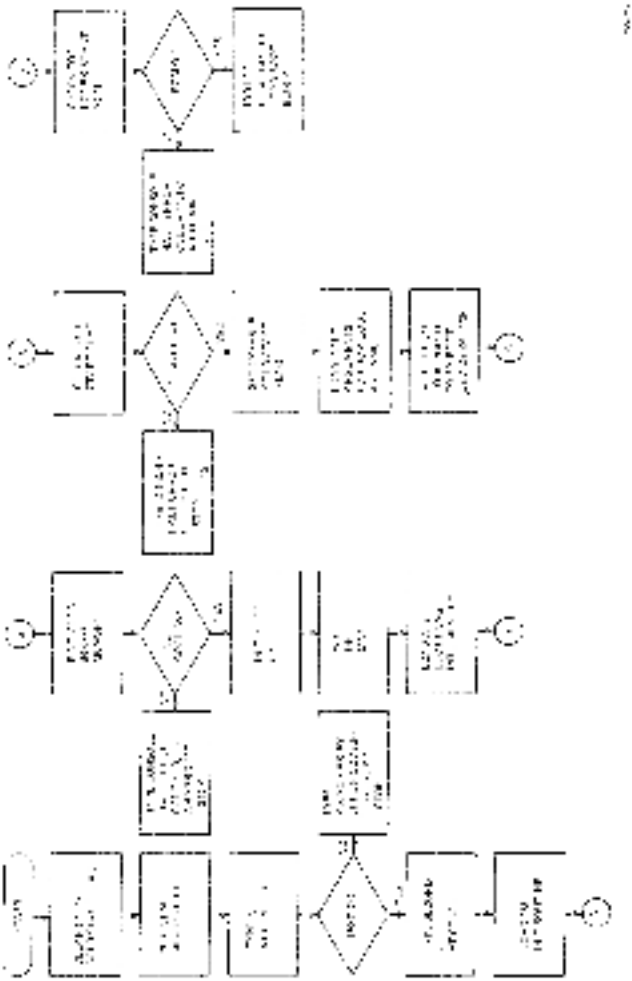
position	description
Power up option (cont)	
BOOT	The console subsystem performs a power up boot sequence and transfers control to the primary bootstrap program. A total reset, as when the RESET switch is pressed, is not performed. The boot is performed from the device selected by the BOOT DEVICE switch.
AUTOBOOT/HALT	On a power fail recovery, the CPU firmware initiates an automatic reboot or a halt to start programs via ODT. If the BOOT is valid, the program that is to be invoked operating system. If not, the system performs a bootstrap sequence from the device selected by the BOOT DEVICE switch.
SAVE	The processor halts and a valid is attempted.
RESTART/HALT	A restart is attempted as for BOOT. If unsuccessful, however, the processor halts.

Boot Device Select (Typical Devices)

Device A	TU58 dual DD
Device B	System fixed disk RP (DB = RM05/06, DM = RM/RP)
Device C	Alternate disk for B/U
Device D	Spare

CONSOLE BOOTSTRAP FLOW





BOOTSTRAP SEQUENCE

The following steps are required to obtain a running version of the VAX-11/750 processor.

1. The operator powers up the VAX 11/750 system.

2. The VAX 11/750 microcode detects power up and follows the power-up strategy selected by the console action switch located on the processor control panel.

   a. If the microcode manual option is selected, it will perform a reset of the internal CPU registers and halt.

   b. If the automatic option is selected, the microcode program control transfers to the console program.

      (1) Secure the console drives (RX0) as the console terminal.

      (2) Execute initialization commands to bootstrap the system by means of the console boot/dump (octal 50) or a user-supplied boot/dump device.

3. The microcode program looks up and executes the boot or dump primary (BOOT). This BOOT is 256 bytes and contains a small routine for the entry and a pointer. The boot routine loads block 0 from the bootstrap and jumps to the boot block entry. The boot routine has the boot block as the BOOT program so as to read arbitrary blocks from the bootstrap device into memory.

4. The boot block contains the logical block address, size, and entry address of the program to be executed in the bootstrap process. This program can be either (1) autoloader (VMB.EXE, when the installed system is the VAX/VMS system, or (2) VMB.EXE, when the hardware device is the console disk).

   a. If the bootstrap operation is performed from the console floppy disk, the program VMB.EXE, the user system image, and the system parameter file must be loaded and start VMS.EXE.

   b. If the bootstrap operation is performed directly from the system disk using VMB.EXE, the microcode program performs the equivalent of a CONNECT to the system parameter file, the user system image, and the executive image, and starts VMS.EXE.

7. VMSKRN is the primary bootstrap program. It contains CPU independent code and CPU dependent routines. It also contains a set of primitive run-time system services for all base-level device and I/O facilities. File system facilities and creation of multi-user environments are left to higher level software. VMSKRN performs the following steps:

   a. Saves the register values and some values calculated from the register values in the restart parameter block (RPB).

   b. Reads the system identification register to determine the processor type and to select the table of appropriate processor-dependent routines for the current CPU.

   c. Tests the IO circuit and pattern of memory. A page frame number (PFN) bitmap is constructed. On the MicroVAX I, the first page, memory is tested for proper, repeatable parity errors. VMSKRN establishes PFN for RPB, I/O buffers by memory present at all memory controller and CPU address types.

   d. Build up any other values, one of the following errors:

      (1) A fatal block in the designated initial system routine (sysa) will be sent into memory and given control.

      (2) A file named [SYSLIB]SYSBOOTLOG.EXE will be sent into memory and given control.

      (3) A file named [SYSMAINT]SYSBOOTLOG.EXE will be sent into memory and given control.

      (4) A file specified by the user to perform an action will be sent into memory and given control.

6. SYSBOOT.EXE is the standard secondary bootstrap program. It provides initialization services for the creation environment. SysBoot performs the following steps:

   a. Asks current customized initial file SYS.EXE.

   b. Looks up the [SYSLIB]SYS. device driver file and system information about it.

   c. If register or map is invalid, prompts the user to specify current system parameter work ages. The user can change the startup command procedure name and modify system parameters using SET or [privileged] parameter file, and parameters before the "current" parameters on the next startup operation.

   d. Sets up SCB, SYSPHD, PCB, and PTE also startup time.

   e. Reads the relocated executive file into a virtual memory.

   f. Activates and initializes the VAX/VMS system.

SOOTSTRAP SEQUENCE (CONT)

7. The system initialization process consists of four stages: INIT, SYSINIT, SYSCONFIG, and SYSLOGON.

   a. INIT is part of SYS.SOS. It performs the following:

      (1) Reads in mapping and size of the I/O system space

      (2) Prints the system announcement message

      (3) If requested by means of the Boot Block, stops at the ADB/DB breakpoint

      (4) Initializes the system I/O pages

      (5) Establishes available Virtual pages (VPN) through use of SHM in the first pass SYSP

      (6) Initializes the system page table for paged and unmapped pools

      (7) Initializes I/O structures using the list of present configured generations in VMB.SYS. To this initial condition of empty adapter control space (only one number of pages which is supplied) and as time permits specific routines to allocate and set up data structures and to perform bus adapter hardware. In addition, for UNIBUS adapters, the 22 bit I/O page of the UNIBUS is mapped.

         Data structures allocated are:

         NAPRn - adapter control block
                 channel request block
                 interrupt dispatch block

         UB.BCB - UNIBUS control block

      (8) Performs ADD/MAP/ALL commands initialized on disks

      (9) Transfers the initial VMB.SYS system device driver into language pool and passes the I/O vector array and boot device controller status register as a serial listing that previously addressed in the PFN.

(.01) Loads the CPU dependent code into the kernel memory pool and links it into the system

(.11) Loads the terminal handler into kernel memory pool and connects its interrupt vectors. Calls the driver init for the system device into memory pool, connects its interrupt vectors, and performs the read of the system disk. The path for the system disk device name is as follows:

device name	Decodes the primitive driver where the device name is stored.
controller	The controller designator is "A", "B", or "C" for the first, second, or third occurrence of this kind of adapter. For example, if the adapter at the system device is the second occurrence, the number is B. (Note that for a generically configured system, it is possible to use the ALTCONTROLLER parameter to override the controller number specified in the SYSGEN.) Otherwise, some data is required from controller to control the system disks across multiple buses.)
unit	Passed from VMB.EXE (qio) request.

(.12) Adds the processors of the present primary (for validity chk) to the universe list

(.13) Performs initialization of privileged drivers

(.14) Makes connection code of init into the pool and executes it. The completion code deallocates space occupied by init (all options by /PROTECT) as the free pool list. The connection code tries to the primary, otherwise a fatal error is (SYSINIT, ACCVIO) occurs as shortcut.

BOOTSTRAP SEQUENCE (CONT)

5. SYSINIT performs the following:

   (1) If necessary or requested, prompts for the time of day

   (2) Writes back system parameters to SYS.DAT

   (3) Creates some logical names

   (4) Sets up swapping and paging files

   (5) Installs the VMS=I/O RMS image and system services (i.e. the supervisor system routines)

   (6) Mounts the system disk (and private volumes)

   (7) Creates the job controller, STARTUP, and [SYSLIB]

   (8) Creates the STARTUP process

6. STARTUP reads input from the startup command procedure, which usually is:

   (1) Creates logical names

   (2) Runs SYS$SYSTEM:SYSGEN to configure to I/O system

   (3) Installs known images

   (4) Invokes [SYSMGR]SYSTARTUP.COM

   (5) and so.

7. SYSTARTUP.COM is an empty command procedure distributed by DIGITAL. The system manager can edit SYSTARTUP to perform site specific startup functions.

8. SYSINIT is run by STARTUP at IPL 0 (user level, RSX11M)

   a. Provides for dynamic loading of new drivers to Executive. (The operator can install drivers are permanently part of the executive image)

   b. provides for a variable of new parameter files (not even as mounted disks)

   c. Creates pagfil, swapfil and resets CWP filer

INPUT ARGUMENTS

The special registers receive the following input arguments from the console software:

R1 - Maximum page address of a PAGEFILE.SYS header (MBZ) (clear otherwise) specified to the BOOT command.

R2 - Page cell address of the INIT.SYS file page descriptor with a CNTRLC or CNTRLY bit set, signalling operation to the BOOT command.

R3 - Device unit number to which subsequent operations are to be performed.

R4 - Software boot control block (0 clear) otherwise specified in the BOOT command.

R5 - Block address + 1280 of the work block of pool memory.

BOOTSTRAP SEQUENCE (CONT)

SOFTWARE BOOT CONTROL FLAGS

Bit	Hex Value	Function
7	80	Initialization flag. Prevents the system from attempting to boot a previous version of the operating system.
	A7	Debug. This bit is passed through to the ROM and causes the boot of the executive program to be initiated in the console monitor.
2	04	Initial breakpoint. If this flag is set, and the executive debugger code is included (SYSGEN option), a breakpoint will occur immediately after executive data tables are initialized.
3	02	Not used on the VAX-11/750.
4	01	Diagnostic boot. This flag causes a boot by the code for the diagnostic subsystem.
5	00	Bootstrap breakpoint. This flag causes the bootstrap to stop at a breakpoint after performing necessary initialization.
6	00	Image header. If this flag is set, the image header from the first block of the boot file will be read. Otherwise, control will transfer to the first byte of the boot file.
7	00	Memory test inhibit. This bit inhibits the testing of memory during bootstrapping.
8	00	Bit 8 reset. Causes the bootstrap to default to the name of the boot file.
9	00	Halt before transfer. Causes a halt instruction to be executed before transfer to the secondary boot file. This option is useful for debugging.

SYSGEN>BOOT

>>> B/1 DEVA

>SYSBOOT> SET START "PROG" ! DEFINE SAT BOOT

SYSBOOT> ?? DISPLAY (CURRENT)

SYSBOOT> CONT (CONTINUE) TO BOOT VMS

>>>B/10 DEVA BOOTING FROM DUA0 ON DEVICE DEVA

DS>

>>> B/100 DEVA

BOOTING FROM DEVICE DEVA

BOOT IMAGE ACTIVATED

SYSTEM IMAGE ACTIVATED

!! A WARNING TEXT !!

DEVICE DUA0

VMD PRIMARY BOOT FAILURES

```
Boot is not possible. Halt CPU.
The CPU indicates a fatal error and the system is stopped.
```

STOPPED - Unknown processor	Processor type not CPU is not a VAX 11/730 or 750. Place CPU switches in HALT, and cycle power to CPU.
Fault or Exception Occurred	Indicates that one of the following exceptions occurred: 1. Access violation 2. Breakpoint to KDB 3. Reserved operand 4. Bugchk trap 5. Bugchk (PANIC)
Error in Microprogram Machine Check	Indicates that a machine check occurred. Check for console halt or any other hardware problem. Look for a flashing error LED on CPU.
STOPPED - Console Error	Self test failure. Check port cables and console system disk is okay for booting.
Error in Loading Secondary File	VMB cannot find [SYSEXE]SYSBOOT.EXE or it has a IO or QIO error. VMB cannot find SYSBOOT.EXE.
STOPPED - Invalid PFN or Secondary Bootblocks	Indicates that SYSBOOT.EXE is not contiguous or invalid file. Reapply or reinstall.
Secondary Error reading Boot File	Indicates a problem reading boot file from disk by SYSBOOT (VMB error code).

CONSOLE COMMANDS

```
Command       Description

G[0-9/?       Enter sense + hide, [close >>] prompt.

CTRL/D        Enter DOS or Unix mode, remove DOS prompt.

>>>E          Examine register

>>>?          Register summary

        Public:

        F [(/M)[FLSH]<SEG>[:OFFSET](COM)

        D [(),A, <I>,(/P),ADDR1 [(/ADD)]ADDR2(CR)

        Qualifiers:  /B    Set size to byte
                     /W    Set size to word
                     /L    Set size to long[word]

                     /P    Physical address space
                     /V    Virtual address space
                     /T    Job    "     "
                     /U    CPU    "

         Address name      The number or physical or virtual
                           address
                   <CR>    Next address
                   <LF>    Next address (Display only)
                   <UP>^   Prev

>>>A          Assembler halt command

>>>I          Processor Initialize command; Invalidate Prefetch
              Queue, Causes Processor INIT and RESET Pins.

>>>P          Trap command; Trap microverify routine.

>>>R name     Start command; performs initialize functions, stores
              specified box address in Job Control Register (JCR).

>>>R(CR)      Start command, performs initialize functions and
              starts program at current contents of the PC.

>>>S          Synchronize command; starts program at current contents
              of the PC without an initialize first pass.

>>>U          Single-step the program a set box PC is loaded.

>>>X          Exit command; boots from device as noted on front
              panel device switch.
```

```
Command         Description

    Syntax:

    > [QUALIFIER]/X,DDDD[=??]

    Qualifiers:     /?              Toggle console or microcontroller
                    /Boot_number    Enable boot control block in RAM
                    /D              Device
                    /F              Adapter code
                    /U              Unit number

    Examples:

        >>>/D 780F      Set device specified by F780.

        >>>/X 2000      Boot from an operating system and initial
                        bootstrap.

        >>>/D 780F      Type hex digit hex number to DA and boot
                        device specified by 780F.

        >>>/X/F 3220            Change 3807 to 3C.
        >>>/D 1003 7810:4A2F    Change location of code to 1675.
        >>>/F 27                Examine route block a register.
        >>>                     performs processor diagnostics.
        >>>/X/S DKA0            Boots diagnostic supervisor from
                                DKA0, although allows error.

>>>X     Primary bootstrap command routine used by
        translating the specified boot device into
        communications with the console to transfer data
        between itself and memory.

        Primary Load:

        >>>X/DEV[A,B,C,D][0,1,2,3][0,1,2,3][0,1,2,3][0,1,2,3][0,1,2,3][0,1,2,3][0,1,2,3][0,1,2,3]

        Address         Starting address of the load
        Count           Number of bytes to be transferred (unsigned
                        16-bit hex number, bit 0=1 is octal
        Checksum        Twos complement checksum of the received
                        string
        Data            bytes of binary data
        Checksum        Data compatible checksum of the data

        Primary Alive:

        >>>X/DEV[A,B,C,D][0,1,2,3][0,1,2,3][0,1,2,3][0,1,2,3]

        Address         Starting address of the alive
        Count           Number of bytes to be transferred (unsigned
                        16 bit hex number, bit 0=1 is octal)
        Checksum        Twos complement checksum of the received
                        string
```

CONSOLE COMMAND ERROR CODES

Code	Description
020	Memory exhausted or corrupt (stack, access violation (ACV), translation not valid (TNV), machine check, bug error, OS parity error, PCB/KCB parity error)
021	Error in command to VMB or SCP
022	Checksum error on ARC load or reload
023	Attempt to boot from unconfigured device (DB, DU, DV, or DB)
024	Initialize not A) at C) at F) boot command

CONSOLE HALT CODES

Code	Description
0	User console command executed
02	FPE/P halt or single recalculation exec (>>>H)
04	Interrupt stack not valid
05	Double bus mcheck error halt
06	Processor halt instruction executed (>>>H)
07	Memory C (0) = 0, halt at reset
08	Memory C (0) = 0, ACL disabled or not present
0A	Change mode instruction executed at interrupt stack
0B	Change mode instruction executed, kernel stack not = 0

BOOT, POWER UP, AND INITIALIZATION HALT CODES

Code	Description
00	Halt instruction on console boot command, boot ROM, or VMB.EXE image
01	Power-up, memory test ERR, /HALT at ABSENT/HALT
02	Power-up, boot start flag value ERR at ABSENT/HALT
03	Power-up, primary CPU not found ERR or memory
04	Power-up, CPU boot, and/or parameters bad, HLT
05	Power-up, bad check sum during boot initialize
06A	Power-up with /HALT at HALT position
FF	Microverify boot failed

MICROVERIFY ERROR CODES

```
Code    Text    Test Name/Error Message

'A'             RAM Test
        280     Bad bit in DATA of ADDRESS
        281     Bad bit in DATA of MASK

'B'             ROM Test
        9A1     Bad bit in CHKS
        9A2     Bad bit in CHKP

'E'             Checksum bad bit test 2
        B01     Error clearing DLMP
        B02     Error filling STEP with ones
        B04     Error clearing CPU
        B07     Error in ILL. OPC also clear
        B08     Error clearing LOP
        B0B     Error filling ILL with ones
        B0L     Error clearing PTEMP
        B0Z     Error filling "MEMP will also

'F'             Misc explicit address test
        B01     Error addressing MEMP3
        B02     Error addressing MEMP2
        B04     Error addressing MEMP1
        B08     Error addressing MEMP0
        B10     Error addressing PTEMP0

'I'             Interp explicit address test
        B41     Error addressing DIMP7
        B42     Error addressing DLMP6
        B44     Error addressing DLMP5
        B48     Error addressing DLMP4
        B50     Error clearing MPTMP

'J'             IPR explicit address test
        B41     Error addressing LMP0
        B42     Error addressing LMP1
        B44     Error addressing LMP2
        B47     Error clearing YPR0
        B50     Error clearing YPR0

'L'             TPR explicit address test
        B01     Error addressing R0
        B02     Error addressing R1
        B04     Error addressing R2
        B08     Error addressing R4
        B10     Error addressing R8
        B2E     Error clearing TPR  R15
```

MICROVERIFY ERROR CODES (CONT)

Code	8042	Test Name/Error Message
'0'		A8/16/GSR bit test
	0E1	Error in A8 (E108)
	0E2	Error in GSR(E110)
	0E7	Error in A8
	0E7	Error in GSR
'1'		Cache A8 bit increment test
	111	Error reading one byte from Ch. 0
	112	Error reading 2 bytes from A0 or incrementing A0 by 1
	114	Error loading an unaligned long word or incrementing A0 by 1
	117	Error incrementing A0 by 4
'2'		RAMP/D-TAG test
	121	Error reading DE107 RAM address 1
	122	Error loading/reading DCAM
	124	Error reading DE105 RAM address 2
	127	Error loading/reading TAGM
	128	Error reading DE105 RAM address 1
	12B	Error loading/reading TAGM
	137	Error loading DE105 RAM address 4
	138	Error loading/read to RAMM
'3'		RAMP/D-TAG test continued
	141	Error reading DE105 RAM address 2
	142	Error loading/reading RAMM
	144	Error reading DE105 RAM address 4
'4'		Cache parity error test
	'41	Failed to get cache parity error
	'42	set machine check error summary register
	.48	set machine check (E)1420)
'5'		TB parity error test
	151	Failed to get group 0 TB parity error
	152	Bad TB group parity error register
	154	Bad machine check error summary register
	158	Failed to get group 1 TB parity error
	1A2	Bad TB group parity error register
	1A4	Bad machine check error summary register
'6'		Control store parity error test
	161	Failed to get control store parity error
	162	Error in control store parity error
'7'		Cache test
	'61	Error writing cache with data. Location not initially 0
	'62	Error writing cache with data. Unable to write 0000

The following lists a description of the commands that can be entered at the transaction source program. They are listed in alphabetical order.

BOOT

B[{device-name}]

Bootstrap the system from the specified device. If you do not specify the device name, the system is bootstrapped using the default bootstrap procedure (DSKB0.CPD). Note that you cannot enter the B command procedure to the BOOT command and you cannot specify this command within a command procedure.

DEPOSIT

D location=value[,...]

Deposits a value in the specified location. The location is interpreted according to the location and size qualifiers. The location qualifiers can be specified as follows:

/G	general variable
/I	internal process register
/P	physical memory

The size qualifiers can be specified as follows:

/B	byte
/W	word
/L	longword

If you do not specify the location and size qualifiers, the default values established by a previous command are used.

EXAMINE

```
EX [qualifier[,qualifier...]] location
```

Displays the contents of the specified locations. The location is interpreted according to the location and size qualifiers; the location qualifier can be expressed as follows:

```
    /G   current radix
    /I   instruction mnemonic
    /P   physical address
```

The size qualifiers can be expressed as follows:

```
    /B   byte
    /W   word
    /L   longword
```

If you do not specify the location and size qualifiers, the default values established by a previous command are used.

HELP

```
HELP [topic [subtopic...]]
```

Displays the list of topics available for help. You cannot specify this command within a command procedure.

LOAD

```
LOAD file_spec [/OPTIONS=value]
```

Loads a file from the boot device into memory, starting at the address specified with the /START qualifier. If you omit the /START qualifier, the file is loaded into memory beginning at the first free address.

START

```
START [value]
```

Transfers control to the value specified. You generally use this command with the LOAD command.

@file-spec

```
@file-spec
```

Executes the name of the command procedure specified. You cannot specify a wildcard character. File-type of .SCP is assumed if you do not specify a file-type of your own. Alternately, you can specify a file-type of your own.

CHAPTER 6
RDM AND MICRODIAGNOSTICS

RDM INSTALLATION

RDM HARDWARE

The following hardware is supplied with the revitrol RDM options:

Option	Hardware (Qty) part	Image
RDM8-V2	RDM module - 1080 Expansion screw Diagnostic LED (1 Serial interface cable (PN 16923) Volume interface cable (PN 954) 2) RDM8 Options Card Guide (EN-RDM8-IG)	Top
RDM8-EA	RDM module - 1080 Programming kit Ethernet connection cable (PN-16923) Three options Card Guide (EN-RDM8-IG) (Model 954 cable is supplied by customer)	Cable

PREINSTALLATION

The following steps must be taken before making the initial network connection to the RDM.

1. The customer's network administrator must be contacted and a system configuration worksheet supplied to the Revitrol Diagnostic Center (RDC).

2. The customer must supply their Internet Service Provider with the following information so the modem can be used.

   a. Model number

   b. Manufacturer's number

   c. Ginger modulation number

   d. FCC registration number

   e. Voice term circuit customer type designation

   f. Telephone number of line/trunk to be initially tested

   Are to supplied by the Customer dialtick choice.

INSTALLATION PROCEDURE

The following is the procedure for installing the DDT module and hardware.

1. Perform an orderly shutdown of the operating system and turn power down the system by turning the console power key switch to OFF.

2. The DDT module is supplied with the console port baud rate set to 300 baud by jumpers on the module. Earlier modules had a switchpack in that location originally removed by the jumpers.

   The baud rate is selected by jumpers W0, W1, W2, W3 (up to) by certain positions (W3=1 through W0 as shown below.

   To attain proper jumper W-3 on the opposite end of the module from the baud rate jumpers.

Baud Rate	W3 or 7(BR-1	W2 or 7(BR-2	W1 or 1/BR-3	W0 or 1/BR-4
300	On	On	Off	On
600	Off	On	Off	On
900	On	Off	Off	On
1,200	Off	Off	Off	On
2,400	On	On	On	Off
3,600	Off	On	On	Off
4,800	On	Off	On	Off
6,400	Off	Off	On	Off
7,200	On	On	Off	Off
9,600	Off	On	Off	Off
19,200	On	Off	Off	Off

3. Install the DDT/B module in slot 6 of the extended memory subsystem.

   Refer to the ROM and memory configurations for the next steps.

4. On the pin side of the CPU backplane, carefully move the following cables from the left side of slot 6 (add numbered slot on the right side of slot 7 (move numbered 9 to), keeping the same vertical placement.

   a. Serial power cable (row A)

   b. Fiber cable (row B)

   c. Console cable (row C)

   CAUTION
   Make sure there are two open pins between the console cable and the console baud rate plugs.

BDM INSTALLATION (CONT)

5. Plug the BDM ribbon cable assembly (H8-5021) into the option I/O cable connector at slot G, on the right side of slot G. Attach the assembly to the backplane so that pin 1 is on top and pin 22 is on the bottom (pin 22 is).

CAUTION

When installing this cable, be sure that it is connected properly and that the two center pins (pin 20 and pin 21) are not connected.

6. Attach the ribbon cable assembly to the I/O port panel using two 6-32 screws provided with the cover plates.

7. Attach the cable tag around the ribbon cable using the wire strap and tighten it to prevent accidental disconnect.

8. Plug the power cable (H8-5021) into the BDM ribbon cable connector in the I/O port panel and secure.

9. Route the system cable as necessary and connect it to the modem. Use a 9-pin socket to 25-pin plug cable that matches a DEC VAX power source.

CAUTION

Modem AC power must not come from the VAX-11/782 terminal power distribution system. This violates UL regulations and cabinet power integrity.

INSTALLATION VERIFICATION

Use the following procedure to verify BDM operation.

1. Run any diagnostic software and reboot system.

2. Power up the system by turning the boot device switch to A. The power-on action switch to HALT, and the power key switch to LOCK.

3. Observe the BDM power-up self-test. After power-up, the fault and reset should turn off for about two seconds. Suspect the installation if the fault indicator does not turn on, or does not turn off, or if there is no console printout of error message with the LED. BDM or BDM cable. If the installation is correct, remove the BDM. Install another BDM or return the cables to their original positions.

1. The power up test is successful, the display terminal prints the following:

```
& &
87982780 16
<>
```

4. Perform the ROM installation tests described later in this chapter yourself or ask the DEC to perform an optional installation verification.

5. For remote testing by the DEC, turn the remote power key switch to LOCKED.

6. Press with the DEC from the printer terminal or by telephone for read-in of the verification.

ROM REMOVAL

Removing the ROM is the reverse of the installation procedure. If removal is necessary, only installation steps 1 through 4 need to be performed in reverse. The power cord and connectors can be left in place.

CAUTION

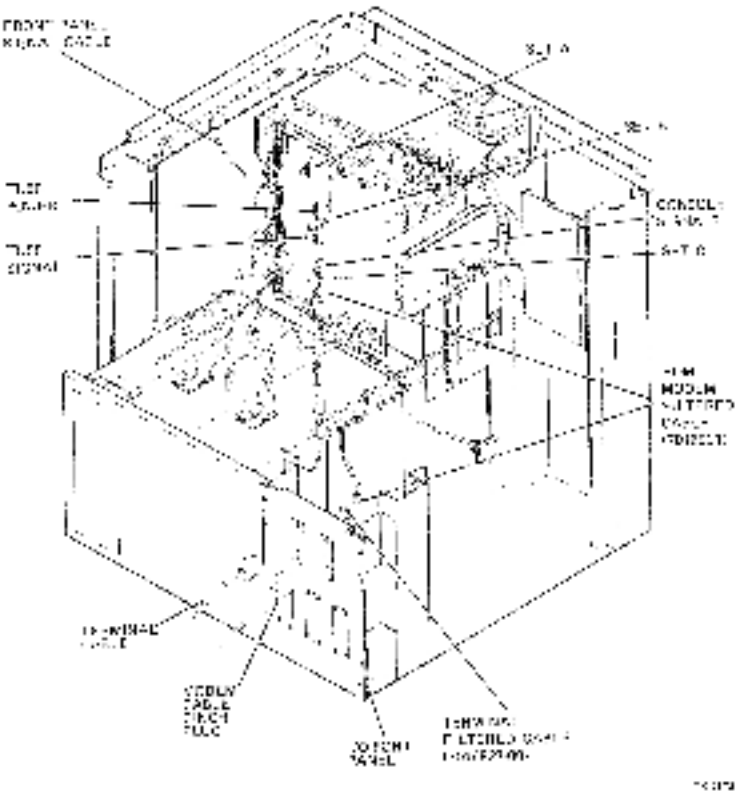
Always power down the system before installing or removing the ROM.

INSTALLATION OUTSIDE THE UNITED STATES

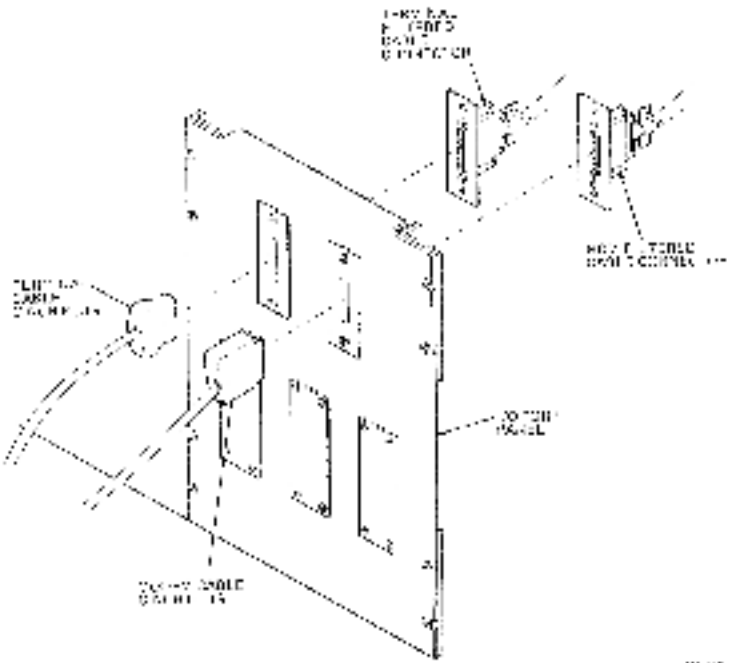
DIGITAL does not provide modems with the options outside the United States. The hardware practices and procedures of a given country must be followed to obtain a modem there. Refer to the KE11B Options Installation Guide (EK-KE11B-IN) for specifications on modems that will work with the KE.

