VAX 8820/8830/8840 System Hardware User's Guide

Order Number: EK-8840H-UG-001

This document was prepared using VAX DOCUMENT, Version 1.0

CONTENTS

CHAPTER 1 INTRODUCTION

CHAPTER 2 VAX 8820/8830/8840 SYSTEM OPERATION

CHAPTER 3 FUNCTIONAL DESCRIPTION

CHAPTER 4 VAX 8820/8830/8840 SYSTEM MAINTENANCE

Appendix A DF112 MODEM INFORMATION

EXAMPLES

FIGURES

TABLES

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 VAX 8820/8830/8840 SYSTEM OVERVIEW

The basic VAX 8820/8830/8840 system is contained in two physical units: a MicroVAX II system, used as the console, and a cabinet arrangement. A typical VAX 8820/8830/8840 system shown in Figure 1-1.

1.2 SCOPE

This guide contains basic hardware operating information and customer care and use instructions. Detailed information pertaining to specific areas of hardware, software, or diagnostics can be found in the documentation referenced in Table 1-1, Table 1-2, and Table 1-3.

Table 1-1 VAX 8820/8830/8840 System Hardware Manuals

Document Title	Document Number
VAX 8820/8830/8840 System Installation Guide	EK-8840I-IN
VAX 8530/8550/8700/8800/8820/8830/8840 System Diagnostic User's Guide	EK-KA88D-UG
VAX 8820/8830/8840 System Site Planning and Preparation Guide	EK-8840P-SP
VAX 8820/8830/8840 System Hardware User's Guide	EK-8840H-UG

Table 1-2 VAX 8820/8830/8840 System Related Hardware Manuals

Document Title	Document Number
Installing & Using the LA75 Companion Printer	EK-OLA75-UG
MicroVAX II Owner's Manual	AZ-GL1AA-MN
MicroVAX II Technical Manual	AZ-GL0AA-MN
TK50 Tape Drive Subsystem User's Guide	EK-OTK50-UG
DF112 Modem User's Guide	EK-DF112-UG
Remote Services Console Field Service Manual	EK-KCRSC-FS
Remote Services Console User's Guide	EK-KCRSC-UG

VAXBI
EXPANSION
CABINET   MEMORY CABINET   CPU CABINET

MICROVAX II
CONSOLE
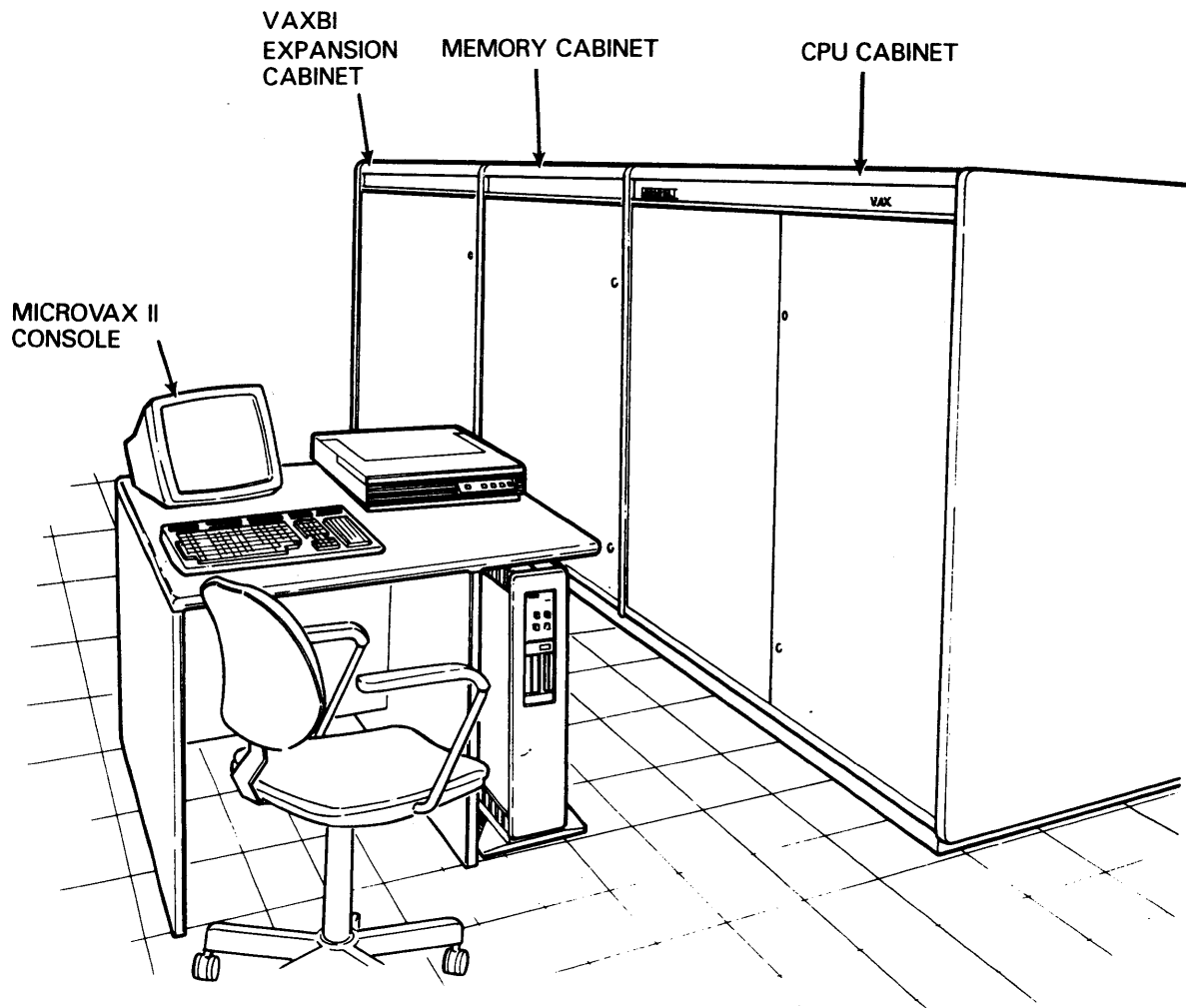
TTB-0257

Figure 1-1   Typical VAX 8820/8830/8840 System

Table 1-3   VAX 8820/8830/8840 System Software Documentation

	Document Title	Document Number
	VAX 8820/8830/8840 Console User's Guide	AA-FJ84A-ME
Volume 1A System Reference	*VAX/VMS Primer*	AA-D030A-TE
	VAX/VMS Summary Description	AA-D022A-TE
	VAX/VMS Information Directory	AA-D016A-TE
	VAX/VMS Release Notes	AA-D015A-TE
	VAX-11 Software Installation Guide	AA-D021A-TE
	VAX/VMS System Services Reference Manual	AA-D018A-TE
Volume 1B System Reference	*VAX/VMS Command Language User's Guide*	AA-D023A-TE
	VAX/VMS Linker Reference Manual	AA-D019A-TE
	VAX-11 Symbolic Debugger Reference Manual	AA-D026A-TE
Volume 1C System Reference	*VAX-11/RSX-11M Programmer's Reference Manual*	AA-D020A-TE
	VAX-11 MACRO Language Reference Manual	AA-D032A-TE
	VAX-11 MACRO User's Guide	AA-D033A-TE
Volume 2A System Procedures	*VAX-11 Common Run-Time Procedure Library Reference Manual*	AA-D036A-TE
	VAX-11 Text Editing Reference Manual	AA-D029A-TE
Volume 2B System Procedures	*VAX/VMS Operator's Guide*	AA-D025A-TE3
	VAX/VMS System Manager's Guide	AA-D927A-TE
	VAX/VMS System Messages and Recovery Procedures Manual	AA-D017A-TE

Table 1–3 (Cont.)   VAX 8820/8830/8840 System Software Documentation

	Document Title	Document Number
	VAX/VMS UETP User's Guide	AA–D643A–TE
	VAX–11 Disk Save and Compress User's Guide	AA–D739A–TE
Volume 3 VAX/VMS I/O	*VAX/VMS I/O User's Guide*	AA–D028A–TE
	Introduction to VAX–11 Record Management Services	AA–D028A–TE
	VAX–11 Record Management Services Reference Manual	AA–D031A–TE
	VAX–11 Record Management Services User's Guide	AA–D781A–TE
Volume 4	*IAS/RSX–11M RMS–11 Macro Programmer's Reference Manual*	AA–0002A–TC
	Introduction to RMS–11	AA–0001A–TC
	RSX–11 RMS–11 Utilities User's Guide	AA–D083A–TC
	PDP–11 SORT Reference Manual	AA–3341C–TC
Volume 5A Optional Software	*VAX–11 FORTRAN IV–PLUS Language Reference Manual*	AA–D034A–TE
	VAX–11 FORTRAN IV–PLUS Language User's Guide	AA–D035A–TE

1.3   INTRODUCTION

The VAX 8820, VAX 8830, and VAX 8840 systems are LSI–based, high–performance systems compatible with all of the VAX family of 32–bit machines. They combine 32–bit architecture with a virtual memory operating system and memory management (typically using the VAX/VMS Operating System).

The VAX 8820 system uses two CPUs; it can be upgraded to a VAX 8830 system (three CPUs) or a VAX 8840 system (four CPUs) by installing the appropiate uprade kits. The VAX 8830 system may be upgraded to a VAX 8840 system by installing the appropriate upgrade kit.

VAX 8820/8830/8840 systems support symmetric multiprocessing (SMP), which allows all CPUs to execute most code paths with no restrictions on processor mode. Symmetric multiprocessing makes more efficient use of multiprocessor hardware than asymmetric multiprocessing, increasing the throughput of the multiprocessor system. The term "primary CPU" designates the CPU responsible for controlling synchronization among the CPUs.

The VAX 8820/8830/8840 system hardware contains a native instruction set (microcode) that provides built–in floating–point, packed decimal arithmetic, and character and string manipulation instructions.

1.4   PHYSICAL DESCRIPTION

Figure 1–2 and Figure 1–3 show the cabinet layout of the VAX 8820/8830/8840 system and the location of the major components. Table 1–4 provides a list of the physical characteristics.
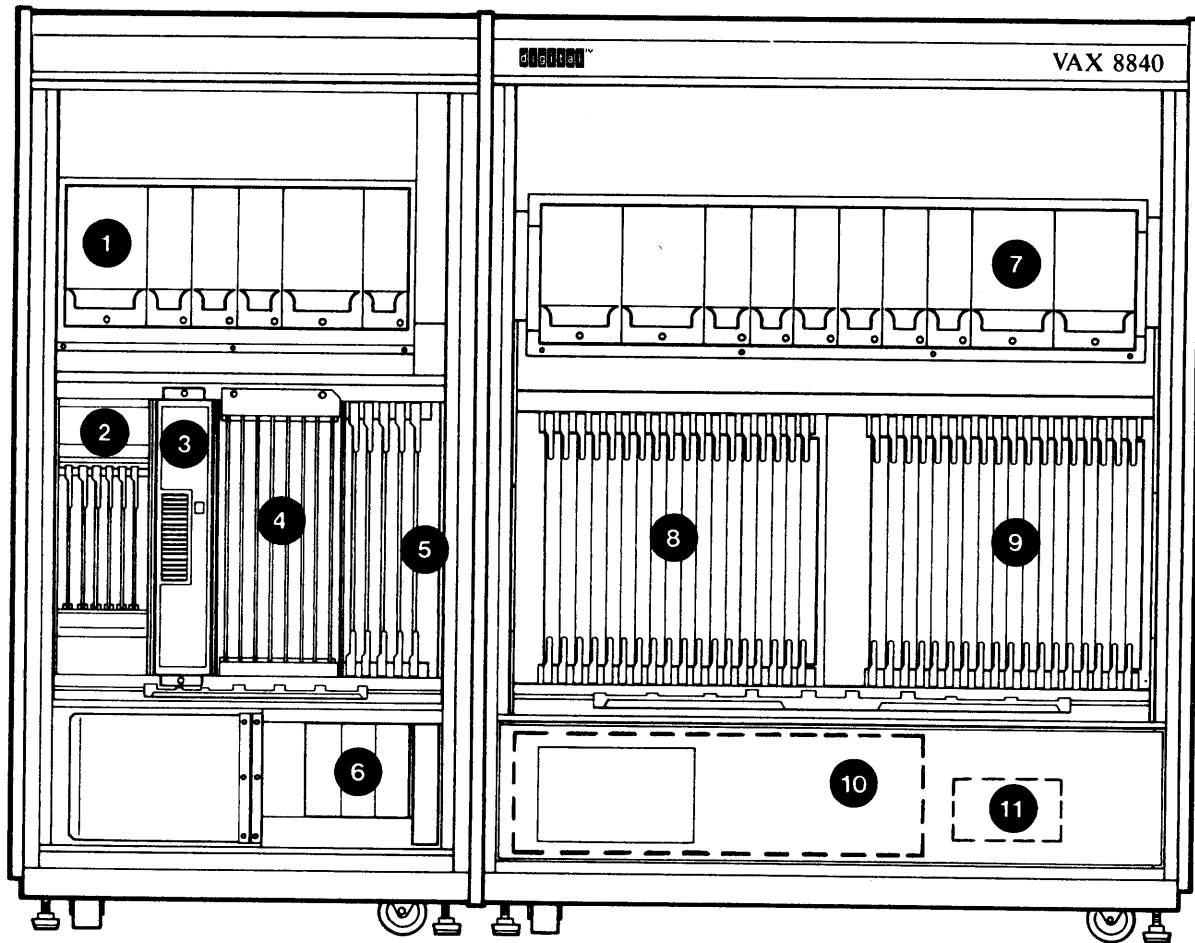
MEMORY CABINET   CPU CABINET

FRONT VIEW

1. MPS POWER REGULATORS.
2. VAXBI
3. BATTERY BACKUP UNIT
4. MEMORY ARRAY MODULES
5. NMI BUS WINDOW (NBW),
   MEMORY CONTROLLER (PMCL),
   AND VAXBI INTERFACE (UP TO
   3 PBIA MODULES).
6. PBOX (3 CSP MODULES AND 1
   ILM MODULE).

7. MPS POWER REGULATORS
8. CPU 2, CPU 3, AND NMI BUS
   WINDOW (NBW).
9. CPU 0, CPU 1, CONSOLE
   INTERFACE (CIM) AND
   CLOCK (PCLK).
10. 880 POWER CONTROLLER
11. SST BOX

TTB-0258

Figure 1-2   VAX 8820/8830/8840 System Cabinet Layout (Front)
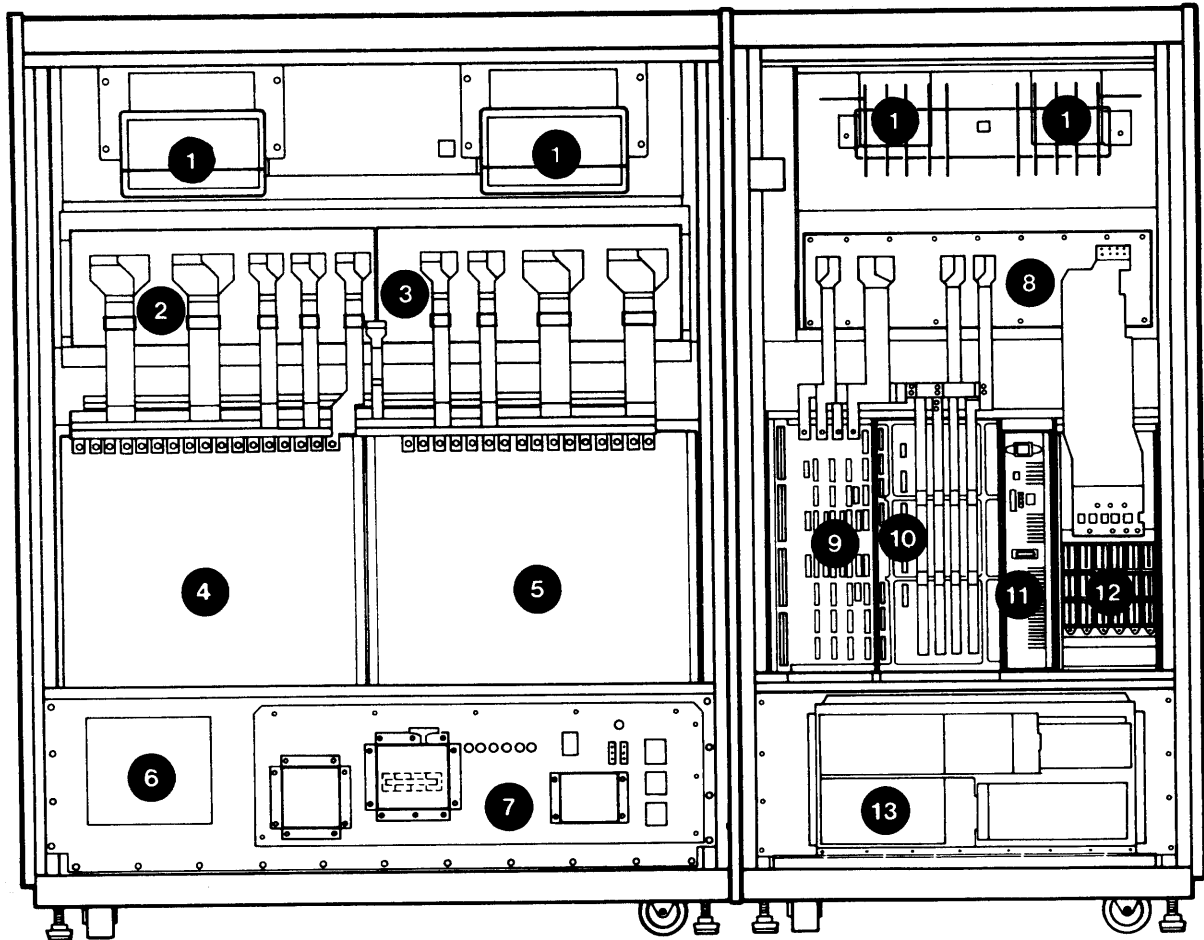
CPU CABINET   MEMORY CABINET

REAR VIEW

1. AIR MOVER DEVICE (BLOWER)
2. MPS BACKPLANE 1
3. MPS BACKPLANE 2
4. CPU 0, CPU 1, CONSOLE INTERFACE (CIM) AND CLOCK (PCLK).
5. CPU 2, CPU 3, AND NMI BUS WINDOW (NBW).
6. SST BOX
7. 880 POWER CONTROLLER

8. MPS BACKPLANE 3
9. NMI BUS WINDOW (NBW), MEMORY CONTROLLER (PMCL), AND VAXBI INTERFACE (UP TO 3 PBIA MODULES).
10. MEMORY ARRAY MODULES
11. BATTERY BACKUP UNIT
12. VAXBI
13. PBOX (3 CSP MODULES AND 1 ILM MODULE).