RDM INSTALLATION (CONT'D)

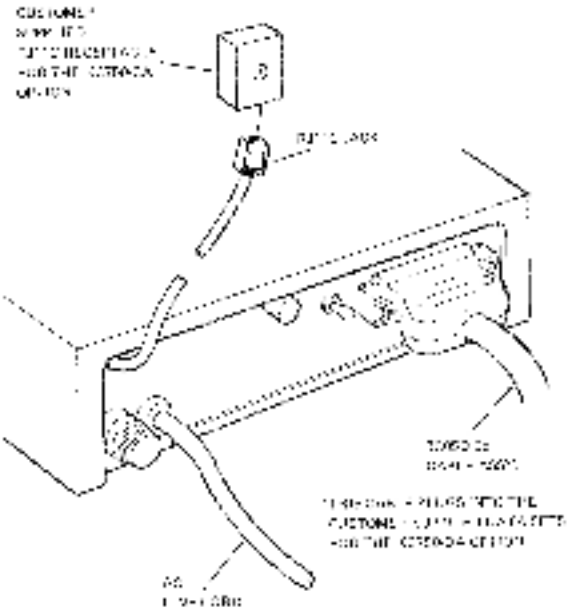
RDM CABLING



FILTERED CABLE INSTALLATION



MODEM CABLING



RDM/MODEM SIGNALS

RDM Pin	DB-25S(2) Female Pin	EIA (RS232-C) Signal Name	EIA Circuit	CCITT Circuit
CN8111	1	Protective Ground	AA	101
CN8112	7	Signal Ground	AB	102
CN8113	4	Transmitted Data	BA	103
CN8114	3	Received Data	BB	104
CN8115	6	Equipment Ready	CC	107
CN8116	5	Clear to Send	CB	106
CN8117	8	Data Set Ready	CF	109
CN8118	22	Data Terminal Ready	CD	108.2
CN8119	20	Ring Indicator	CE	125
CN8110	2	Carrier Detect	CF	109
	(Unused)	Frame Ground	AA	101

SELECTED CPU BACKPLANE SIGNALS

Pin	Signal Name
J-2A Terminal	
CN8211	Ground
CN8214	RDM Terminal Serial Out to CPU4 Ser. #1 In
CN8212	RDM Terminal Serial In to CPU4 Serial Out
CN8215	Comm. & Spec. Cable A 5
CN8213	Cable's Spec Cable B 5
CN8216	Comm. & Spec. Cable B 5
CN8217	Cable's Spec Cable A 5
P-CN8	
P-CN2	TCP1 Serial Out (signals between TCP1 and RDM)
P-CN3	TCP1 Serial In (signals between RDM and TCP1
P-CN4	EIA CC Serial Out (signals between RDM and CPU 1)
P-CN5	EIA CC Serial In (signals between RDM and CPU)
Front Panel	
AC-SN1	The front panel cable is plugged into CC1 (Cable No.1) with cable CC-1. An LED indicating ON-LINE.
RDM Specific	
CN8111	+5V DC (+5 - (2)(+))
CN8115	GA SC/SP 5 (Active)
CN8117	+5V DC (1 (receive)
CN8116	+5V MSOTH (DCIR (receive)

ROM INSTALLATION TESTS

The following section of tests are run in the order given.

VAX CPU TEST

To verify CPU operation, five VAX ROM tests are executed. Start the TEST mode from operation. The screen displays a prompt as follows:

```
>>> TEST
?(RET)
```

Cache is disabled and the user option indicates. For the 8700 should fail if any on the boot parameters while E>>>A is loaded. The diagnostic runs repeated y until it is stopped by CTRL/C. The following screen appears for processing.

```
>>>TEST 23 (RET)
>>> 0.0.0.2
4%

KV630 - 6-8 Done!
M7668A - 4.8 Done!
CPU0.0
8200 from 00
>>>
```

In this 4 steps, a script for more values. The KV630 diagnostic should run if test value LuCuse being stopped. Display may have to be typed more than once to stop KV630.1

Next, use the VAX firmware to write and read test data to the VAX memory locations. Type the following to run memory.

```
>>>T/L 2 40-FF
>>>1 + FFFFFFFF(FF)
>>>2 + 55555555(FF)
>>>3 : 55555555(FF)
>>>4 : AAAA AAAA(FF)
>>>5 : 00000000(FF)
```

Next, type the following to execute memory.

```
>>>E 2(RET)
                    F    >00:00FF     FFFFFF00
>>>6<FF>
                    F    >00:00FF     FFFFFFFF
>>>5<.E>
                    D    >00:0000     55555555
>>>4<FF>
                    D    >00:0000     AAAAAAAA
>>>T 2L/FF5<.E>
                    D    >00:0000     55555555
>>>E4<.E>
                    D    >00:0000     00000000
```

If the selected data does not agree with the expected result, the test will fail.

VAX MEMORY BUS TEST

```
This test verifies the ability of the MEM  buffers  to  read  and
write  data  with  the  VAX  memory  (from  the VAX CPU port module to the
memory  array  and  back).

If  all  the  tests  listed  below  the ROM console would be halted.

    >>>CONTINUE
    HALT

Testing VAX memory.

    MEM>X 2(0,2)
                        P     00000000    00003367
    MEM>R/PERT>
                        P     00000000    FFFFFFFF
    MEM>R/PERT>
                        P     00000000    AAAAAAAA
    MEM>R/PERT>
                        P     00000000    55555555
    MEM>R 1/P2>/PERT>
                        P     00FFFFFF    01010101
    MEM>R/ALL>
                        P     FF00FF00    87654321


Loopback to memory.

    MEM>T @ 8FFFF(81/1E1)
    RUNNING 1 PASSED
    MEM>D /WIDE %%X0000%%X8FFF>
    RUNNING + PASSED/ERRORS>

Read/write memory.

    MEM>T @<FFF>
                        P     00000000    FFFFFFFF
    MEM>T(81)
                        P     00000001    00000000
    RUNNING 2FFFF-8FFF>
                        P     00FFFFFF    00000000
    RUNNING/ERR>
                        P     01010101    AAAAAAAA
    RUNNING
```

To the data read from memory is not the same as data written into
memory, the test has failed.

VAX CONTROL STORE PARITY CHECK

This test exercises the RDM's ability to know whether the VAX clock is running, to stop the VAX clock, and to check the control store parity. (The VAX memory bus test should be run before this test.)

If the test is working properly, it will use the clock VAX control store parity while the VAX clock is running. In order to see if the RDM can detect the VAX is not, ask it to do a parity check while the VAX's clock is running.

```
>>>D/P/V RDM
    CSS DIAGNOSE
>>>C
```

Check that the RDM can stop the VAX clock.

```
RDM>ST/VAX/RTN
    CPU STOPPED CASE 3000      KSET 200F

In this display, 0 stands for some number.
```

Now test the RDM's ability to check the VAX control store parity. (An error has been added at address 0000.)

```
RDM>LOAD 2/RRT>
PARITY  /HOOK CASE 1FED
0 MS
```

To any valid address between 0 and the address, a parity error has been detected in that address of the VAX control store.

MICROBREAK POINT AND TRACE

This test exercises the RDM logic that stores the VAX clock when the CPU executes the microinstruction at a preset address called the microbreak point. The test also exercises the RDM's ability to clock VAX control store addresses. A trace is a listing of the 32 addresses of the last 32 microinstructions executed, in the order of their execution. (The VAX control store parity check should be run before this test.)

Reset, initialize the CPU as in Item.

```
RDM>RST/RRT>
#3

B:80/38 To

>>>D/L/2
RDM>
```

```
;;; Use alternate port ...

        LDA#$0 %00000
        STA

;;; Alternate port happens to be the address of the 6522's
;;; auxiliary control register.  Thus, the CPU sees a variety
;;; of, the ROM's write-protect bit, the CPU clock.

Return to the VIA's output state.  The ACR bits 4 through 1 are
in the default.

        LDA#%01010000
        STA $600B,X     ; ACR ...

Write 0 to even horizontal address and AT&T  returns the address
of the next location to be written.

Just check that the VIA does a valid write cycle.

        RTS?T=$0000

                $0A0    $A9B    ; ACR ...
                $0A1    $0A
                $0A2    $85
                ...     ...
                %0A0    $F00
                %0A0    $F06
                $0A0    $F08
                %0A0    $F04
                $0A0    $F06
                ...     ...
```

Stray time standard names should be displayed in a ... to the above directory, or a table to indicate that a continuation of choices, patterns takes place. Also, a representa- ... ; machine... memory ... and ORG to the address of the next directive action to be taken [PC].

Note that the address of the top alternative be executed [BASE] is the address at the earliest ... which alternative actions. The ...Expression ... and also use the loop to solve the CPU waits for a earliest values.

MICRODIAGNOSTIC RUN TEST

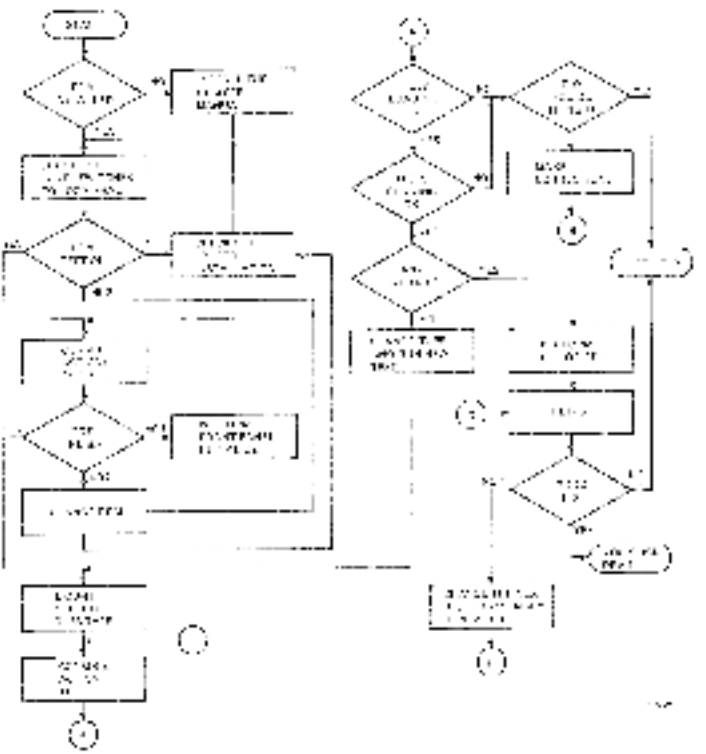
The next ROM installation test runs a microdiagnostic program on the 2650. Unlike the other tests, this one runs from the system ROM rather than from RAM. The microdiagnostic program is the same one that runs automatically when the system is first powered up. It checks the 2650 registers, the PSU, the RAM memory and the ALU.

```
RUN TEST/ROM
MICRODIAGNOSTIC

ERROR-PSU-PSL REG=XX:XX
01, 02, 04, 08, 10, 20, 40, 80, 81, 82, 84, 88, 90, A0,
A1, A2, A4, A8, B0, B1, B2, B4, B8, C0, C1, C2,
C4, C8, D0, D1, D2, D4, D8, E0, E1, E2, E4, E8,
F0, F1, F2, F4, F8, FC, FE, FF.
END OF DATA XX

ERROR-ALU
00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B,
0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23,
END OF DATA XX

? CONTINUE
ROM>
```

Note that the PSU prompt is displayed when the system is ... the microdiagnostic starts.

If the printout is not the same ... as just given and the test has been performed correctly, replace the new with a spare. If spare is not available, remove the existing ROM and return. The workplace earlier to the original position.

ROM TROUBLESHOOTING FLOW



RDM COMMANDS (CONT)

RDM CONTROL KEY FUNCTIONS

Key	Function
CTRL/J	Enter RDM console command mode
CTRL/P	Enter CPU access command mode
CTRL/U	Abort current command line
CTRL/S	Send a printable ASCII character
CTRL/R	Retype current command line
CTRL/C	Control return to function (exit RDM or a program)
CTRL/B	Display CPU status in octal
CTRL/Z	Enable CPU halted to octal mode

RDM CONSOLE COMMANDS

Command	Function
B/W?	Examine - Prompt for following command switches:
	B/I <ADDRESS> Byte mode is byte B/W <ADDRESS> Byte mode is word B/L <ADDRESS> Byte mode is longword B/P <ADDRESS> Examine RDM status registers and next addresses in I/O page of RSTS or RSTS
	/N or N Previous next address ^ Examine previous address = Open same last examined or deposited as the address
D/W?	Deposit - Prompt for following command switches:
	D/B <ADDRESS> <DATA> Byte mode is byte D/W <ADDRESS> <DATA> Byte mode is word D/L <ADDRESS> <DATA> Byte mode is longword D/P nn C <ADDR> Deposit data in next address D = <DATA> Deposit data in previous address D ^ <ADDR> Open data last examined or deposited at the address
DEPOSIT	Initialize - Prompt for a keyboard emulator to program CPU. Input a hung condition. Prompts ACLO and HALT signals. (May be used also to reinitialize to terminate keyboard recovery problems.)
EXAMINE	Octal - Initiate system into octal mode Done - Terminate mode to CPU program code. Step - Single-instruction. If set, to single set.
PROGRAM	Initialize - Initiate system into RDM console mode to RDM control mode. System is at normal level of the current bus. Initiate CPU program mode, single instruction mode. Enable RDM console mode whether the CPU is running or halt. Step-other - Single, set, perform program.

RDM CONSOLE COMMANDS (CONT)

[illegible]	[illegible]
RDMDIA	DMA = enables bulk data between local and [illegible] terminal during an RD [illegible].
RDMHST	Show - displays 8086 CPU operating status.
RDMHSTVER	Show version - displays firmware rev alan level of the firmware residing in the RDM.
RDMHELP	Number = particularly useful [illegible] given until [illegible] is typed. [illegible] displays a brief description of [illegible].
RDMHELP <command>	Number param = particularly useful specified [illegible] until CTRL/C is typed. [illegible] displays a brief description of the [illegible].
RDMINIT <drive#>	parity zero = sets CRC parity check [illegible] specified drive. Drive = A, B, C, or D. Zero = [illegible] drive address [illegible] is either in hex, or [illegible] must be preceded by [illegible] command. Any may be checked by specifying address [illegible] format. [illegible] [illegible] the [illegible] [illegible].
RDMSTAT	Stat = Shows [illegible] status.
RDMTEST	Test = shows [illegible] and physical [illegible] RD clock [illegible]. Displays address or alternate [illegible] currently latched in CPU and address of the next [illegible].
RDMTEST 1	Loop [illegible] = alternate alternate [illegible] clock [illegible]. Displays current address latched by CPU and data to be [illegible] the next RD clock [illegible].
RDMWIN	[illegible] = [illegible] the [illegible] [illegible].
RDMWIN	[illegible] = [illegible] [illegible] the current [illegible] address stored in the RDM [illegible] control area [illegible]. [illegible] [illegible] display of [illegible] [illegible]. [illegible] must be [illegible] [illegible].
RDMWIN <address>	Hexadecimal = [illegible] the CPU and [illegible] external [illegible] to [illegible]. [illegible] read [illegible] and checks parity on the [illegible] data. [illegible] [illegible] [illegible] [illegible]. [illegible] [illegible] [illegible] [illegible], to disabled.

ROM CONSOLE COMMANDS (CONT)

Command	Function
BOOT/A <address>	Attempts to boot from disk using the boot address specified. If no address is given, the default boot address is used.
BOOT/C <address>	Attempts to boot from disk using the boot address specified. Before booting, the CPU registers are set to the values specified in the command.
BOOT/D	Boots from disk using the default boot address.
BOOT/F	Boots from floppy using the default boot address.
BOOT/H	Boots from hard disk using the default boot address.
BOOT/M	Boots from memory using the default boot address.
BOOT/S	Boots from serial port using the default boot address.
BOOT/T	Boots from tape using the default boot address.
BOOT/X	Boots from network using the default boot address.
CLEAR	Clears the screen.
CLOCK	Displays the current date and time.
CONFIG	Displays the current configuration.
DIR	Displays a directory of files on the current drive.
DIR/A	Displays a directory of files on the current drive, including hidden files.
DIR/D	Displays a directory of files on the current drive, including deleted files.
DIR/H	Displays a directory of files on the current drive, including hidden files.
DIR/S	Displays a directory of files on the current drive, including system files.
DIR/W	Displays a directory of files on the current drive, in wide format.
DISK	Displays information about the disk in the current drive.
DISK/F	Displays information about the floppy disk in the current drive.
DISK/H	Displays information about the hard disk in the current drive.
DISK/M	Displays information about the memory disk in the current drive.
DISK/S	Displays information about the serial disk in the current drive.
DISK/T	Displays information about the tape in the current drive.
DISK/X	Displays information about the network disk in the current drive.
DRIVE	Displays the current drive.
DRIVE <drive>	Changes the current drive to the specified drive.
ERASE <filename>	Erases the specified file.
ERASE/A <filename>	Erases the specified file, including hidden files.
ERASE/D <filename>	Erases the specified file, including deleted files.
ERASE/H <filename>	Erases the specified file, including hidden files.
ERASE/S <filename>	Erases the specified file, including system files.
ERASE/W <filename>	Erases the specified file, in wide format.
EXIT	Exits the ROM console.
FORMAT <drive>	Formats the specified drive.
FORMAT/F <drive>	Formats the specified floppy drive.
FORMAT/H <drive>	Formats the specified hard drive.
FORMAT/M <drive>	Formats the specified memory drive.
FORMAT/S <drive>	Formats the specified serial drive.
FORMAT/T <drive>	Formats the specified tape drive.
FORMAT/X <drive>	Formats the specified network drive.
HELP	Displays a list of available commands.
HELP <command>	Displays help for the specified command.
LOAD <filename>	Loads the specified file into memory.
LOAD/A <filename>	Loads the specified file into memory, including hidden files.
LOAD/D <filename>	Loads the specified file into memory, including deleted files.
LOAD/H <filename>	Loads the specified file into memory, including hidden files.
LOAD/S <filename>	Loads the specified file into memory, including system files.
LOAD/W <filename>	Loads the specified file into memory, in wide format.
MEMORY	Displays the current memory configuration.
MEMORY <address>	Displays the memory at the specified address.
MEMORY <address> <value>	Sets the memory at the specified address to the specified value.
MEMORY <address> <value> <count>	Sets the memory at the specified address to the specified value, for the specified number of bytes.
MEMORY <address> <value> <count> <step>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step.
MEMORY <address> <value> <count> <step> <mask>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask.
MEMORY <address> <value> <count> <step> <mask> <offset>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset.
MEMORY <address> <value> <count> <step> <mask> <offset> <repeat>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset, repeating the specified number of times.
MEMORY <address> <value> <count> <step> <mask> <offset> <repeat> <delay>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset, repeating the specified number of times, with the specified delay.
MEMORY <address> <value> <count> <step> <mask> <offset> <repeat> <delay> <flags>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset, repeating the specified number of times, with the specified delay, with the specified flags.
MEMORY <address> <value> <count> <step> <mask> <offset> <repeat> <delay> <flags> <comment>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset, repeating the specified number of times, with the specified delay, with the specified flags, with the specified comment.
MEMORY <address> <value> <count> <step> <mask> <offset> <repeat> <delay> <flags> <comment> <end>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset, repeating the specified number of times, with the specified delay, with the specified flags, with the specified comment, ending at the specified address.
MEMORY <address> <value> <count> <step> <mask> <offset> <repeat> <delay> <flags> <comment> <end> <filename>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset, repeating the specified number of times, with the specified delay, with the specified flags, with the specified comment, ending at the specified address, and saves the memory to the specified file.
MEMORY <address> <value> <count> <step> <mask> <offset> <repeat> <delay> <flags> <comment> <end> <filename> <type>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset, repeating the specified number of times, with the specified delay, with the specified flags, with the specified comment, ending at the specified address, and saves the memory to the specified file with the specified type.
MEMORY <address> <value> <count> <step> <mask> <offset> <repeat> <delay> <flags> <comment> <end> <filename> <type> <attributes>	Sets the memory at the specified address to the specified value, for the specified number of bytes, with the specified step, using the specified mask, with the specified offset, repeating the specified number of times, with the specified delay, with the specified flags, with the specified comment, ending at the specified address, and saves the memory to the specified file with the specified type and attributes.

ROM CONSOLE ERROR CODES

```
Code      Description

TFTP Function Errors

TAP:01    Tape LOAD - Device timeout
TAP:02    Tape LOAD - Server Error UART
TAP:03    Tape LOAD - Data not ready display
TAP:04    Tape LOAD - Receive overflow
TAP:05    Tape Checksum error received
TAP:06    Tape data error maximum exceeded
TAP:07    Tape OS ERR packet invalid operation value
TAP:08    Tape Invalid packet received
TAP:0A    Tape file not found
TAP:0B    Tape directory error
TAP:0C    Tape A ACK received (not normal or fatal)
TAP:0D    Tape read length error (not all blocks 512)
TAP:0E    Tape bad packet number
TAP:0F    Tape bad operation code
TAP:10    Tape access violation
TAP:11    Tape block not found
TAP:12    Tape data packet error
TAP:13    Tape write protected error
TAP:14    Tape cartridge not present
TAP:15    Tape bad file header
TAP:16    Tape serial operation (end of medium)
TAP:17    Tape diagnostic failure

Console terminal errors

TRM:01    Terminal BART - Device timeout
TRM:02    Terminal BART - Error from UART
TRM:03    Terminal BART - Full not ready displayed
TRM:04    Terminal BART - Receive overflow
TRM:05    Data not STX/A received
TRM:06    Terminal command input buffer overloaded
TRM:07    Terminal CTRL/P received
TRM:08    Terminal command input larger than buffer
TRM:09    Terminal recvbk line CRC error
```

ROM CONSOLE ERROR CODES (CONT)

Code	Description
CPU Errors	
UNIBUS	Nonexistent memory
UNIBUS	Parity error
UNIBUS	Read data substitute
General Errors	
ABORT	Error detected; operation aborted
COMMAND	ASCII code not recognized command
ROM	ROM refused RAM power-up self test
RAM	RAM failed ROM power up self test
RUNNING	Operation already in progress
ROM	Invalid operation code in packet
RAM	Memory LOAD - Device timeout
RAM	Memory LOAD - Block [out] [of]
RAM	Memory LOAD - Data was really [wrong]
RAM	Memory LOAD - Memory available
Ethernet	Ext DEUA - Device timed out
Ethernet	Ext DEUA - Error from the [bus] DEUA
Ethernet	Ext DEUA - Data not ready dropped
Ethernet	Ext DEUA - Receive overflow

VAX- [illegible] diagnostic programs are available under DEC 300/400
levels of operation.

Level 1 - VMS operating system based diagnostic programs
These run without the diagnostic supervisor.

DEC/X SMP-32, T-TAC, DSA

Level 2R - Diagnostic supervisor based programs. Any diagnostic
[illegible] which have an RP3 based driver

Peripheral diagnostics not supported by the
supervisor in the shared RAM area

System diagnostic control program

Level 3 - Diagnostic supervisor based program. Will run on
processors under VMS or in the standalone mode.

Disk diagnostic program

Extended and optional-by-level peripheral
diagnostics

Level 4 - Diagnostic supervisor based programs that can only
be run in the standalone mode.

Extended level peripheral diagnostics
Memory-level peripheral diagnostics
CPU cluster diagnostics

Level A - Miscellaneous corrective/service programs that run without the
supervisor.

Various initialization aids (loads the basic in-
structions necessary to run the supervisor)

Level 5 - Console based diagnostics that run only in the
standalone mode.

Microdiagnostics
Console operating program
Microcode/ECO programs

Device	Name	Num.	Description	Table Reference Number
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
[illegible]	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
			[illegible]	
	[illegible]	1	[illegible]	[illegible]
[illegible]	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
			[illegible]	[illegible]
[illegible]	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
[illegible]	[illegible]	1	[illegible]	[illegible]
			[illegible]	[illegible]
[illegible]	[illegible]	1	[illegible]	[illegible]
[illegible]	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]
	[illegible]	1	[illegible]	[illegible]

[illegible]

[illegible]	Name	Level	Description	For Cause or Further Information
[illegible]	TVTAA	2	VAX 11/[illegible] Tape Reliability	5,31,33
	TVTCA	3	[illegible] Subsystem [illegible] Optional	33
[illegible]	TVTAA	2	VAX 11/[illegible] Tape Reliability	5,34,35
DNSB	[illegible]	3	VAX disk Thrashing	8
	TVTCA	3	VAX TK50/TU58 Diagnostic	33
	[illegible]	3	VAX [illegible] Diagnostic	33
[illegible]	TVTAA	2	VAX 11/[illegible] Tape Reliability	5,36,37
	TVTCA	3	VAX TK50/TU58 Diagnostic	33
	[illegible]	3	DN11 Quick Test	10,11
	EVTCC	3	VAX Diagnostic	33
[illegible]	EVTAA	2	VAX VV/[illegible]/TUS8 Disk Reliability	9,28,34
[illegible]	EVTAC	3	VAX Disk Formatter	9
TU81	EVTBC	3	TU81 BOOT/DIAG Journal Test	33
	EVTTA	3	VAX DTE/[illegible]/87 Repair	47
[illegible]	KFTAA	2	VAX UV/[illegible]/TU58 Disk Reliability	9,30,39
	KFTCC	3	VAX Disk Formatter	9
	EVTBA	3	DJ/DZ Front End	47
	EVTBA	3	RX02 Exhaustive	35
	EVTBT	3	KLESI Data Path Continuity	37
[illegible]	TVTAA	2	VAX 11/[illegible] Tape Reliability	5,28,29
	EVTCC	3	VAX Disk Formatter	9
	EVTEA	3	VAX 8x00 Subsystem Diagnostic	33
TU11	EVNAA	2	VAX Nanotape Data Reliability	33
	EVNAB	3	VAX TU11 Diagnostic	33
[illegible]	KVTCA	3	VAX Magtape Data Reliability	33
TU77	EVNAB	3	VAX [illegible] Drive Exerciser	33
[illegible]	[illegible]	3	VAX [illegible] Control Display	35
[illegible]	[illegible]	3	VAX Magtape Data Reliability	33
	EVSAA	3	[illegible] Exhaustive Log	33

NOTES: Solve [illegible] [illegible] VAX diagnostic listing first [illegible] if the [illegible] diagnostic [illegible] [illegible] cause [illegible] [illegible].

```
Command          Function

MICMON           Return control from MICMON and return to DEV
                 monitor level.

MICDIAG          Diagnose - perform TEST 0 report on each DEV
                 diagnostic control block (CCB).

MICDIAG PARAMETER=VALUE
                 Diagnose - initializes program control flags and
                 starts execution loop of test programs in DCS for
                 specified number of passes.

MICDIS [/SECTION] ADDRESS[:ADDR2]  [-n]
MICDIS [/SECTION] ADDRESS/[COUNT] [ADDR2]
                 Disassemble - disassembles 8086 machine code
                 into assembly source.

MICDIAG PARAMETER=VALUE [-n]
                 Diagnose - conditionally executes specified test
                 suite designated by an error in DCS/CC.

MICDIAG EX       Diagnose - runs diagnostic module TCKAPL.TYK.

MICDIR           Directory - reads a DOS disk directory and lists
                 files.

MICDIR           Loop - sets loop flag and loops on a program
                 detected and reported error.

MICDSA FL XX     Set flag - sets specified program control flag.

MICDSC FL XX     Clear flag - clears specified program control
                 flag.

MICDSP FL        Show flags - displays current status of the
                 program control flags.

MICDSP V         Show vbus - displays current signal status on the
                 visibility bus (VBUS).

MICDUS ADDRESS [LENGTH]
                 Dump memory - displays contents of specified
                 address.
```

```
Command          Function

S[/#N] or .      Steps through VCS microinstructions or N clock
                 cycles.

P[/#N] or ;      Runs a CPU cycle (instruction), stepping
                 by or to the N clock cycle or N function of the
                 phase clock.

R[/#N] or ?      Set step instruction = clock through memory
                 instructions to current check.  If [/#N] is
                 specified, step function starts when the
                 activation of beep=pp is ready to be executed.
                 If [/#N] is not specified, stepping begins at
                 the next instruction of the current user
                 running a loop (J) or subroutine (JS) command.

A[=val #C: <I/O-ADDRESS>]
                 Clear Stop/Interrupt   Use address if
                 specified, generates Stop# sync pulse on clock C,
                 pin CN1 (1888 has no value to designate external
                 Pulse outputs with N or not stop (J) current
                 address address I/O address.

W[/#N] CN: <LOC-ADDRESS> <TTT-MMM>
                 Set control file bit = value specified control
                 bit to the specified I/O address (bits
                 <15:8>).

W[/#N] CN: <I/O ADDRESS> <TTT-MMM>
                 Clear control file bit = value specified control
                 file bit at the specified I/O address.

[N]=VAL <REGISTER> or
W[/#N] <REGISTER-NUMBER>
                 Examine the following processor registers:

                 AC      Virtual  address register (VAR)
                 MA      Memory address register (MAR)
                 PC      CPU program counter (PC)
                 PS      PC Lookup register (PCLU)
                 R[=n]   Block registers 0 through 7F
                 R'[=n]  Whole registers 0 through 7
                 PS      Processor status register (PSR)
                 MD      Memory data register (MDR)
                 MB      Memory data register (MDR)
                 SR=n    Status and control registers 0 through
                 SP      Start of stack
                 JT      Jump vectors
```

MICROMONITOR PROGRAM CONTROL FLAGS

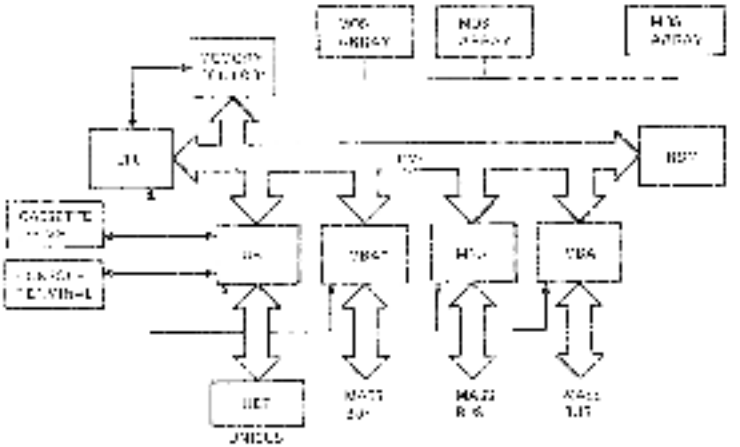
Flag	Function
HALT	Halt on error - control to the monitor on a program detected error.
LOOP	Loop on error - loops on program detected error. Also set by Micro Display Utility (MDU) if the HALT flag is not set and the MDU is not in continuous loop.
	This loop may include opcode instructions and DCS instructions from the buffer or the [ERROR] instruction in the calling test. If the LP flag is not on a single step occurs. (DO [RETURN] then exit loop on micromicroinstructions in DCS). Also must be clear to inhibit error messages.
NER	No error report - inhibits error messages.
SELL	Self to error - error out on the first, then every [LOOP parameter of, an error.
SP	[with P value] - when used with LOOP flag, program does not loop error on this instruction but loops until between the [ERROR and [PAUSE pseudo instructions.
QA	Quality assurance - each test reports as if an error was detected.
VP	Video - monitor or the test video and variable.
SA	Signature analysis - used with signature analyzer to help diagnose faults. Loop occurs on test to indicate whether or not an error occurs.
	Also, provides two sync points on the backplane:
	Start/stop window, slot 6, pin C25 Clock pulse, slot 6, pin C23
	The signature analyzer analyzes test points by displaying a value (signature) at the output pattern is steady. This value can be associated with the one a fixed, good problem to enable fault trace.

VISIBILITY BUS (VBUS) SIGNALS

Bit Number (Octal)	Pin Number (Dec)	Signal	Print Out Name
00	03	CBUS PARITY ERROR L	CBPE L
01	04	CBUS PARITY ERROR H	CBPE H
02	05	CB SPEAK PAR H	CBPE H
03	06	CBUS STOP ERROR L	CBSE L
04	07	CBUS BUSY LATCH H	CBSY H
05	08	CBUS ADD PEND L	CBAD L
06	09	CBUS TO INT GRANT L	CBTI L
07	10	CBUS CON BUSY L	CBCB L
08	11	MICRO VECTOR 6 H	MIC06 H
09	12	MICRO VECTOR 5 L	MIC05 L
0A	13	MICRO VECTOR 4 H	MIC04 H
0B	14	MICRO VECTOR 3 L	MIC03 L
0C	15	MICRO CPE FIFO FULL L	MICF L
0D	16	MICRO DISABLE L	MICD L
0E	17	MICRO EXTENDED MODE IN L	MICE L
0F	18	MICRO PROG INIT L	MICI L
10	19	MICRO MICRO RQ L	MICR L
11	20	MICRO MEM STALL H	MICM H
12	21	MICRO OPCODE VALID H	MICV H
13	22	MICRO DP REQ H	MICD H
14	23	MICRO CPER DATA INT L	MICD L
15	24	MICRO MB PCE TER INT L	MICB L
16	25	MBC [A-D], BLADE 0	MICB0
17	26	MBC [A-D], BLADE 1	MICB1
18	27	DPMIC INTERN PERIPH H	DPMI H
19	28	DPMIC DC SHORT L	DPMI L
1A	29	DPMIC IDLE L	DPMI L
1B	30	DPMIC MEMORY SHADOW H	DPMI H
1C	31	DPMIC DIAGNOSTIC TL BLKI H	DPMI H
1D	32	DPMIC DP PARITY ERROR H	DPMI H
1E	33	DIMIC LP DET L	DIMI L
1F	34	DIMIC DEL CP H	DIMI H
20	35	DIMIC IO.EZ 0 L	DIMI L
21	36	DIMIC IO.EZ 1 L	DIMI L
22	37	DIMIC DPP PHOOL H	DIMI H
23	38	DIMIC TIMER INT L	DIMI L
24	39	DIMIC DSIAT 0 H	DIMI H
25	40	DIMIC DSIAT 1 H	DIMI H
26	41	DIMIC MCS TMR L	DIMI L
27	42	DIMIC DEBG INIT L	DIMI L