TTB-0259

Figure 1–3   VAX 8820/8830/8840 System Cabinet Layout (Back)

Table 1–4   VAX 8820/8830/8840 System Physical Characteristics

Characteristic	Parameters		
Dimensions	**CPU Cabinet H9650 DWHB (4 CPUs)**	**Memory Cabinet H9654–EA**	**VAXBI Expansion Cabinet H9654–EC/ED**
Height	152 cm (60 in)	152 cm (60 in)	152 cm (60 in)
Width	118 cm (46.5 in)	76 cm (30 in)	76 cm (30 in)
Depth	76 cm (30 in)	76 cm (30 in)	76 cm (30 in)
Weight			
50 Hz	487 kg (1074 lb)	305 kg (673 lb)	308 kg (680 lb)
60 Hz	487 kg (1074 lb)	305 kg (673 lb)	245 kg (540 lb)

Environmental

Heat Dissipation: VAX 8840 system – 37.3k BTU/hr

VAX 8830 system – 31.9k BTU/hr

VAX 8820 system – 26.6k BTU/hr

VAXBI Expander cab. – 6.5K BTU/hr

Maximum
Temperature Rise:
    3°C/hr (5.5°F/hr)

Temperature
Range:
    Operational – 15°–32°C (59°–90°F)
    Non–operational – –40°–66°C (–40°–151°F)

Humidity:
    Operational – 20–80 % RH (non–condensing)
    Non–operational – 10–95 % RH (non–condensing)

Cooling system:
    Type – Air–moving device
    Air Source – Filtered ambient air
    Drive – Three–phase, 208 V ac, 60Hz; 460 V ac, 50Hz; induction motor
    Air Mover – Quad–inlet, dual–outlet centrifugal blower

Table 1–4 (Cont.)   VAX 8820/8830/8840 System Physical Characteristics

Characteristic	Parameters
Electrical	
Maximum Power Consumption:	12.6 kW
Input Requirements:	Voltage:
	60 Hz – Three–phase 208 V ac (180–222 V rms)
	50 Hz – Three–phase 380 V ac (331–407 V rms)
	50 Hz – Three–phase 416 V ac (360–443 V rms)
	Frequency Tolerance:
	60 Hz +/– 3 Hz
	50 Hz +/– 3 Hz

1.4.1   Enclosures

The VAX 8820/8830/8840 system is housed in a MicroVAX II pedestal enclosure, an H9650 cabinet, and an H9657 cabinet.

The MicroVAX II pedestal enclosure (system console) contains:

- MicroVAX II processor with 5 Mbyte of memory

- VT320 console (terminal and keyboard)

- Winchester disk with controller

- Removable streamer tape drive and controller

- DZQ11 asynchronous multiplexer (remote console and printer)

- QC module which connects by cable to the H9650 cabinet

The H9650 cabinet, usually called the CPU cabinet, contains:

- Power controller and solid–state transformer box

- Modular power system

- Cooling system

- Module card cage (NMI backplane for 4 CPUs)

The H9657 cabinet, usually called the memory cabinet, contains:

- PBox and modular power system

- Battery Backup Unit

- Module card cage (memory controller, I/O)

- Module card cage (memory arrays)

- Module card cage (6 VAXBI slots)

- Cooling system

Two VAXBI expansion cabinets can be added to the system to house devices and device controllers on up to six VAXBIs.

CHAPTER 2
VAX 8820/8830/8840 SYSTEM OPERATION

2.1 INTRODUCTION
This chapter covers the VAX 8820/8830/8840 system console and system operation and provides a brief summary of the console commands.

2.2 SYSTEM SETUP AND PREPARATION FOR USE
Check that the VAX 8820/8830/8840 system is set up and configured properly. Refer to the following documents:

- *VAX 8820/8830/8840 System Site Planning and Preparation Guide* (EK–8840P–SP)

- *VAX 8820/8830/8840 System Installation Guide* (EK–8840I–IN)

- *VAX 8820/8830/8840 Console User's Guide* (AA–FJ84A–ME)

2.2.1 Input Power
Three amber LEDs on the 880A/B Power Controller panel will be lit when input power is available to the system. There is one LED for each phase: L1, L2, and L3. The location of these LEDs on the panel is indicated in Figure 2–1. Before turning the system power on, check that these three LEDs are lit, indicating that three-phase power is available to the system.
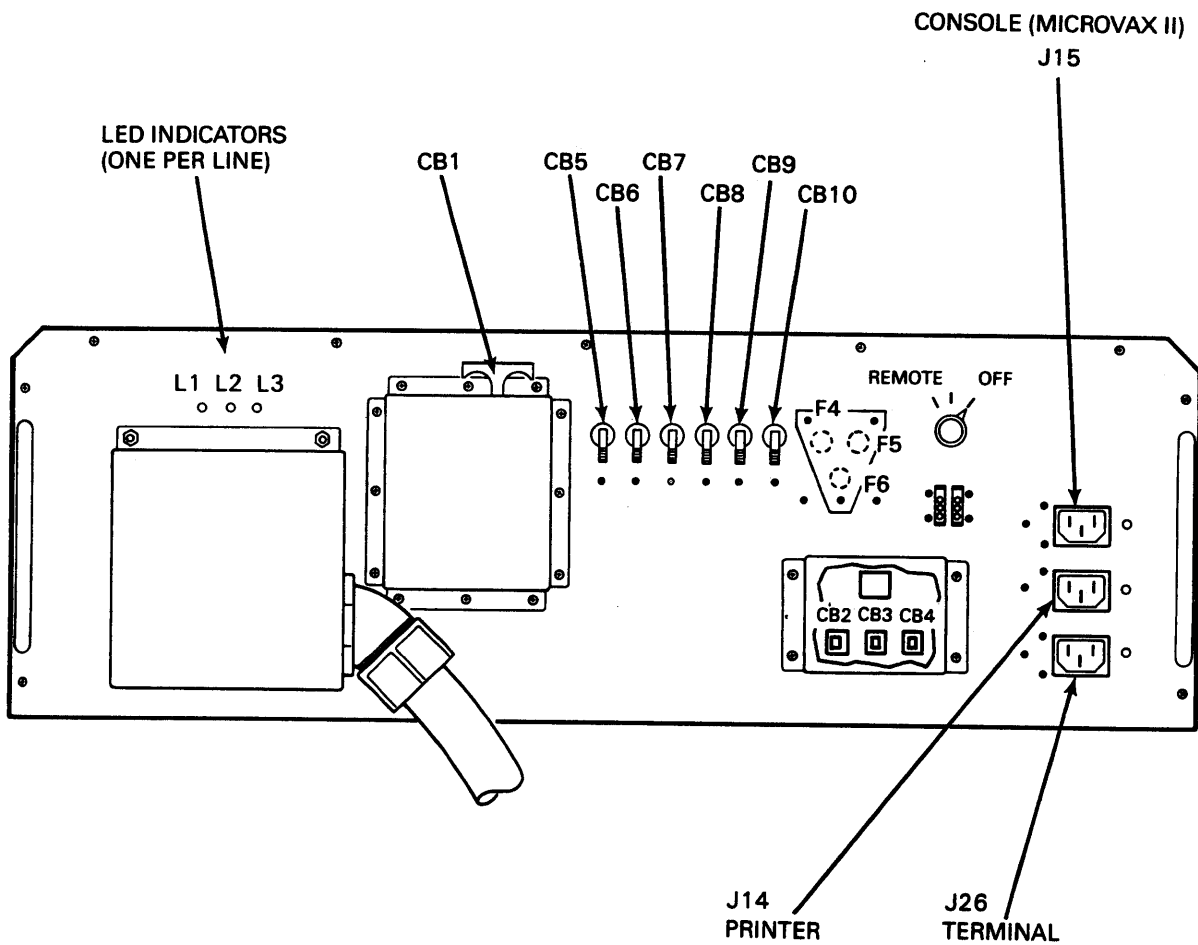
Figure 2-1   880 A/B Power Controller Panel

2.2.2 Primary System Power

Primary power is applied to the VAX 8820/8830/8840 system in three steps:

1. Turn on peripheral power breakers

2. Turn on main power breaker (CB1)

3. Turn on console power

2.2.2.1 Power Breakers

The VAX 8820/8830/8840 system has ten circuit breakers as listed in Table 2-1.

Table 2-1   VAX 8820/8830/8840 System Circuit Breakers

Breaker	Voltage	Unit(s) powered
1	3-phase ac	L1,L2,L3 to SST box.
2,3,4	1-phase ac (L1,L2,L3)	Dranetz Power Monitor port
5	1-phase ac (L1)	Console (MicroVAX II)
6	1-phase ac (L2)	Printer
7	1-phase ac (L2)	Local Console Terminal
8	1-phase ac (L1)	Spare
9	1-phase ac (L1)	Spare
10	1-phase ac (L1)	PBox and Battery Backup Unit

The circuit breakers are located on the 880A/B Power Controller panel which is at the lower right side of the rear of the CPU cabinet. The panel is illustrated in Figure 2-1.

Turn the circuit breakers on in two steps:

1. With CB1 in the OFF position, turn circuit breakers 5 through 10 to the ON position. Circuit breakers 2 through 4 are used by qualified field service personnel to connect the Dranetz power monitor to the main system input power; circuit breakers 2 through 4 should be in the OFF position during normal operation.

2. Turn CB1 to the ON position.

2.2.2.2 Starting the Console Subsystem

The console subsystem is started in two steps:

1. Turn the power on.

2. Run the console subsystem software.

The console subsystem receives power from the VAX 8820/8830/8840 system cabinet when the main and console circuit breakers (CB1 and CB5) are in the ON position. The console power-on switch shown in Figure 2-2 must be placed in the ON or "1" position to turn on console power.

CONSOLE POWER SWITCH
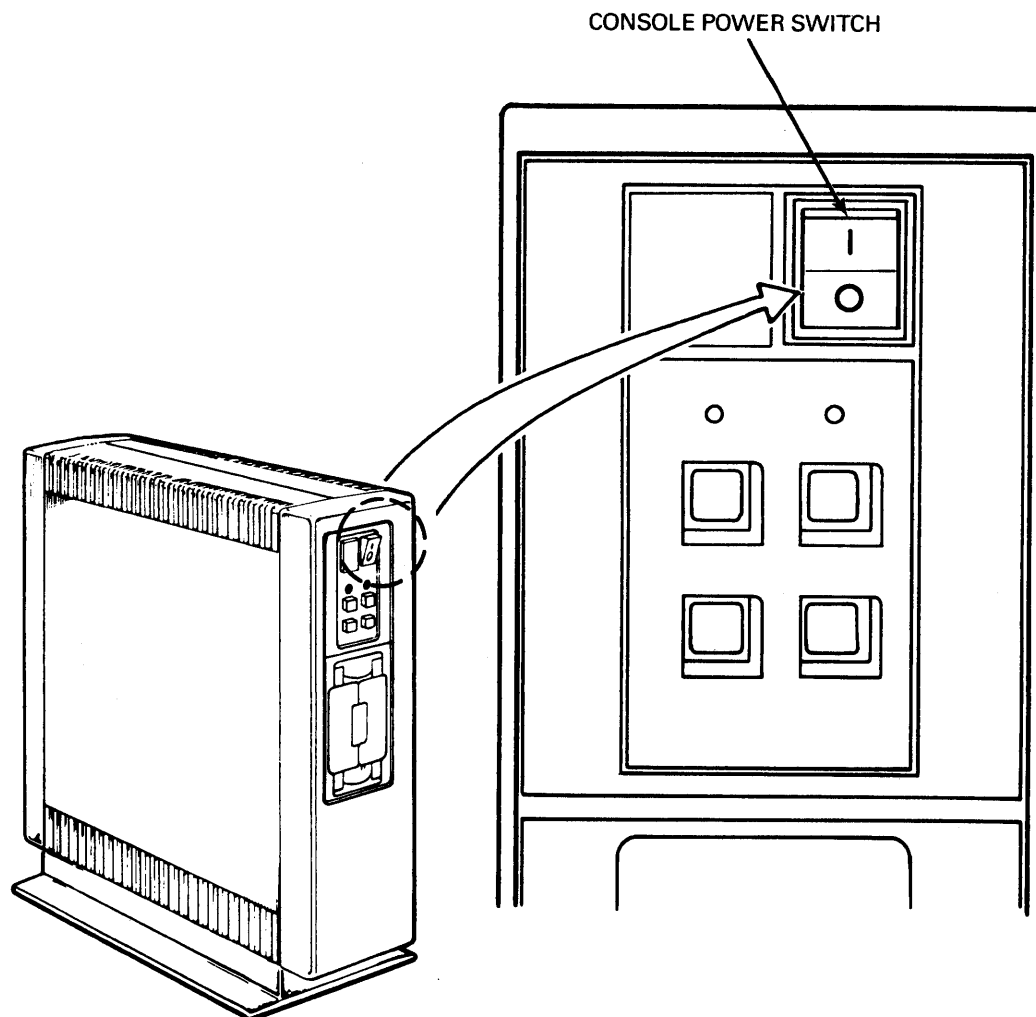
TTB-0261

Figure 2-2 Location of the Console Power-On Switch

The console subsystem software is started by the following steps (see Example 2–1):

1. MicroVAX II automatically boots when power is turned on.

2. Operator is logged into VAX/VMS Operating System on MicroVAX II.

3. The POLARIS program runs @VERIFY to check power status.

4. The POLARIS program displays the local console prompt: PS–OFF>.

```
KA630-A.V1.3

Performing normal system tests.

    7..6..5..4..3..

Tests completed.

Loading system software.

    2..1..0..

    MicroVMS Version V4.5  3-Sep-1986 20:00

The Console system is now executing the startup procedure.

    Define System Logical Names...
    Starting DECnet...
%RUN-S-PROC_ID, identification of created process is 00000025
%RUN-S-PROC_ID, identification of created process is 00000027
    Starting QFueue Manager...
    Starting Batch Queue...
    Misc Installs...
Job START_CONSOLE (queue SYS$BATCH, entry 13) started on SYS$BATCH
%SET-I-INTSET, login interactive limit = 64, current interactive value=0
    VAX-xxxx

    Copyright 1987 Digital Equipment Corporation

POLARIS — VAX-xxxx Console Program — Version 0.1240
Sunday  4-JAN-1987 15:34:24.72     OPA0
    VAX-xxx Console EMM Subprocess — Version 0.155
    VAX-xxx Console Printer Subprocess — Version 0.31
%Console Auto Startup Enabled

******************************
    <Remote Console enabled>
******************************
%EMM-I-POWER_OFF, System Power is Off

    The Current Date and Time is Sunday 4-JAN-1987 15:34:42.67
%EMM-I-POWER_OFF, System Power is Off
% Command file aborted at line 20
PS-OFF>
```

TTB-0367

Example 2–1   Console Display of Console Subsystem Startup

WARNING
High voltage and power is present in the VAX
8820/8830/8840 system. If for any reason there
is a need to open the cabinets, be sure to remove
power from the system. Only qualified field service
personnel should open or enter the system cabinets.
Consult with the system manager before powering
down the system.

2.2.3.1 Power On

The power–on sequence is initiated by entering the POWER ON command at the console terminal. This command turns on the solid–state transformer (generates 300 V dc) and air movers 1 and 2:

PS–OFF>POWER ON

The power–on sequence can also be initiated by a VAX 8820/8830/8840 system powerfail recovery. The console software determines if the software is recovering from a "console only" power failure. It determines this by examining the VAX power status via the environmental monitoring module in the power system.

The power–up procedure requires that the console send commands to the EMM units to apply system power in a particular sequence. Table 2–2 shows the sequence in which the power supply modules are activated. Figure 2–3 illustrates the MPS unit letter identification code.

EMM CONTROL LETTER DESIGNATION

MOD E1	MOD E1	MOD D1	MOD D1	EMM 1	MOD C1	MOD D1	MOD D1	MOD E1	MOD E1
H7180	H7180	H7187	H7187	H7188	H7186	H7187	H7187	H7180	H7180
−5.2 V	−5.2 V	−2.0 V	−2.0 V	EMM	+5.0 V	−2.0 V	−2.0 V	−5.2 V	−5.2 V

CPU CARD CAGE (FRONT VIEW)

EMM CONTROL LETTER DESIGNATION

MOD F2	EMM 2	MOD B2	MOD C2	MOD E2	MOD D2
H7189	H7188	H7186	H7186	H7180	H7187
BI	EMM	+5.0 V	+5.0 V	−5.2 V	−2.0 V

MEMORY/IO CARD CAGE (FRONT VIEW)

TTB-0262

Figure 2–3   MPS Unit Letter Identification

Table 2–2   Module Power Supply Turn on Sequence

Order	Module	Power Application
1	J	+300 V to 4 MPS cardcage 300 V buses
2	B2	+5.0 V to memory
3	C2,C1	+5.0 V to NMI devices (CPU & memory)
4	F2	+5.0 V, +12 V, −12 V, −5.2 V, −2.0 V to VAXBI
5	E1,E2	−5.2 V to NMI devices (CPU & memory)
6	D1,D2	−2.0 V to NMI devices (CPU & memory)

2.2.3.2 System Initialization

After secondary power has been turned on, the system may be initialized by entering @SYSINIT on the console:

PS–HW–0>@SYSINIT

This verifies and initializes the system, and loads the VAX 8820/8830/8840 system microcode. The SYSINIT.CMD command file is shown in Example 2–2.

```
!+
! SYSINIT.CMD - For a VAX xxxx system
! REV 1.0
!
! This SYSINIT.CMD has been specifically tailored
! for interactive command use.
! We suggest that you do not edit this file unless
! you wish to load USER WCS.
!-
!----------------------------------------------------------------------------------------------------------------
DISABLE ERROR                             ! If error, abort
CLEAR RESTART__FLAGS                      ! Always allow booting
CONSOLE                                   ! Set command level
SET DEFAULT HEX                           ! Set up logging
SET TERMINAL LOCAL
SHOW TIME                                 ! Put time stamp in logfile
!
POWER STATUS                              ! Display current power state
!
IF NOT POWER__ON THEN ABORT
                                          ! If power is not on exit
                                          ! system initialization procedure.
!
INITIALIZE/HARDWARE                       ! Init to base hardware level
SET CLOCK NORMAL                          ! Set clock period
!
VERIFY MODULE__PLACEMENT                  ! See if any CPUs are there
        IF NOT STATUS THEN ABORT          ! If no CPUs, quit.
!
SELECT CPU ALL                            ! For whatever CPUs are there, ...
!
SENSE REVISION CPU                        ! Read CPU board revisions
!+
! Power up diagnostics are in the command file POWERUP.CMD
! located in directory [PSTAR]. To inhibit power up diagnostics
! ENABLE QUICK__BOOT.
IF NOT QUICK__BOOT THEN @[PSTAR]POWERUP.CMD
```