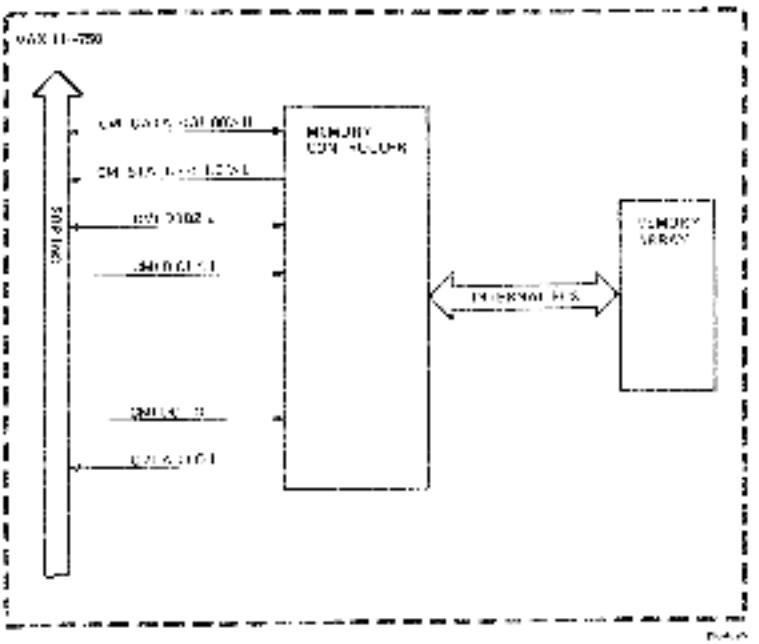
CHAPTER 7
BLOCK DIAGRAMS

VAX-11/750 BASIC DIAGRAM

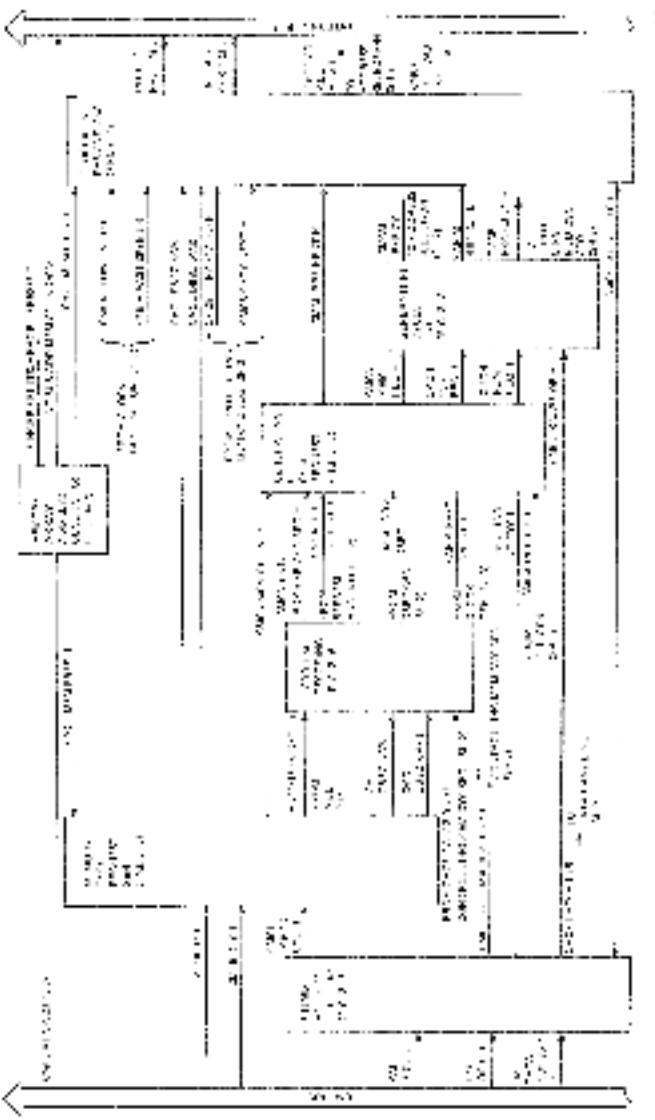


VAX 11/750 SYSTEM DIAGRAM

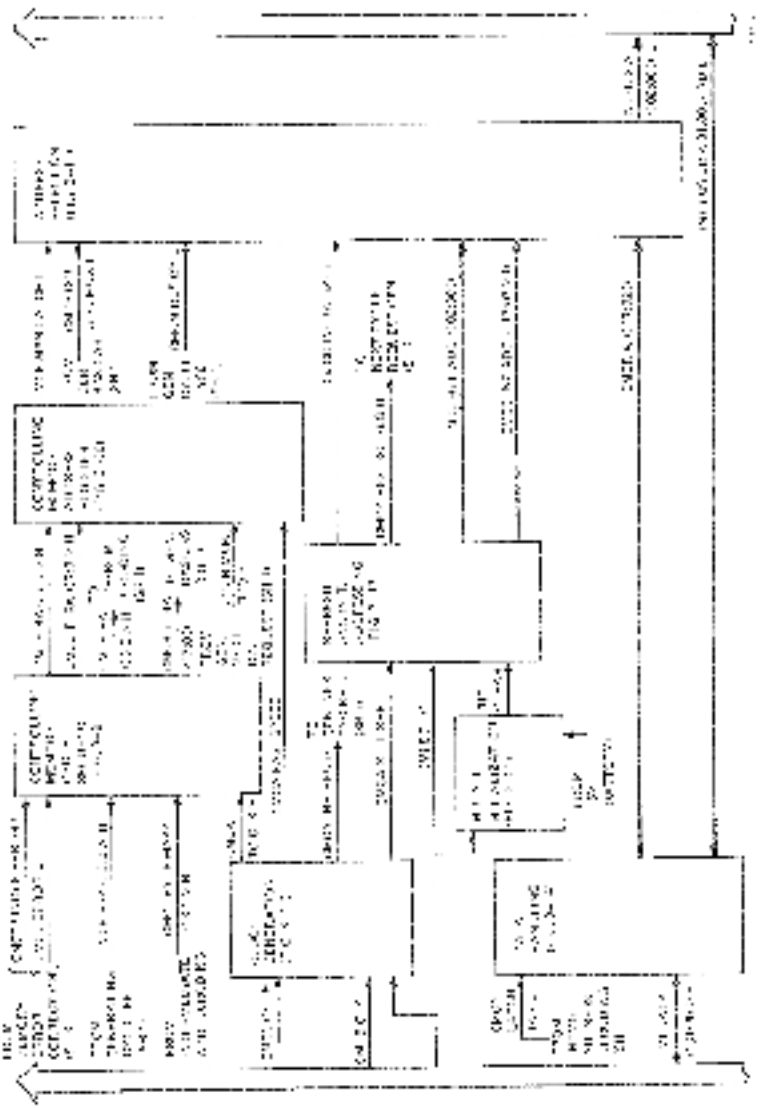
CMC (MS750) BASIC DIAGRAM



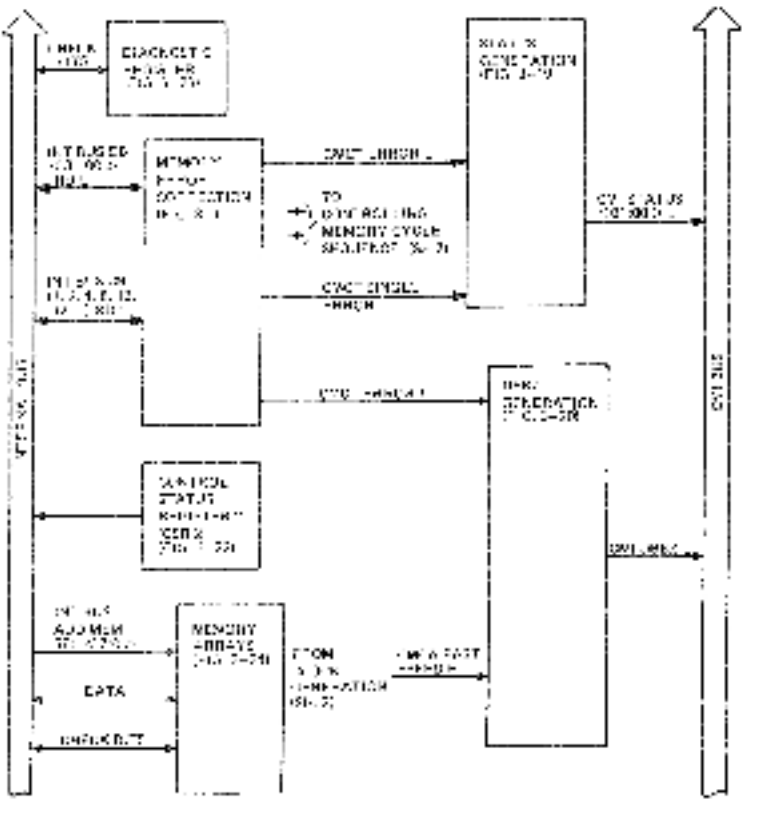
CMC (MST50) BLOCK DIAGRAM, PART 1



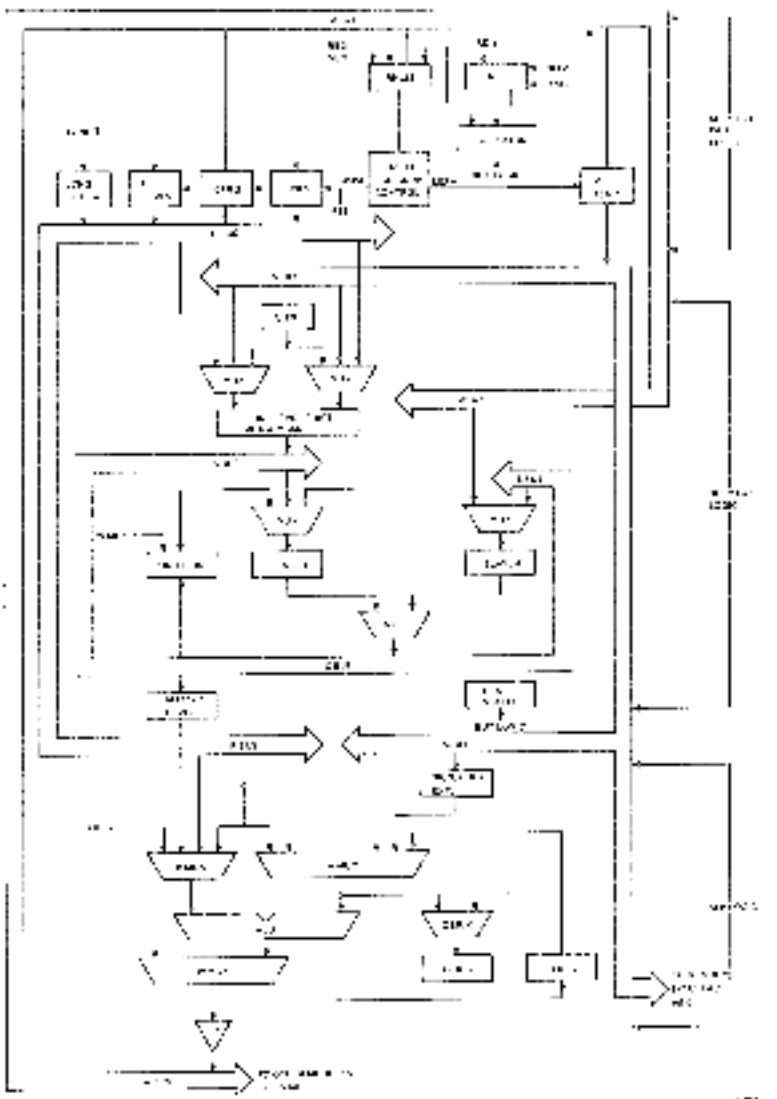
CMD (MS750) BLOCK DIAGRAM, PART 2



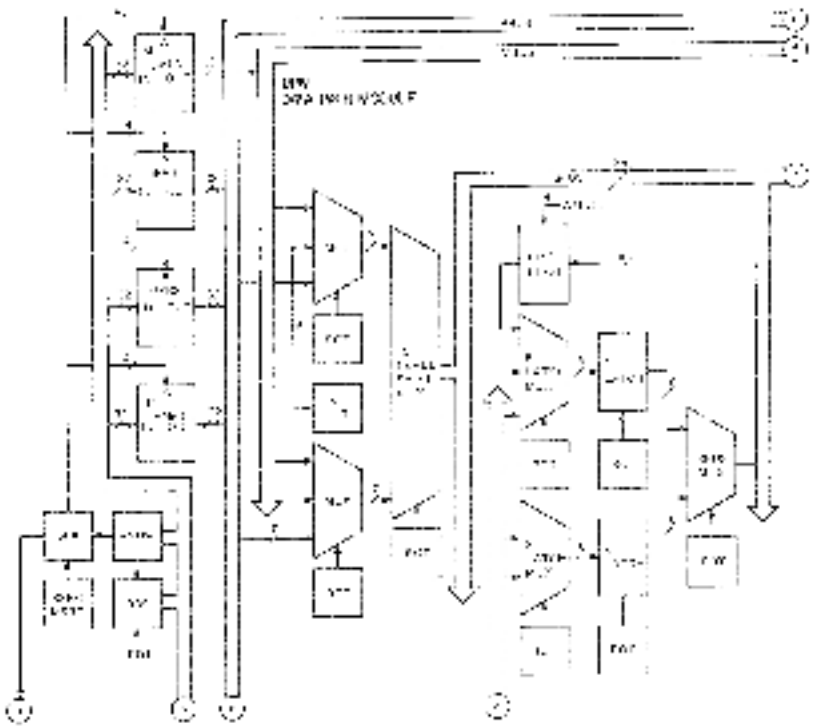
CMC (MS750) BLOCK DIAGRAM, PART 3



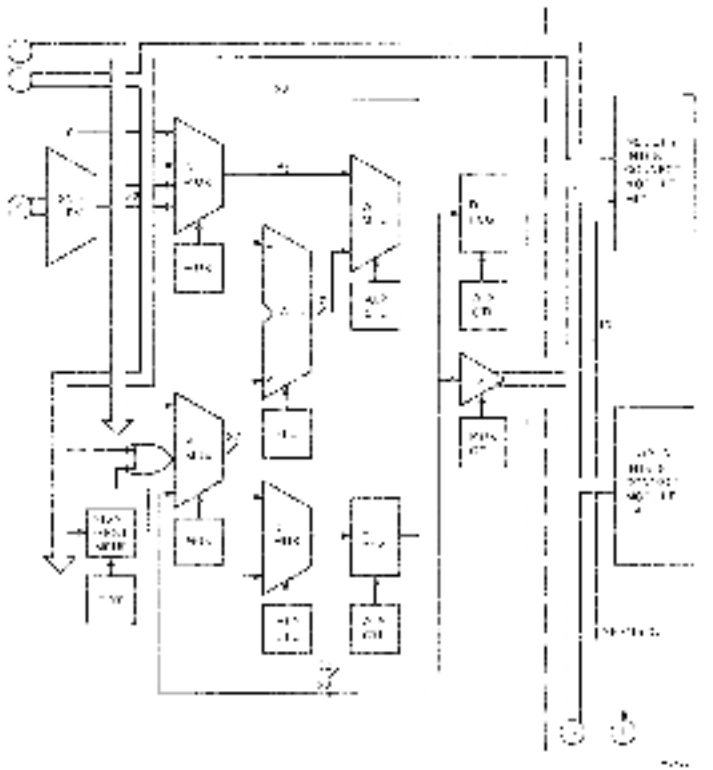
DPM BASIC DIAGRAM



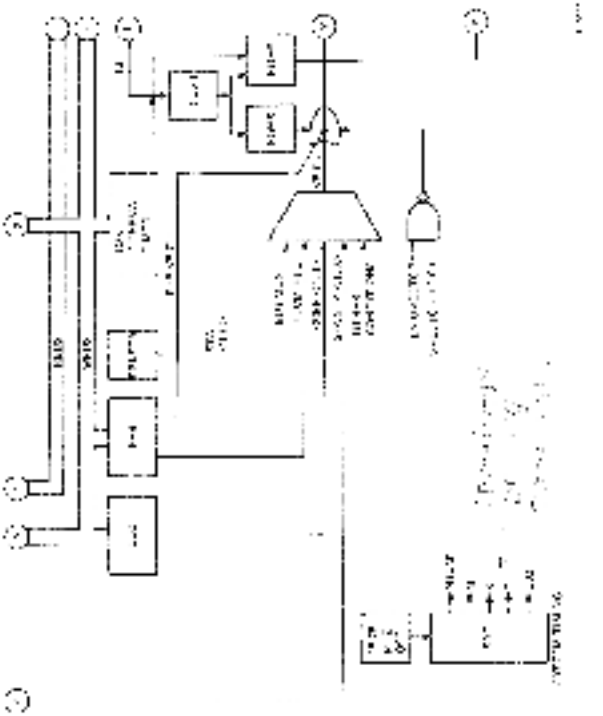
DPM BLOCK DIAGRAM, PART 1



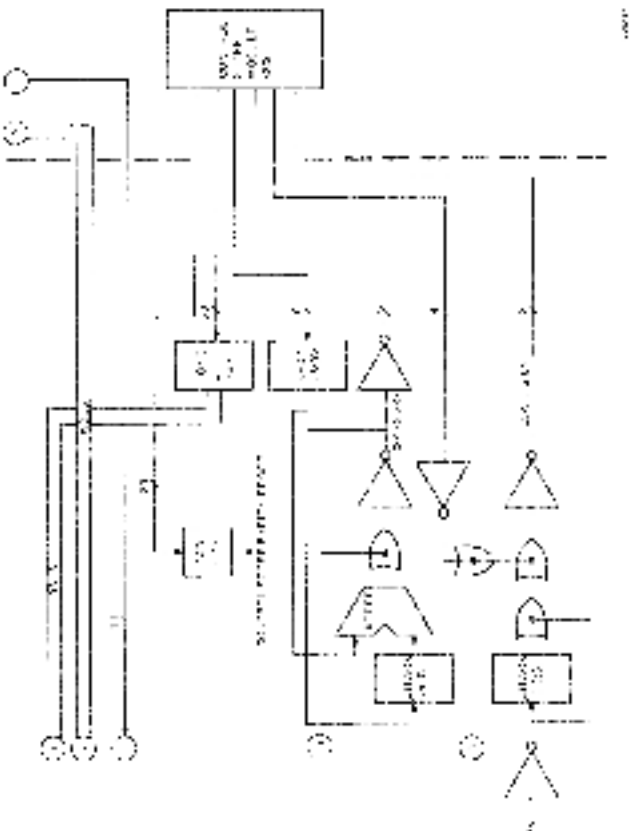
DPM BLOCK DIAGRAM, PART 2



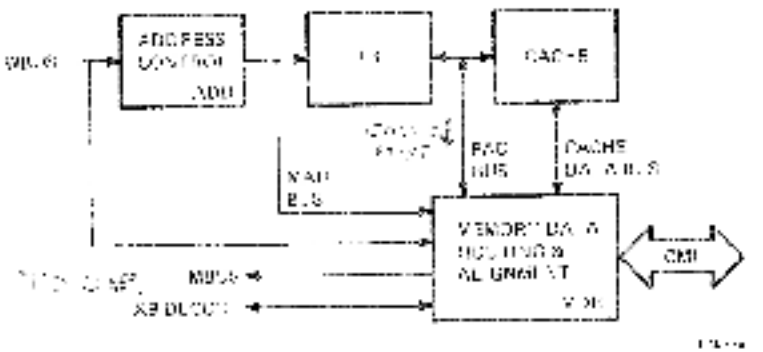
OPM BLOCK DIAGRAM, PART 3



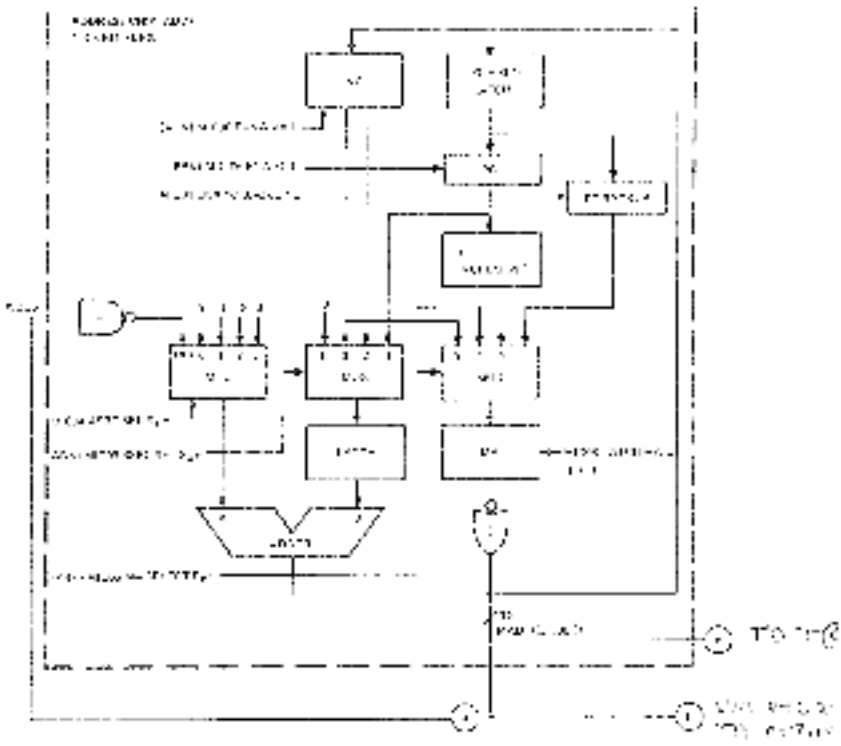
DPM BLOCK DIAGRAM, PART 4



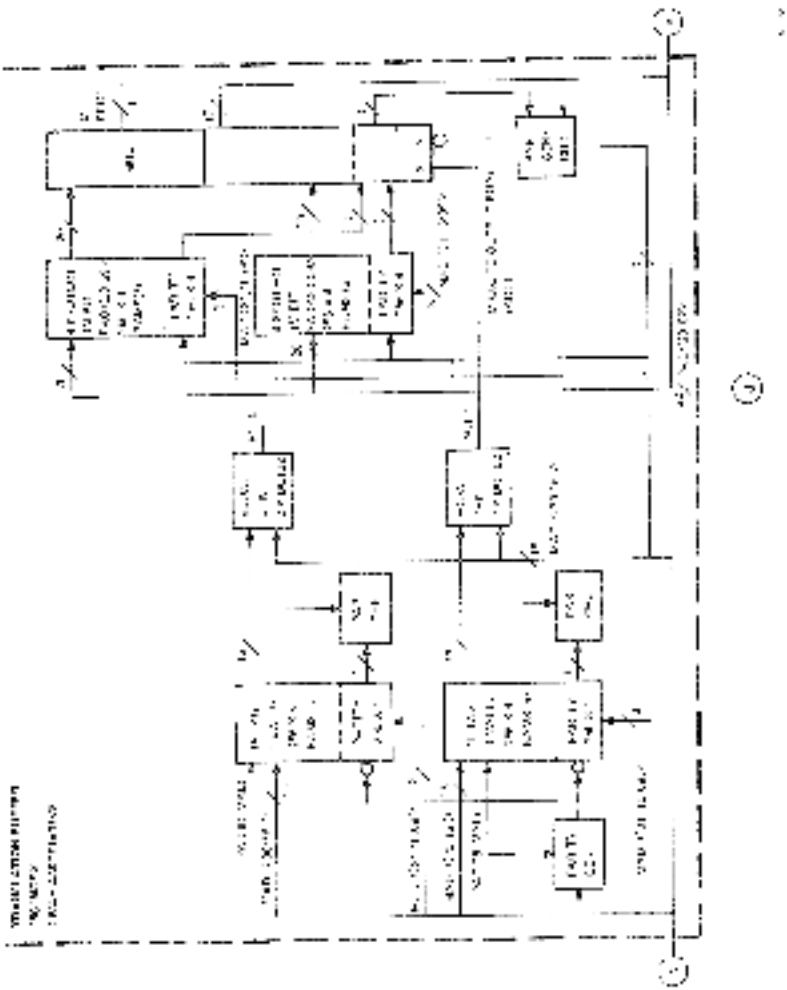
MIC BASIC DIAGRAM



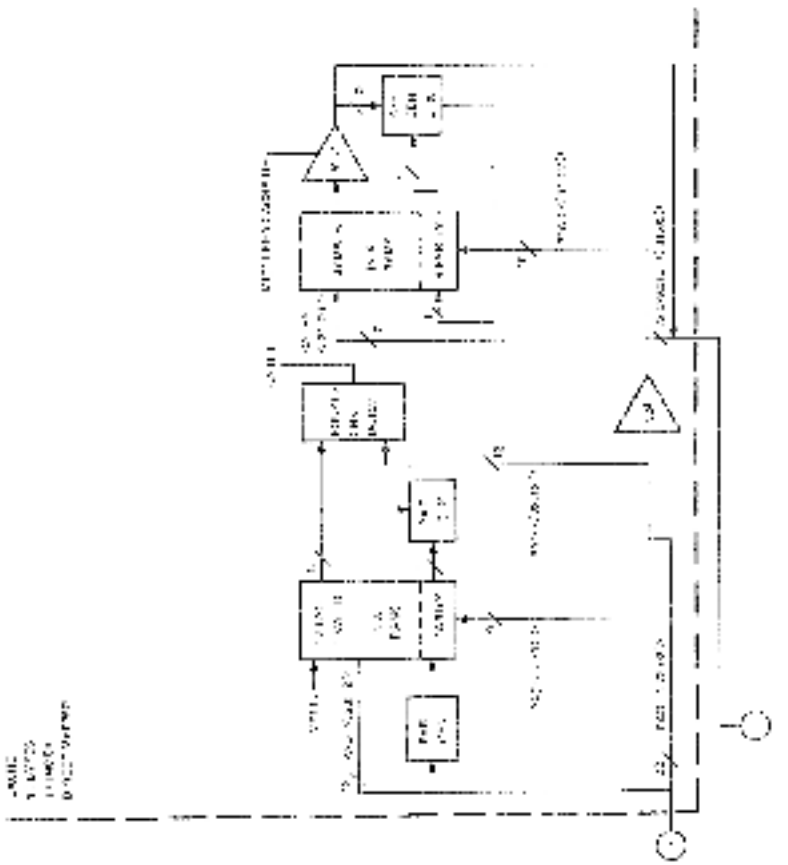
MIC BLOCK DIAGRAM, PART 1



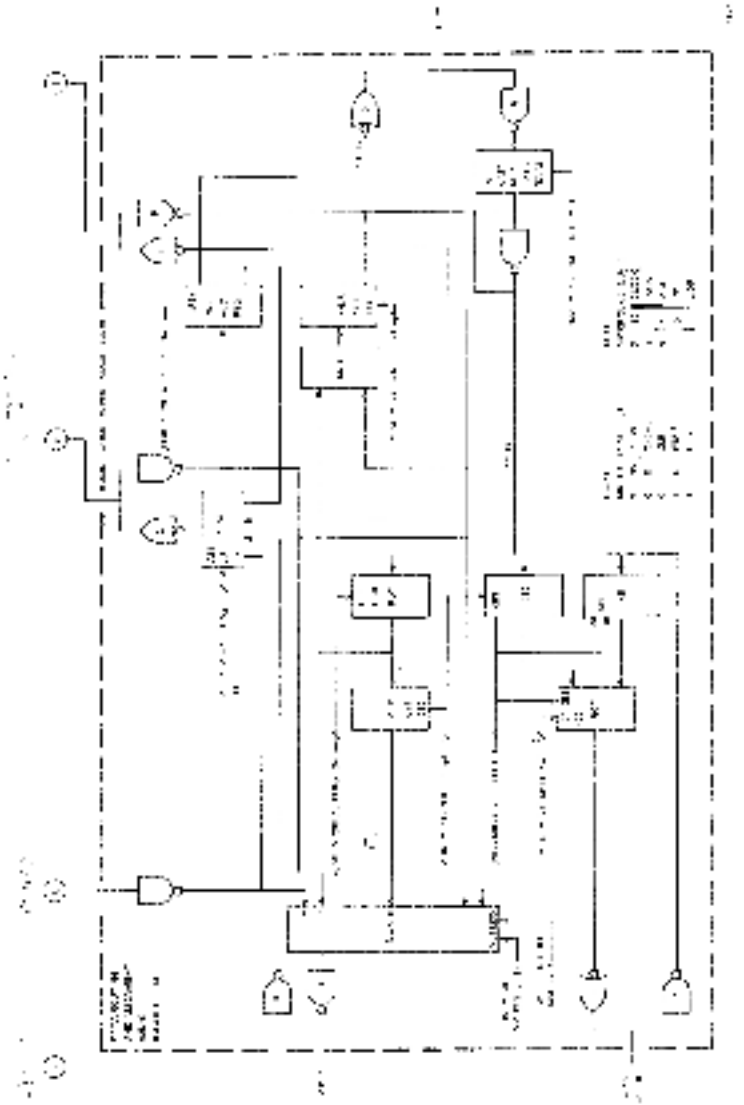
MIC BLOCK DIAGRAM, PART 2



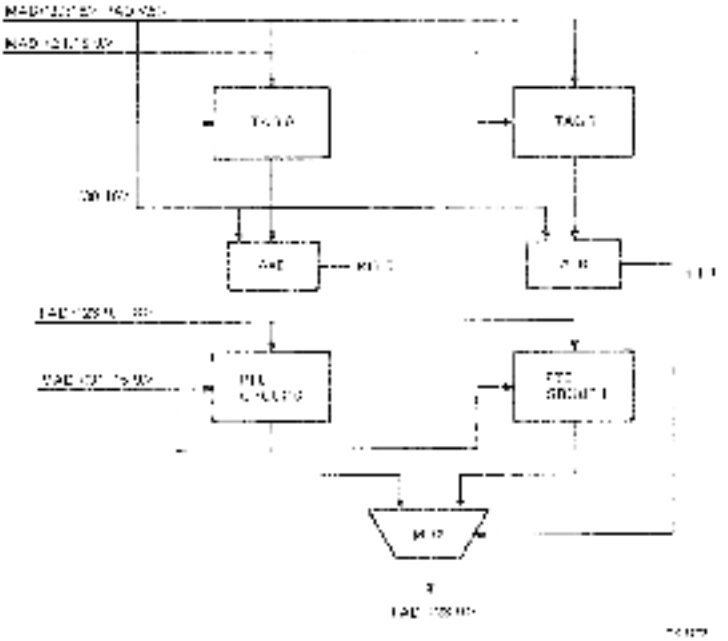
MIC BLOCK DIAGRAM, PART 3



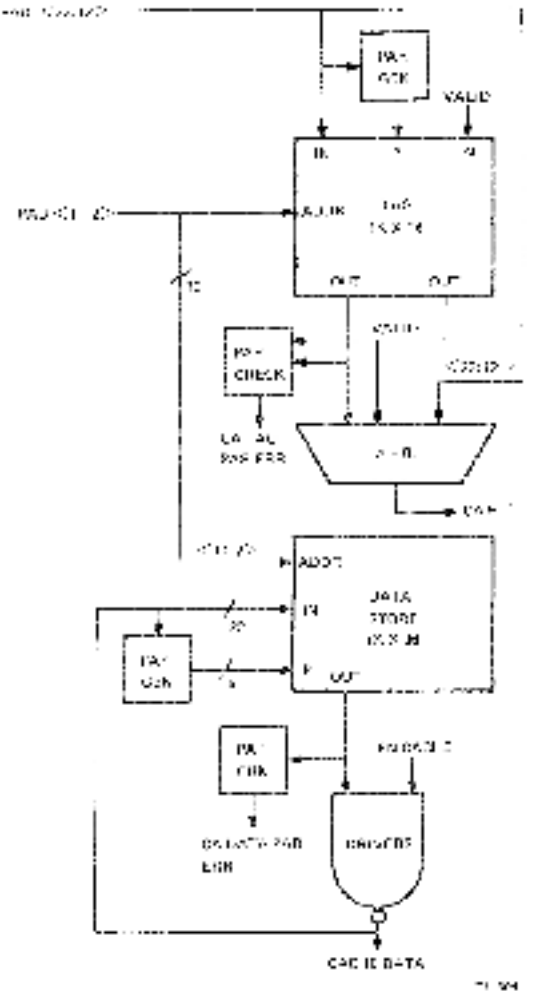
MIC BLOCK DIAGRAM, PART 4



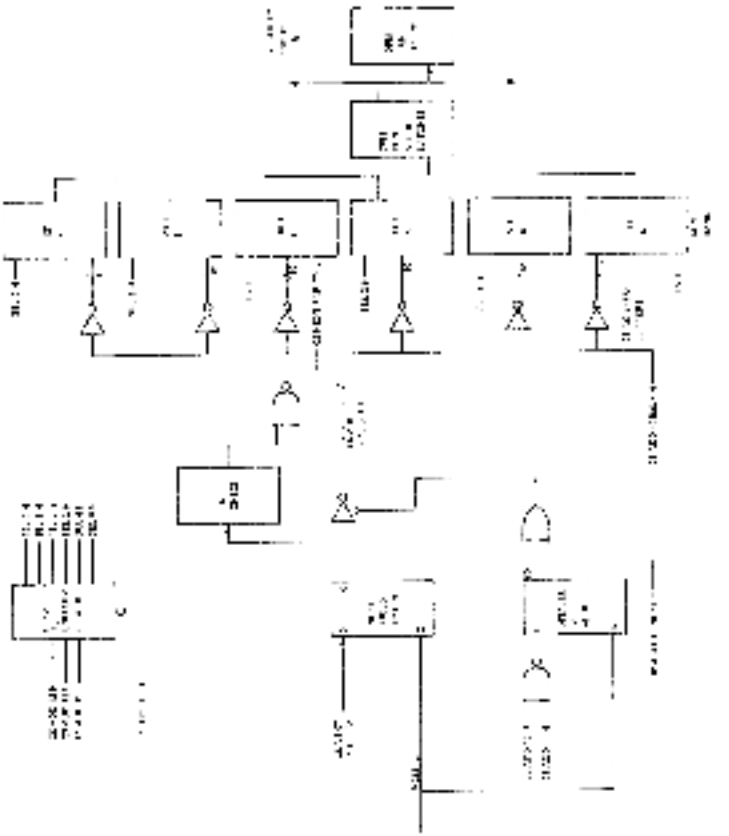
TR FUNCTIONAL DIAGRAM



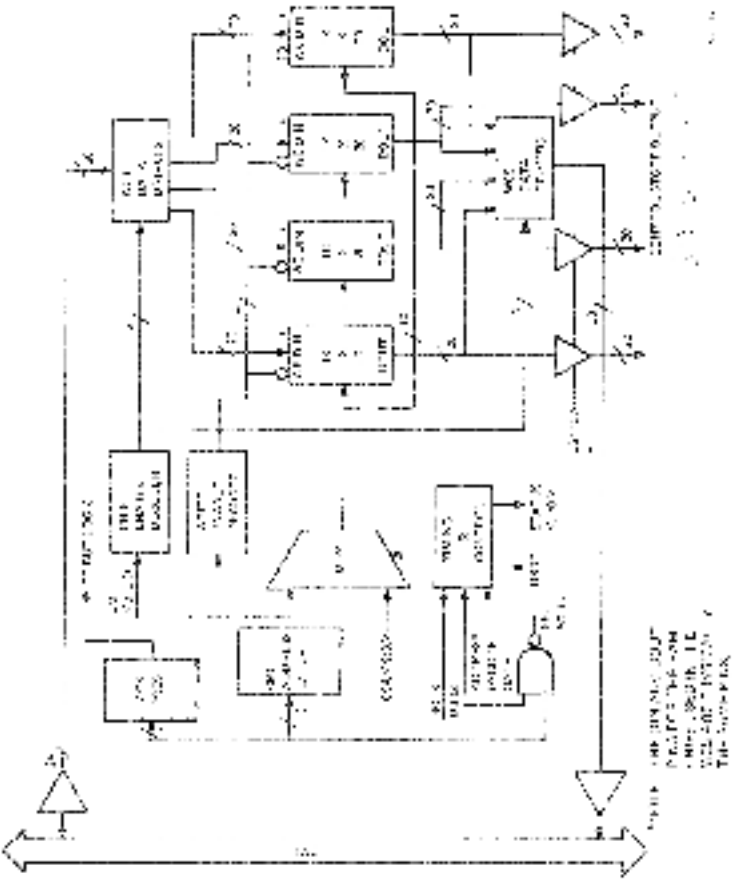
CACHE FUNCTIONAL DIAGRAM



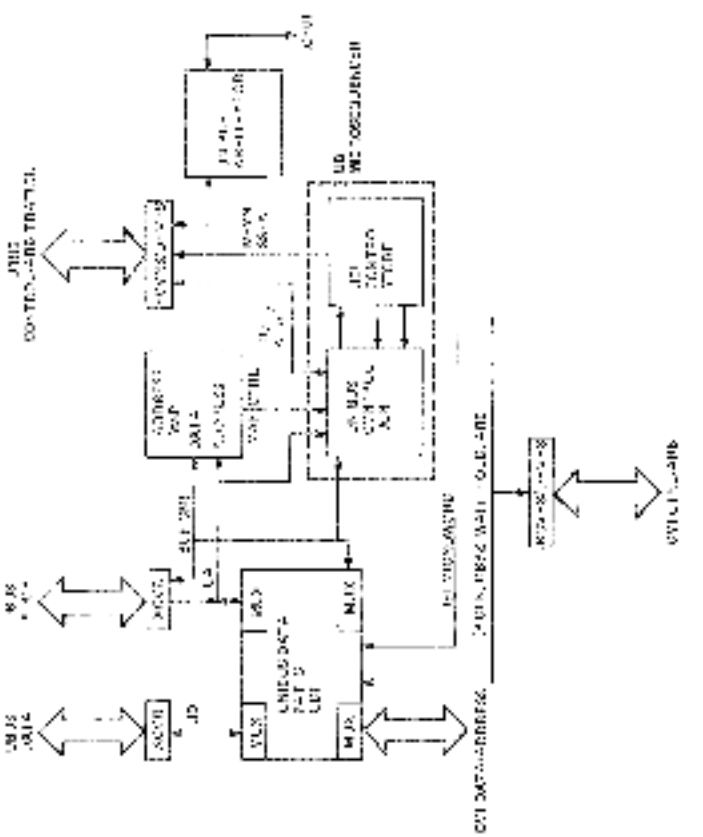
CCS BLOCK DIAGRAM



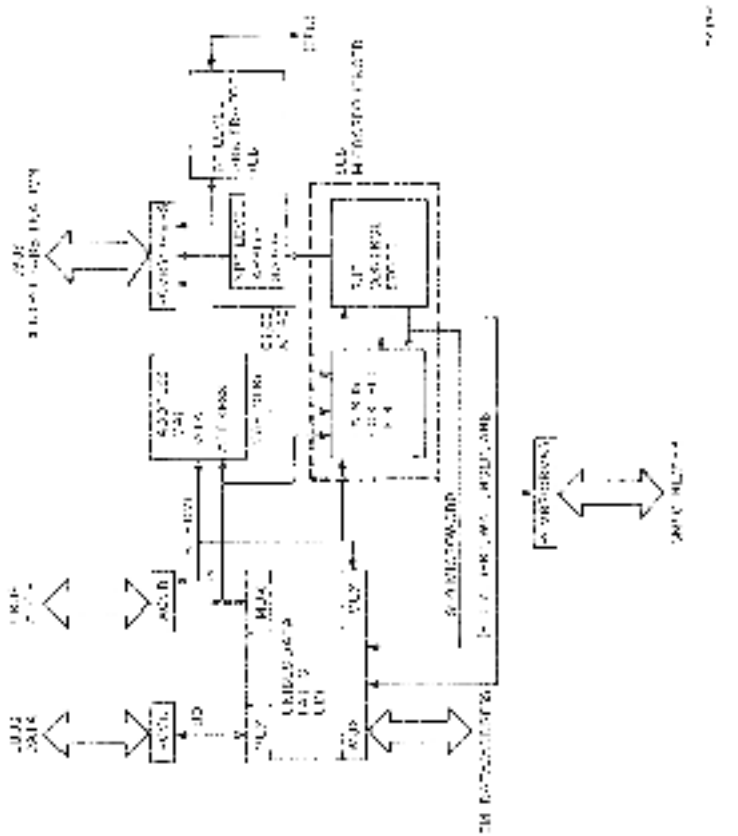
WCS (XU750) BLOCK DIAGRAM



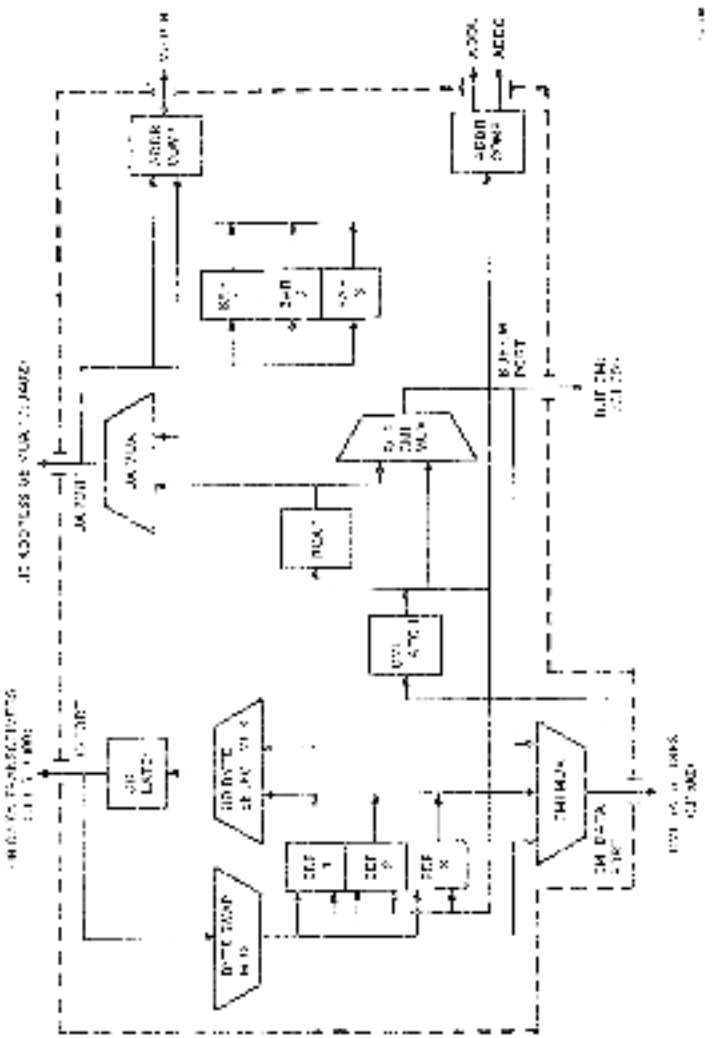
URI BLOCK DIAGRAM



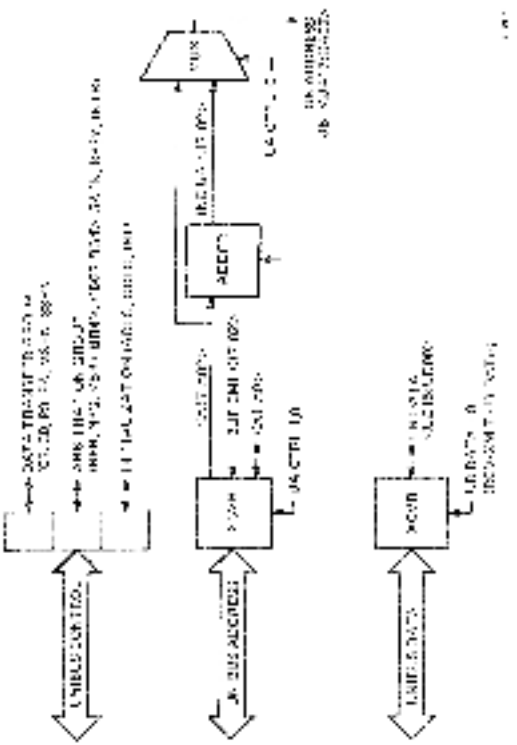
SUB (DW750) BLOCK DIAGRAM



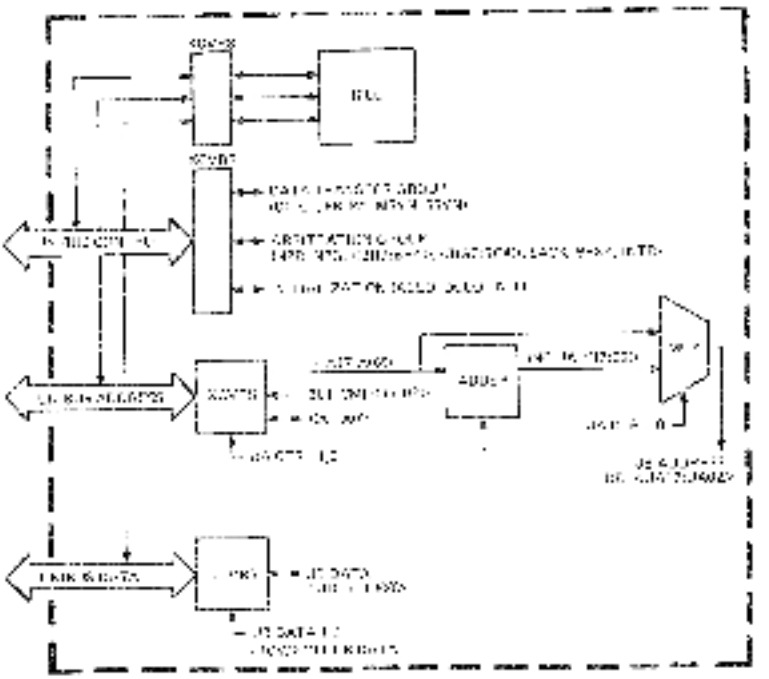
UDP DATA FLOW BLOCK DIAGRAM



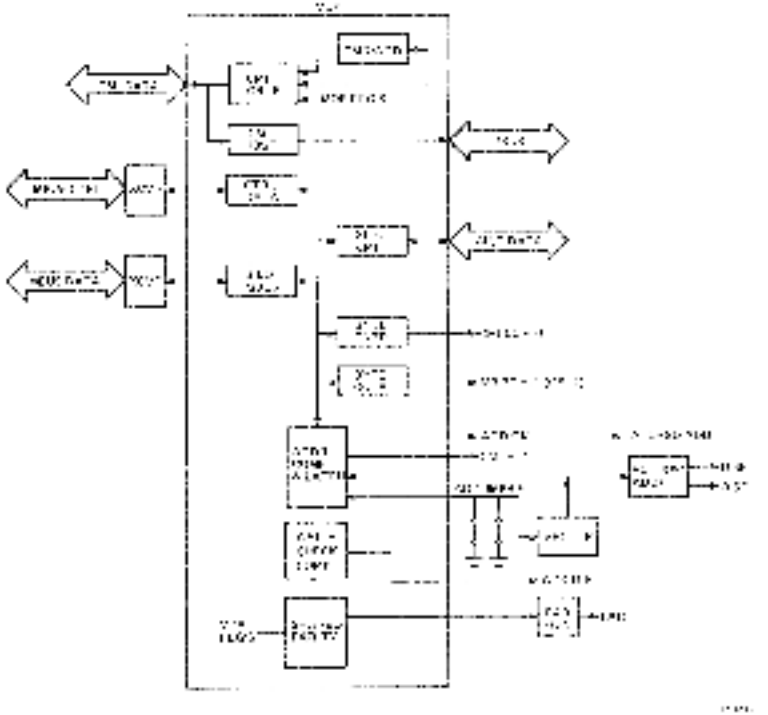
UBI INTERFACE TO FIRST UNIBUS LINES



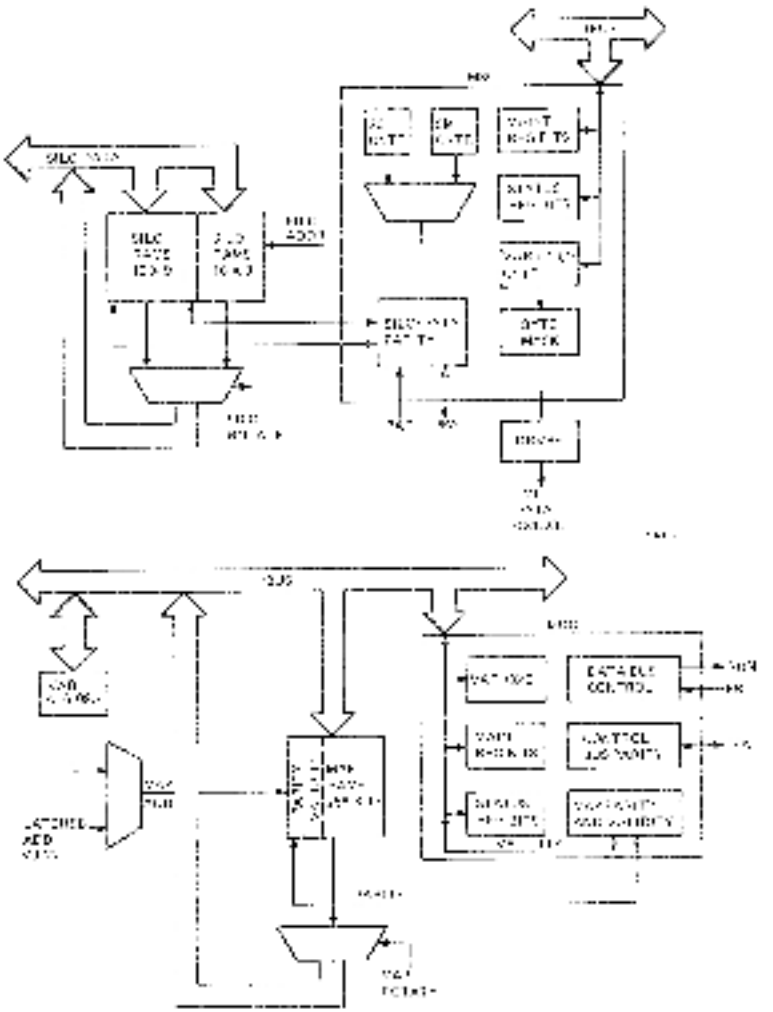
SUB INTERFACE TO SECOND UNIBUS LINES



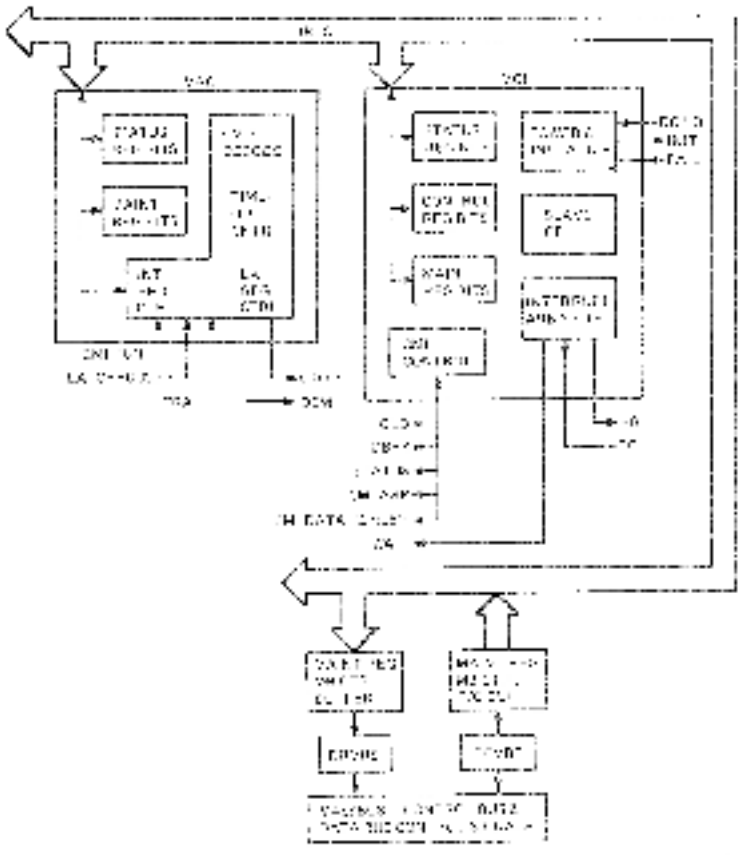
MBA (RH750) BLOCK DIAGRAM, PART 1



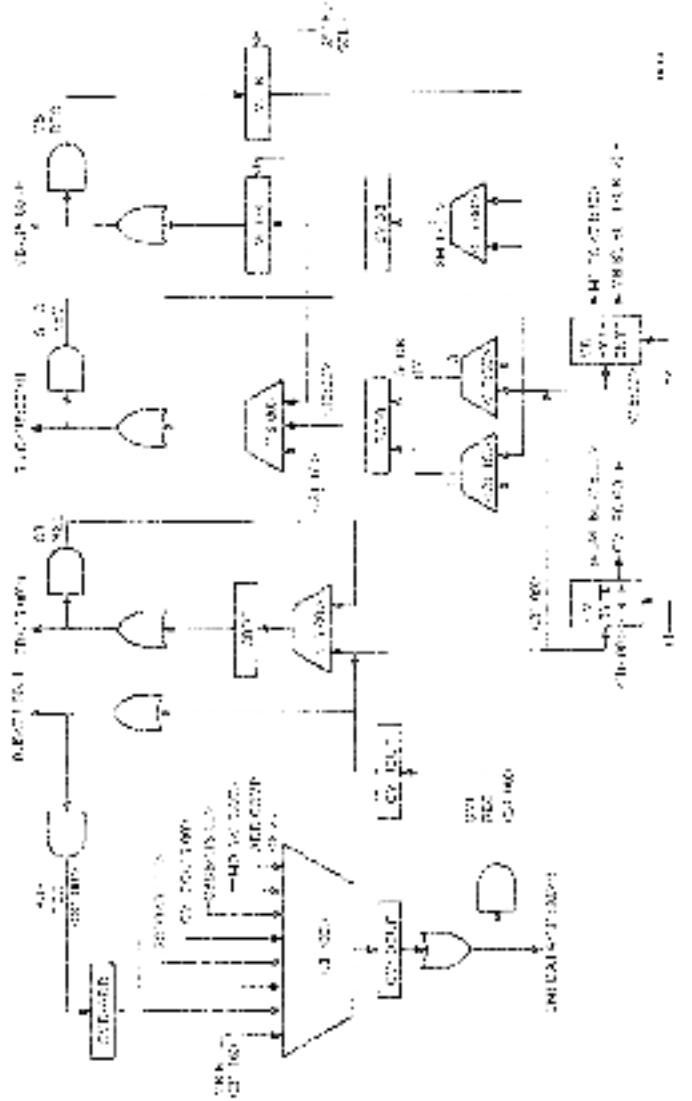
MBA (RH750) BLOCK DIAGRAM, PART 2



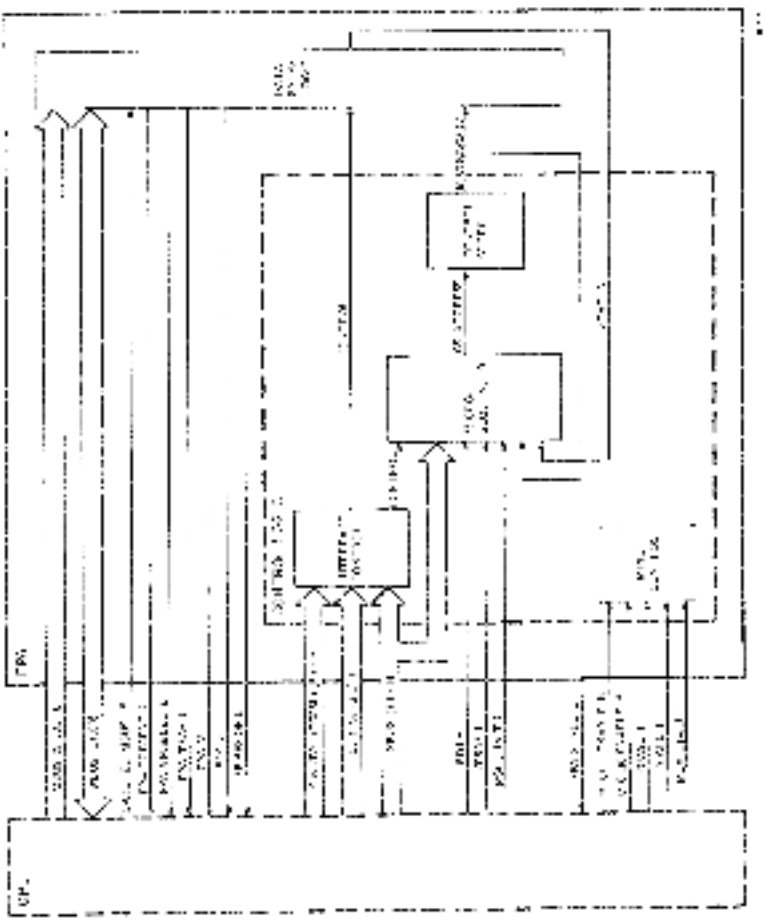
MBA (RH750) BLOCK DIAGRAM, PART 3



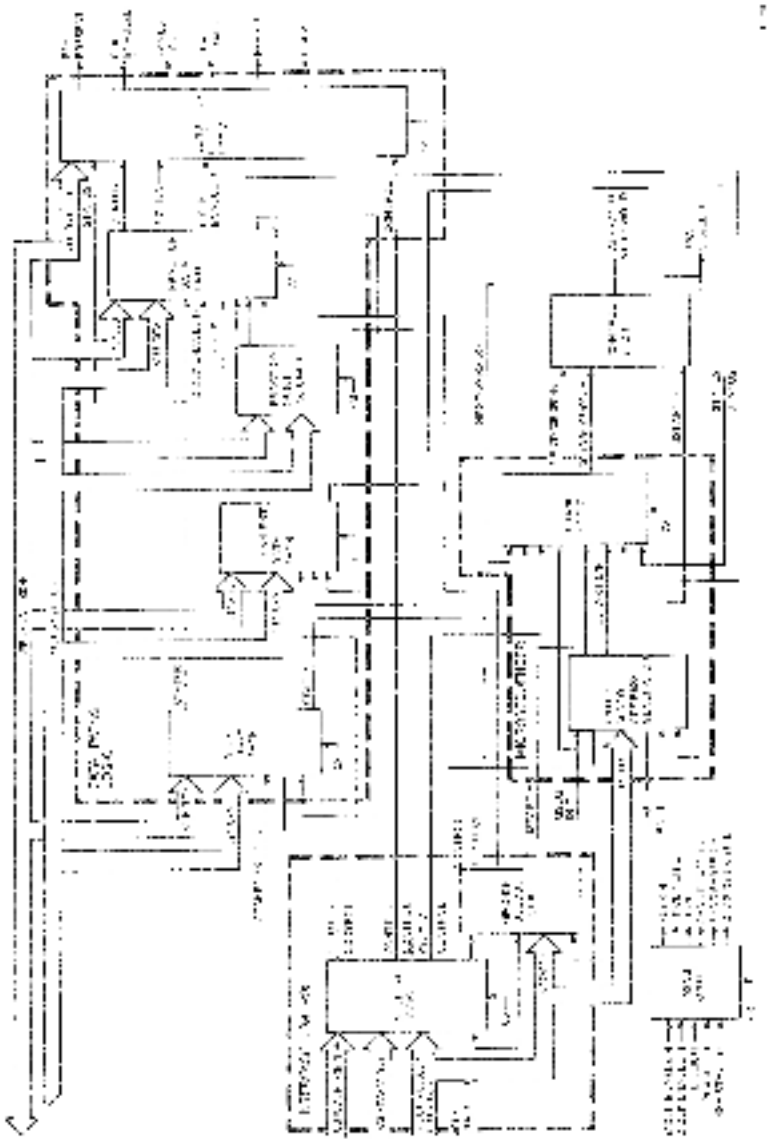
MOP DATA FLOW BLOCK DIAGRAM



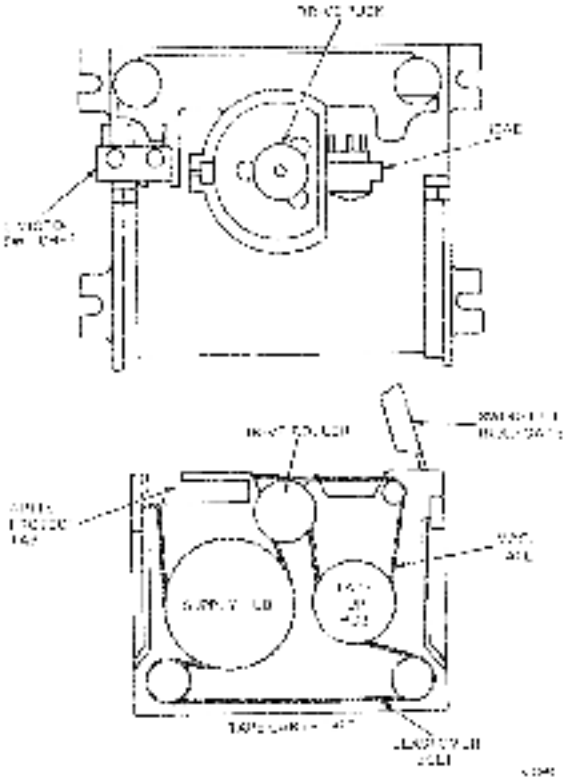
FPA (FP750) BASIC DIAGRAM



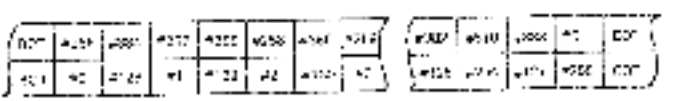
FPA (FP750) BLOCK DIAGRAM



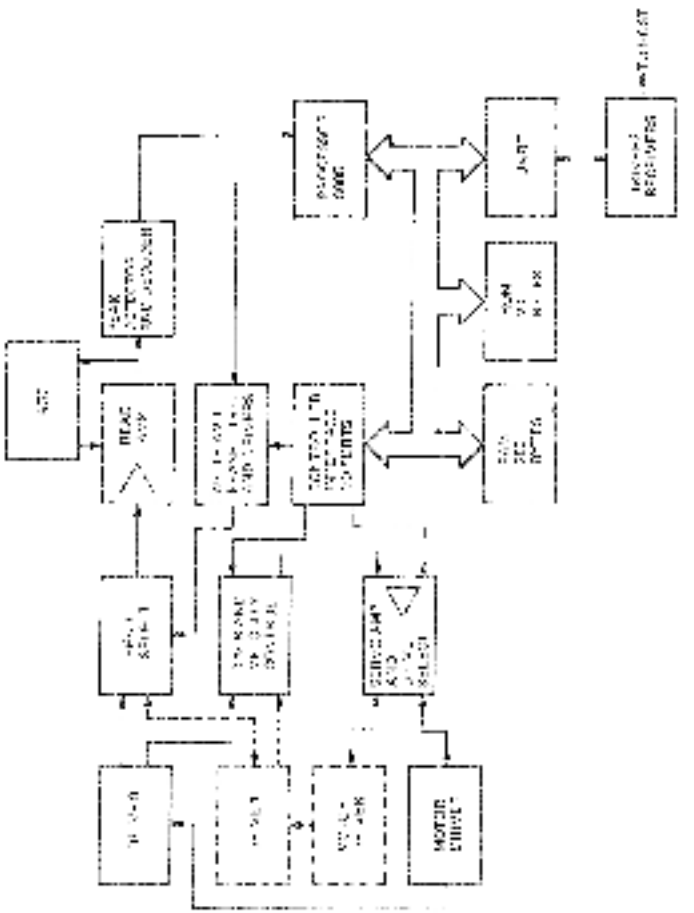
TU58 DRIVE/CARTRIDGE DIAGRAM



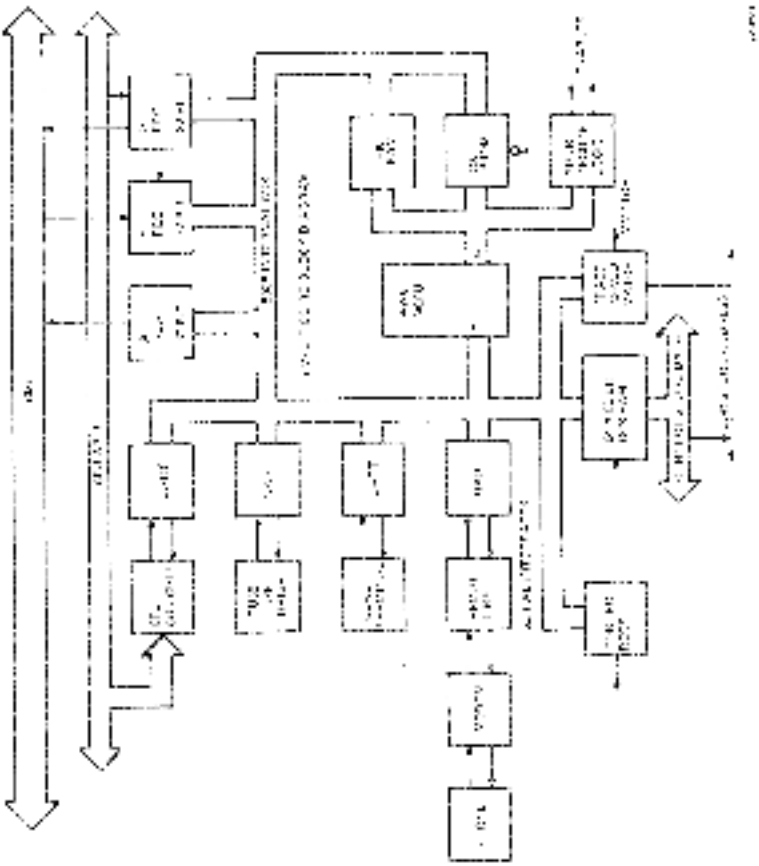
TU58 CASSETTE TAPE BLOCK LOCATIONS



TMS8 BLOCK DIAGRAM

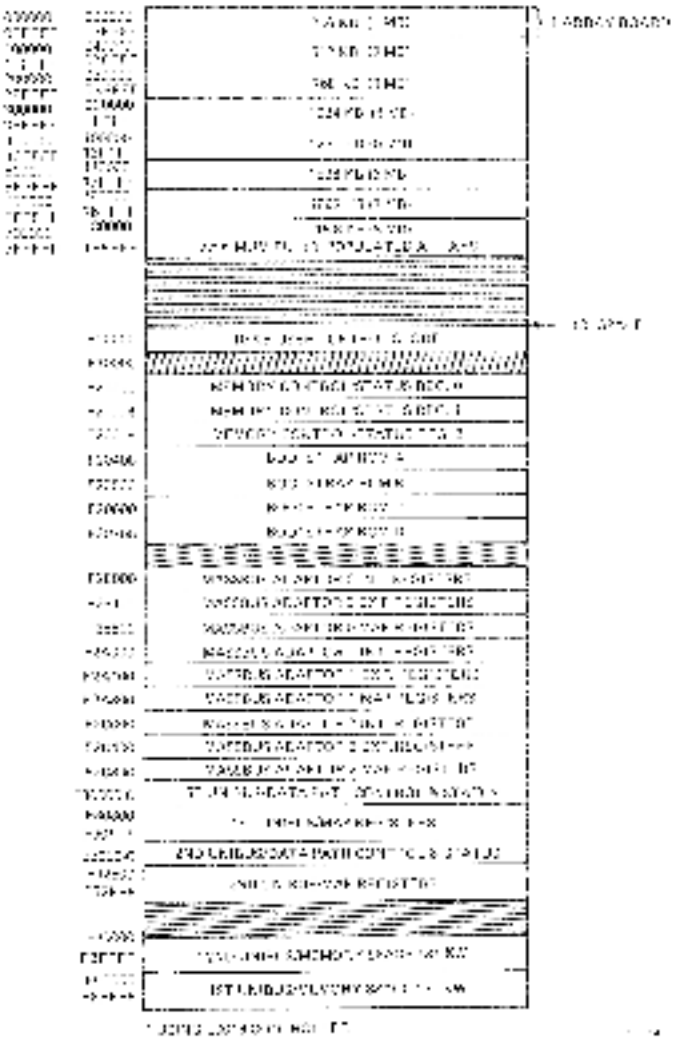


ROM BLOCK DIAGRAM



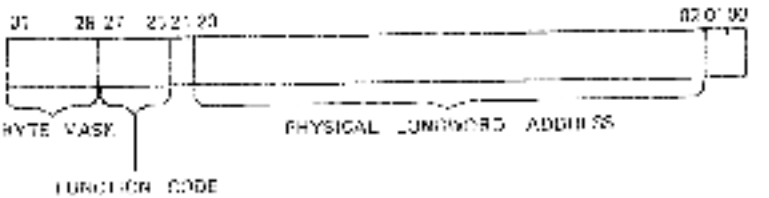
CHAPTER 8
CMI, UNIBUS, AND MASSBUS

CMI PHYSICAL ADDRESS MAP



CMI ADDRESS AND DATA FORMAT

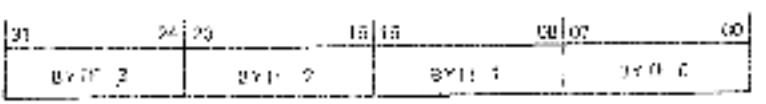
CMI ADDRESS FORMAT



CMI BYTE MASK AND FUNCTION BITS

Byte Mask Bits	Byte(s) Valid for Transfer	Function Bits 27 26 25	CMI Operation
[illegible]	Byte 3 valid	0 0 0	Read
[illegible]	Byte 2 valid	0 0 1	Read Lock
[illegible]	Byte 1 valid	0 1 0	Read Modify
[illegible]	Byte 0 valid	0 1 1	(Undefined)
		1 0 0	Write
		1 0 1	Write Unlock
		1 1 0	Write Modify
		1 1 1	(Undefined)

CMI DATA FORMAT

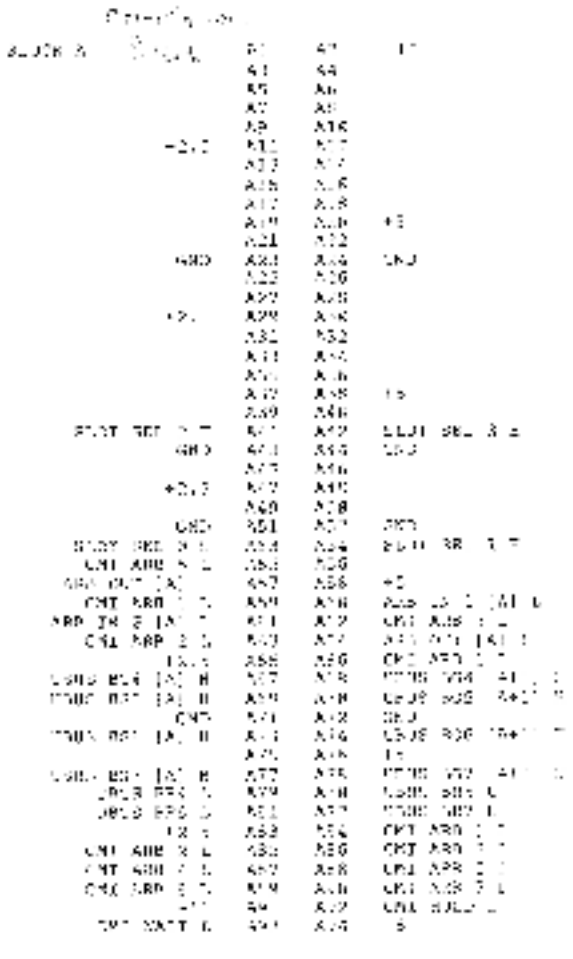


CMI SIGNAL DESCRIPTION

Signal Lines	Description
Timing Group	
> CLK L	A CLK L is generated by the C.I.U. to synchronize all system activities. One > CLK cycle is like one rising edge of a CPU C.I.U. clock. A CLK is low for one-half of the cycle.
Control/Address and Data CPU B	
CMI DATA (31-00)	The CMI data lines are first received by a device that has arbitration for the CMI. One assigned control or other. The new system generates control and address information on the A cycle of the CMI clocks format, are asserts used for one T CLK cycle. Data is then transferred on the lines in the CMI data bits). Bits 31:08 of the CMI address are ignored since four bytes (one longword) of data are transmitted as bits lines are the valid data is referred to the byte mask.
MODE CMI DATA signal is the associated with a higher ... it a nominal +3.7 V (typ.). All other CMI signals are asserted +5 V nominal ground level.	
Data Bus busy (BUSY)	BUSY is first asserted by the device for one T CLK cycle while it clocks the CMI address of the slave or a CPU cell timer. BUSY is then asserted by the slave. Until the data transfer is complete. If the slave is currently ready to respond, it may not assert BUSY.
HOLD	Hold is used to temporarily block all CMI activities. Cache for example, requires extra D.M.A. cycles to perform a invalidation. Cache is one part of the DMA transaction and does not exist. D'HOLD
SYNC	SYNC is asserted by a CMI subsystem to initiate a processor synchronization. It is held until the sync cycle operation is performed.

Signal Lines	Description
Priority Arbitration Group	
CARBITRATE	A ARB level is assigned to each subsystem that is used to gain control of the CMI. If more than one higher priority level is not asserted when a subsystem asserts its own ARB level, the subsystem assumes control of the CMI, asserting DBSY and the low-order address of a block subsystem.
	If also is a higher priority level is asserted, the subsystem asserts its own priority bit to hold off subsystems of lower priority until it gains control. Priority on the CMI is assigned as follows:
	ARB7 - REF - highest priority ARB6 - processor ARB5 - reserved ARB4 - IPI ARB3 - MBA 3 or DMA ARB2 - MBA 2 or VIAX (all LSI) ARB1 - MBA 1 or XMA1 (all SSI) ARB0 - lowest priority (used for deadlock resolution)
Status Group	
STATE <1:0>	Status is transferred by a drive to initiate the condition under which it returns data to the caller. Status bit combinations are encoded as follows:
	Status Bit 1 0 0 0 DVM - The buffer (slot) is access pointer to start reading (slave to CPU) or write data. 0 1 ECC - Read data returned to the master carried an uncorrectable error. 1 0 CCC - Corrected read data was returned to the master. 1 1 NT ERR - The read data bit is wrong.

CMI SIGNAL PIN ASSIGNMENTS ON OPTION SLOTS



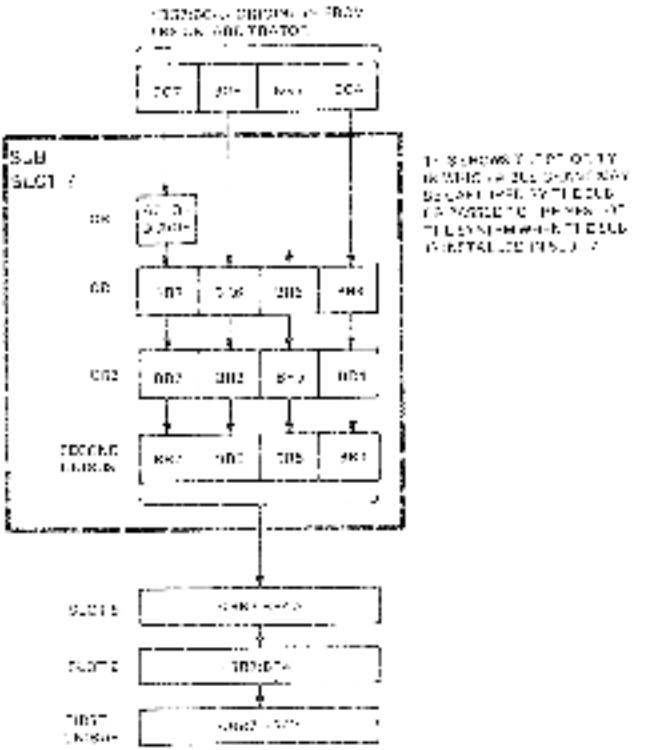
CMI SIGNAL PIN ASSIGNMENTS ON OPTION SLOTS (CONT)

SLOT 0	CMI DATA 0 H	A1	B	-12		
	CMI DATA 1 H	A2	B1	CMI DATA 1 H		
	CMI DATA 2 H	A3	B2	CMI DATA 2 H		
	CMI DATA 3 H	A4	B3	CMI DATA 3 H		
	CMI DATA 4 H	A5	B4	CMI DATA 4 H		
	CMI DATA 5 H	A6	B5	CMI DATA 5 H		
	CMI DATA 6 H	A7	B6	CMI DATA 6 H		
	CMI DATA 7 H	A8	B7	CMI DATA 7 H		
	+2.5	A9	B8	+2.5		
	CMI DATA 8 H	A10	B9	CMI DATA 8 H		
	CMI DATA 9 H	A11	B10	CMI DATA 9 H		
	CMI DATA 10 H	A12	B11	CMI DATA 10 H		
	CMI DATA 11 H	A13	B12	CMI DATA 11 H		
	CMI DATA 12 H	A14	B13	CMI DATA 12 H		
	GND	A15	B14	GND		
	CMI DATA 13 H	A16	B15	CMI DATA 13 H		
	CMI DATA 14 H	A17	B16	CMI DATA 14 H		
	+2.5	A18	B17	+2.5		
	CMI DATA 15 H	A19	B18	CMI DATA 15 H		
	CMI DATA 16 H	A20	B19	CMI DATA 16 H		
	CMI DATA 17 H	A21	B20	CMI DATA 17 H		
	+2.5	A22	B21	+2.5		
	CMI DATA 18 H	A23	B22	CMI DATA 18 H		
	CMI DATA 19 H	A24	B23	CMI DATA 19 H		
	CMI DATA 20 H	A25	B24	CMI DATA 20 H		
	CMI DATA 21 H	A26	B25	CMI DATA 21 H		
	+2.5	A27	B26	+2.5		
	CMI DATA 22 H	A28	B27	CMI DATA 22 H		
	CMI DATA 23 H	A29	B28	CMI DATA 23 H		
	CMI DATA 24 H	A30	B29	CMI DATA 24 H		
	CMI DATA 25 H	A31	B30	CMI DATA 25 H		
	+2.5	A32	B31	+2.5		
	CMI DATA 26 H	A33	B32	CMI DATA 26 H		
	CMI DATA 27 H	A34	B33	CMI DATA 27 H		
	CMI DATA 28 H	A35	B34	CMI DATA 28 H		
	CMI DATA 29 H	A36	B35	CMI DATA 29 H		
	CMI DATA 30 H	A37	B36	CMI DATA 30 H		
	CMI DATA 31 H	A38	B37	CMI DATA 31 H		
	+2.5	A39	B38	+2.5		
	CMI DATA PAR H	A40	B39	CMI DATA PAR H		
	GND	A41	B40	GND		
	CMI STATUS 0 L	A42	B41	CMI STATUS 0 L		
	CMI STATUS 1 L	A43	B42	CMI STATUS 1 L		
	GND	A44	B43	GND		
	CMI STATUS PAR L	A45	B44	+5		
	+2.5	A46	B45	LA750		
	+2.5	A47	B46	+5		
	GND	A48	B47	GND		
		A49	B48			
		A50	B49			
		A51	B50			
		A52	B51			
		A53	B52			
		A54	B53			
		A55	B54			
		A56	B55			
		A57	B56			
		A58	B57			
		A59	B58			
		A60	B59			
		A61	B60			
		A62	B61			
		A63	B62			
		A64	B63			
		A65	B64			
		A66	B65			
		A67	B66			
		A68	B67			
		A69	B68			
		A70	B69			
		A71	B70	GND		
	SLOT 1 AVAILABLE H	A72	B71	+5		

UMI SIGNAL PIN ASSIGNMENTS ON OPTION SLOTS (CONT)

FLOPPY 0	BDISK	C1	C2	C3
		C3	C4	
		C5	C6	
		C7	C8	
		C9	C10	
		C11	C12	
		C13	C14	
		C15	C16	
		C17	C18	
		C19	C20	
		C21	C22	
		C23	C24	
		C25	C26	
		C27	C28	
		C29	C30	
		C31	C32	
		C33	C34	
		C35	C36	
		C37	C38	
		C39	C40	
		C41	C42	GND
	+5	C43	C44	GND
	BDISK (CLOCK)	C45	C46	+5
	+2.5	C47	C48	INTR(5) INTR(7)
	+2.5	C49	C50	+5
	+5	C51	C52	GND
		C53	C54	
		C55	C56	
		C57	C58	
		C59	C60	
		C61	C62	
		C63	C64	
		C65	C66	
		C67	C68	
		C69	C70	
		C71	C72	
		C73	C74	
		C75	C76	
		C77	C78	
		C79	C80	
		C81	C82	
		C83	C84	
		C85	C86	
		C87	C88	
		C89	C90	
		C91	C92	GND
	BDISK (CLOCK)	C93	C94	+5

BUS GRANT CHAIN AND CONTINUITY JUMPERS



BUS GRANT CHAIN AND CONTINUITY JUMPERS (CONT)

Each of the Unibus backplane option slots is hardwired to one BR priority level for bus arbitration. The priority level is determined by the slot position. The priority level is as follows:

Bus Grant Level	Backplane Slot Jumpers
BR7	A07 to A08
BR6	A05 to A06
BR5	A03 to A04
BR4	A01 to A02

CAUTION

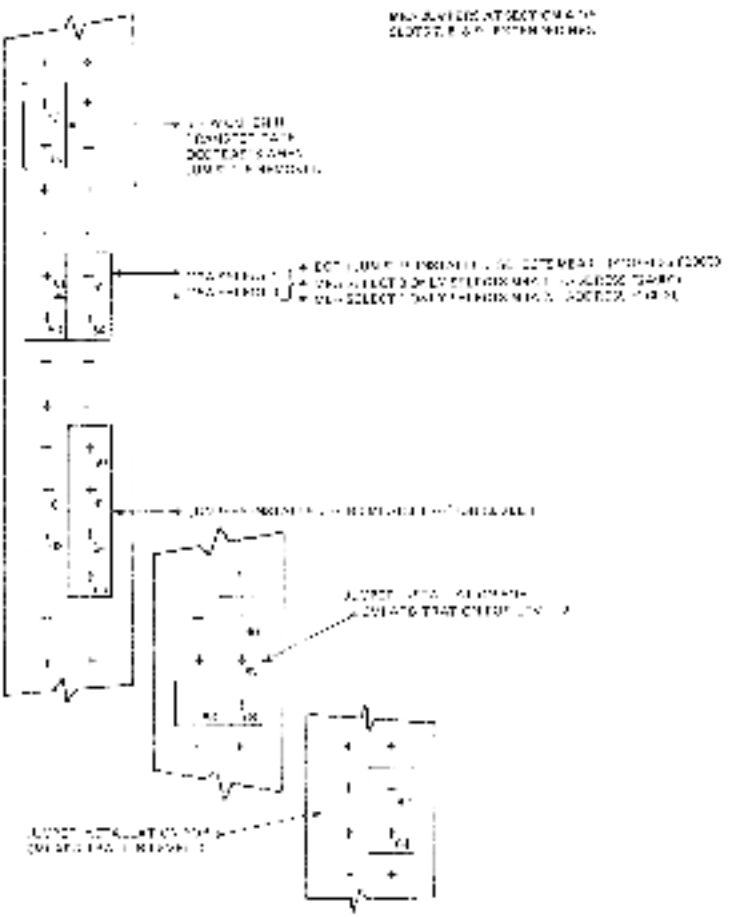
Before the CPU is unpacked for installation, refer to the Owner Installation Procedure that describes the use of a grounding mat and provides data to prevent static discharge damage.

Grounding must first be made between a snap fastener on the mat and a good ground on the system. Connection to this main floor ground must be by the installer to an alligator clip that is fastened to the mat.

The mat is placed on the VERTICAL mat, and grounded. The module is then removed from the box, the protective covering removed, and the module laid flat on the VERTICAL mat. This brings it to the same potential as the CPU.

With the wrist strap still in place, the Unibus module is plugged into slot 7 (or next cabling cable), and be sure the second UNIBUS at first priority in the bus grant chain.

MBA INSTALLATION JUMPERS



MBA INSTALLATION JUMPERS (CONT)

MBA Select Jumpers

```
To Select MBA          Install Jumper
As Follows             Between

MBA 3                  A53 to A57 (Jumper)
                       A54 to A58 (Jumper)

MBA 2                  A51 to A55
                       (A52 Removed)

MBA 1                  A50 to A54
                       (A53 Removed)

Slave (SL) Jumper      A47 to A48 (Jumper normally in place).
                       Removing this jumper reduces bus
                       rate to 4.5 MB/s (the rate is fixed)
                       and disconnects an RC1 before bus
                       is not and other is in slave
```

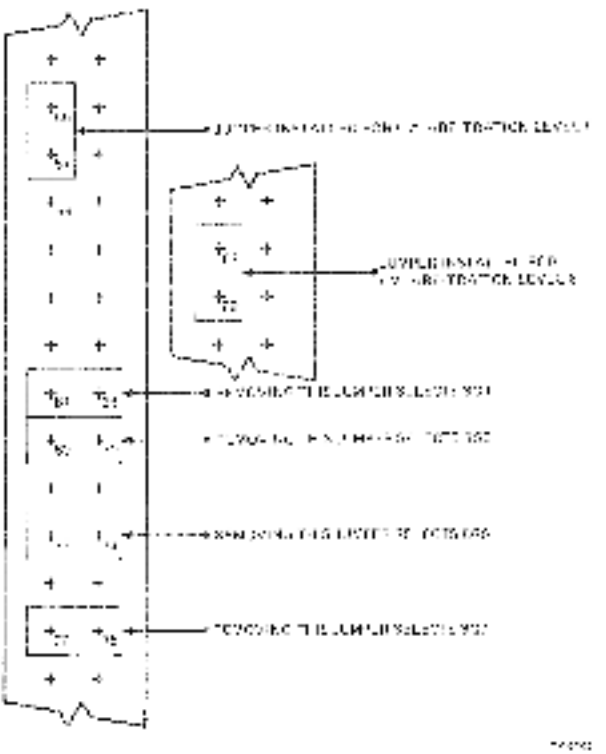
MBA CMI Arbitration Jumpers

```
To Select    For       Install Jumper
ARB Level    MBA       Between

ARB 3        MBA 3     A14 to A12 (MBA ARB <2> to CMI ARB <2>)

ARB 2        MBA       A14 to A11 (MBA ARB <1> to CMI ARB <1>)
                       A12 to A10 (CMI ARB <3> IN)

ARB 1        MBA 2     A14 to A07 (MBA ARB <0> to CMI ARB <0>)
                       A11 to A09 (CMI ARB <2> IN)
                       A12 to A08 (CMI ARB <3> IN)
```

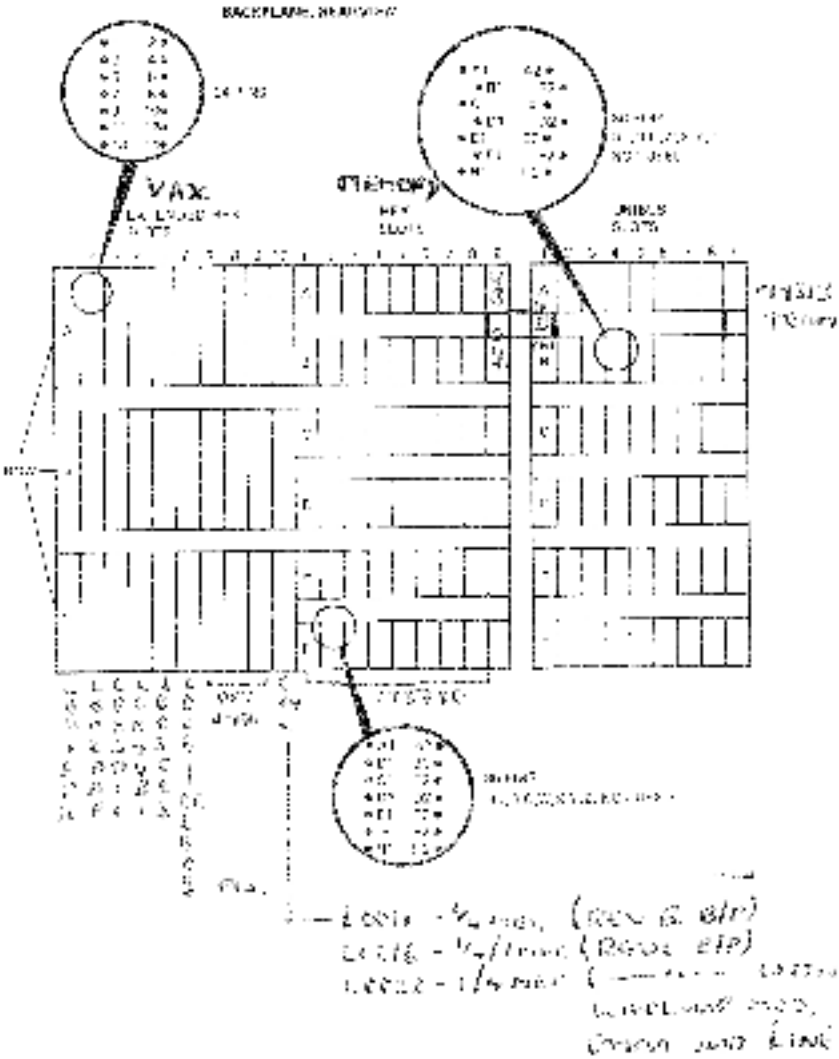
MBA Interrupt Priority Plug

```
To Set MBA Bit         Use Priority Plug
BR Priority Level      Part Number

7                      54-09772-02
6                      54-09773-02
5                      54-09774-02
4                      54-09775-02

Unused* BR level (Only Required)
```