TTB-0263

Example 2–2 (Cont.)   SYSINIT.CMD Command File

```
!-
CONSOLE
SET CLOCK ON
INITIALIZE/KNOWN-STATE                       ! Put all cpus in known state
SELECT CPU ALL                               ! Reset CPU selection
LOAD WCS/CPU ALL UCODE.BIN                   ! Load ISP microcode,
        IF NOT STATUS THEN ABORT             ! Exit if any error is encountered
LOAD CACHECONROL/CPU ALL CCODE.BIN           ! Cache microcode,
        IF NOT STATUS THEN ABORT             ! Exit if any error is encountered
LOAD IBDECODER/CPU ALL DRAM.BIN              ! IBOX decoder tables.
        IF NOT STATUS THEN ABORT             ! Exit if any error is encountered
SET CLOCK OFF                                  Stop clocks to start microcode
START/WCS/CPU ALL CSM-INIT                   ! Start the CPUs!
        IF NOT STATUS THEN ABORT             ! Exit if something went wrong
!                                            !
LOAD SDF/CPU ALL SDFDEF.BIN                  ! Load microcode constants
        IF NOT STATUS THEN ABORT             !
!+
! Remove the comment flags on the following lines
! if you are loading user WCS
!
! LOAD UWCS/CPU ALL USERWCS.BIN              ! Must come after SDF is loaded
! SET CLOCK OFF                              ! Clocks off to start microcode
! START/WCS/CPU ALL CSM-INIT                 ! Restart microcode with user WCS
!       IF NOT STATUS THEN ABORT             !
!                                            !
!-
INITIALIZE/CPU/CPU ALL                       ! Initialize hardware states & VAX regs
        IF NOT STATUS THEN ABORT
disable error                                !
INITIALIZE/IO                                ! Reset I/O devices
INITIALIZE/NBI                               ! Initialize NBIs
SENSE REVISION MCL                           ! Read memory controller revision
SENSE REVISION NBI                           ! Read NBI adapter revision
REVISION/COMPARE                             ! Verify revisions
REVISION/SAVE                                ! Save present revisions
write nbw1fsr 2000                           ! clear interlock busy
read nbw1clr                                 ! clear interrupts
write nbw0fsr 2000                           ! clear interlock busy
read nbw0clr                                 ! clear interrupts
read memcsr5                                 ! clear nmi fault
!
INITIALIZE/CPU/CPU ALL                       ! Load SID and rev IPRs
        IF NOT STATUS THEN ABORT
```

TTB-0264

Example 2-2 (Cont.)   SYSINIT.CMD Command File

```
!+
! Before memory can be initialized, a memory configuration file
! must have been created.
 This command will create the file memconfig.dat if it doesn't exist.
!-
INITIALIZE/MEMORY/CLEAR                         ! Write 0's to all of memory
!+
! Initialize halt code on each CPU
!-
SET CPU 0
if cpu__available then DEPOSIT SDF 0CA 3        ! Powerfailure halt code
SET CPU 1
if cpu__available then DEPOSIT SDF 0CA 3        ! Powerfailure halt code
SET CPU 2
if cpu__available then DEPOSIT SDF 0CA 3        ! Powerfailure halt code
SET CPU 3
if cpu__available then DEPOSIT SDF 0CA 3        ! Powerfailure halt code
!+
! Attempt to restart or boot the primary
!-
SET CPU PRIMARY                                 ! Single-cpu operations from here on
CONSOLE/SAVE                                    ! Save console status
CLEAR RESTART-FLAGS                             ! Clears WARM and COLD restart flags
!
!End of SYSINIT.CMD
show el 1
disable echo
```

TTB-0265

Example 2–2 (Cont.)   SYSINIT.CMD Command File

2.2.3.3 System Boot

After secondary power has been applied, the system may be booted (VAX/VMS Operating System loaded and started). The boot sequence can be initiated using the default file DEFBOO.CMD or from a particular command file such as BCIBOO.CMD or BDABOO.CMD. Samples of these three files are shown in Example 2-3, Example 2-4, and Example 2-5.

The command file DEFBOO.CMD, used to boot the VAX/VMS operating system from the console storage device, is shown in Example 2-3. Use the command:

    PS-CIO-0>BOOT

```
! DEFBOO.CMD - Default Boot Command file
! Rev Num 1.0
!
! This command procedure will be executed automatically during system
! initialization or following a power failure or an error halt if you
! have enabled the AUTO__BOOT keyswitch. It will also be executed when
! the command BOOT is entered.
!
! Boot DDDnnn/R5+##
!
! Replace DDDnnn with the name of the boot command procedure that you
! wish to execute.
!       DDD   device name: BCI, BDA
!       nnn   boot device unit number

BOOT BCI/R5=1
```

TTB-0266

Example 2-3   DEFBOO.CMD Command File

2-12

```
BCIBOO.CMD
! REV 1.0
!
! Command procedure to BOOT the VAX/VMS Operating System from
! an HSC over the BCI.
!
! The following register deposits must be done before executing
! this command procedure or must be edited to correspond to the
! hardware configuration:
! R1 - Bus address information
! R2 - CI port # of HSC(s) to which drive is ported
! R3 - device unit number
CLEAR DEBUG
SET TERMINAL LOCAL                          !Set up logging
SET DEFAULT HEXADECIMAL
SET CPU PRIMARY                             !Select CPU that will be the primary
IF NOT HALTED THEN HALT                     !Halt it and avoid warning messages
EXAMINE SDF 0CA                             !Retrieve halt code
DEPOSIT AP @                                !Move halt code to AP
INITIALIZE/CPU/cpu=all                      !Initialize CPU
INITIALIZE/MEMORY/clear                     !Init memory
INITIALIZE/IO                               !Clear BI errors, NMI and BI node state info
DEPOSIT PHY/L 3E000000 020F00F0             !Clear interlock timeout in mem CSR0
EXAMINE PHY/L 3E000010                      !Read mem CSR4 (clears interrupts)
EXAMINE PHY/L 3E000014                      !Read mem CSR5 (clears NMI fault)
DEPOSIT PHY/L 3E000008 F0000000             !Clear RDS error bits in mem CSR0
DEPOSIT PHY/L 3A000000 2000                 !Clear interlock busy in NBW1FSR
EXAMINE PHY/L 3A000808                      !Read NBW1CLR (clears interrupts)
DEPOSIT PHY/L 38000000 2000                 !Clear interlock busy in NBW0FSR
EXAMINE PHY/L 38000808                      !Read NBW0CLR (clears interrupts)
EXAMINE PHY/L 3E000014                      !Read mem CSR5 (clears NMI fault)
DEPOSIT R0 %x20                             !CI boot device type code
DEPOSIT R1 %x02                             !Boot dev bus adr:<3:0>=BI node #,<5:4>=BI
                                            #
DEPOSIT R2 %x03                             !CI port # of HSC:<7:0>=CI port # of HSC,
                                            !<15:8>=CI port # of 2nd HSC if dual ported
DEPOSIT R3 %d193                            !Unit # of drive, decimal radix
DEPOSIT R4 0                                !Not applicable
DEPOSIT R5 %x01                             !Software boot control flags and
                                            !root directory # in <31:28>. Defaults
                                            !to SYS0 or unrooted directory
IF P1 THEN DEPOSIT R3 %P1                   !Override above if given in boot cmd
IF P2 THEN DEPOSIT R5 %P2                   !Display disk number
EXAMINE R3
EXAMINE R5                                  !Display R5 parameter
FIND/MEM                                    !Find 64kb of memory, set cold start bit
IF NOT STATUS THEN ABORT                    !Boot if find was successful
EXAMINE SP                                  !Show address of working memory + %X200
LOAD/START @ VMB.EXE                        !Load VMB into good mem + %X200
EXAMINE SP                                  !patch vmb
START @                                     !Start executing VMB
```

TTB-0267

Example 2–4   BCIBOO.CMD Command File

```
!ENABLE ECHO
! BDABOO.CMD
! REV 1.0
!
! Command procedure to BOOT VMS from a BDA disk.
!
! NEXT__PRIMARY is expected to point to the CPU that is to be used
! as the primary CPU.
!
! The following register deposits must be done before executing this
! command procedure or must be edited to correspond to the hardware
! configuration:
! R1 - Bus address information
! R2 - <31:24> = optional controller letter specifier, remaining bits
!                    depend on device type in R0
! R3 - device unit number
!
CLEAR DEBUG
SET TERMINAL LOCAL                          ! Set up logging
SET DEFAULT HEXADECIMAL
SET CPU PRIMARY                             ! Select CPU that will be the primary
IF NOT HALTED THEN HALT                     ! Halt it and avoid unnecessary warning messages
EXAMINE SDF 0CA                             ! Retrieve halt code
DEPOSIT AP @                                ! Move halt code to AP
INITIALIZE/CPU/CPU=ALL                      ! Init CPUs, copy next__primary to current
INITIALIZE/MEMORY/clear                     ! Init Memory
INIT/IO                                     ! Clear BI errors and nmi and BI node state info
DEPOSIT PHY/I 3E000000 020F00F0             ! Clear interlock ttimeout in mem CSR0
EXAMINE PHY/I 3E000010                       ! Read mem CSR4 (clears interrupts)
EXAMINE PHY/I 3E000014                       ! Read mem CSR5 (clearrs nmi fault)
DEPOSIT PHY/I 3E000008 F0000000             ! Clear RDS error bits in mem CSR0
DEPOSIT PHY/L 3A000000 2000                 ! Clear interlock busy in NBW1FSR
EXAMINE PHY/L 3A000808                       ! Read NBW1CLR (clears interrupts)
DEPOSIT PHY/L 38000000 2000                 ! Clear interlock busy in NBW0FSR
EXAMINE PHY/L 38000808                       ! Read NBW0CLR (clears interrupts)
EXAMINE PHY/L 3E000014                       ! Read mem CSR5 (clears NMI fault)
DEPOSIT R0 %x21                             ! BDA boot device type code
DEPOSIT R1 %x15                             ! Boot dev bus adr:<3:0>=BI node #,<6:4>=BI
!DEPOSIT R1 %x44                            ! Boot dev bus adr:<3:0>=BI node #,<6:4>=BI #
DEPOSIT R2 0                                ! <31:24>=optional controller letter specifier
DEPOSIT R3 %d100                            ! Unit # of drive, decimal radix
DEPOSIT R4 0                                ! Not applicable
DEPOSIT R5 %x81                             ! Software boot control flags and
                                            ! root directory @ in <31:28>. Defaults
                                            ! to sys0 or unrooted directory
                                            ! R5: %x81 - don't test mem,sysboot
                                            ! R5: %x87 - don't test mem,sysboot,breakpoints
                                            ! R5: %xA7 - don't test mem,sysboot,all break
                                            ! R5: %x1 - sysboot
                                            ! R5: %x0 - normal
```

TTB-0268

Example 2–5   BDABOO.CMD Command File

```
IF P1 THEN DEPOSIT R3 %P1              ! override above if given in boot cmd
IF P2 THEN DEPOSIT R5 %P2              ! ''
EXAMINE R3                             ! Display disk number
EXAMINE R5                             ! Display R5 parameter
FIND/MEM                               ! Find 64kb of memory; set cold start bit
IF NOT STATUS THEN ABORT               ! Boot if find was successful
EXAMINE SP                             ! Show address of working memory + %X200
LOAD /START= @ VMB.EXE                 ! Load VMB into good mem + %X200
!
!examine phy/1 3e94                     !patch vmb
!deposit .  00400011                    !patch vmb
EXAMINE SP                             !patch vmb
!
START @                                ! Start executing VMB
```

TTB-0269

Example 2–5 (Cont.)   BDABOO.CMD Command File

2.2.3.4 System Initialization and Autoboot

The automatic startup procedures initialize the system and then automatically boot the system using the command file AUTO_SYSINIT.CMD. The command file is shown in Example 2–6.

```
!+
! AUTO__SYSINIT.CMD - For a VAX xxxx system
! REV 1.0
!
! This SYSINIT.CMD has been specifically tailored
! for automatic startup operation.
! We suggest that you do not edit this file unless
! you wish to load USER WCS.
!-
!----------------------------------------------------------------------------------------------------------------------
DISABLE ERROR                              ! If error, abort
CONSOLE                                    ! Set command level
SET TERMINAL LOCAL                         ! Set up logging
SHOW TIME                                  ! Put time stamp in logfile
!                                          ! Display current power state
POWER STATUS
!
IF POWER__ON THEN GOTO 100
IF NOT AUTO__POWERON THEN ABORT

                                           ! If power is not on and
                                           ! AUTO__POWERON keyswitch not enabled
                                           ! exit system initialization procedure.
!
DTEXT "*** Turning ON System Power ***"
POWER ON                                   ! keyswitch is enabled, turn on power.
       IF NOT STATUS THEN ABORT            ! Abort if POWER ON failed
                                           ! If power is not on and AUTO__POWERON
100:
!
SET DEFAULT HEX
INITIALIZE/HARDWARE                        ! Init to base hardware level
SET CLOCK NORMAL                           ! Set clock period
!
VERIFY MODULE-PLACEMENT                     ! See if any CPUs are there
       IF NOT STATUS THEN ABORT            ! If no CPUs, quit.
!
SELECT CPU ALL                             ! For whatever CPUs are there, ...
set primary 0
set next__primary 0
set cpu 0
if cpu__available then goto 100
set primary 1
set next__primary 1
set cpu 1
if cpu__available then goto 100
set primary 2
set next__primary 2
set cpu 2
if cpu__available then goto 100
set primary 3
set next__primary 3
set cpu 2
if cpu__available then goto 100
dtext "*** No CPUs Available ***"
abort
100:
!
SENSE REVISION CPU                         ! Read CPU board revisions
```

TTB-0270

Example 2–6   AUTO_SYSINIT Command File

```
!+
! Power up diagnostics are in the command file POWERUP.CMD
! located in directory [PSTAR]. To inhibit power up diagnostics
! ENABLE QUICK-BOOT.
IF NOT QUICK-BOOT THEN @[PSTAR]POWERUP.CMD
!-
CONSOLE
SET CLOCK ON
INITIALIZE/KNOWN-STATE                       ! Put all cpus in known state
SELECT CPU ALL                               ! Reset CPU selection
LOAD WCS/CPU=ALL UCODE.BIN                   ! Load ISP microcode,
        IF NOT STATUS THEN ABORT             ! Exit if any error is encountered
LOAD CACHECONTROL/CPU=ALL CCODE.BIN          ! Cache microcode,
        IF NOT STATUS THEN ABORT             ! Exit if any error is encountered
LOAD IBDECODER/CPU=ALL DRAM.BIN              ! IBOX decoder tables.
        IF NOT STATUS THEN ABORT             ! Exit if any error is encountered
SET CLOCK OFF                                ! Stop clocks to start microcode
START/WCS/CPU=ALL CSM-INIT                   ! Start the CPUs!
        IF NOT STATUS THEN ABORT             ! Exit if something went wrong
WAIT/HALT/CPU=ALL                            ! Wait for CPUs to HALT
LOAD SDF/CPU=ALL SDFDEF.BIN                  ! Load microcode constants
        IF NOT STATUS THEN ABORT             !
!+
! Remove the comment flags on the following lines
! if you are loading user WCS
!
! LOAD UWCS/CPU=ALL USERWCS.BIN              ! Must come after SDF is loaded
! SET CLOCK OFF                              ! Clocks off to start microcode
! START/WCS/CPU=ALL CSM-INIT                 ! Restart microcode with user WCS
!       IF NOT STATUS THEN ABORT             !
!                                            !
!-
INITIALIZE/CPU/CPU=ALL                       ! Initialize hardware states & VAX regs
        IF NOT STATUS THEN ABORT             !
disable error
INITIALIZE/IO                                ! Reset I/O devices
INITIALIZE/NBI                               ! Initialize NBIs
REVISION/COMPARE                             ! Compare revisions
REVISION/SAVE                                ! Save revisions
SENSE REVISION MCL                           ! Read memory controller revision
SENSE REVISION NBI                           ! Read NBI adapter revision
disable error
write nbw1fsr 2000                           ! clear interlock busy
read nbw1clr                                 ! clear interrupts
write nbw0fsr 2000                           ! clear interlock busy
read nbw0clr                                 ! clear interrupts
read memcsr5                                 ! clear nmi fault
!
INITIALIZE/CPU/CPU=ALL                       ! Load SID and rev IPRs
        IF NOT STATUS THEN ABORT
```

TTB-0271

Example 2-6 (Cont.)   AUTO_SYSINIT Command File

```
!+
! Before memory can be initialized, a memory configuration file
! must have been created.
! This command will create the file memconfig.dat if it doesn't exist.
!-
INITIALIZE/MEMORY                                    ! Write 0's to all of memory
                                                     ! if memory were not backed up

!¢
! Initialize halt code on each CPU
!-
SET CPU 0
if cpu__available then DEPOSIT SDF 0CA 3             ! Powerfailure halt code
SET CPU 1                                            ! Powerfailure halt code
if cpu__available then DEPOSIT SDF 0CA 3
SET CPU 2                                            ! Powerfailure halt code
if cpu__available then DEPOSIT SDF 0CA 3
SET CPU 3                                            ! Powerfailure halt code
if cpu__available then DEPOSIT SDF 0CA 3
!¢
! Attempt to restart or boot the primary
!-
SET CPU PRIMARY                                      ! single-cpu operations from here on
CONSOLE/SAVE                                         ! Save console status
IF AUTO__RESTART THEN @[PSTAR]RESTAR.CMD             ! Attempt to restart if AUTO__RESTART
                                                     ! enabled
IF AUTO__BOOT THEN @[PSTAR]DEFBO0O0.CMD              ! If AUTO__RESTART disabled, and
                                                     ! AUTO__BOOT enabled attempt to boot
                                                     ! Otherwise, EXIT.
CLEAR RESTART-FLAGS                                  ! Clears WARM and COLD restart flags
!
!End of SYSINIT.CMD
show el 1
disable echo
```

TTB-0272

Example 2–6 (Cont.)   AUTO_SYSINIT Command File

2.2.3.5 System Power Down

An intentional powerdown is initiated at the console by entering:

PS–CIO–0> POWER OFF

The MPS power regulators are turned off in the sequence indicated in Table 2–3.

WARNING
The POWER OFF command *does not* eliminate
all power from the VAX 8820/8830/8840 system.
Primary ac power and some dc power are still
present within the system.

Table 2–3 Module Power Supply Turn off Sequence

Order	Module	Power Application
1	D1,D2	–2.0 V to NMI devices (CPU & memory)
2	E1,E2	–5.2 V to NMI devices (CPU & memory)
3	F2	+5.0 V, +12 V, –12 V, –5.2 V, –2.0 V to BI
4	C2,C1	+5.0 V to NMI devices (CPU & memory)
5	B2	+5.0 V to memory
6	J	+300 V to 4 MPS cardcage 300 V buses

2.2.3.6 Powerfail and Standby Modes

Power failure recognition by the environmental monitoring modules in the power system will result in an interrupt to the VAX 8820/8830/8840 system via the console interface. The EMM units will be directed to shut off the MPS power units in the sequence indicated in Table 2–4. This action can also be initiated by the operator at the system console by entering:

PS–CIO–0>POWER STANDBY

Table 2–4 Power Fail and Standby Modes MPS Turn Off Sequence

Order	Module	Power Application
1	D1,D2	–2.0 V to NMI devices (CPU & memory)
2	E1,E2	–5.2 V to NMI devices (CPU & memory)
3	F2	+5.0 V, +12 V, –12 V, –5.2 V, –2.0 V to BI
4	C2,C1	+5.0 V to NMI devices (CPU & memory)

WARNING
The POWER STANDBY and powerfail modes
do not **eliminate all power from the VAX**
8820/8830/8840 system. Primary ac power and
some dc power are still present within the system,
as well as backup power in backup mode and
standby power in standby mode.

When the system is in STANDBY or powerfail mode, power is present within the system. In STANDBY mode the J regulator is active and supplies 300 V dc to the four MPS 300 V dc buses, and the B2 regulator module (+5.0 Vdc) is active. In powerfail mode the system BBU unit is active and supplies 300 V dc to the four MPS 300 V dc buses, and the B2 regulator module (+5.0 V dc) is active.

2.2.3.7 Warm Restart
A warm restart uses the same procedures as the power–on sequence for turning on the system power. The
SYSINIT.CMD file exits to the restart command file RESTAR.CMD. A restart parameter block (RPB) must be
available for a warm restart, since an intact RPB indicates that memory data is valid. If a warm restart is not
possible, the VAX 8820/8830/8840 system executes the default boot command file DEFBOO.CMD. Example 2–7
shows the RESTAR.CMD file.

```
! RESTAR.CMD - For a VAX xxxx System
! REV 1.0
!
!This command file is invoked after a HALT, if a reboot has not been
!requested by the VAX, and after power failure recovery if AUTO_RESTART
!is enabled.
!
!The following tests are required:
!
IF NOT AUTO_RESTART THEN IF AUTO_BOOT THEN @£TAR°FBOO
                            ! Execute defboo if
                            ! auto_restart is not enabled but auto_boot is.
IF NOT AUTO_RESTART THEN abort
                            ! if neither auto_restart nor auto_boot are
                            ! enabled then remain halted
!                           ! Set up logging
SET TERMINAL local
SET DEFAULT HEXADECIMAL
!
!
EXAMINE SDF 0CA              ! Retrieve HALT code
DEPOSIT AP @                 ! Put halt code into the AP
EXAMINE PC                   ! Retrieve old PC
DEPOSIT R10 @                ! Put old PC into R10
EXAMINE PSL                  ! Retrieve old PSL
DEPOSIT R11 @                ! Put old PSL into R11
INITIALIZE/cpu               ! Initialize processor.
EXAMINE AP
DEPOSIT SDF 0CA @
DEPOSIT R0 0                 ! Clear unused register.
DEPOSIT R1 0                 ! Clear unused register.
DEPOSIT R2 0                 ! Clear unused register.
DEPOSIT R3 0                 ! Clear unused register.
DEPOSIT            R4 0
! Clear unused register.      DEPOSIT
R5 0                         ! Clear unused register.
enable error
FIND/RPB                     ! Locate restart parameter block and put addr + 200
                             ! in SP and contents of RPB+4 into PC
IF NOT STATUS THEN IF AUTO_BOOT THEN @[PSTAR]DEFBOO
                             ! If an RPB was not found and
                             ! auto_boot is enabled go try to reboot
IF NOT STATUS THEN abort
                             ! If an RPB was not found and auto_boot is
                             ! Disabled then remain halted
disable error                ! Get start address
EXAMINE PC
START @                      ! go
```

TTB-0273

Example 2-7   RESTAR.CMD Command File

2.3 OPERATOR/CONSOLE INTERACTION

The console, a MicroVAX II system (in the pedestal configuration), is used to initialize, operate, and shut down the VAX 8820/8830/8840 system. The console prompts and commonly used command language statements are described in this section.

2.3.1 Console Command Language Display Prompts

Table 2–5 describes Console Command Language prompts that may appear on the local and remote display screens.

Table 2–5   Console Command Language Prompts

Prompt	Description
>>>	MicroVAX II power is on and MicroVAX II self–tests have been completed. The MicroVAX II system is booted from this prompt.
PS>	Indicates that access to the VAX 8820/8830/8840 system hardware has not been attempted yet.
PS–OFF>	Indicates that the EMM and the QC module have been verified, but that system power is turned off.
PS–HW–n>	Indicates that system power is on and that the console connected CPU is number n.
PS–CIO–n>	The system console is in console I/O mode with Console Support Microcode (CSM) loaded into the connected CPU which is number n.
PS–OS–n>	The console has returned to console I/O mode from console program mode because a CTRL P was typed at the terminal. CPU n is the connected CPU.

2.3.2 Console Commands

The Console Command Language (CCL) is the interface between the operator and the control and monitoring capabilities of the console subsystem. Table 2–6 provides a short list of frequently–used console commands. Refer to the *VAX 8820/8830/8840 Console User's Guide* (AA–FJ84A–ME) for a detailed description of all console commands.

Table 2–6 Frequently Used Console Commands

Command	Description
@file_spec	Opens a specified file and processes the records within the file as commands. (The default file extension is .CMD.)
BOOT [device_name]	Executes the command file device_nameBOO.CMD. If no device name is specified, the file DEFBOO.CMD is used.
CLEAR item	item can be any one of the following: ACCUMULATOR RESTART_FLAGS SCREEN SOMM SWITCHES TOMM
CONSOLE	Returns the console to top command level from a subcommand level such as DIAG or EMM. This command always returns to or stays at the top command level to avoid an inadvertent exit from the console program.
CONTINUE	In console mode, CONTINUE starts macrocode execution at the address currently present in the macro program counter (PC). Processor initialization is not performed. The processor state is changed from HALTED to RUNNING. In diagnostic mode, CONTINUE continues the execution of the diagnostic that just halted on error, or was interrupted by CTRL/C. Execution will resume at the point at which it was stopped.
DEPOSIT location	Replaces the contents of a specified location. If no data size or address space qualifiers are specified, the defaults will be those used in the most recent EXAMINE, DEPOSIT or SET DEFAULT command. Initially the defaults are PHYSICAL, LONGWORD, and address 0. If no address is specified, "+" is assumed.
DIAGNOSE	Causes the console to enter the Diagnostic Support mode. The CONSOLE or EXIT commands will return to console mode.
DISABLE item	Establishes console parameters that are retained across power failures, restarts, and reboots. The parameters established by DISABLE are changed by the ENABLE command. Item can be: AUTO BOOT AUTO POWERON AUTO RESTART CONFIRMATION CPU cpu_list ECHO ERROR LOG PRINTER TEXT_OUTPUT
ENABLE item	Establishes console parameters that are retained across power failures, restarts, and reboots. The parameters established by ENABLE are changed by DISABLE.

Table 2-6 (Cont.)   Frequently Used Console Commands

Command	Description
EXAMINE location	Examines the contents of a specified location. If no data size or address space qualifiers are specified, the defaults will be those used in the most recent EXAMINE, DEPOSIT, or SET DEFAULT command. Initially the defaults are PHYSICAL, LONGWORD, and address 0. If no address is specified, "+" is assumed.
EXIT	Terminates the current subprogram and returns control to the console program. In EMM or diagnostic mode, returns to CONSOLE mode.
FIND [/CPU:cpu] [/MEMORY] [/RPB]	The console will look for good memory or a restart parameter block for the indicated CPU. The defaults items are the connected processor and the restart parameter block.
FIND /MEMORY	Requests Console Microcode (CSM) to search main memory starting at address 0 for a page-aligned 64-kbyte block of good memory.
FIND /RPB	Requests CSM to search physical memory for a valid restart parameter block. The search leaves the contents of memory unchanged.
HALT [CPU:cpu_list]	Attempts to stop macrocode execution in all of the specified processors, beginning at the next macro instruction boundary. The state of each of these processors changes from RUNNING to HALTED. If a processor is malfunctioning, it may never try to execute the next macro instruction and therefore never halt. The HALT command produces a message reporting which processors are successfully halted and the contents of the PC for each one of them. If a processor is already halted when the HALT command is issued, a warning message is returned.
HELP [help_topic]	Help_topic is a keyword about which to provide help information. This command allows you to display the function, format, arguments, and switches of any console command.
IF	Allows a command to be conditionally performed. If the condition specified is met, the remainder of the line is performed. If the condition specified is not met the remainder of the line is ignored.
INITIALIZE [/CONSOLE] [/CPU:cpu_list] [/IO] [/MEMORY] [/CREATE_CONFIGURATION_FILE]	Sets the appropriate part of the system to a defined initial state.
LOAD [filename]	In console mode, LOAD loads data into the specified space. The data comes from the file specified. If no optional qualifiers are specified, the command defaults to loading memory starting at physical address 0. In diagnostic mode, the specified diagnostic program is loaded into console memory.

Table 2–6 (Cont.) Frequently Used Console Commands

Command	Description
MICROSTEP [/CPU:cpu_list] [count]	Bursts the clock for the specified number of microcycles (one by default), displays the micro–PCs for all affected CPUs, and enters space bar step mode. Pressing the space bar at this prompt causes another single–cycle micro– cycle to be executed and the new micro–PCs to be displayed. Any non–space character exits the space bar step mode and becomes the first character of a new console command line.
NEXT [/CPU:cpu_list] [count]	Executes macroinstructions for the specified count (one by default), displays the PC for all affected CPUs, and enters space bar step mode. Pressing the space bar at this prompt causes another macroinstruction to be executed and the new PCs to be displayed. Any non–space character exits the space bar step mode and becomes the first character of a new console command line.
POWER (ON I OFF I STANDBY I INITIALIZE I STATUS)	Changes the state of the power system by issuing commands to the EMM. The power may be in one of four states:
	1. OFF – All CPU regulators and memory power are turned off.
	2. ON – All CPU regulators and memory power are turned on. The system can function.
	3. STANDBY – CPU power is turned off, memory power remains on.
	4. INITIALIZE – Communication with the power system is established and default parameters loaded.
	See Tables 2–2, 2–3, and 2–4 for the sequence in which power goes to the on, off, standby, or powerfail states.
	STATUS – Report on the current power state.
QUIT	Terminates the console program and returns to the MicroVMS prompt.
REPEAT command	Sets an internal flag indicating that the command line be continuously executed. REPEAT mode is terminated by typing CTL/C.
RUN	In diagnostic mode, the specified diagnostic program is loaded into Console memory and started.
SET	Sets parameters for clocks, CPU defaults, EMM power margins, etc.
	CLOCK – ON, OFF, SLOW, FAST, NORMAL
	CPU:cpu
	MARGINS (regulator) [NORMAL I HIGH I LOW]
	SOMM upc_address – loads the micro–pc into the micromatch register.
	TERMINAL (LOCAL, NOLOG)
	TOMM upc_address – Loads the micro–pc into the micromatch register.

Table 2–6 (Cont.) Frequently Used Console Commands

Command	Description
SHOW	Displays information about certain console and system parameters such as:

CLOCK	CONSOLE
CPU	DEFAULTS
ENABLES	LOGGING
MARGINS	MEMORY
POWER	STATUS
T1_TEMPERATURE	T2_TEMPERATURE
T4_TEMPERATURE	T4_TEMPERATURE
T5_TEMPERATURE	T6_TEMPERATURE
T7_TEMPERATURE	T8_TEMPERATURE
TERMINAL	TIME

Command	Description
START	Operates differently depending upon the mode in which the console is running.

Console mode:

START/MEMORY [address]   Starts execution of a program on the specified processor at the specified address.

START/WCS [address]   Starts execution of microcode in the WCS at the WCS address specified (between 0 and 3FFF hex).

Diagnostic mode:

START   Starts a diagnosic program which is already loaded.

[PASSES=passcount]   [/SECTION=section_name

[TEST=first_test[:last_test]]   [/WCS address]

Command	Description
VERIFY MODULE_PLACEMENT	Checks that each CPU module is in the correct slot. If all the modules for a CPU are correctly placed, the CPU is marked present. If not, the CPU is marked not present and the missing or misplaced modules are reported.

2.4 CONSOLE DISPLAYS AND LOG FILES

The console display for the VAX 8820/8830/8840 system is the console subsystem's VT320 video terminal and/or the remote console terminal. Console sessions at both display terminals are recorded in the console log file. The console log file is maintained on the console storage device.

The console display is composed of console subsystem prompts, operator commands, and console subsystem responses. The text which is echoed back from the system while the user is in program I/O mode is also shown (some entered information, such as passwords, is not echoed by the system and is therefore not displayed). Example 2–8 shows an example of character flow during a local operation.

```
PS-HW-0>VERIFY MODULE_PLACEMENT
PS-HW-0>SHOW CPU
   Current CPU = 0, Current Primary = 0, Next Primary = 0
   Auto — Poweron:disabled, Boot:disabled, Reboot:disabled,
   Restart:disabled
   CPU 0 = Enabled, Present, Available
PS-HW-0>SHOW STATUS
   Power is On, BBU is Disabled
   Clock is Off, Clock period = 70 ns

      CPU    0   1   2   3
   Enabled = Y   Y   Y   N
   Present = Y   Y   Y   Y
   Select  =
   Avail   =
   Status  =
      Current CPU = 0
      Current Primary = 0
      Next Primary = 0
PS-HW-0>SEL CPU 0
PS-HW-0>@SYSINIT
PS-HW@> SET EL 1
PS-HW@> !+
PS-HW@> !  SYSINIT.CMD — For a VAX xxxx system
PS-HW@> !  REV 1.0
PS-HW@> !
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                     (runs the SYSINIT.CMD file)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
PS-CIO-0>
```

TTB-0368

Example 2–8   Console Display Showing a Local Operation

2.4.1 System Log File

The console can maintain a log file of console display activity on the console subsystem storage device (disk). The log file is kept so that activity may be reviewed at a later time. All of the data that appears on the display screen during console–mode operations, as well as the output from the VAX 8820/8830/8840 system, is saved in the log file.

Console entries made during program mode are echoed from the system and are saved in the log file. Entries that are not echoed are not saved.

2.4.1.1 Starting the log file

When starting to log console activity, the operator may specify either system format (compact and comprehensive), or ASCII format (less compact and abbreviated). The default format is system format. The ASCII–format log file is PCON.LOG while the system–format log file is PCONLOG.SYS. The operator may also enable or disable logging. The commands to select the format and to enable/disable logging are:

        PS–CIO–0> ENABLE LOG/SYSTEM
        PS–CIO–0> ENABLE LOG/ASCII
        PS–CIO–0> DISABLE LOG/SYSTEM
        PS–CIO–0> DISABLE LOG/ASCII

2.4.1.2 Size and Number of Log Files

The size and versions of log files which are kept can be specified. The size is specified in blocks with the default size being 1000 blocks. The number of versions to be kept can vary from 1 to 99; the default number of versions is 5. The commands used to specify the size and number of log files are:

        PS–CIO–0> SET LOG/SIZE=100
        PS–CIO–0> SET LOG/KEEP=10

The operator may examine the status of logging with the following console command:

        PS–CIO–0> SHOW LOG

2.4.1.3 Examining the Log File

The operator may examine any of the ASCII format log files by entering "TYPE log_file" or by editing the file with the EDT editor. The system–format log files may not be examined directly, but may be converted to ASCII format files and then examined. The CONVERT command is used to convert a log file to an ASCII–file. The default condition is the current log file. Use:

        PS–CIO–0> CONVERT log_file
        PS–CIO–0> CONVERT/CURRENT

2.5   THE STAR PROCESS

The STAR process allows a MicroVAX II user terminal to display graphic representations of system status, become the local console, or become a system user terminal.

The STAR process becomes active and handles a user terminal when the person at the user terminal logs into the VAX/VMS Operating System on the MicroVAX II and issues the command "$STAR". The local console terminal must be logged in and running the POLARIS program.

When the user terminal runs the STAR process it has the ability to cause special graphic system status displays such as module utilization and power subsystem component locations to appear on the terminal. The user terminal must be a VT3xx, a VT2xx, or a VT1xx with the advanced video option (up to 132 characters per line). The following commands can be issued when the user terminal is running the STAR process:

- SET TERMINAL USER

- SET TERMINAL CONSOLE

- SET TERMINAL NOCONSOLE

- EXIT

The user terminal may become a remote console terminal by issuing the command "SET TERMINAL CONSOLE". The prompt for the remote terminal is PS–CC–HW–n> or PS–CC–CIO–n>. PS–CC–HW–n> indicates that the power is on but that the console support microcode (CSM has not yet been loaded into the CPU control store. PS–CC–CIO–n indicates that CSM has been loaded, and the CPU is ready to perform operations such as examining memory locations. All local console terminal commands may be given from the remote terminal except for commands which change power state.