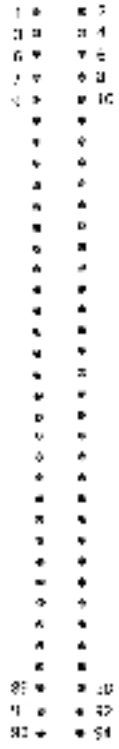
RESERVED ARBITRATION JUMPERS



MODULE BLOCK LAYOUT



MODULE PIN BREAKOUTS

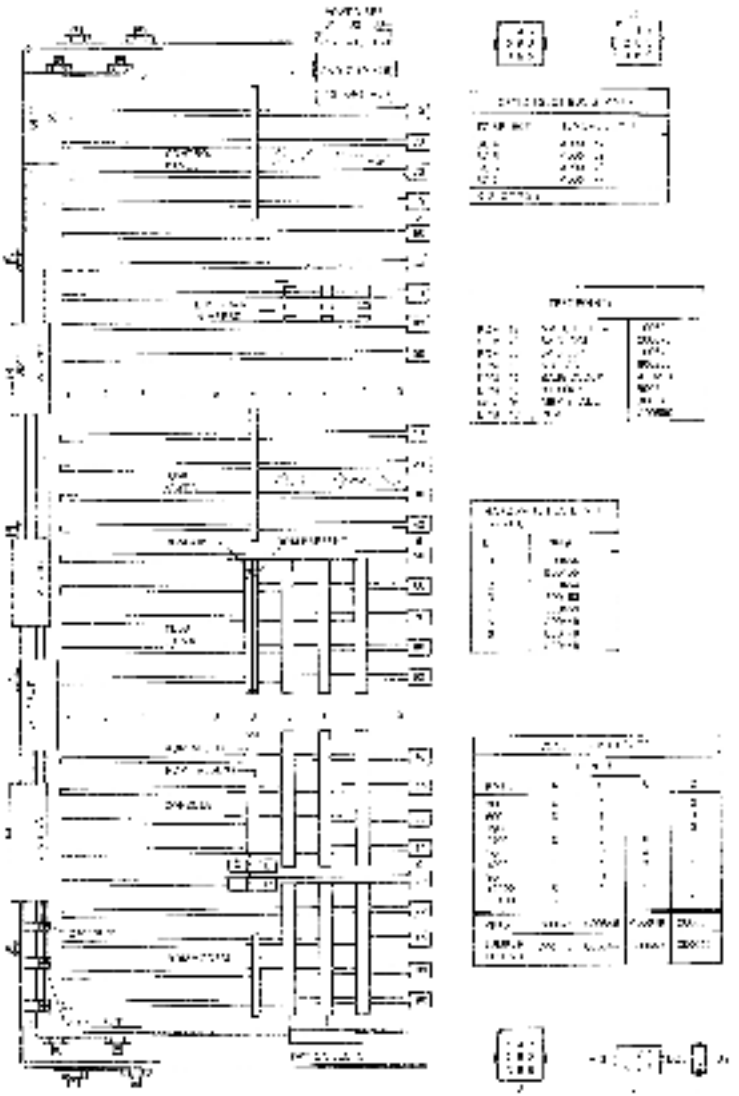


HEX PIN BREAKOUT

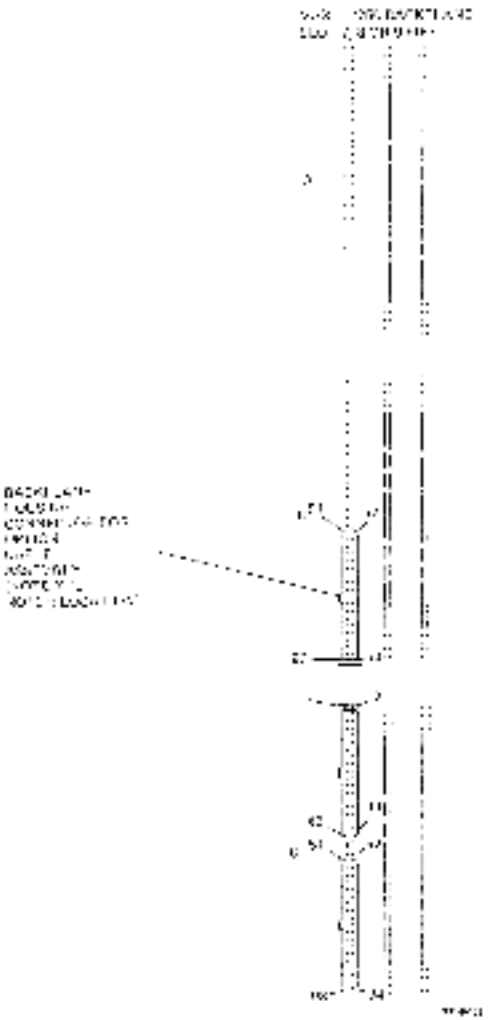
UNIBUS PIN BREAKOUT

EXTENDED HEX
PIN BREAKOUT

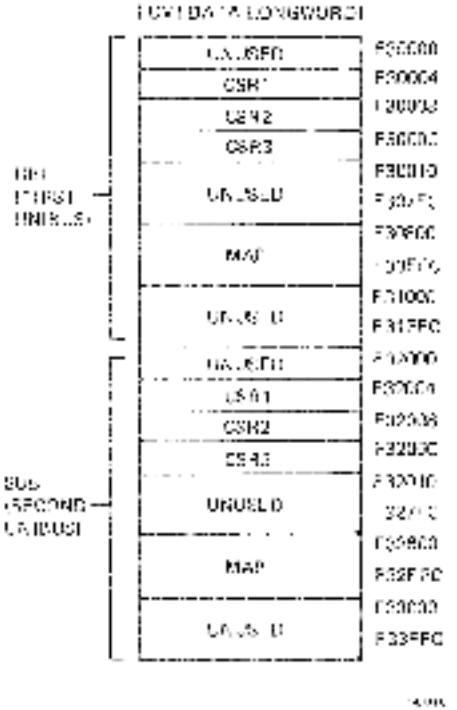
CPU BACKPLANE FROM PIN SIDE



BACKPLANE CONNECTOR HOUSING INSTALLATION DIAGRAM



UBI AND SUB PHYSICAL ADDRESS SPACE



```
UBI        PIR        Register
Address    Address    Name        Description/Contents

F30304     F17A84     CSR 1       CSR for UBI 1
F30308     F17A88     CSR 2       CSR for UBI 2
F3030C     F17A8C     CSR 3       CSR for UBI 3

F30 0N     F32800     MAP 00      Physical page address
F30x04     F32824     MAP 01
F30x08     F32828     MAP 02      The 1 X 8kb (16kbytes via
E30x0C     F3282C     MAP 03      extension to Jupiter/Ptr 32
F30x10     F32830     MAP 04      four through the range of
E30x14     F32834     MAP 05      8kb-P2F (hex).
F30x18     F32838     MAP 06
F30x1C     F3283C     MAP 07      The physical UBI address may
F30x20     F32840     MAP 08      be obtained by adding the
    .          .          .       MAP  relative number in the
    .          .          .       previous base address.
    .          .          .
    .          .          .
F30F00     F32F00     MAP 0F0
F30F04     F32F04     MAP 0F4
F30F08     F32F08     MAP 0F8
F30F0C     F32F0C     MAP 0FC

FF3000     FF0000     UB.BUS      Important base address for
                                  "UNIBUS I/O devices.
    .          .          .
    .          .          .       The UBI number of a device
    .          .                  is obtained by converting
    .          .          .       the octal device address to
    .          .          .       hexadecimal and adding it to
    .          .          .       the physical base address.
    .          .          .

FFBFFC     FFBFFC     UB.BUS

E3F040     E3E480     UBA/XBAR    UBI address register
E3F044     FF7407        .        is UBIA register
E3F048     E3E484        .        "EXT-CSR. (CSR C.0)
FFF407     FF7404     I.CSR/LPEL  LOOP-CSR. "EXT PROG"
                                  data register option
```

(All UBI address ranges are listed with the PROM software. Including the PROM data register that contains PROM card revision.

FIXED DEVICE ADDRESS AND VECTOR ASSIGNMENTS

Assigned UNIBUS devices normally occupy vector addresses
through 774 octal (252 through 508 decimal).

Floating UNIBUS devices may be assigned vector addresses
through 774 octal (252 through FF hexadecimal).

UNIBUS Device	Address Base	Address Size	Vector Base	Vector Size	No. of Units/ Comments
AR11 (1)	772610	8	70	2	1st unit
AR11	772620	8			4 extra units
AR11	772640	4			1 unit
ADH1	772710	4	130	2	1
AJP11	772440	8			1
APU11-A	772460	2	120	4	1
APU11	772470	2	110	4	1
A-11	772662	8			1
BA11 (BA-	77+62e	3			1
BDV11-BJA	77+640	1			1
BJA11-EJK	772002	250			1
BK-V2 YA	772062	32			1
BK702-YB	772102	37			1
BK702-YC	772202	37			1
BK702-YD	772302	37			1
BK11A YA	772002	120			1
BK11A-YB/YC	772002	250			1
BXV11			114	2	1 CSR class
BPV11	772150	2			1 CSR class
BQ-CV/CB11	772160	2	260	4	1
CR11	764472	4	170	2	1 CSR device
CP8			114	2	
CP811	764002	4	220	2	1 CSR device
DC11/Base	764002	1024			1
DC11	774002	4			32
DL A-D	777562	8			1
DL,pseudo	762002	4			1
DP211			210	2	
DT/DTV 1-A/B	772342	6	224		1 - console
DT/DTV11-A/B	772342	6			16
DT11-E/F/S	772010	4			31
DJV11-A	770510	4			4
DJV11-B	770500	4			16
DJV11-A	770500	10			4
DL 1-A	772e46	2	060	4	1st LDC
DL 1-A	772560	4			1 - console
DT 1-E	772e0e	6			16
DU11-BA/DA	772500	6			16 - serial
DX11	772360	4			16
DH11-AA	772760	8			16
DR 1-DA	772e0e	2			64

FIXED DEVICE ADDRESS AND VECTOR ASSIGNMENTS (CONT)

UNIBUS Device	Address Base	Address Size	Vector Base	Vector Size	No. of Units/Comments
[illegible]	[illegible]	4			[illegible]
[illegible]	[illegible]	[illegible]			[illegible]
[illegible]	[illegible]	4	[illegible]	2	2
[illegible]	[illegible]	4			
[illegible]	767736	[illegible]			[illegible]
[illegible]	[illegible]	4	[illegible]	2	[illegible]
[illegible]	764120	8			[illegible]
[illegible]	[illegible]	64			[illegible]
[illegible]	777420	[illegible]			[illegible]
[illegible]	775200	[illegible]			[illegible]
[illegible]	[illegible]	16			[illegible]
[illegible]	[illegible]	4	[illegible]		[illegible]
Floating Address	[illegible]	[illegible]			
Floating Vectors			[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]			1
[illegible]		[illegible]	[illegible]	2	
[illegible]	[illegible]	4			[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	4	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	2	[illegible]
[illegible]	[illegible]	[illegible]			[illegible]
[illegible]	[illegible]	[illegible]			6
[illegible]	[illegible]	4	[illegible]	4	16
[illegible]	[illegible]	4			1 - [illegible]
[illegible]	[illegible]	[illegible]			1 - [illegible]
[illegible]	[illegible]	[illegible]			1
[illegible]	[illegible]	1			1
[illegible]			[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]			1
[illegible]			[illegible]	4	
[illegible]	[illegible]	1	[illegible]	[illegible]	1
[illegible]	[illegible]	6	[illegible]	[illegible]	1
[illegible]	[illegible]	4			1
[illegible]	[illegible]	2	[illegible]	4	1
[illegible]	[illegible]	8	[illegible]	2	[illegible]
[illegible]	[illegible]	8	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	8	[illegible]	2	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	8			[illegible]
[illegible]	[illegible]	[illegible]			[illegible]
[illegible]	[illegible]	[illegible]			[illegible]
[illegible]	[illegible]	32	[illegible]	[illegible]	2
[illegible]			[illegible]	[illegible]	
[illegible]	[illegible]	8			[illegible]
[illegible]	[illegible]	16			1

*pending assignments

FIXED DEVICE ADDRESS AND VECTOR ASSIGNMENTS (CONT)

ENTRY/ Device	Address Base	Address Size	Vector Name	Vector Size	No. of Lines/ Controller
[illegible]	[illegible]	[illegible]			8
KY32/	[illegible]	2			1
KY30/-SA	[illegible]	2[illegible]			1
KW3U/ SA	[illegible]	[illegible]			1
PLI/	[illegible]	22			1 - KW32/BF-11
KMC/	[illegible]	1			[illegible]
KA[illegible]-DA	[illegible]	64			1
KDA50-Q	[illegible]	[illegible]			1
KA[illegible]/MSV11	[illegible]	1			[illegible]
RQDX1/	[illegible]	4			1
DL11	[illegible]	8			1
DZ11-A	[illegible]	4	[illegible]	8	[illegible]
DZ11-B	[illegible]	4	[illegible]	8	[illegible]
DC11/DX11	[illegible]	4	[illegible]	8	1
DH11	[illegible]	16	[illegible]	4	1 - DZ11 device
DTE-11	[illegible]	[illegible]			1
DL-11 Receiver			[illegible]	1	
DL-11 XMT Buffer			[illegible]	1	
DTE-11 Reserved			[illegible]	2	
Interrupts					
DTE-11 Receive/ Transmit Trap			[illegible]	1	
DL-11 I/O Trap			[illegible]	1	
DTE-11 Packet Int			[illegible]	2	
DE[illegible]-11 FPT Trap			[illegible]	1	
DTE-11 DUAP Trap			[illegible]	1	
DTE-11 VLAG			[illegible]		
DR11	[illegible]	4			1
RF11	[illegible]	8	[illegible]	2	1
Reserved	[illegible]	32			1
Reserved	[illegible]	8			1
Reserved	[illegible]	8			1
Reserved	[illegible]	4			1
Reserved	[illegible]	8			1
Reserved	[illegible]	8			1
Reserved	[illegible]	32			1
Reserved	[illegible]	4			1
RC25 (1)	[illegible]	256			1
RC25 (2)	[illegible]	256			1
RL11	[illegible]	8	[illegible]	2	1
RP03/RP11-P[illegible]	[illegible]	32	[illegible]	2	All RP/RM/TU
RF11/RC11	[illegible]	8	[illegible]	2	1
RS11/RX01	[illegible]	8	[illegible]	2	1
RL11/211	[illegible]	8	[illegible]	2	1
RC211	[illegible]	8	[illegible]	2	1
RK611/3/04/67	[illegible]	32	[illegible]	2	1 RK7[illegible]
RM02/03/05	[illegible]	32	[illegible]	2	1 RM7[illegible]
RF11	[illegible]	16	[illegible]	2	1
RS03/RS04	[illegible]	16	[illegible]	2	1 - RF11/RB11

UNIBUS Device	Address Octal	Address Hex	Vector Octal	Vector Hex	No. of Unibus Interrupts
RX11/211	777170	1	154	2	1
RX11/211	777170	4	154	2	1
MSV11-P Front Panel			160	2	
MSV11-P Diagnostic Failure			274	2	
System Software Reserved			310		
System Software Reserved			340	8	
TA 11/TU11 A	777730	7	260	2	1
TS 11	772526	5	224	2	1
LSIA	772540	16	240	2	1 - SHARED
Thermal	772380	17			
RH11/TM11	77 626	8	224	2	1
DPV8	764002	6			1
DS11	772520	2	224	2	1
LINK11-CT	772540	14	224	2	1 - SHARED
TM78	772520	7			1
TM78	772440	37	260	2	1 - SHARED
DJ8	772 56	2	150	2	DEC Systems
DZ11	77 770	2	234	2	
UDC Users	77 563	1			255
DS1	772570	2			
CAB,UB Map	772202	64			
Cluster - Reserved for Diagnostic			164	2	
Cmpr/CCS Reserved			330	4	
Cmpr/DEC Reserved			270	4	
VSV11	772RG2	4			4
VT40	772RG2	17			1
VZ52	772562	1..			2
VT11	772537	7	120	2	1

FLOATING ADDRESS UNIBUS DEVICES

Order	Device	Address	Size
1	DL11		4
2	DT11		8
3	TU10		4
4	DJ/DL11		4
5	DC11		4
6	DR11		4
7	DS11		4
8	DM11		4
9	TU12		4
10	AR11		4
11	LP11		4
12	VP1		4
13	DR11		4
14	DM77		4
15	DJ/DH11		4
16	DMA1-7		8
17	DR11-C		4
18	KW11		4
19	DX11		4
20	DT11-M		4
21	DL-11 E		4
22	DR11-AC		4
23	AD01		4
24	DR11		4
25	DM11-AB		8
26	LK8/VNA		4
27	DEA		2
28	DM11		16

FLOATING VECTOR UNIBUS DEVICES

Entry	Device	Size
0	[illegible]	4
1	[illegible]	4
2	[illegible]	4
3	[illegible]	4
4	[illegible]	4
5	[illegible]	4
6	[illegible]	4
7	[illegible]	4
8	[illegible]	4
9	[illegible]	4
10	[illegible]	4
11	[illegible]	4
12	[illegible]	4
13	[illegible]	4
14	[illegible]	4
15	[illegible]	4
16	[illegible]	4
17	[illegible]	4
18	[illegible]	4
19	[illegible]	4
20	[illegible]	4
21	[illegible]	4
22	[illegible]	4
23	[illegible]	4
24	[illegible]	4
25	[illegible]	4
26	[illegible]	4
27	[illegible]	4
28	[illegible]	4
29	[illegible]	4
30	[illegible]	4
31	[illegible]	4
32	[illegible]	4
33	[illegible]	4
34	[illegible]	4

Entry	Device	Size
35	[illegible]	4
36	[illegible]	4
37	[illegible]	4
38	[illegible]	4
39	[illegible]	4
40	[illegible]	4
41	[illegible]	4
42	[illegible]	4
43	[illegible]	4
44	[illegible]	4
45	[illegible]	4
46	[illegible]	4
47	[illegible]	4
48	[illegible]	4

Signal Lines	Description
Data Transfer Lines	
BS valid (BS7:SSYN)	The eight bus address lines are asserted by the master device to select a slave on the basis of device register addresses. <BS7:BS1> address a 128 word (256) addressing space. BS0 selects upper or lower byte of the word addressed.
UB Data (D15:D00)	The data lines are enabled to transfer data information between the master and the slave.
Control (C1:C0)	Two control signals are created by the master device to control the slave in one of four data transfer operations. Valid signals are specified with respect to the master as follows:
	C1 C0
	0 0 Sync To (DATI) -- valid if C1 = 0 is considered to the master from the slave.
	0 1 Data In (DATI) -- a DATI cycle is followed by a DATO or DATOB to the same location.
	1 0 Data Out (DATO) -- a word of data is transferred from the master to the slave.
	1 1 Data Out Byte (DATOB) -- a byte of data is transferred from the master to the slave. The upper or lower byte is specified by <BS0>.
Parity (PA,PB)	Two parity signals are supplied by the slave on a DATI transfer. If (either (or both) of the normally used pair is not utilized):
	PB PA
	0 0 No Error
	1 0 Parity Error
	0 1 Unspecified
	1 1 Unspecified
Master Sync (MSYN)	MSYN is asserted by the master to indicate to the slave that valid address and control information along with data on a Data/BI is present on the Unibus.
Slave Sync (SSYN)	SSYN is asserted by the slave in response to MSYN from the master. It indicates to the master that data has been placed (DATI) or that DATO data is received (i.e., the UB data lines).

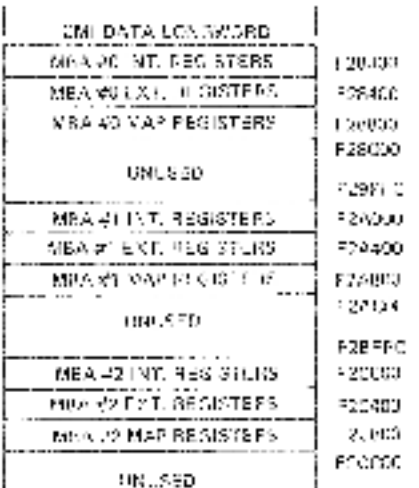
grant lines	(see also)
priority arbitration lines	
bus master (BBSY)	BBSY is asserted by an I/O device for a DMA transfer directly with memory. The processor must synchronize to BBSY before
non-processor grant (NPG)	NPG is the processor response to the NPR. It indicates to the I/O device that it will be the next master when the current master has released BBSY.
bus request (BR4-7)	BR is asserted by an I/O device for a processor interrupt. (Procedure information is included for
bus grant (BG4-7)	BG is the processor response to a BR. It informs the I/O device that it will be the next master. Only the highest request receives a BG grant at any one time.
	SSYN
	All UNIBUS signals are asserted at a nominal ground (low). SSYN must be asserted at a nominal +3.5 V (high).
select acknowledge (SACK)	SACK is asserted by the RPP or DR requesting device that has received a grant. (All devices must see new bus signals when SACK is low before becoming master. Selected slaves must respond.)
bus busy (BBSY)	BBSY is asserted by the current master to indicate that the bus is in use.
initialize (INIT)	INIT is asserted as a new of used by an initializing control that has received a (grant). It informs the UNIBUS as initial system address is present on the UNIBUS. (INIT is also used as the output of BUSY.)

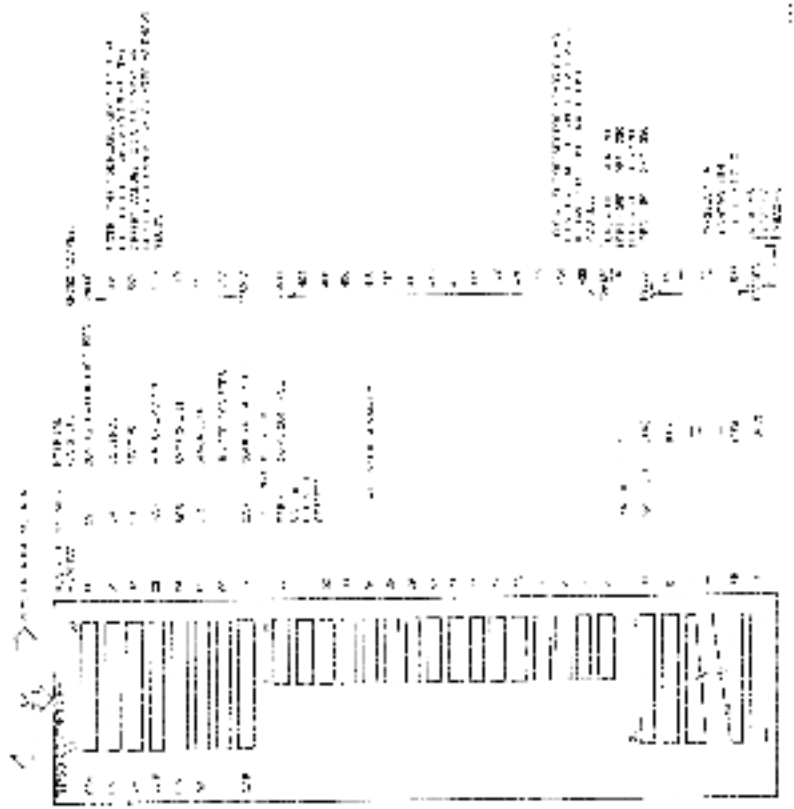
UNIBUS SIGNAL DESCRIPTION (CONT)

Signal Line	Description
Initialization Lines	
Initialize (INIT)	INIT is asserted by the UB and is going at the BUF and DCLO is asserted at the first INTR. (on a system power-up) is also set by a program. To provide assurance that devices will initialize their operation at INIT.
DC Low (DCLO)	DCLO is available when each system power supply and is asserted if an out-of-tolerance condition occurs. DCLO on the first INTR should wait to both the first and second DCLO. DCLO on the second INTR returns DCLO status to the BUF if the CPU attempts to access a second failed device.
AC Line Low (ACLO)	ACLO warns of an impending power failure. ACLO on the first INTR initiates the power-fail routine sequence and may be used in multiprocessor devices to terminate operations and save data. ACLO on the second INTR returns the BUF to normal and the processor if the ACLO is on. If the ACLO is not.

Pin	Signal Name	Pin	Signal Name
A01	[illegible]	B01	[illegible]
A02	[illegible]	B02	[illegible]
A03	[illegible]	B03	[illegible]
A04	[illegible]	B04	[illegible]
A05	[illegible]	B05	[illegible]
A06	[illegible]	B06	[illegible]
A07	[illegible]	B07	[illegible]
A08	[illegible]	B08	[illegible]
A09	[illegible]	B09	[illegible]
A10	[illegible]	B10	[illegible]
A11	[illegible]	B11	[illegible]
A12	[illegible]	B12	[illegible]
A13	[illegible]	B13	[illegible]
A14	[illegible]	B14	[illegible]
A15	[illegible]	B15	[illegible]
A16	[illegible]	B16	[illegible]
A17	[illegible]	B17	[illegible]
A18	[illegible]	B18	[illegible]
A19	[illegible]	B19	[illegible]
A20	[illegible]	B20	[illegible]
A21	[illegible]	B21	[illegible]
A22	[illegible]	B22	[illegible]
A23	[illegible]	B23	[illegible]
A24	[illegible]	B24	[illegible]
A25	[illegible]	B25	[illegible]
A26	[illegible]	B26	[illegible]
A27	[illegible]	B27	[illegible]
A28	[illegible]	B28	[illegible]
A29	[illegible]	B29	[illegible]
A30	[illegible]	B30	[illegible]
A31	[illegible]	B31	[illegible]
A32	[illegible]	B32	[illegible]
A33	[illegible]	B33	[illegible]
A34	[illegible]	B34	[illegible]
A35	[illegible]	B35	[illegible]
A36	[illegible]	B36	[illegible]
A37	[illegible]	B37	[illegible]
A38	[illegible]	B38	[illegible]
A39	[illegible]	B39	[illegible]
A40	[illegible]	B40	[illegible]
A41	[illegible]	B41	[illegible]
A42	[illegible]	B42	[illegible]
A43	[illegible]	B43	[illegible]
A44	[illegible]	B44	[illegible]
A45	[illegible]	B45	[illegible]
A46	[illegible]	B46	[illegible]
A47	[illegible]	B47	[illegible]
A48	[illegible]	B48	[illegible]
A49	[illegible]	B49	[illegible]
A50	[illegible]	B50	[illegible]
A51	[illegible]	B51	[illegible]
A52	[illegible]	B52	[illegible]
A53	[illegible]	B53	[illegible]
A54	[illegible]	B54	[illegible]
A55	[illegible]	B55	[illegible]
A56	[illegible]	B56	[illegible]
A57	[illegible]	B57	[illegible]
A58	[illegible]	B58	[illegible]
A59	[illegible]	B59	[illegible]
A60	[illegible]	B60	[illegible]
A61	[illegible]	B61	[illegible]
A62	[illegible]	B62	[illegible]

MBA PHYSICAL ADDRESS SPACE





MBA REGISTER OFFSETS

MBA PHYSICAL DASF ADDRESSES

MBA Number	Physical Base Address (Hex)	Valid Address Bits*			
		31	30	29	28
MBA 3	20038400	1	0	0	0
MBA 2	20038200	1	0	0	0
MBA 1	20038100	1	0	0	0
MBA 0	20038000	1	0	0	0

*Bits 27 through 0 are 0.

MBA INTERNAL REGISTER OFFSETS

Qb Access Internal Register	MBA to NPR Physical Base Address
00 - Configuration/Status (CSR)	Bit Read Only Returned, value as zero
04 - Control (CR)	Bit Read/Write
08 - Status (SR)	Bit Read Only
0C - Virtual Address (VAR)	Bit Read/Write
10 - Byte Count (BCR)	Bit Read/Write
14 - Diagnostic (DR)	Bit Read/Write
18 - Reserved for Customer	Bit Reserved for MBA private use by CPU
1C - Reserved/Addressable	Bit Read Only

MBA MAP REGISTER OFFSETS

Qb Access MAP Register	MBA to NPR Physical Base Address
MAP 0	Bit
MAP 1	Bit
--	The MBA MAP register address -- are accessed in increments of
--	Each of this is equal to 8
--	(Octal) MAP (next).
MAP 7F	Bit
MAP 80	Bit

MBA EXTERNAL (MASTER) REGISTER OFFSETS

Drive Register	Drive unit number							
	0	1	2	3	4	5	6	7
00	400	440	500	540	600	640	700	740
01	402	442	502	542	602	642	702	742
02	404	444	504	544	604	644	704	744
03	406	446	506	546	606	646	706	746
04	410	450	510	550	610	650	710	750
05	412	452	512	552	612	652	712	752
06	414	454	514	554	614	654	714	754
07	416	456	516	556	616	656	716	756
10	420	460	520	560	620	660	720	760
11	422	462	522	562	622	662	722	762
12	424	464	524	564	624	664	724	764
13	426	466	526	566	626	666	726	766
14	430	470	530	570	630	670	730	770
15	432	472	532	572	632	672	732	772
16	434	474	534	574	634	674	734	774
17	436	476	536	576	636	676	736	776
20	440	500	540	600	640	700	740	1000
21	442	502	542	602	642	702	742	1002
22	444	504	544	604	644	704	744	1004
23	446	506	546	606	646	706	746	1006
24	450	510	550	610	650	710	750	1010
25	452	512	552	612	652	712	752	1012
26	454	514	554	614	654	714	754	1014
27	456	516	556	616	656	716	756	1016
30	460	520	560	620	660	720	760	1020
31	462	522	562	622	662	722	762	1022
32	464	524	564	624	664	724	764	1024
33	466	526	566	626	666	726	766	1026
34	470	530	570	630	670	730	770	1030
35	472	532	572	632	672	732	772	1032
36	474	534	574	634	674	734	774	1034
37	476	536	576	636	676	736	776	1036

Note that offset value, when added to the MBA Control Base address, is equivalent to the Drive Unit and Selector Select on the MASSBUS Control Bus.

Signal Line	Description
Data Bus Section	
Data Bit Lines (D15:D00)	The parallel data path between the MBA and the drive. The MBA can deliver 16 bits in output operations and the drive may return 18 bits in input operations. The 18 bit format is used by the MBA for writing or reading data with parity checking.
Data Bit Par Lo (DPL)	DPL is the odd-parity bit for data bus data. It is supplied by the MBA on write to drive, input operations. It is supplied by the drive on read from drive (MBA) or write check (MBA) operations.
BDV	When a data transfer command is written to a drive's control register, the drive considers it by the data bus and asserts RDY. BDV is then asserted by the MBA to start transfer. BDV is placed by the drive at the end of each block or sector on the trailing edge of SEC. If BDV is not issued, the transfer on read runs to the next block or sector. If not, the operation terminates.
Recovered Error (ERR)	ERR is asserted by the drive when the recovered data in data transfer markers and is connected to the data bus is written. If the drive is not able to use ERR because of an error condition, the MBA times out and ERR is set in the MBA status register. ERR is asserted on the trailing edge of the last SEC of the operation.
End of Block	BDV is asserted by the drive for two microseconds after the first bit of each block or sector. BDV may be asserted earlier by error conditions where it is necessary to terminate the operation immediately.
Exception Occurred (EXC)	EXC is asserted by the drive if an abnormal condition occurs during DRV transfers. EXC is asserted at or before RDY and is negated on the trailing edge of SEC. Also, on the assertion of EXC, the EXC in the MBA status register.
	EXC disables those drives in the active drive unibus DMA translator from acting on any new signals by the MBA. Inc. as set on drive that not assert RDY during data transfers will MBA descriptors until the operation will been aborted or terminated. In the drive is no longer active.

Signal Line	Description
Data Strobe (DS(L))	DS(L) is generated by the active drive to control the data bus data clocking in the MBA. On read from drive (DEI or write check (WCK) operations the drive changes data on the assertion of DS(L). The MBA then clocks data in for acquisition of BUS.
Write Clock (WCK(L))	The MBA receives WCK and returns it to the drive as WCKI to allow drive (WCK) transaction. The MBA changes data on the negation of WCK. The drive then clocks data on the assertion of WCK(L).

Control Bus Section

Control Bus Data (CBI(15:0))	The parallel data path transfers control and status information between the MBA and the drive as CPU reads or writes to a drive's control/status logic.
Control Bus Parity (CPA)	CPA is the odd-parity bit for control bus data. It is asserted by the MBA on a CPU write to a drive's control or after a initialize. It is asserted by the drive on a CPU read of a drive's control or status register.
Control In Drive (CIDR)	CIDR is asserted by the MBA when the transfer is a CPU write (from the MBA to the drive). CIDR is negated by the MBA when the transfer is a CPU read (from the drive to the MBA).
Drive Select (DS(2:0))	The MBA asserts a three-bit code to select one of eight possible disk drives.
Register Offset (RS(4:0))	The MBA asserts a five-bit code to select a register within the drive.
Initiate (INI)	INI is asserted by the MBA to perform a control bus transfer with a drive. It is asserted after the RS, DS, and CIDR lines are settled (one clock before data lines are settled in the transfer in a CPU write).
Transfer (TRA)	TRA is asserted by the drive to transmit or receive data from the MBA. It is asserted when the drive clocks before to drive data to the selected register on a CPU write. It is asserted after drive is control data - as selected on the control bus (all lines) from the selected register on a CPU read. It is negated when the acquisition of data has been received from the MBA.

MASSBUS SIGNAL DESCRIPTION (CONT)

Signal Line	Description
Attention (ATTN)	ATTN is asserted by any inactive drive as a result of a change of status such as the completion of a motion command or an error. ATTN is also asserted by a previously active drive when it powers up or is plugged into a powered on drive. ATTN is a request for service to processor interrupt. The attention active (ATA) status bit for each drive is set (by the attention summary register) to determine the request originator.
Initialize (INIT)	INIT is asserted by the MBA to initiate a system reset of all drives. It is asserted as a system power-up, or by setting a one in the INIT bit of the MBA control register. It must not be written while one is set since the active drive will abort execution of the current command and perform drive clear sequence. INIT is used only for the MBA that is control. It asserts the internal clear signal for that MBA.
Poll	POLL is asserted by the MBA to indicate that a program level interrupt exists or that the MBA is in the maintenance mode. While POLL is asserted, all drives ignore execution of the INTR or DVA signals.

I/O cable	pins		polarity	signal name
cab a A	A	1	-	MBS DB0
	B	2	+	
	C	3	-	MBS DB1
	D	4	+	
	E	5	-	MBS DB2
	F	6	+	
	H	7	-	MBS DB3
	J	8	+	
	K	9	-	MBS DB4
	L	10	+	
	M	11	-	MBS DB5
	N	12	+	
	P	13	-	MBS DB6
	R	14	+	
	S	15	-	MBS DB7
	T	16	+	
	U	17	-	MBS DB8
	V	18	+	
	W	19	-	MBS DB9
	X	20	+	MBS DB10
	Y	21	-	MBS DB11
	Z	22	+	MBS DB12
	a	23	-	MBS DB13
	b	24	+	MBS DB14
	c	25	-	MBS DB15
	d	26	+	MBS DB16
	e	27	-	MBS DB17
	f	28	+	MBS DB18
	h	29	-	MBS DB19
	j	30	+	MBS DB20
	k	31	-	MBS DB21
	l	32	+	MBS DB22
	m	33	-	MBS DB23
	n	34	+	MBS MBE
	p	35	-	
	s	36	+	MBS SDN
	t	37		
	u	38		
	JL	40		Ground
	JZ	41	GND	Ground

*alternate pin assignments

NOTE

MASSBUS cab ax are terminated as shown.

MASSBUS CABLE PIN ASSIGNMENTS (CONT)

I/O Cable	Pair		Polarity	Signal Name
Cable B	A	1	-	BASS 00
		2	+	
	B	3	+	BASS 01
		4	-	
	C	5	-	BASS 02
		6	+	
	D	7	+	BASS 03
		8	-	
	E	9	-	BASS 04
		10	+	
	F	11	+	BASS 05
		12	-	
	H	13		BASS 06
		14		
	J	15	-	BASS 07
		16	+	
	K	17	+	BASS 08
		18	-	
	L	19		BASS 09
		20		
	M	21	-	BASS 10
		22	+	
	N	23	+	BASS 11
		24	-	
	P	25		BASS 12
		26		
	R	27	+	BASS 13
		28	-	
	S	29	-	BASS 14
		30	+	
	T	31	+	BASS 15
		32	-	
	U	33	-	BASS 16
		34	+	
	V	35	+	BASS 17
		36	-	
	W	37	-	BASS 18
		38	+	
	X	39	+	BASS 19
		40	-	
	Y	41	-	BASS 20
		42	+	
	Z	43	+	BASS 21
		44	-	
	AA	45	-	BASS 22
		46	+	
	BB	47	+	BASS 23
		48	-	
	CC	49	-	BASS 24
		50	+	
	DD	51	+	BASS 25
		52	-	
	EE	53	-	BASS 26
		54	+	
	FF	55	+	BASS 27
		56	-	
	HH	57	-	BASS 28
		58	+	
	JJ	59	+	BASS 29
		60	-	
	KK	61	-	BASS 30
		62	+	
	LL	63	+	BASS 31
		64	-	
	MM	65		SPARE
		66		SPARE
	NN	67		SPARE
		68	GND	Ground

Additional pin assignments

NOTE

Massbus cables are installed as labeled.

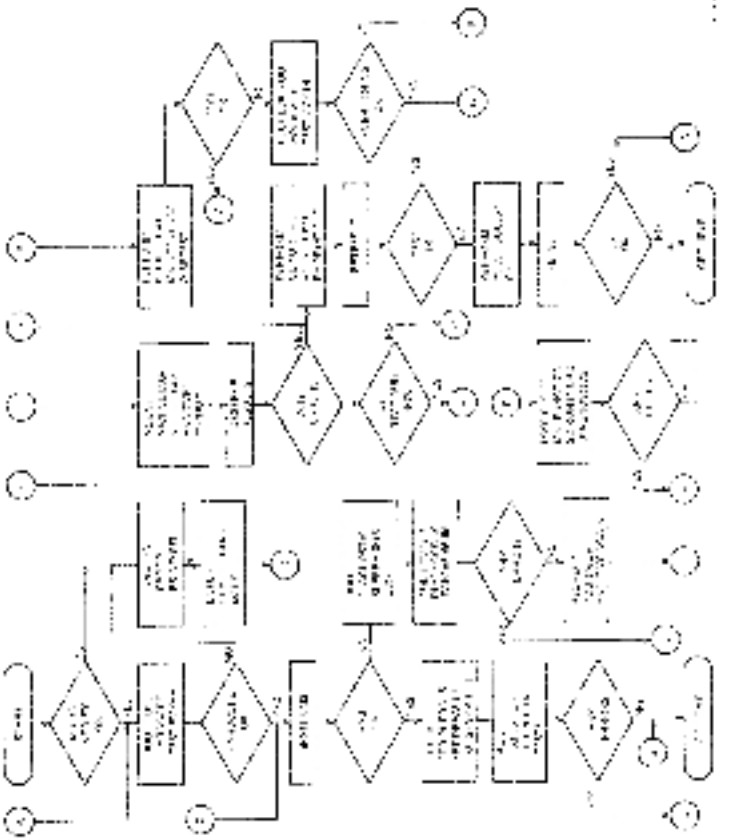
I/O Cable	Pin #	Polarity	Signal Name
Cable A	A 1		BA00 L12
	B 2		
	C 3	+	BA01 L11
	D 4	-	
	E 5	-	BA02 L10
	F 6	+	
	H 7	+	BA03 L09
	J 8	-	
	K 9		BA04 L08
	L 10	+	
	M 11	-	BA05 L07
	N 12	-	
	P 13		BA06 DPA
	R 14	-	
	S 15	-	BA07 L06
	T 16	-	
	U 17	-	BA08 L05
	V 18	-	
	W 19	+	BA09 L04
	X 20	-	
	Y 21	-	BA10 L03
	Z 22		
	AA 23	+	BA11 L02
	BB 24	-	
	CC 25		BA12 L01
	DD 26	+	
	EE 27	-	BA13 L00
	FF 28	-	
	HH 29	-	BA14 P01
	JJ 30	-	
	KK 31	-	BA15 P02
	LL 32	-	
	MM 33		BA16 P03
	NN 34	+	
	PP 35	-	BA17 DPP
	RR 36		
	SS 37	-	BA18 P04
	TT 38		
	UU 39	Ground	BA19 GND
	VV 40	Ground	BA20 GND

*Alternate pin assignments

NOTE
Multiple cables are installed as one set.

CHAPTER 9
TROUBLESHOOTING AIDS

SYSTEM TROUBLESHOOTING FLOW



```
First sample - The following sequence of console commands causes
the UCT to execute a single NPR read [DATO] from/to) first memory.

1.  >>>D/I 17 1                     Initialize CPU
    >>>D/M/P 832184 1               Program UBA: BDP 1
    >>>D/A/P 832180 0A20B4C8        Put in UBA MAP address 8 for
                                    validity, BDP 1, PFN 1B87
    >>>D/L/P 1B87 1234ABCD          Load data in memory 1B87
    >>>D/M/P 772467 6               Set UCT Bus address to 6
    >>>D/M/P 772461 1               Set UCT GO bit to 1.

2.  Note the results:

    >>>/a EEE467                    Examine bus/data registers
                                    (should return 5678 AB CDEF)

3.  Investigate the bus address register:

    >>>D/M/P 772470 A               Set U-C bus address to A
    >>>D/A/P 2FF470 ?               Get UCT BAR (bit 0:1)

4.  Look at the results:

    >>>D/M 772467                   Examine UCT data register
                                    (should = 1234 AB CDEF)

Second sample - This sequence causes the IPEC to execute a single
NPR read [DATO] from/to) ZERO memory.

1.  >>>D/I 17 1                     Initialize CPU
    >>>D/M/P 832184 1               Program BDP1 UBA
    >>>D/A/P 83A180 0A20B4C8        Put in UBA MAP address 8 for
                                    validity, BDP 1, PFN 1B87
    >>>D/L/P 1B87 12345678          Load data in memory 1B87
    >>>D/M/P 772467 8               Set UCT bus address to 8
    >>>D/M/P 772461 1               Set UCT GO bit

2.  Look for the results:

    >>>E/W 2BF467                   Examine UCT data register
                                    (should return 5678 AB CDEF)

3.  Optional: The bus address register:

    >>>D/M/P 772470 2               Set UCT bus address to 2
    >>>D/M/P 772461 1               Set UCT GO bit

4.  Look for the results:

    >>>E/W 2BF467                   Examine UCT data register
                                    (should return 1234 AB CDEF)
```

```
The bit having substitutions can be made in diagnostic programs to test
other data path and MAP address combinations.

MAP address:  UPI       UBP

              E18000    E32780      MAP physical page address
              E18004    E32784      (MAP locations are contiguous
                 .         .         in increments of four)
                 .         .
              E187F8    E33FFC
              E187FC    E33FFF

Map Paths     DUP     = 8823A780    (direct data path)
              DTE 1   = 8628C780
              DTE 2   = 8648C780
              DTE 3   = 8658C780

CTR Error:    USI       BIT

              T3A2780   2120228     Bits low (BR not rec'd)
              F5A1E4    F57827      USI 1 for DTE
              F1A7A8    F32C78      CTR 1 for DTE 2
              F32C87    F7FAA2      CTR 7 for DTE 3

Register
Address:      UET       IPEC

              E7F467    F81481      Task address register
              E7F463    F81783      Map fail location
              E7F464    E87FA3      MSCP
              E7F465    E87FA7      Error log, CTR fail
                                    fail location

Further operations can be performed using another one of
the following control registers bits:

    CSR7/     = DLL IDC bit
    CSR(15)   = CC (CI) free buffer bit
    CSR(11)   = <UC1F536> DLL DC/DAPA bits

          /CI>  <DR>   Function

           0     0     DA01
           0     1     DA11
                 2     DA10
                 3     DA00

To do a DA20 or DA28 you will either program, load the bit into the data
register if it was set for memory address bits and send it back to the
processor register as above.
```

```
Filex can be used to copy files from CP/M to disk or from disk to
CP/M.  The user is used to build new files or make existing files
have new contents.  It provides user input prompts and
help screens to guide the user through the process.

Disk to Disk.  To copy from the default disk drive to a CP/M
diskette, do the following (FILEX name only [OK] and some not yet
completed modes.)

     A B:=A:*.*[OK]
     A:FILEX[OK]
     A:FILEX>DIR
     * A:=B:*.*[OK]
     * DIR A:=B:*.*[OK]
     * XXX FLX
     A:FILEX >A:*.*                [File type A;copy = opt out.]
     A:FILEX >A:*.COM/SE           [Same type A;copy = opt out.]
     A:FILEX >A:*.ASM    [Device filename/ext/DB
          [to device]       [from any dat

               XX Control Codes

               ^P - Dump mode (ASC, HEX, DMP, DEV, OFF)
               ^E   Enter operating mode
               ^R   Reverted Binary (ASC, HEX, DEV, LST)
               ^A - Formatted ASCII (ON, OFF, ATTRIBUTES)
               ^T - Continuous [Tile To File; File Dump Line
                    File to type ASC always continuous
               ^S   Still Format (Byte)
               ^U   Still Format (Line)

     Exit:           return execution to caller(CCP)
     ^C

page to disk (BASIC) - To move from a file directory to the default
disk, do the following:

     A B:=A:*.BAS[OK]
     A:FILEX>DIR A:*.BAS[OK]
     A:FILEX>XXX.T
     * B:=A:*.BAS[OK]
     * DIR B:=A:*.BAS [FORMAT]
     * XXX FLX
     A:FILEX>A:*.BAS = A:*.BAS[OK]
          [to device]     [from device]

     A:FILEX      return execution to caller(CCP)
     ^C
```

THE WRITEBOOT UTILITY

The WRITEBOOT utility may be used to write a new boot block on any bootable disk. A bootable operation is usually performed from the boot block which is block 0 of the system disk. If block 0 is bad, the system can only be booted from the disk. When the system is running again, the bad block can be revitalized with WRITEBOOT.

NOTE

You operator must have LOG_IO privilege to run the WRITEBOOT facility.

To call WRITEBOOT, type the following command:

```
$ RUN SYS$SYSTEM:WRITEBOOT
```

The program then asks three questions.

1. Target system device:

   Enter the name of the device that will have its boot block written and that must also be bootable if you DISABLE the automatic bad block replacement. Specify the device's system device name (Dnn:) or its physical device name.

   ```
   ddcu:    device type
     n:     controller designation
     u:     unit number
   ```

   An example: 'DUA0:' is RA82 unit 0 on a KDB50 controller A.

2. Value for boot file index:

   To override the boot block, press N<RET> (default = Y) .

3. Value: Load address of primary bootstrap in hex:

   To override the boot block, press RETURN (default is 200).

WRITEBOOT then writes the boot file specified on the first line of the address specified on the third line.

```
WRITEBOOT [/y] [/n] [d:][path]filename.ext [/b:sss] [/t:tt] [/s:ss] [/hd:hh]
[/cylinders:ccc] [/p:pp] [/d:dd] [/sectors:sss] [/multitrack:mm]

    Assigns system boot sector to file specified.

        [drive:] = Logical or physical drive to write boot sector to.

    Enter ^BREAK^ to abort file input.  Enter ^Z to end.

        [path:] = Optional path to file if not in current directory.

    Enter load address of primary bootstrap in hex:

        [/b:sss] is 7C00 if option is omitted.

Level 1 utilities and 2 compatible are the only bootable programs
from DOS/65.  Most bootable programs are initially read at 0400 and
written to a fixed location each using block descriptors:

Filename.ext    Type    Load Address    Description

SYSTEM.SYS      .       200             Resident system
IO.SYS          .       700             IO and cache
SYSTEM.SYS      .       1980            Alternate Operation
Boot16.bin      .       0700            Boot16 option
```

MACHINE CHECKS

The CHIERR bit of ICR, if set, indicates multiple Dstream cache parity errors. These indicate any disk error or operator sequence error. Machine checks also the system error log is examined, although there is routine using with the system.

CHIERR is only valid when the system exits at the visible clock value after one of the following:

- Memory error
- TB parity error
- Cache parity error
- Control store parity error

CHIERR overwrites the previous MCHECK register of the 11/780 system. The MCHECK register has a type identical that follows the logic flows. Here are full explanations memories below the macro code for an EXEC from this register. The formed machine checks are included here for illustration.

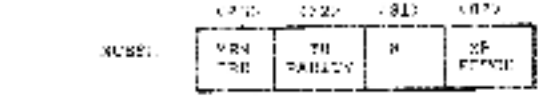
TRANSLATION BUFFER OR BUS ERROR

This machine check was caused by mapping a translation page with a corrupt system service and then accessing that page in user mode.

```
Exception PC     200734 C

Error PSL        3 01C00000    (interrupt priority level = 0
                               Previous mode = User
                               Current mode = User

Summary code     C0070070      Translation buffer or bus error

VA (opt. ref     200730F0
PC at error      C00200EF
MBZ              00070000
SBZ              00070000      Cpu mode = User
                               Virtual
                               Read

BTB              00020020
TBUPD            00070020
CADR             00070020
BUS              00070020      Memory error

SBIMR            80200000      Control processor
                               Bus error
```

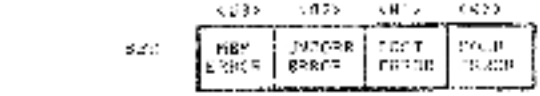
In the parity/ECC machine check logout, supplied to the VAX 11/730, an individual address is pushed only for reads. This is because the physical address latch is continuously within the arrays and there is no direct means of obtaining the physical address.

The MCHK subtype may be used to determine what type of exception condition occurred. The bits are defined as follows:



If bit <0> is set in the mask byte, an error occurred during access for read data. The sub register logout may then be read to determine the type of memory error.

The sub register will indicate what type of error occurred. Bit <0> indicates a memory write bit <02> indicates an uncorrectable error. Correctable errors will interrupt (on processor level OCUB bit will log to error while uncorrectable causes machine checks.



If the uncorrectable error bit is set bit <1> will be set. If set, the address latched is for a reference to a nonexistent location. If the VS_Utility/Error Diag is the system, it is a read. Some type of memory error caused the machine check.

The MCHKSUBTYP is used to determine what the error was. If machine check was not caused by a memory error or PE parity error, a cache parity error is the only other possibility. The CDAL contains the data or bad parity information.

Several other machine checks may occur on the VAX-11/730.

4. Memory parameter = 03	This error occurs only if the initialization console RAM is not replaced at IPCR time. The machine only writes it there is no CS parity error and the MBZ field is 0. The most significant byte is used as the IPR 3A number.
5. Memory parameter = 05	This is a microdetected parity error that occurs in the M console logic. It clears the error and continues. This only happens if there is no CS parity error.

The following paragraphs illustrate the last possible occurrence of machine checks, the control store parity error, when the VAX-11/750 machine check logout does not have the right to access certain errors occurred passed on the stack. This information cannot be saved because the VAX-11/780 does not have a CSE error register like the VAX-11/780. It is possible, however, to recognize interrupts CS parity error machine checks by using the SBR as follows.

```
MOVL    SBR,R0          ; start the application
MOVL    #^X80000000,R1
ADDL    R0,R1
CMPL    @#^X20080000,R1 ; see what type of error occurred at CS address 8000
BEQL    NEXT            ; return to system if not
RSB                     ; continue running application
```

When the CS parity error occurs, the following routine is performed at the console.

```
>>>CRAM 1F0:0FFF:0FFF:0FFF  ;Set BPT
>>>
>>>E/P 1F0                ; When V-AXP command will permit you to see if
                          ; there is where the microinstruction has been
>>>
>>>DEP 2004               ;Set BPT
        2003              ; Jump to the microinstruction should the CS
        2002              ; parity error
        2001
         .
         .
         .
        2000              ; Set microinstruction again
>>>
```

Review this routine to be certain that the proper values in the hard files. Changing the VAX80 machine will usually prevent this type of problem. The following is a description of the CS parity error machine check.

CONTROL STORE PARITY ERROR

```
This machine check was caused by detecting a single bit error in
a microinstruction stored in the WCS.  A D and B floating-point
arithmetic operation were performed and when the microinstruction
with bad parity was executed, the error machine check occurred.

Exception PC      80004e00

Error PSL         03c00384      1 bit
                                interrupt priority level = 0
                                Pascal is mode = User
                                current mode = User

Summary code      c0820007      Control store parity error

CA last TB0       80010400
PC at error       80004e00
MDR               ffffffff
CPD               00870000      CPD also    Other
                                virtual
                                Good

P1CS              80c80020
IPDER             80c80020
CASE              00870001      Cache bit

P1CS              80c80020
WCSER             80c80020      Control Sequencer
                                Address register = 0000
```

On this type of exception check, all other information could not be put. The actual problem was that a control store parity error occurred and there is no way to determine what happened.

All that can be done is the machine check error summary register (80c80020) bold that byte 2 containing the bit(s) 'Par'. Then also the opcode of the instruction that caused the machine check. Use either the opcode in the trace of the PC/PSL in memory or look at the location in memory where the instruction that was executed caused parity error.

GENERAL GUIDELINES

Software Performance Reports (SPRs) provide feedback to DIGITAL on problems with software and provide help to customers with questions they have during

The following guidelines cover the information that should be provided with a new SPR. Depending on the problem, this information will vary in quality and content. Remember that the more pertinent the information included, the easier it will be to resolve the problem.

1. Scenario

Describe what you were doing when the problem occurred. Provide as much detail as possible. Explain what commands you entered, what displays you saw, what results you got, and what you expected to see. Specify the hardware configuration and the software version. Include relevant excerpts from batch logs or terminal sessions that show exactly how the problem was encountered. Supplying only the output produced by the problem is not enough. The problem may be caused by an interaction between various system components, such as privileges, logical names, foreign commands, DCL symbols, or command procedures. Some or all of the steps you entered to reproduce the problem may be required for DIGITAL to duplicate the problem.

Sometimes a command procedure is the easiest way to document the steps to reproduce a problem. The following command procedure demonstrates how to use this method:

```
$SHOW LOGICAL/FULL
$SHOW SYMBOL/ALL/GLOBAL

$DEFINE SYS$OUTPUT MYLOGFILE.LOG
$ SET VERIFY
$SHOW /ALL
$SHOW /SPECIAL
```

2. Don't Include Too Much

The more clutter you remove, the easier it is for DIGITAL engineers to find the problem. For example, if the execution of a very large program causes a problem, you may want to isolate the program so that it includes only the code that causes the problem. This could be a small program that demonstrates the problem. This solution has two benefits: first, the more you trap logic errors, and second, the software engineer looking into the problem does not have to understand unnecessary material.

1. Machine Readable Files

If possible, supply any software needed to reproduce the problem. This may include source programs, image files, sample data, or scripts processors. If source programs are submitted, also include any libraries or related files referenced. These files must be provided in machine readable format. The sample media is 1.44Mb diskette and the user wants to include with the SPR.

If the problem involves a system error, then also the system dump.

The data should be written with an ASCII editor like vi or NOT wordprocessor. But beware, the following commands will copy the system dump I is to an 8Mb diskette.

```
$ dir *.dat;  DUMP1
$ DUMP2 *.dat;  DUMP2
$ DUMP2 /output=disk:[sales]sales.dmp *.dat;
$ DUMP1 disk:
```

To copy files to the remote system, use the following commands:

```
$ run sys$system:stdnet
SYSNET> connect
[At this time, receive the terminal emulation and
please a suitable value in the range 1 thru 10.]

$ init test;  $ dpfapp
$ DUMP1 test;  $ DUMP2
$ CREATE/FDL=SYS$SYSTEM:SYSGEN.FDL [.DUMP]
$ DUMP2 MYSYS.DAT;*/SAVE_BOOT [SYS]/DUMP
$ DISMOUNT test;
```

When reading variable data in previous format, the user should initialize the record parameters that were used to write the data and the parameters used to read it. Other record format problems will result if this is not done. For example, using RMS without the FDL option will create a dump file that is completely unusable, because RMS will initialize all bytes of zero in a file.

All ASCII variable length records will be padded with one will be used to fix the number.

c. System Environment

Every computer site has a different type of workload. Some products only appear under certain conditions. For example, some disk drive different classes of users a through each possibilities. These disks may have problems that other a has do not. This information can be very important in isolating the problem, especially for system hangs or crashes.

The user should describe any special software workload being used. Any unique hardware devices or user written drivers should also be documented.

Software version numbers should be included. For example, if there is a problem with accessing local symbols during a DEBUG session, the version numbers of DEBUG and all compilers/assemblers used should be noted first.

If any patches other than those maintenance updates are being used, they should be mentioned in the SPR.

d. User Analysis (Optional)

Optionally, the user may include an analysis of the problem. Any useful miscellaneous information should be included such as: "Although cycle happening, the problem can't now be reproduced" or the version only. This portion does not count

e. Problem Specific Clarification or Instruction

There are different types of problems with regard different kinds of information. This portion could also be used for different types of problems.

Problem	Information to include
System Diagnostic Trace	A machine readable copy of the system diag. file MUST be included. Without this the SDE will by-and-large NOT be able to locate or usually even run trouble through. Information to resolve the problem.
	A copy of the error log at the time of the error should also be included because many system problems are highlighted by hardware errors.
Machine check	In a machine check, include a copy of the error log.
	A machine readable copy of the system dump file should also be included.
System hang	When a system appears "hung" the response on any terminal, the system should be actually crashed and the system dump file retrieved with the SDE.
	When the system is shut down in this way, the console listing is very important and should be obtained with the LPR.
	On the VAX-11/780 console terminal, enter:
	``` ^P HALT ^C(BOOT) ```
	On the VAX 11/750 console terminal, enter:
	``` ^P >>> >>L ? > > > >>>D/G 0 FFFFFFFF >>>B/2 1F0000 ? ```

	data will remain the system or any check to see if encryption by VMS developers or a forced erase. A description of the circuit paying here and should also be the need.
Execution	If the user encounters a problem with system, include if applicable values of the system parameters. These can be obtained by invoking SYSGEN and entering path and SHOW/ALL and SHOW/SPECIAL commands. A machine readable copy of the source program showing the problem plus libraries, logical files, and I/O if the should also be available, if possible. Include a copy of the error log at the time of the problem.**
Devices	For any suspected device or device driver error, include a copy of the error log at the time of the problem.**
Files	If the problem appears to be with a file, a description [directory/FULL on each file and its directory should be included. If pertinent, include a machine readable copy of the file (ies).
Reproducibility	For a problem that is intermittent or that cannot be reproduced, include a copy of the error log at the time of the problem.**

Public	Information to Public
Command language interpreters	When writing an SPR on a command language interpreter, it is important to show all symbols (such as GLOBAL and LOCAL) and logical names (such as LOGICAL/ALL) defined at the system
Job controller	If the job controller aborts, it will print a message on the console and write a file named SYS$LOGIN:JOBABORT.DMP. The user should include the console printout and a patch as readably copy of the SYS$SYSTEM:*.EXE files.
Libraries	If the user encounters a problem with the LIBRARIAN, include the following information:
	1. A readable copy of the library (text)
	2. Machine readable copies of all input files to the library
	3. Information (DIFFERENCES/BRIEF) on the library file
	4. Information (LIBRARY/LIST/FULL) on the library contents
	If the problem can be duplicated with (, include the scenario and any commands in use).
Linker	If the user encounters a problem with the LINKER, include machine readable copies of the object files and libraries used in the link along with a full log (/DEBUG/MAP) file.
Others	For a DECnet problem, supply configuration of the systems involved in the problem. This information must include the version numbers of the operating systems and DECnet, the hardware on each system, and the path taken by the packet between the non-VMS systems, if applicable.

Problem	Information to Provide
Questions	If you need support a problem with the Pershing driver, provide the following information: 1. A list of terminal characteristics (i.e. TERM.PRM) 2. The type of terminal 3. The type of cable (if any) 4. Any special terminal configuration If the problem involves create file access, it is often useful for the individual to know if the use of a different account or with the source and destination nodes reversed.
Compiler/Assembler	If the user encounters a problem with the compiler or a compiler, the use the source program that caused the problem. It is very important to the DIA staff to also have the data (if any) entered by the source program, also. It is especially important to list the types of the symbols when creating SPRs on compilers. Include the version number of the compiler and the version number of the operating system.

*The .DAT data file (SYSSYSTEM:SYSDATA.DAT) and the Executive output from the DIA utility, should be included. Terminal output usually does not contain all the information needed to solve the problem.

**If a .DAT file is (SYSSYSTEM:SYSDATA.DAT), not the Executive output from the DIA utility, should be included. Executive output often does not contain all the information needed to solve the problem.

The following are priority explanations should be used as a guideline for determining how priority is set for SPRs:

1. Mass promotion work cannot be run. Functionality that are not basic of a major new of system system will not be, nonessential applicable cannot be used as intended.

2. Mass production work cannot be run. Certain specifications are not used as anticipated, installation has inefficient process capacity.

3. A production work can be run with some impact on users, application name interaction required, easy procedures, publications received but inital action but process capacity.

4. All production work can be run with no application impact on users. Problem can be easily bypassed, steps bypass procedure exists.

5. No system modifications needed to return to normal production. suggestion, compatibility, documentation error.

GENERAL NOTES

VAX-11/750 TROUBLESHOOTING TIPS

1. When troubleshooting, place the console power-on action switch in the HALT position. If you do not, the CPU will try to reboot if the CPU halts for any reason.

2. If the console prompt does not appear when power is [illegible] [illegible] the [illegible] [illegible] [illegible] [illegible] should appear.

   [illegible] loop waiting for [illegible].

   [illegible] by the [illegible] supply. The memory controller [illegible] all memory locations have been cleared. [illegible] [illegible] is very useful if the CPU is hung [illegible] loop.

3. [illegible] try to examine and deposit [illegible] [illegible] console prompt [illegible] [illegible] examine and deposit [illegible] [illegible] present.

   [illegible] memory [illegible] from the console prompt, [illegible] and see if it is [illegible] from the [illegible] prompt. This [illegible] whether the CMI and memory are working or not. [illegible] [illegible] cannot be [illegible] from the [illegible] prompt.

   The [illegible] [illegible] always present on the [illegible] [illegible] it should be possible to [illegible] location [illegible] the internal programmable [illegible] [illegible] console default [illegible] [illegible] physical [illegible] [illegible] [illegible] [illegible] [illegible] address [illegible] [illegible] binary address. (See Chapter [illegible] [illegible])

4. [illegible] parity [illegible] the [illegible] by typing [illegible] from the [illegible] prompt. [illegible] [illegible] [illegible] parity [illegible] [illegible] starting with location [illegible] [illegible] with [illegible] parity error at [illegible] address [illegible] [illegible] [illegible] [illegible] problem in the [illegible] [illegible].

s. Always run ... DFU ... Microdiagnostics before the interdiagnostics (see Chapter 6, DEC Installation Manual).

t. Note: It is that the system has the 'check register' bit set and it is 'check' bits are available. This can be checked by examine of the S.P register (SSP:/I SR <RET>). The bit 15(14) of the SSR register are set to [clear]. (The register layouts are current as of the date of this document).

```
Bit 15(14) = Type check (the VAX 11/730   3)
Bit 4(3)(2) = Microcode control of interrupt V = 0;
Bit 0(1)(0) = parity check on microcode.   3)
```

Refer to Power Bulletin, 1-6, 171, and 287 for latest ECO to date.

u. The current revisions for microdiagnostics are:

```
EC/EA  -   Rev.  A 2     Floating/Fixed Partner (MTCMDC)
EC/EB  -   Rev.  B.1     Virus ECO
EC/EC      Rev.  C.2     Virus ECO
```

v. EC/AX does not test the first memory array module. For this, EVA-xx Alpha Level 2 tested by the MTC microdiagnostics.

w. Further revisions of EC/AX may not report a single bit errors, or find some dual addressing errors. Check periodically to find out when the updated version will be available.

x. Cache parity errors in the VAX-11 /730 cause a machine check to occur at location 1C8E (4). This can be treated as a hard error. In the VAX-11/780, the VAX 11/780 Cache will attempt to obtain good data from memory on a cache parity error.

DW750 INSTALLATION

The DW750 is typically installed in slot 7 of the reserved slots for the CPU. In order to make room for the DW750 and to place the BR level priority in the new grant chain.

CAUTION

Refer to the DW750 installation procedure that specifies use of the V7.0SDAY and DIAG to prevent access from public diagnostics. Also see figures and tables in Chapter 3 about the switches, the quick chain and jumpers by jumpers and NPR installation summaries.

If one or more DW750 KA650 adapters are already in place, they must be moved into one slot and one CPU slot level to allow the following steps to.