The flowchart shown in Figure 2–4 shows the user terminal states.
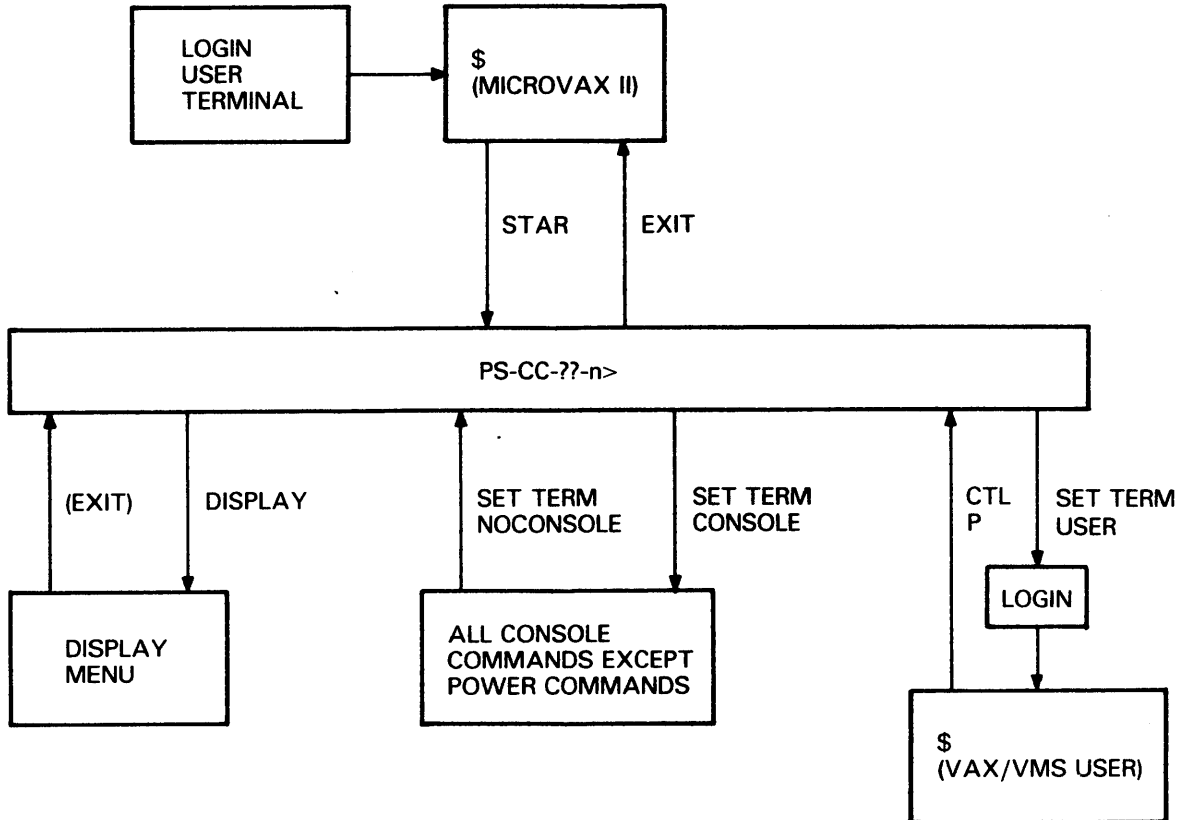
TTB-0315

Figure 2–4   MicroVAX II User Terminal State Flowchart

CHAPTER 3
FUNCTIONAL DESCRIPTION

3.1 OVERVIEW
The VAX 8820/8830/8840 system configuration, shown in Figure 3–1 consists of:

- The console subsystem

- Two CPUs for the VAX 8820 system, three CPUs for the VAX 8830 system, and four CPUs for the VAX 8840 system

- The memory subsystem with one to eight 64–Mbyte arrays

- The power system

- One, two, or three VAXBI interconnect adapters

The VAX 8820/8830/8840 system is interconnected through a synchronous backplane bus called the memory interconnect (NMI). The NMI provides the system with a communications path between the CPU(s), memory, and adapters that connect to the VAXBI interconnects. The NMI is physically separated into two parts: one part in the CPU cabinet connected via a cable and two NMI Bus Window (NBW) modules to the second part in the memory cabinet.

> **NOTE**
> The VAX 8820/8830/8840 system must contain at least one VAXBI channel. It can be expanded to contain up to six VAXBI channels (using expansion cabinets).
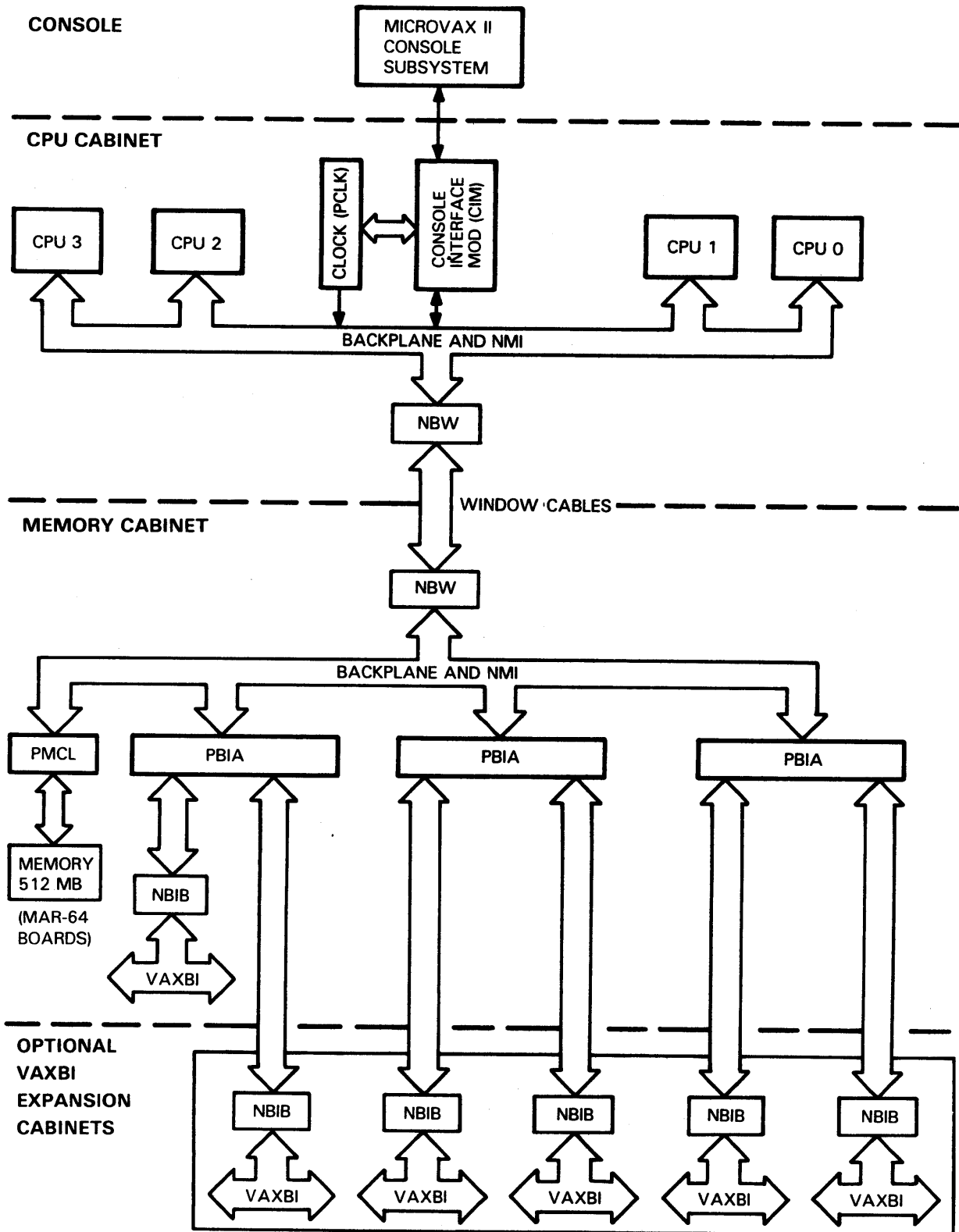
Figure 3–1   VAX 8820/8830/8840 System Configuration

3.2 CONSOLE SUBSYSTEM

The console subsystem, shown in the simplified block diagram in Figure 3–2, controls the power sequencing, loading of control stores, and general operation of the VAX 8820/8830/8840 system. The console subsystem is composed of hardware and software: The hardware part of the subsystem is grouped around the MicroVAX II, which is the "brains" of the subsystem. The software part is made up of the MicroVMS operating system and the console applications procedures which implement the console commands.
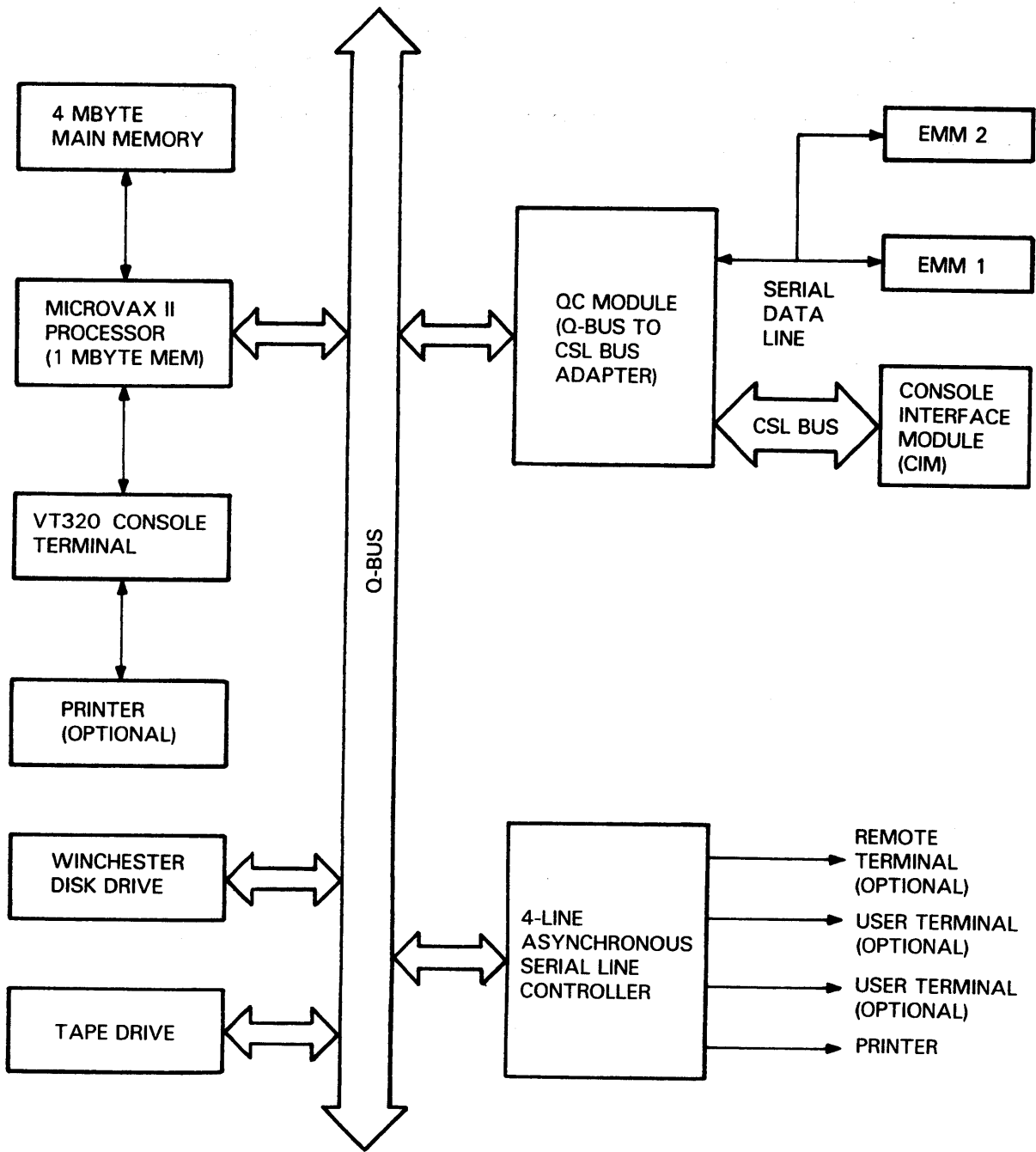
Figure 3–2   Console Subsystem Block Diagram

TTB-0275

3.2.1 Console Subsystem Hardware

The console subsystem hardware consists of a MicroVAX II system, a QC module, a cable, and a CIM module. The MicroVAX II system communicates with the QC module via the Q–bus. The QC module then communicates with the CM module and the EMM module (one in the CPU cabinet and one in the memory cabinet) over a 4.5–meter (15–foot) cable. The CIM module communicates with the IBox in each of the four CPUs, the VBus, and the PCLK module.

3.2.1.1 The MicroVAX II System

The MicroVAX II system, used in its pedestal configuration, has a 32–bit VAX processor, 5 Mbyte of main memory, a VT320 terminal, a Winchester fixed disk, a removable tape drive, and a communications interface to a printer and optional user terminals (up to three). Refer to the *MicroVAX II Owner's Manual* (AZ–GL1AA–MN) and the *MicroVAX II Technical Manual* (AZ–GL0AA–MN) for further information.

The MicroVAX II processor is contained on a single quad–height board which resides on the MicroVAX Q–bus. It includes the MicroVAX 32–bit central processor, memory management, floating point, one Mbyte of onboard memory, a Q–bus interface, a Q–bus map for DMA transfers, an interval timer, boot and diagnostic facilities, a console serial line, and a time–of–year clock with battery backup.

The console has one Mbyte of memory on the processor module and four Mbytes of memory on a separate memory board.

The VT320 terminal consists of a detachable, low–profile keyboard and a monochromatic video monitor. It includes 15 programmable function keys, multi–national character set support, and plain–language setup menus. The monitor includes a built–in bidirectional printer port. Characters are displayed as a 15–by–12 cell (80–column) matrix or a 9–by–12 (132 column) matrix. The 14–inch non–glare flat screen CRT is available with white, green, or amber phosphor. A terminal/host programmable 25th status line can be optionally used to display system or host status. Advanced video features include double–width and double–height characters, blinking, bold, underline, and normal or reverse video.

The fixed disk drive is an RD53 5–1/4 inch, 71 Mbyte, Winchester drive. It is mounted in the MicroVAX II pedestal cabinet and is controlled by the RQDX3 controller. It holds console subsystem software, system diagnostic test programs, system boot software, and a system message log.

The TK50 removable tape drive, a 95 Mbyte removable–media streaming tape, is suitable for distribution of console software updates and archiving data. A single TK50 tape is capable of holding the contents of an entire RD53 disk. The TK50 is mounted in the MicroVAX II cabinet; its interface requires a half–slot of Q–bus space. Refer to the *TK50 Tape Drive Subsystem User's Guide* (EK–OTK50–UG) for further information.

The optional remote console uses a VT2xx or VT3xx terminal and keyboard, controlled via the DZQ11 asynchronous line controller. This console allows remote diagnosis and control of the VAX 8820/8830/8840 system .

The DZQ11 asynchronous serial line controller is a four–line, asynchronous multiplexer that provides local or remote interconnect to EIA RS–232/423 devices. The DZQ11 operates at program–selectable speeds up to 9600 baud in full–duplex, with limited modem control on each line. The DZQ11 is used in the VAX 8820/8830/8840 system to support an LA75 printer and three user ports (optional). Any one of the three user ports can be used by a terminal (remote or local) which can be logged in as a local or remote (via a modem) operator's console.

An optional printer may be connected to the console subsystem through a serial printer port on the back of the MicroVAX II processor. The printer is not a standard part of the console subsystem.

3.2.1.2 The Q–bus Control Module (QC, M7553)

The Q–bus control module, commonly called the QC module, is a quad–height Q–bus module which interfaces the MicroVAX II Q–bus to the VAX 8820/8830/8840 system EMM and to the CSL bus. The QC module represents a nexus on the Q–bus bus corresponding to 2 kbyte of I/O addresses beginning at 764000 (octal).

The system side of the QC module has two interfaces: an RS–422 serial line and the CSL bus. The RS–422 serial line is used to communicate with the two EMM modules, one in the CPU cabinet and another in the memory cabinet. The CSL bus is used to communicate with the CIM module on the CPU cabinet backplane.

A 4.5–meter (15–foot) cable is used to carry both the CSL bus and the RS–422 serial line from the console to the VAX 8820/8830/8840 system.

3.2.1.3 The Console Interface Module (CIM, F1032)

The console interface module, commonly called the CIM module, is located near the physical center of the CPU logic backplane. Its principal function is to enable the CPUs and the console to communicate via the CSL bus.

There are data paths to the instruction decoder modules of each CPU on the system side of the CIM module. These paths are used by the console to load system microcode, access system registers, and perform file transfers between the system and the console.

The CIM module provides the console access to the clock module (PCLK) so that the console may exercise some control over the system clock. The console may cause the clock to be stopped, started, stepped any number of A–clock cycles, or stopped at preselected program breakpoints.

The CIM module has access to the 32 channels of the system visibility bus (VBus). The VBus is designed to provide the console with access to selected latches within the CPUs for diagnostic purposes. Each channel is a serial path from one of the 32 modules of the four CPUs.

The CIM module can also access backplane revision information and the system serial number.

3.2.2 Console Subsystem Software

The console subsystem software operates in two modes: operator mode, where the console operator may control and monitor the CPU, and program I/O mode, where the console is used like a user terminal. The transient power–up and boot state may be considered to be a third mode of operation.

3.2.2.1 Power–up and Boot

When the machine is first turned on, power is first applied to the system power supply control logic (ILM) and the environmental monitor module (EMM), and then to the console subsystem. The EMM and the console subsystem perform self–test routines to verify their own proper operation.

3.2.2.2 Console I/O Mode

The console operator may use the console in console I/O mode to control and monitor the halted VAX 8820/8830/8840 system. The console I/O mode prompts are listed and described in Table 2–5.

3.2.2.3 Program I/O Mode

Program I/O mode can only be used under the VAX/VMS operating system, ULTRIX operating system, VAX Diagnostic Supervisor (VDS), etc. The console terminal emulates a user terminal while in this mode: The terminal passes all input to the CPU software and displays output from the CPU. The prompt for the program I/O mode is the prompt of the software running at the time, such as the DCL "$" prompt.