Example 1: NPR Failure After Installation

Slot	Device	Base Address	CPU NPR Level
7	NPR 3	[illegible]	NPR 3
8	NPR 3	[illegible]	NPR 3
9	-	[illegible]	NPR

Example 2: More Slot DW750 Installation

Slot	Device	Base Address	CPU NPR Level
7	DUP	[illegible]	NPR 3
8	NPR 0	[illegible]	NPR 3
9	NPR 1	[illegible]	NPR 3

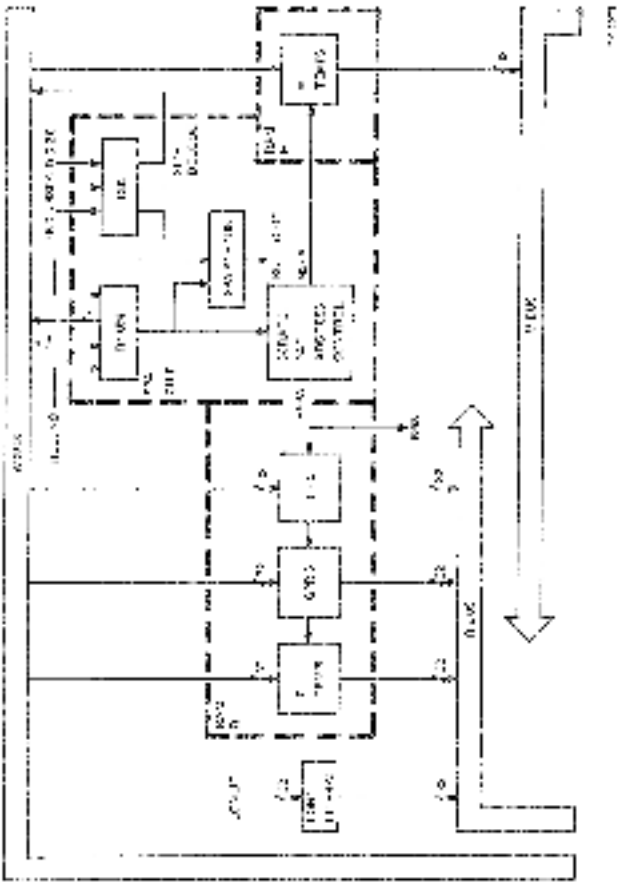
UNIBUS EXERCISER (UBE) ON THE SECOND UNIBUS

The DW750 UBA module may be plugged into any CPU slot in the second UBA adapter as in user module and DIAGNOSTIC

UBA Base CSR Address: 172520 (17760320 octal)

UBE Vector: 510

CHAPTER 10
CHARTS AND MACROS



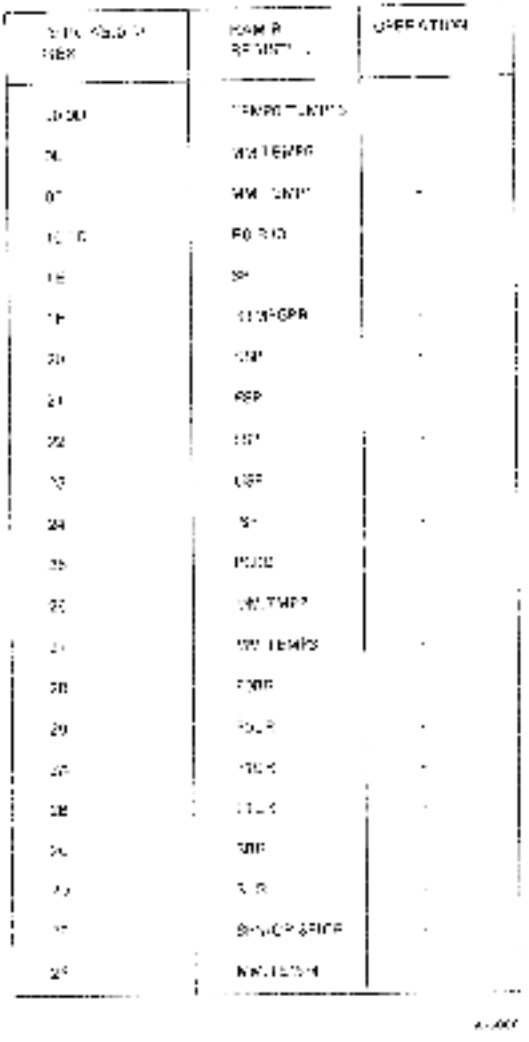
REG NO	REG	ASSIGNMENT/PURPOSE
0	00	DUAL PORT TEMP 0
1	01	DUAL PORT TEMP 1
2	02	DUAL PORT TEMP 2
3	03	DUAL PORT TEMP 3
4	04	DUAL PORT TEMP 4
5	05	DUAL PORT TEMP 5
6	06	DUAL PORT TEMP 6
7	07	DUAL PORT TEMP 7
8	08	RSTACK TEMP 8
9	09	RSTACK TEMP 9
10	0A	RSTACK TEMP 10
11	0B	RSTACK TEMP 11
12	0C	RSTACK TEMP 12
13	0D	RSTACK TEMP 13
14	0E	MEMORY MANAGEMENT TEMP 1
15	0F	MEMORY MANAGEMENT TEMP 2

REG NO	REG	ASSIGNMENT/PURPOSE
0	10	GPR 0
1	11	GPR 1
2	12	GPR 2
3	13	GPR 3
4	14	GPR 4
5	15	GPR 5
6	16	GPR 6
7	17	GPR 7
8	18	GPR 8
9	19	GPR 9
10	1A	GPR 10
11	1B	GPR 11
12	1C	GPR 12
13	1D	GPR 13
14	1E	STACK POINTER
15	1F	REG 15 USED TEMPORARY

PRIVILEGED IPR AND MTEMP FUNCTIONS

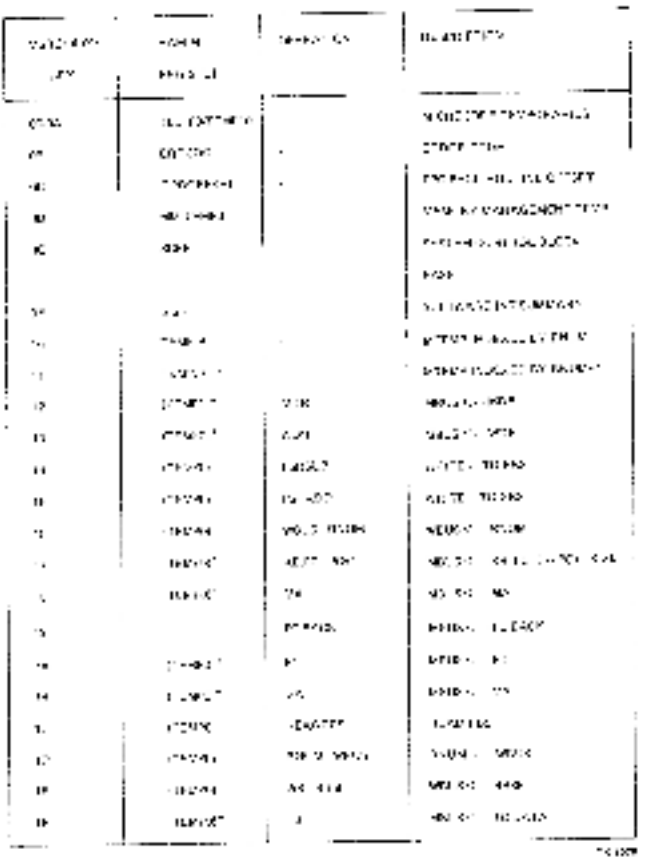
REGNO	MTPR	ASSIGNMENT/PURPOSE
0	00	KERNEL STACK POINTER
1	01	EXECUTIVE STACK POINTER
2	02	SUPERVISOR STACK POINTER
3	03	USER STACK POINTER
4	04	INTERRUPT STACK POINTER
5	05	PROCESS CONTROL BLOCK BASE
6	06	MEMORY MANAGEMENT TEMP 2
7	07	MEMORY MANAGEMENT TEMP 3
8	08	P0 BASE REGISTER
9	09	P0 LENGTH REGISTER
A	0A	P1 BASE REGISTER
B	0B	P1 LENGTH REGISTER
C	0C	SYSTEM BASE REGISTER
D	0D	SYSTEM LENGTH REGISTER
E	0E	RESERVED TO DIGITAL
F	0F	MEMORY MANAGEMENT TEMP 4

REGNO	MFPR	ASSIGNMENT/PURPOSE
0	00	SCAN PORT TEMP 0
1	01	SCAN PORT TEMP 1
2	02	SCAN PORT TEMP 2
3	03	SCAN PORT TEMP 3
4	04	SCAN PORT TEMP 4
5	05	SCAN PORT TEMP 5
6	06	SCAN PORT TEMP 6
7	07	SCAN PORT TEMP 7
8	08	RESERVED TO DIGITAL
9	09	RESERVED TO DIGITAL
A	0A	RESERVED TO DIGITAL
B	0B	INSTRUCTION, CHMK FAULTS & ARITH TRAP
C	0C	TRANSLATION BUFFER OFFSET
D	0D	MEMORY MANAGEMENT TEMP 0
E	0E	SYSTEM CONTROL BLOCK BASE
F	0F	SOFTWARE INTERRUPT REQUEST

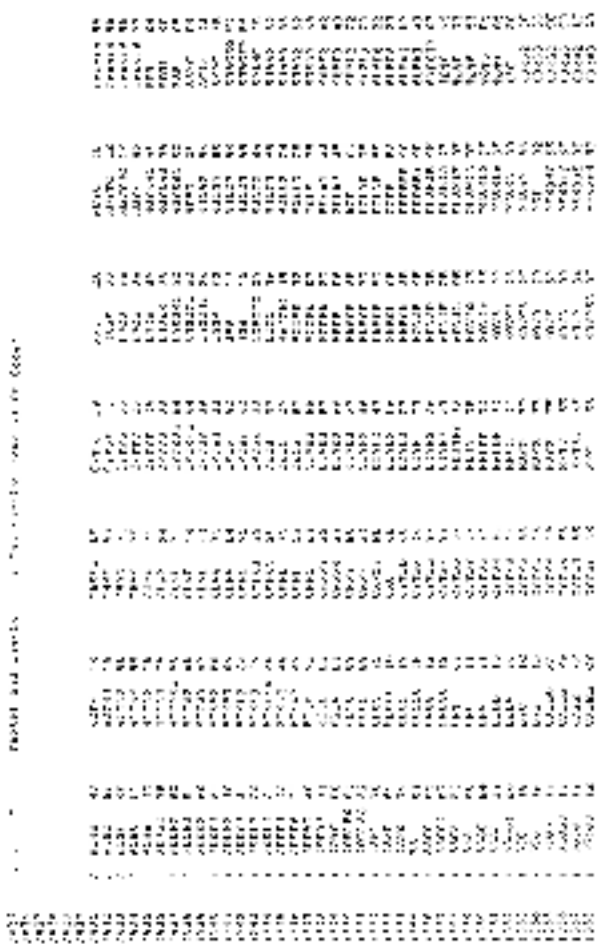


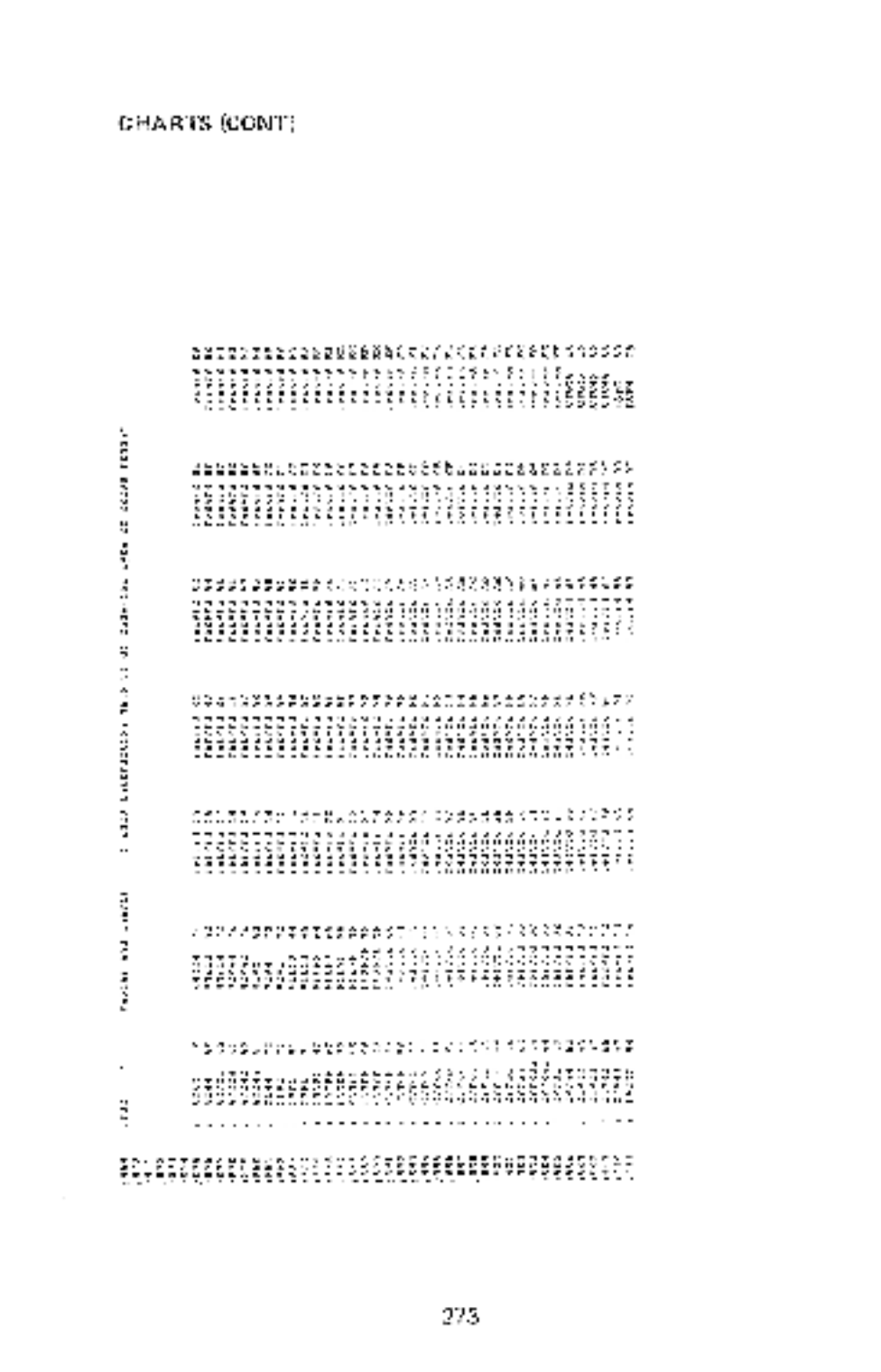
RST 0 ASSIGNMENTS (CONT)

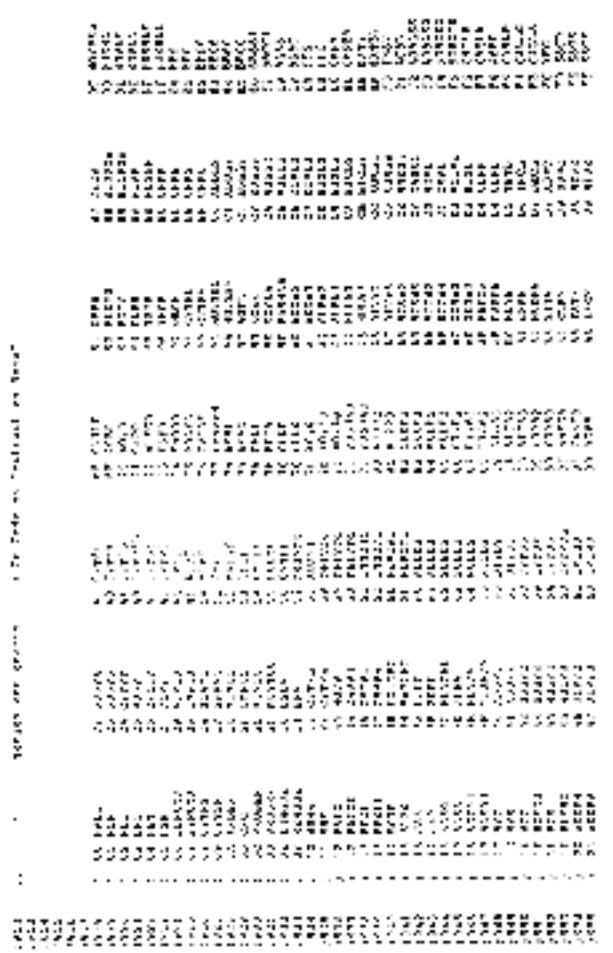
HEX	REGISTER	
[illegible]	[illegible]	[illegible]
91	TEMP 9	
92	D,E	*
93	H,L	
94	<TEMP0>	
95	<TEMP1>	DATA
96	<TEMP2>	TEMP
97	<TEMP3>	ADDRESS
98	TEMP,PSW	
99	[illegible]	*
9A	[illegible]	
9B	[illegible]	
9C	[illegible]	
9D	[illegible]	*
9E	[illegible]	
9F	[illegible]	*

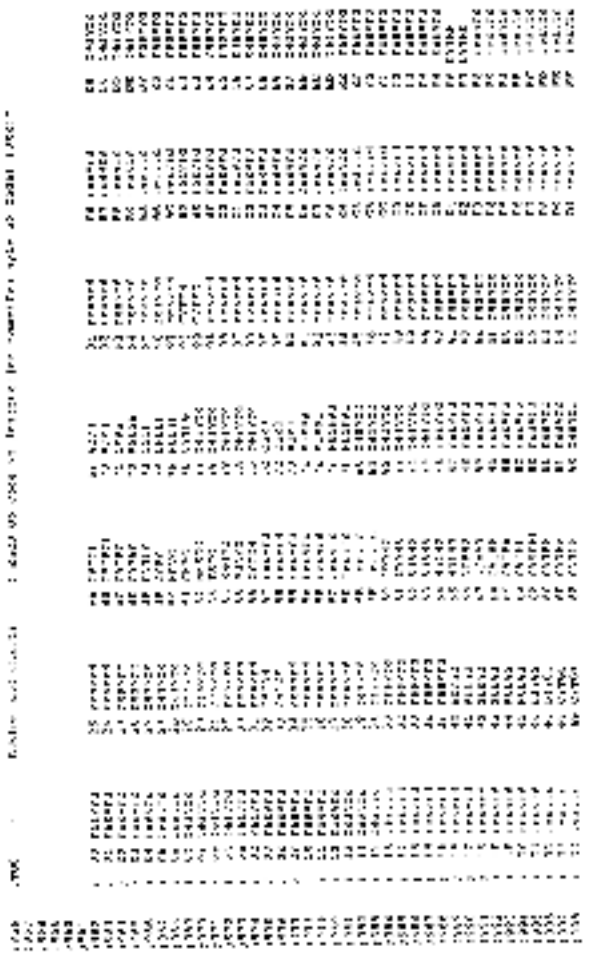


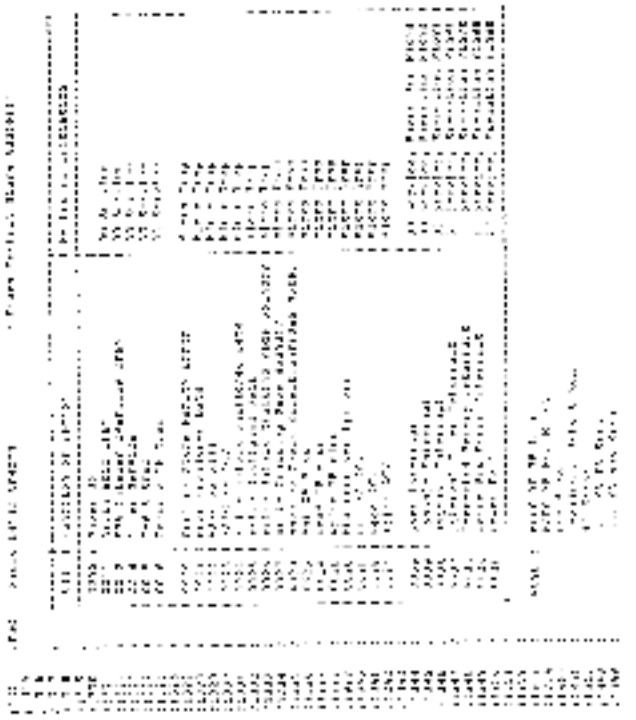
```
1    ZZZZZZ.ZZZ            ZZZZZZ .ZZZ )      ZZZZZZ .ZZZZZZ    .ZZZZ ZZZ  Z.ZZ  ZZZZZ ZZZZ * ZZZZZ
1    ZZZZZZ.ZZZ            ZZZZZZ.ZZZ

                                          ZZZZ      ---) ZZZZZZZZZZZZZZ
                                          ZZZZ      ---) ZZZZZZZZZZ ZZ.ZZ
                                          ZZZZ      *     ZZZZZZ, ZZZZZZZZZZZZZ, Z. Z. ZZZZZZZZ,  Z, Z. ZZZZZZZZZ
                                          ZZZZ
ZZZZ    ZZZZZZ
ZZZZ    ZZZZ    *      ZZZZZZZZ ZZZZZZZZ*
ZZZZ    Z ZZ    ZZZZ ZZ ZZZZ ZZZ ZZZ ZZ
ZZZZ    Z Z     ZZZZZZ ZZZZ ZZZZZZZZZZZZZZ ZZZ ZZZ Z ZZZ ZZZZZZ
ZZZZ    Z ZZ    ZZZZZZZZ ZZZZ ZZZZZZZZZZZZZ
ZZZZ    Z ZZ    ZZZ ZZZZZZZ ZZZZZ ZZZZZZ ZZZ ZZZZZZZ ZZZ ZZZZZZ ZZZZZZZZZZ.
ZZZZ    Z ZZ    ZZZZZZ, ZZZZZZZ.
```

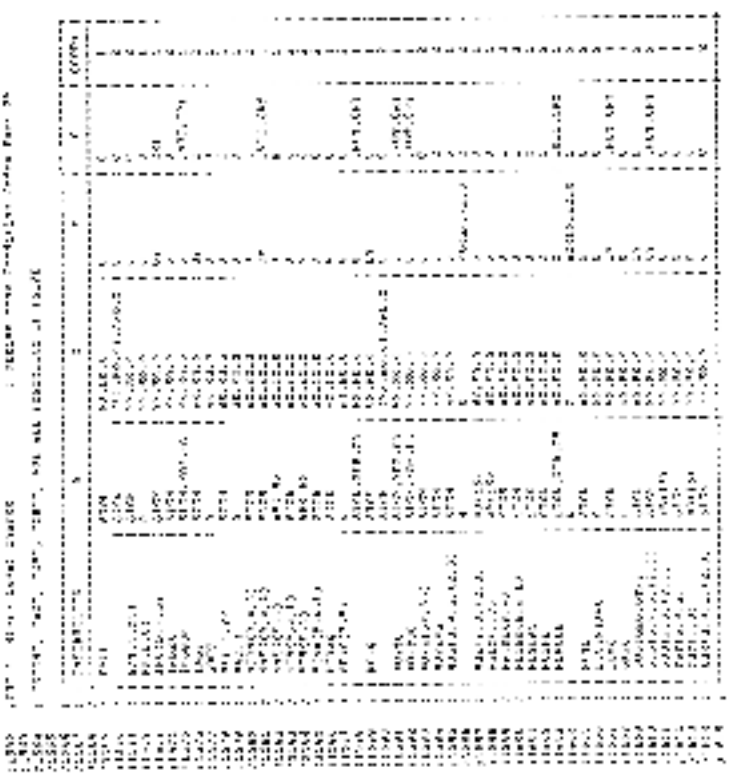


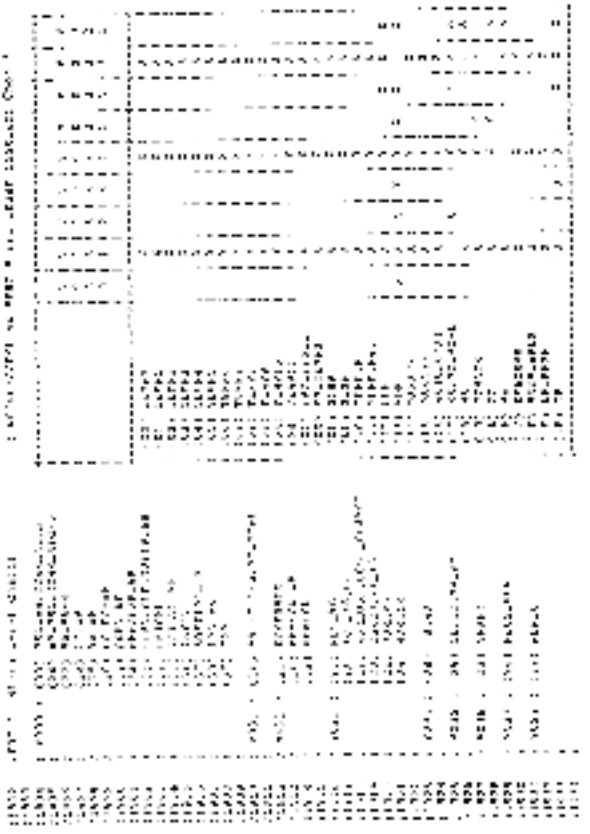




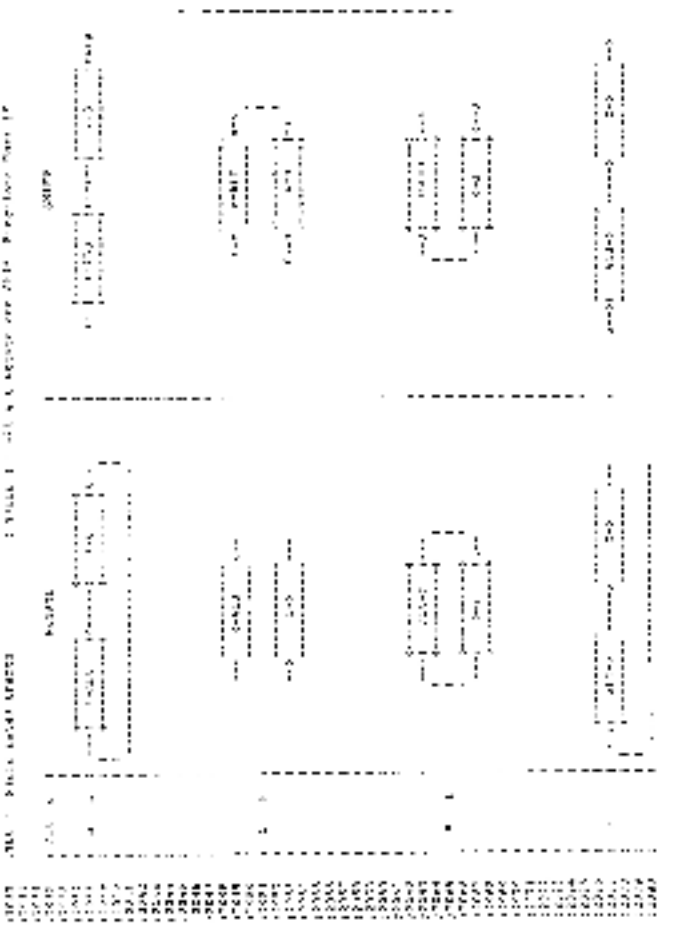


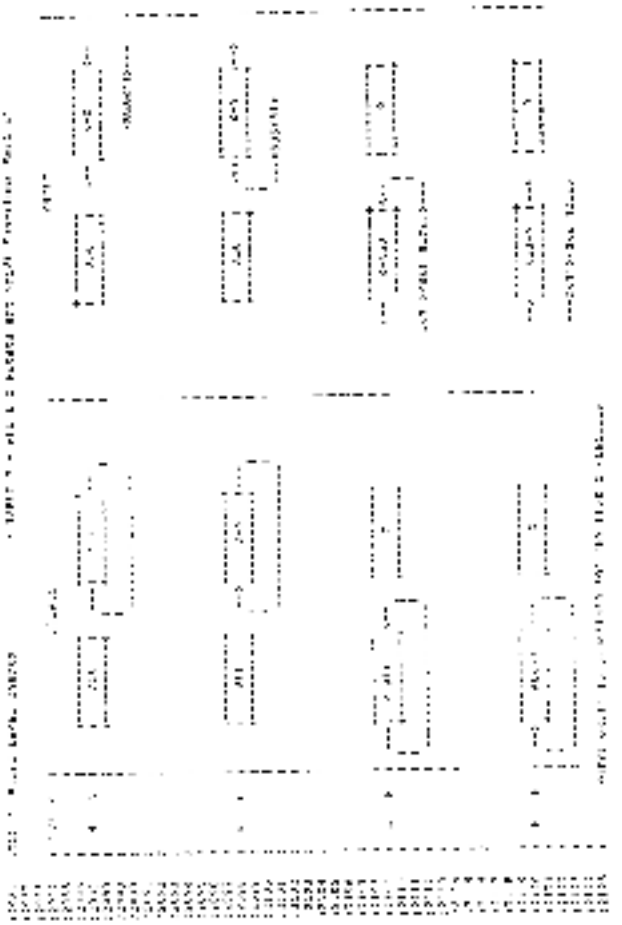






CHARTS (CONT'D)





CHARTS (CONT)

```
.MACRO
.ENDM
.MACRO
.ENDM
.MACRO
.ENDM
```

MACROS (CONT)

MACROS (CONT)

```
```

```
.MACRO  ,,,
```

```
```

11/780 - STAR (1 km) TEL
11/750 - COMET (0.1 km) EXCL G/A
11/730 - NEBULA.

COMET BUS, 24 BIT ADD
16 MEG OF MEMORY. TOP 1 MEG
USED AS I/O SPACE
FC0000 - FFFFFF - UNIBUS ADD SP

MICROVOLPY AT BOOT, 96 LOWER STARTS
96% - LOWER 4K.

DATA PATH PC - 32 BIT 4 GIG VA
PA - 24 BIT - 16 MEG PHYS ADDRESS
TB TRANSLATE FROM VA TO PA.

7 LEVELS OF ARBITRATION (7 IS HIGHEST)
7 - UNIBUS (USER/KERNEL DIAG)
6
5 } NOT USED
4 - UBA (CONTAINS INT LOGIC)
3
2 } FOR OPTIONS
1
0 - CPU.

LEVELS SET BY LINKS.
B.12 LEVELS ARE 4 5 6 7

NOTES

4 Bits = Nibble
8 " = Byte
16 " = Word
32 " = Doubleword
64 " = Quadword

NOTE

Instructions fetched on Longword Boundaries are executed on Byte Boundaries.

(3) How [illegible] memory,
Disable Cache

>>> D/I 25 1 (Default 1 in the Reg 25). Work in Byte mode

>>> E/G 0 = Examine Gen Purpose Register 0 - F.

>>> B DUA0

Dev. Channel, Drive

B = [illegible] I/O mode
(till P goes to
>>> - [illegible] I/O

DS >
ROM $$ >
] Program I/O mode.

NOTES

AVAILABLE DIAGNOSTICS

DEC TRU64 SYSTEM EVENT
ERROR REGISTER WEB ERROR LOG

DEC UNSUPPORTED DIAGS

UNIQUE DIAG SUPPORT EC5AA-GXE
RL ESSAA (OR 8/16 GEN.)
DS> - FOUNDATIONAL TEST:
RASP/SYNC
RDIO3/DX
RMB0
RPCL/7
RAD7
RXCI/2

EVSBA - AUTOSIZER
RL EVSBA

DS> HELP DEV DPAE1
LISTS AVAILABLE TESTS FOR THAT DEVICE

DS> HELP V5 EVSBA

NOTES

Missiles.

Max cable length = 160'

SID.

R/O lines Sectionals	Sectionals

Procedure req. R/O sys req

Possibly use req. 8 - this allows

Operation of VMS.

Req 8 Ø - Req B/R. Logic

8 A - Req - Logic

8 C - Req - Logic

NOTES

NOTES

Your comments and suggestions help us improve our documentation effort to provide the quality and usefulness of our publications.

What is your overall rating of this manual? Very good Good Fair Poor

What do you like most about this manual?

What do you like least about this manual?

Does this manual satisfy the need you had when you purchased the system? Yes No

Do you find errors in this manual? If so, specify by page.

Please indicate the type of user/reader that you most nearly represent.

Assembly language programmer
Higher-level language programmer
Occasional programmer (experienced)
User with little programming experience
Student programmer
Other

Name Date
Title Company
Address
City State Zip Code
or
Country

Thank you for your cooperation.

Digital Equipment Corporation
Accessories and Supplies Group
P.O. Box 10826
Merrimack, New Hampshire 03054

Attention: Publications Group, ZK

FOLD HERE

Order No. EK-VAXMH-HB

Fold here

Do Not Tape Please Keep Area Clear

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

Digital Equipment Corporation
International Services-Customer Assurance
12 Crosby Drive, BO1-2/B5
Bedford, MA 01730