3.3 CENTRAL PROCESSING UNIT

The VAX 8820/8830/8840 system processor, shown in the block diagram in Figure 3–3, consists of three functional units and associated data transfer buses. The functional units and data buses are listed and described in Table 3–1.

```
                    ┌─────────────────────────────┐
                    │      CONSOLE SUBSYSTEM       │
                    └─────────────────────────────┘
                                   ↕
                    ┌─────────────────────────────┐
                    │  CLOCK AND CONSOLE INTERFACE │
                    └─────────────────────────────┘
    ┌──────────────────────────────────────────────────────────┐
    │  VISIBILITY BUS                      CLOCKS               │
    │                                                          │
    │              CONSOLE  DATA                               │
    │                                                          │
    │          ┌──────────────────────────┐                   │
    │   MICRO  │     INSTRUCTION BOX       │    CACHE          │
    │   WORD   └──────────────────────────┘    DATA           │
    │               IBOX                        BUS            │
    │               DATA                                       │
    │               BUS                                        │
    │          ┌──────────────────────────┐                   │
    │          │      EXECUTION BOX        │                   │
    │          └──────────────────────────┘                   │
    │              VA             CACHE                        │
    │   WBUS       BUS            ALU                          │
    │                            BYPASS                        │
    │          ┌──────────────────────────┐                   │
    │          │       CACHE BOX          │                   │
    │          └──────────────────────────┘                   │
    │                                              CPU        │
    └──────────────────────────────────────────────────────────┘
                              ↕
                    ⟨═══════ NMI ═══════⟩
```
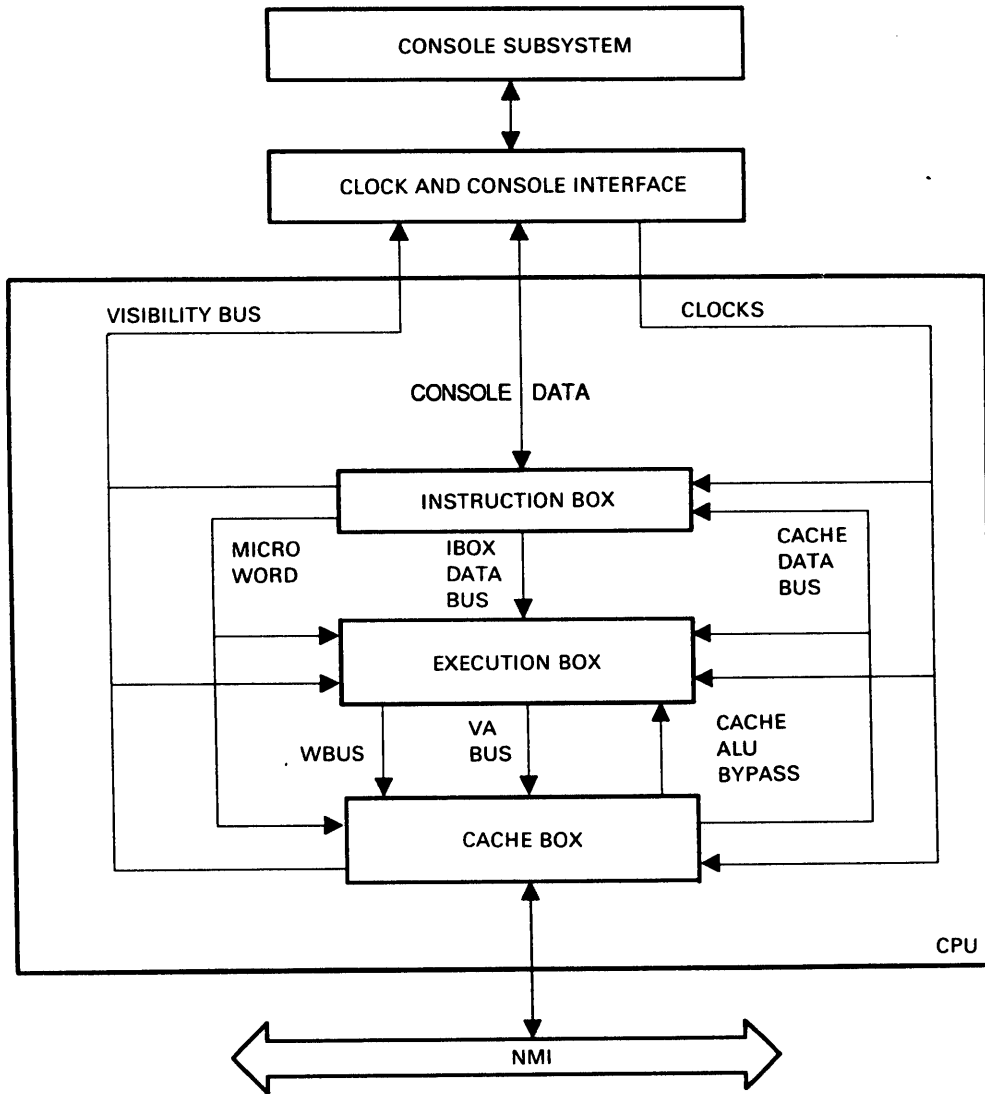
SCLD-78

Figure 3–3   VAX 8820/8830/8840 System Processor Block Diagram

Table 3–1 Functional Units and Data Buses

Functional Unit/Bus	Description
Instruction Box (IBox)	CPU microcode store and control. Consists of the writable control store (WCS), the decoder (DEC), and the sequencer (SEQ) modules.
Execution Box(EBox)	Processes data received from the cache and instruction boxes. Performs arithmetic, logical, and bit–shift operations. The EBox consists of the data slice modules (SLC0 and SLC1) and the shifter (SHR) module.
Cache Box (CBox)	Contains the cache, translation buffer, and the interface to memory and I/O. The cache is a 64–kbyte physical index, direct mapped and buffered write–through cache. The translation buffer is a 1K direct mapped cache of virtual–to–physical address translations. The CBox consists of the address data path (ADP) and cache control sequencer (CCS) modules.
Virtual Address Bus	Data path for transferring virtual addresses from the execution box to the cache.
Cache Data Bus	Data path from the cache to the execution box and instruction parser.
Write Data Bus	Data path for write data from the execution box to the cache.
Cache/ALU Bypass Bus	Bypass register data that is scheduled to be written into a register. The register does not have valid data yet.
Visibility Bus (VBus)	Slow–speed data bus that allows the console to access internally–latched data in the CPU modules.
Instruction Buffer Data Bus	Data path for transfer to the execution box. The data consists of byte, word, and longword address displacements, absolute addresses, and immediate data. Branch displacements and literals are also transferred over the instruction buffer data bus.

3.4 CLOCK MODULE (PCLK)

The clock module contains all of the hardware necessary to generate, control, and distribute the system timing to all system modules. Figure 3–4 shows a simplified block diagram of the clock generation logic on the clock module.
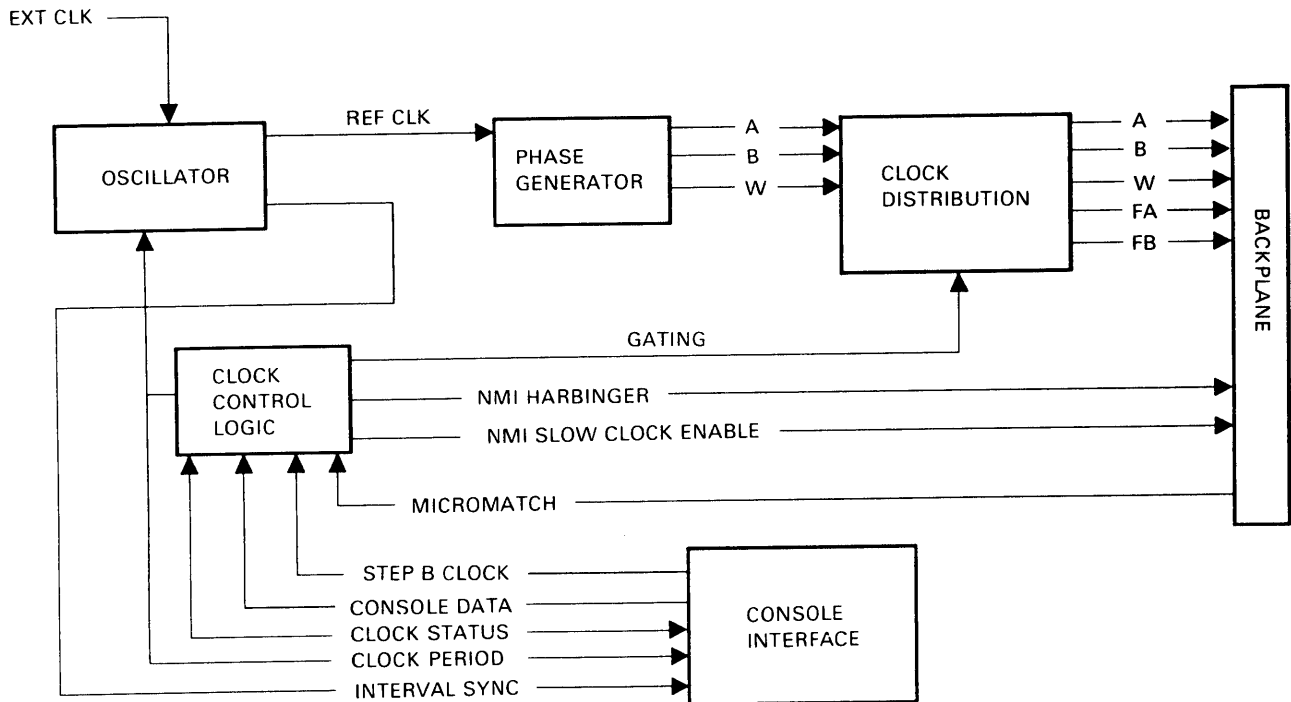
SCLD-83

Figure 3–4   Clock Generation Block Diagram

The basic timing source of the clock module is a 250–kHz oscillator and a phase–locked loop used by the phase generator to produce two non–overlapping clock phases.

The phase generator output consists of phases A, B, and W. The W phase signal is similar to the A phase, but longer in duration. The A and B clock phases are gated in the distribution logic by signals from the clock control logic to produce gated system clocks that can be started, stopped, or burst by the clock control logic.

The distribution logic also produces free–running (ungated) clocks. Table 3–2 lists and identifies the clock signals generated and distributed by the clock module.

Table 3-2 Clock Signals

Signal Name	Characteristics
A and B CLK	Main system clocks used by the CPU modules and the I/O adapters to sequence and synchronize operations.
W CLK	Free–running system clock used in RAM write operations of CPU modules SLC0 and SLC1.
FA and FB CLK	Free–running system clocks used by the memory controller and I/O adapters to sequence and synchronize operations; also used by the console interface to control and monitor the CPUs when clocks are stopped.

The console subsystem controls the clock generation logic with operator–initiated commands by means of the console interface. Three registers within the clock logic are used for control of the system clocks by the console; a fourth register provides status information for the console operator. The following registers are used:

- Clock control

- Clock period

- Burst count

- Timeout and status

Console control of the clocks allows the system operator to:

- Start and stop the clocks

- Burst the clocks

- Single–step the clocks

- Enable a clock stop on a micromatch

- Change the clock period

- Disable clock stalls

- Control the NMI timeout clock (NMI Slow Clock Enable)

3.5 MEMORY (MBOX)

System memory system (MBox) consists of a memory control logic (PMCL) module and one to eight MAR64 64–Mbyte memory array boards. Figure 3–5 shows a simplified block diagram of the MBox with one MAR64 array board.

NOTE
The standard VAX 8820/8830/8840 system main memory can be expanded to 512 Mbytes by installing MAR64 64–Mbyte memory array boards. This memory expansion upgrade can be effected by adding up to eight (total) MAR64 memory arrays.
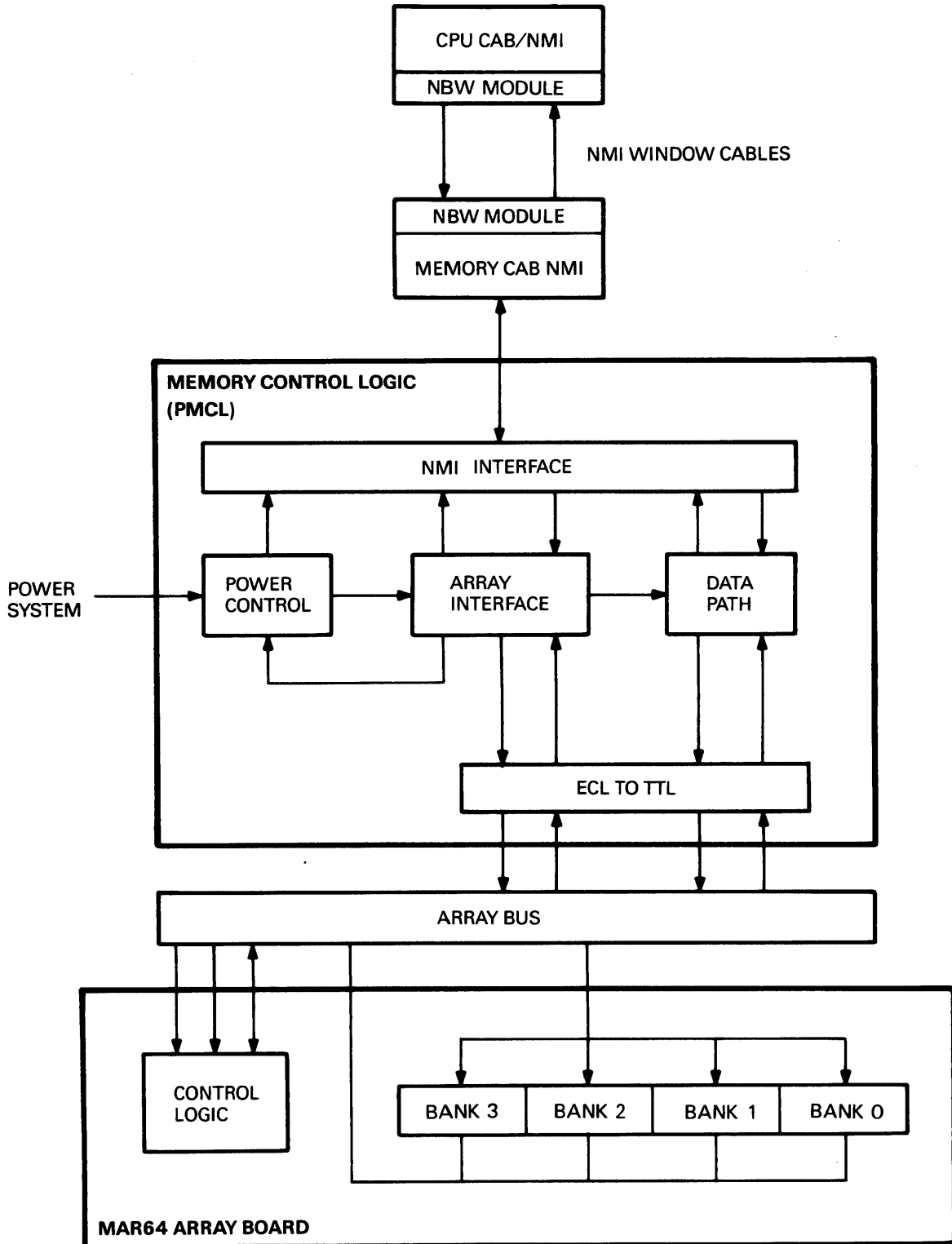
Figure 3–5   Memory Subsystem Block Diagram

3.5.1 Memory Control Logic

The PMCL provides control and a communications interface between the NMI and the memory array boards. The single PMCL has the capability of controlling a maximum of eight array boards, and can monitor operations on three arrays simultaneously. Table 3-3 describes the operations that the PMCL performs on the MAR64 array.

Table 3-3   PMCL Command Operations

Operation	Description
Longword Write	Writes a data longword and seven ECC check bits into one of the MAR64 array banks.
Longword Read	Reads a data longword and seven ECC check bits from one of the MAR64 array banks.
Octaword Read	Reads four data longwords and associated check bits from a MAR64 array.

3.6   SYSTEM BUSES

The VAX 8820/8830/8840 system has three system-level buses that provide a path for data transfers and status information.
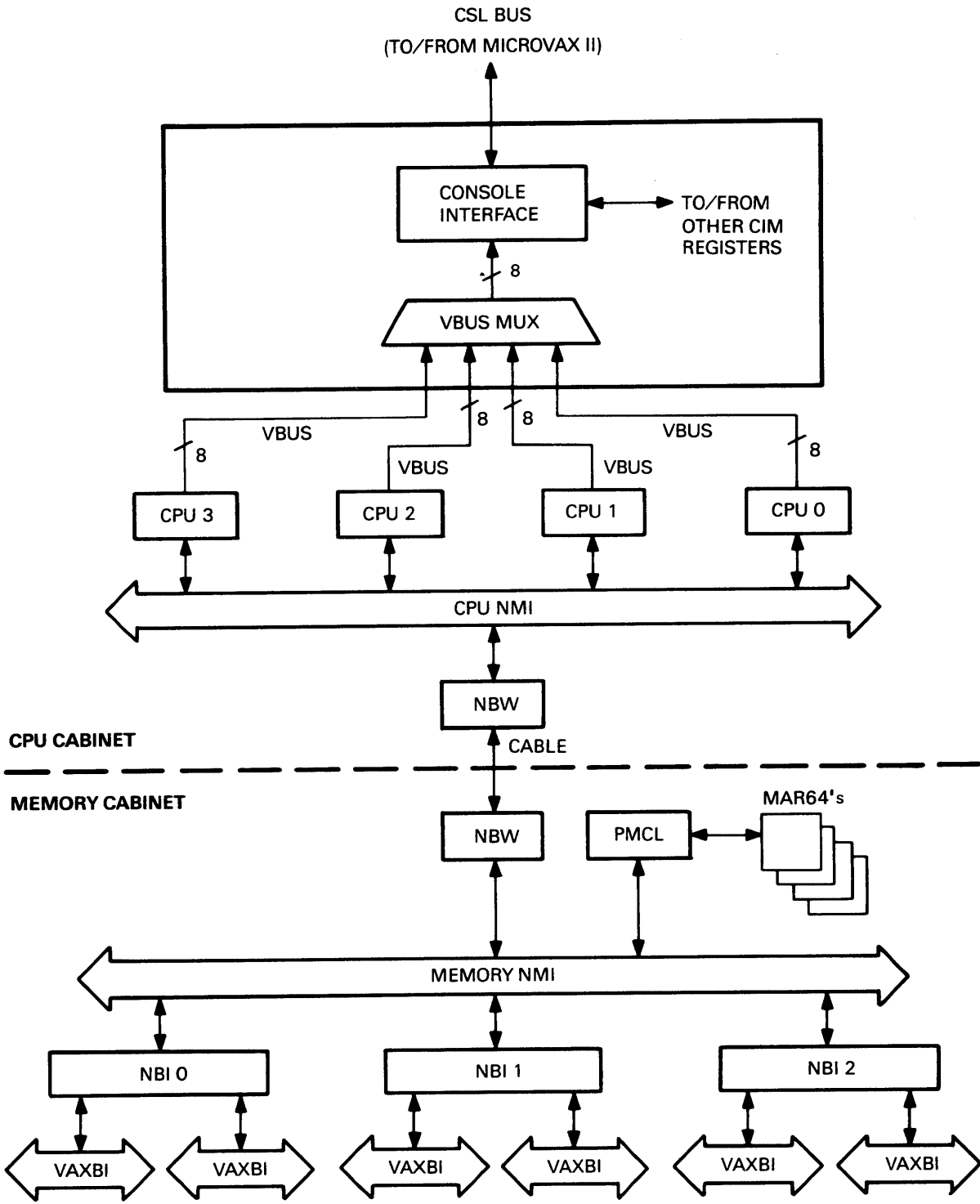
- VAX 8820/8830/8840 system memory interconnect (NMI)

- VAX backplane interconnect (VAXBI)

- Visibility bus (VBus)

The system memory interconnect (NMI) is a backplane bus that interconnects the major system components and allows transfer of data between the connected units. The NMI is separated into two physical parts: the CPU NMI and the memory NMI. The two NMIs are connected by two NBW modules and a cable. During normal operation it is not obvious to the CPUs that the NMI is in two parts.

The VAXBI interconnect is the I/O bus that provides a connection point between the system and external device adapters.

The visibility bus provides the console operator with diagnostic access to internal latch contents. The VBus is also used to perform error checks during initialization.

Figure 3-6 shows the major system buses.

Figure 3–6   System Buses

TTB-0277

3.6.1 Memory Interconnect (NMI)

The NMI, shown in Figure 3–7, is a synchronous backplane bus that interconnects the following major components of the VAX 8840/8830/8820 systems:

- 2, 3, or 4 CPUs (VAX 8820, 8830, or 8840 system)

- 2 NMI bus window modules (NBWs)

- Memory system controller (PMCL)

- NMI–to–VAXBI adapters (PBIA, up to 3)
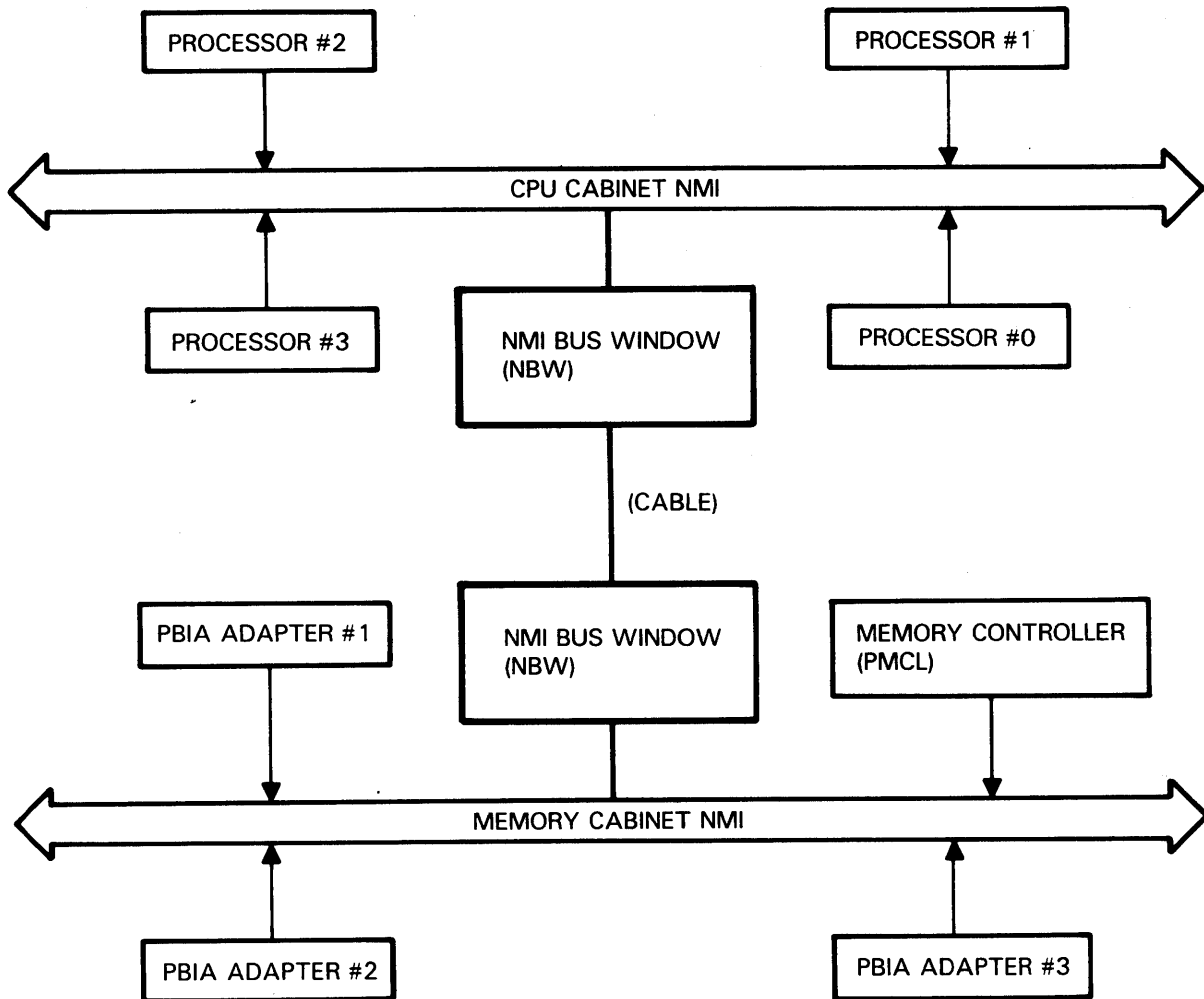
TTB-0278

Figure 3–7   NMI Block Diagram

Table 3–4 lists the primary functions performed by the NMI, and provides a brief description of each of the functions.

Table 3-4 NMI Primary Functions

| Function	Description
Write Transaction	Supports longword, quadword, and octaword write.
Read Transactions	Supports longword, octaword, and hexword read.
Memory Read/Write Operations	Allows the CPUs and I/O adapters to access memory through bus read/write transactions.
I/O Register Read/Write Operations	Allows the CPUs to access I/O registers in the memory, I/O adapters, and I/O devices through bus read/write transactions.
Interrupt Handling	Transmits interrupt requests generated by the memory and I/O adapters to the CPUs.
System Synchronization	Provides system clocks to all nexus (hardware blocks physically connected to the NMI).
System Initialization	Allows the console to initialize all nexus.
Power-Fail Warning	Provides ACLO and DCLO signals to all nexus.

3.6.2 VAXBI Interconnect

The VAXBI interconnect is the I/O bus for the system and is connected to the system through the NMI-to-VAXBI (NBI) adapter, as shown in Figure 3-8. Each NBI adapter can interface a maximum of two VAXBI interconnects and provides for a maximum of six VAXBI interconnects to the system when three NBI adapters are installed.
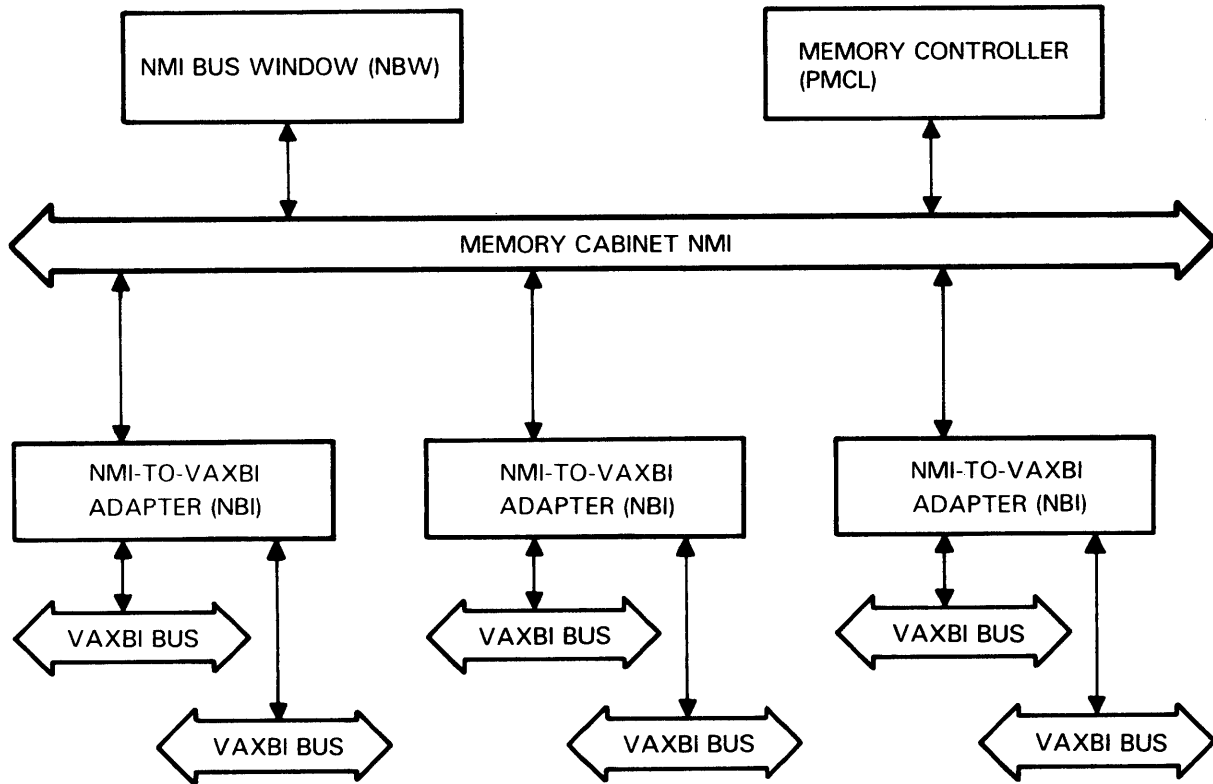
TTB-0279

Figure 3–8   VAXBI Block Diagram

The VAXBI is a 32–bit wide synchronous bus that interconnects a maximum of 16 VAXBI interfaces (VAXBI nodes) with logical addresses of 0 through F (hex). The address of a node is determined by an ID plug inserted on the backplane.

NOTE
Address 0 usually is reserved for the NBI node.

The remaining nodes consist of I/O device controllers and bus adapters interfacing the system's I/O devices to the VAXBI interconnect.

Table 3–5 identifies and describes the basic functions performed by the VAXBI interconnect.

Table 3–5 VAXBI Functions

| Function	Description
Memory Read/Write Operations	Allow DMA transfers between an I/O device on the VAXBI and main memory through bus read/write transactions.
I/O Register Read/Write Operations	Allows the CPUs to access I/O registers in the I/O devices on the VAXBI through bus read/write transactions originating on the NBI.
Interrupt Handling	Enables I/O devices on the VAXBI to interrupt a CPU through bus INTR transactions directed to the NBI node.
System	Allows nodes to assert a reset line and initialize a simulated VAXBI powerfail sequence generated by the NBI.
Powerfail Warning	Provides ACLO and DCLO signals to all nodes.

3.6.3 Visibility Bus (VBus)

The visibility bus, shown in Figure 3–9, is a slow–speed bus consisting of sixteen data lines and two control lines. The VBus allows the console operator to read internally latched data in the CPU modules during the execution of microdiagnostics and system initialization. The VBus is used when the system clocks are stopped. Major functions performed by the VBus include:

- Monitoring the state of the CPUs during the execution of microdiagnostics or in response to commands entered at the console during system debugging

- Verifying CPU module installation and revision during system initialization

- Ensuring that control store parity errors do not occur when loading microcode during system initialization

The console controls and reads the VBus by means of two registers located on the clock module's console interface. The VBus control and access registers perform the following functions:

- Select the VBus input channel

- Step the clocks that operate the VBus

- Send serial VBus addresses to the CPU modules

- Halt the operation of the VBus address shift register
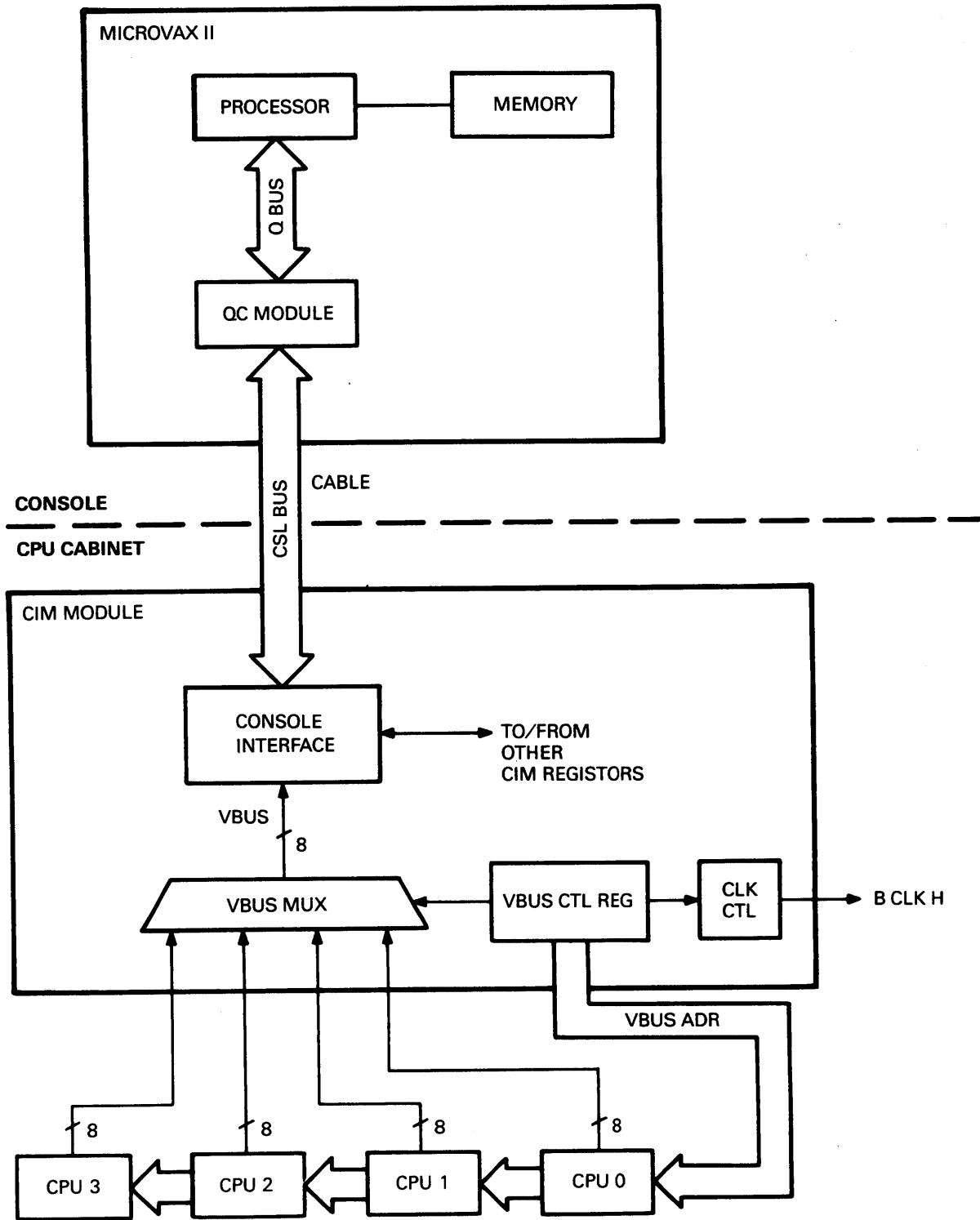
Figure 3–9   VAX 8820/8830/8840 System Visibility Bus (VBus)
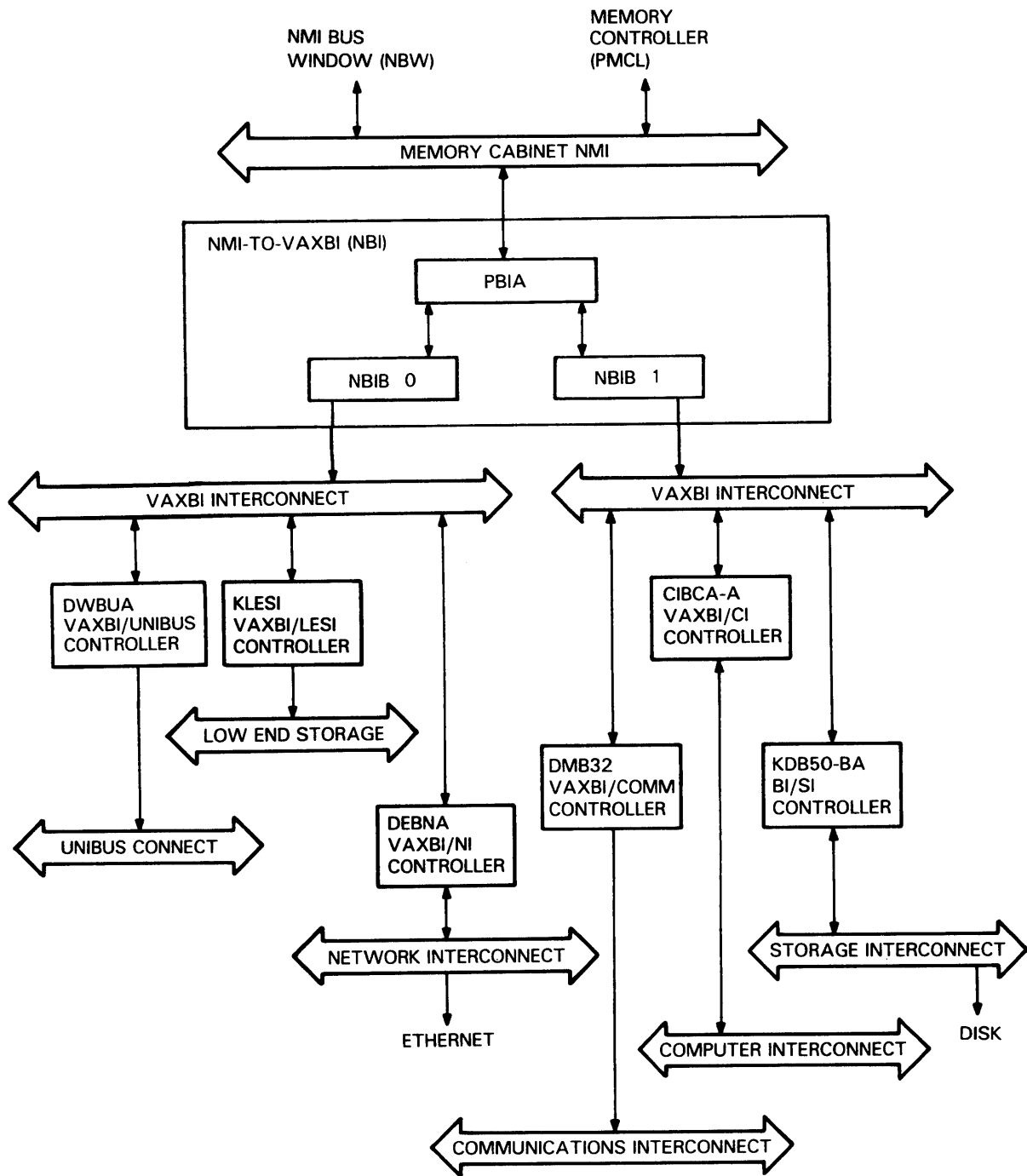
3.6.4 VAXBI Interconnect and I/O Adapters

The system memory interconnect–to–VAXBI interconnect (NBI) adapters provide a connection point for the VAX 8820/8830/8840 system to the VAXBI interconnect. Additional optional adapters can be connected to the VAXBI interconnect (see Figure 3–10) to allow connection of the CPUs to other I/O devices. Table 3–6 lists and identifies some of the optional adapters that can be connected to the VAXBI interconnect.

The NMI–to–VAXBI (NBI) adapter, shown in Figure 3–11, consists of a PBIA module that interfaces to the NMI, and one or two NBIB modules. The NBIB modules interface to the VAXBI interconnect through the use of a VAXBI interface chip (BIIC).

The PBIA module contains the NMI nexus registers, and the NBIB module contains the VAXBI connecting registers. Table 3–7 and Table 3–8 list and identify the registers on each module.

Table 3–6   Optional VAX Bus Interconnect Adapters

Adapter	Function
KDB50–BA	VAXBI Bus–to–Storage Interconnect disk controller.
DEBNA	VAXBI Bus–to–Network Interconnect controller. Uses Ethernet.
CIBCA–A	VAXBI Bus–to–Computer Interconnect controller.
CIBCA–B	VAXBI Bus–to–Computer Interconnect controller.
KLESI	VAXBI Bus–to–Native tape controller.
DMB32	VAXBI Bus–to–Native asynchronous communications controller.
DRB32–M	VAXBI parallel interface.
DRB32–W	VAXBI DR11–W Emulator module plus DRB32–M.
DRB32–E	"Long–line" module plus DRB32–M.
DWBUA	VAXBI Bus–to–UNIBUS Controller. UNIBUS support limited to: LP11, DR11–W, and DUP11.

Figure 3–10   VAXBI Optional Adapters
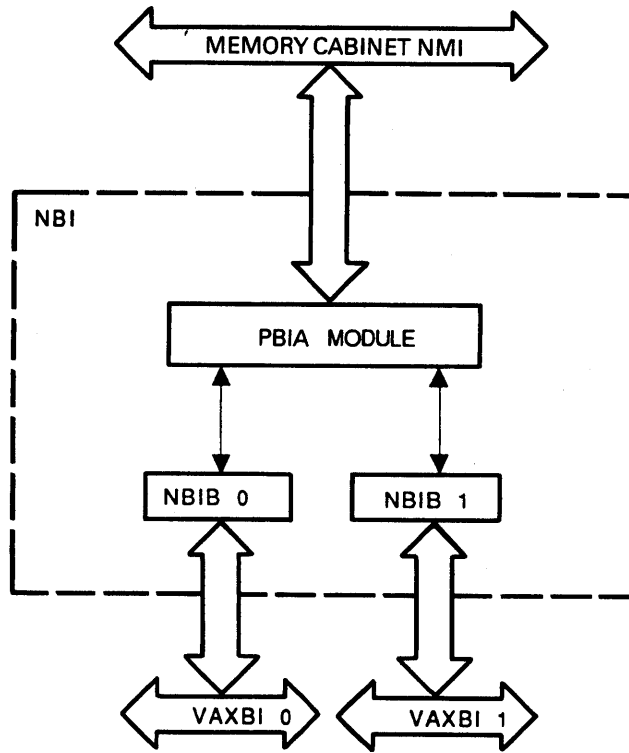
TTB-0281

TTB-0282

Figure 3–11   NMI to VAXBI Interconnect Adapter (NBI)

Table 3–7   PBIA Registers

Address	Register
2X080000	Control/Status 0 (CSR0)
2X080004	Control/Status 1 (CSR1)
2X080008	VAXBI 0 Stop Register (BI0I)

X = 0 = PBIA 0
  = 4 = PBIA 1
  = 8 = PBIA 2

Table 3-8 NBIB Registers

Address	Register
2X080000	Device Type
2X080004	VAXBI Control/Status
2X080008	Bus Error
2X08000C	Error Interrupt Control
2X080010	Interrupt Destination
2X080014	IP Interrupt Mask
2X080018	IP Interrupt Destination
2X08001C	IP Interrupt Source
2X080020	Starting Address
2X080024	Ending Address
2X080028	BCI Control
2X08002C	Write Status
2X080040	User Interrupt Control

X = 0 = PBIA 0/BI 0
 = 2 = PBIA 0/BI 1
 = 4 = PBIA 1/BI 0
 = 6 = PBIA 1/BI 1
 = 8 = PBIA 2/BI 0
 = A = PBIA 2/BI 1

3.7 POWER SYSTEM

The power system provides the voltages necessary to operate the CPUs, memory, and the VAXBI interconnect. Three-phase ac utility power is used as the primary source for the system. The required ac and dc voltages are developed using the power modules and voltage regulators located within the VAX 8820/8830/8840 system cabinet. Figure 3-12 shows a simplified block diagram of the power system, and Figure 3-13 indicates the physical location of the power system components.

3.7.1 880A/B Power Controller

The 880A/B controller is the main ac input module for the power system. Power is received from the main circuit breaker and is distributed to the other system components by the 880A/B controller. The power controller distributes the power as shown in Table 3–9.

Table 3–9   Power Distribution by the 880/B Power Controller

Unit	Power
CPU cabinet	
SST Box	Switched three–phase
Console	Unswitched single–phase
Air mover	Switched three–phase
Memory cabinet	
PBox	Unswitched single phase
BBU	Unswitched single–phase
Air Mover	Switched three–phase
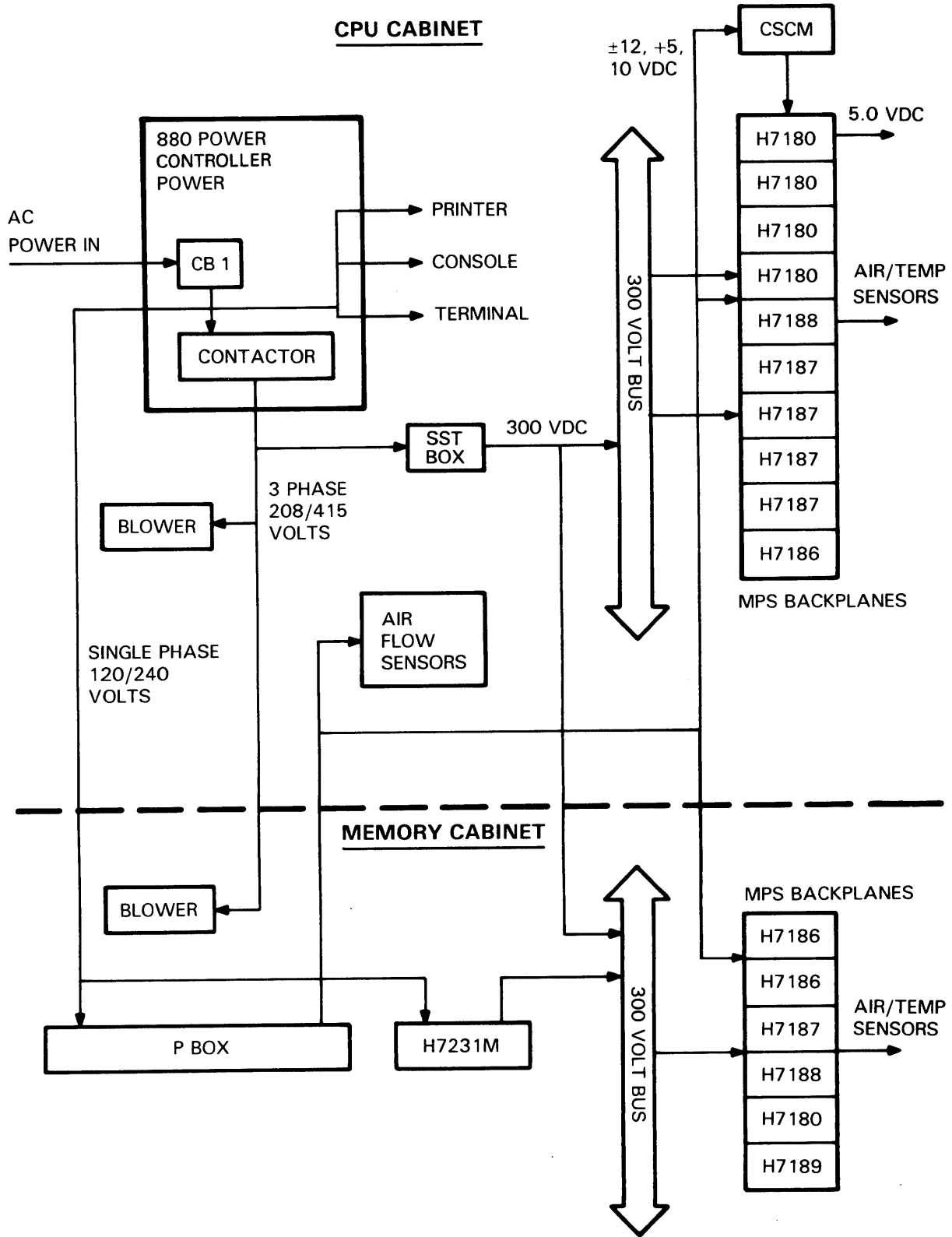
Figure 3–12   Power System Block Diagram

TTB-0283

MEMORY CABINET

BLOWER					
MPS-3					
F2	EMM2	B2	C2	E2	D2
H7189	H7188	H7186	H7186	H7180	H7187
BI		+5 V	+5 V	-5.2 V	-2 V

T8	T7	AF1	T6

BI	BBU	MAB	NEMO

T5	PBOX			
	CSP3	CSP2	CSP1	ILM

AIR FILTER

CPU CABINET

BLOWER					
MPS-2			MPS-1		
E1	D1	EMM1	C1	D1	E1
H7180 H7180	H7187 H7187	H7188	H7186	H7187 H7187	H7180 H7180 [CSCM]
-5.2 V	-2 V		+5 V	-2 V	-5.2 V

AF2	T3	AF1	T2

STAR	POLR

T1	
880 AC INPUT MODULE	SST BOX

AIR FILTER
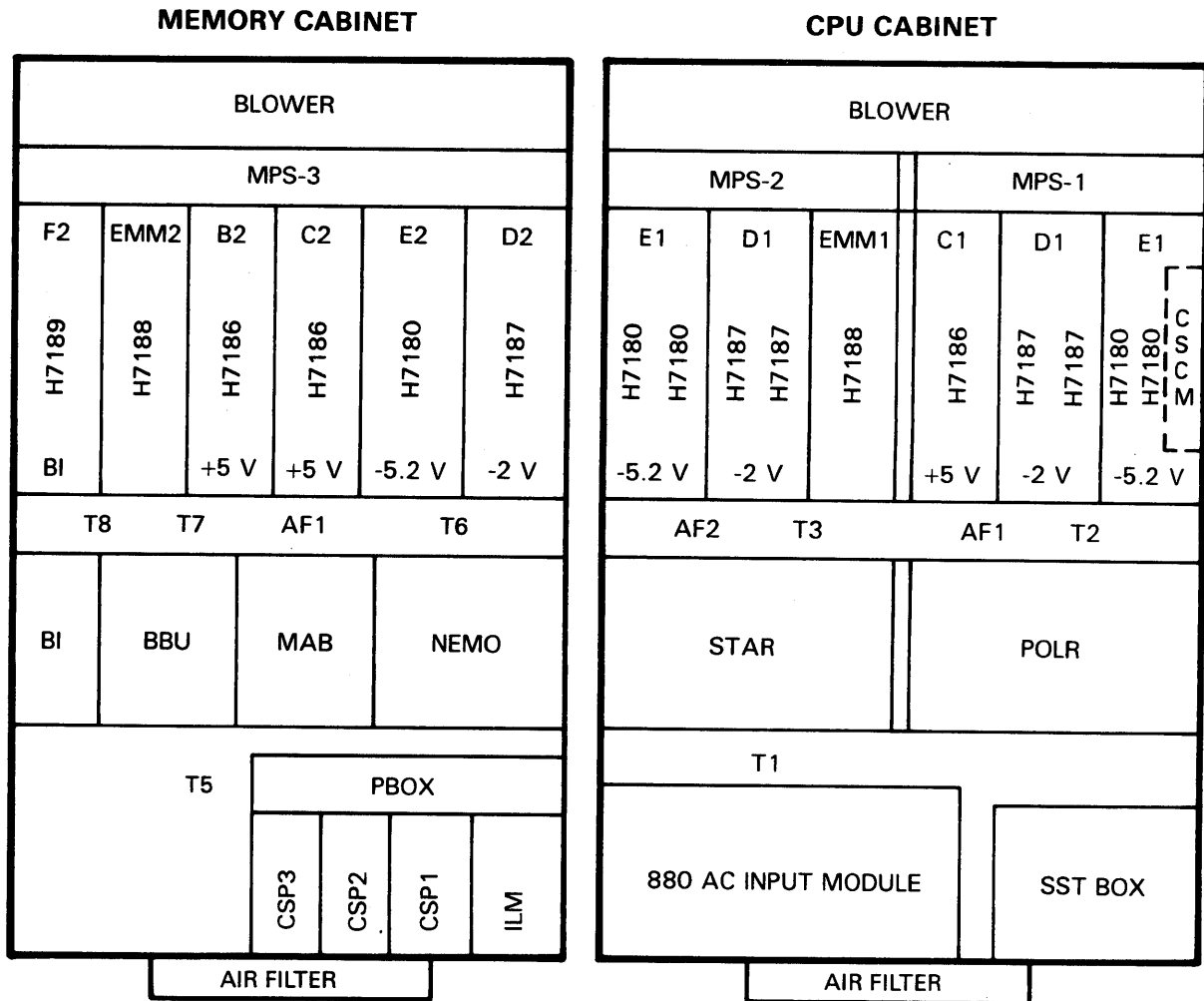
TTB-0284

Figure 3–13   Physical Location of the Power System Components

T1, T5 = Ambient Air Temperature Sensors
T2, T3, T6, T7, T8 = Exhaust Air Temperature Sensors
T4 not connected
AFn = Air Flow Sensors

3.7.2 Solid–State Transformer Box (SST)

The solid–state transformer box converts three–phase ac power to 300–V dc power and sends the 300 V dc power to the four 300 V dc MPS buses. Systems designed for 50–Hz operation require a buck converter, which takes the output of the SST (600 V dc for the 50–Hz, 416/380–V ac configuration) and changes it to 300 V dc.

3.7.3 PBox Port Conditioner

The PBox is a multifunction power assembly containing four modules: three CSP modules and one ILM module.

The primary purpose of the CSP modules (Control and Startup Power, H7060A) is to provide basic power to the control modules of the power subsystem. Upon receiving startup power, these modules begin controlling operation of the power subsystem.

The primary purpose of the ILM module (H7061A) is to provide an interface and to condition signals passing between the various modules of the power subsystem.

3.7.4 Module Power Supplies

The module power supplies (MPS) are a group of dc power modules and three backplanes. Two MPS backplanes are located above the CPU card cage in the CPU cabinet; the third is located above the logic card cages in the memory cabinet. The MPS backplanes contain the regulated dc power supplies that provide the operating power for the CPUs, NBWs, memory, and VAXBI(s).

3.7.5 Environmental Monitoring Module

The environmental monitoring module (EMM) is a microprocessor–based unit that monitors the power and environmental conditions within the system. The EMM responds to console control commands during power–up and power–down sequencing, initialization, and battery backup operations. The console controls the power system through the EMM.

3.7.6 Battery Backup Unit

The battery backup unit (BBU) gives the power system a method of providing backup power to the CPU memory during an ac power failure. The BBU contains a 48–V dc rechargeable battery pack, charging circuit, and a dc–to–dc converter. The converter provides 300 V dc for up to ten minutes during the backup mode.

4.1 INTRODUCTION

This chapter contains console troubleshooting procedures that will provide you with assistance before calling Field Service. A procedure for setting up a remote diagnostic session also is included.

4.2 PREVENTATIVE MAINTENANCE

The only preventative maintenance required on the VAX 8820/8830/8840 System is to clean the filters. The system has two metallic filters located in the bottom of the system cabinet. Both filters should be removed and cleaned at 60-day intervals. Filter service should only be performed by trained and qualified personnel. If the system is under warranty, or is covered by a standard Digital service contract, preventative and corrective maintenance can be performed only by authorized DIGITAL service personnel.

WARNING
**The system cabinets contain potentially dangerous
voltages as well as static-sensitive components.
Access to the interior of the cabinets using the
cabinet opening tool (DEC PN 12-26339) for any
type of maintenance should be restricted to trained
and qualified service personnel.**

4.3 CONSOLE SUBSYSTEM HARDWARE SELF-TESTS

The console subsystem contains a complete set of hardware self-test programs that check all of the major console system functions. If a problem is found, the self-test program identifies the nature of the problem. The self-test is completely automatic and runs each time the console is powered up. The console terminal will appear as shown in Example 2-1 when the tests complete without failure.

The MicroVAX II powerup self-test sequence consists of 16 ROM based tests, which are run when MicroVAX II power is turned on. They are numbered from 0 to F (hex). The tests are run starting at test F and ending with test 0. Tests F through A cause no display at the console terminal. When test 9 (Identifying Console Terminal) runs successfully, a heading which reports the version of the CPU in the MicroVAX II (KA630) is displayed on the terminal. Test number 8 does not cause a display but successful completion of each test from 7 down to 0 causes the number of the successful test to be displayed on the console terminal. The self-tests, in order of their execution, are:

    F. Waiting for DC OK
    E. Waiting for P OK
    D. Running CHECKSUM Test on CPU ROM
    C. Searching for RAM Memory Required for CPU ROM Programs
    B. Reading KA630-A IPCR Register (accesses Q22-Bus)
    A. Testing VCB01 Video Console Display (if present)
    9. Identifying Console Terminal
    8. Language Inquiry or CPU Halted
    7. Running Data Tests on RAM Memory
    6. Running Address Tests on RAM Memory
    5. Running Tests That Use Q22-Bus Map to Access Local Memory

4. CPU Instruction and Register Tests
3. Running Interrupt Tests
2. Searching for Bootstrap Device
1. Bootstrap Device Found
0. Testing Completed

4.4 CORRECTING SIMPLE PROBLEMS

Table 4–1 is a quick reference source for correcting simple problems.

Table 4–1 Correcting Simple Problems

Symptom	Possible Condition	Corrective Action
Nothing happens when the console power switch is turned on.	The power cord is not connected.	Reconnect the power cord to the VAX 8820/8830/8840 system cabinet.
	The MicroVAX II circuit breaker has tripped.	Turn MicroVAX II power off and reset the circuit breaker.
	Power is not received from the system CPU cabinet.	Verify that circuit breaker 5 is in the ON position.
	The VAX 8820/8830/8840 system has entered the automatic shutdown phase.	Attempt a power up of the VAX 8820/8830/8840 system.
		If the power up attempt is unsuccessful, contact a Digital Field Service technician.
		Refer to the *Console User's Guide* for power–up procedures.
Nothing appears to happen when the power switch is ON, but the fan is running.		Call Field Service.

4.5 WHAT TO DO IF YOU HAVE TO CALL FOR SERVICE

Digital Equipment Corporation has a central service point in your area to help you get your system running with a minimum of trouble. Before calling for service, please do the following:

1. Write down the serial number of your system.

2. Summarize the problem and make a note of what you were doing when the system failed.

3. If at all possible, place the service call in close proximity to the system. The service person may ask you to reproduce some of the failing procedures in order to obtain additional information.

4. Be prepared to set up a remote diagnostic session if necessary. See section 4.6 for remote diagnosis procedures.

Call Digital Equipment Corporation at one of the following telephone numbers:

Location	Telephone Number
Belgie/Belgique	02–24 26 790
Bundesrepublik Deutschland	089–95 91 66 44
Canada	(800)–267–5251
Danmark	04–30 10 05
Espana	1–73 34 307
France	1–687 31 52
Italia	02–61 75 381 or 02–61 75 382
Japan	(03)–989–7161
Nederland	1820–31 100
Osterriech	222–67 76 41 App. 444
Suisse/Schweiz	01–81 69 270
Suomi	90–42 33 32
Sverige	08–98 88 35
United Kingdom	0256–59 200
United States	1–(800)–DEC–8000

4.6 REMOTE DIAGNOSIS

Remote diagnosis provides a method for troubleshooting the VAX 8820/8830/8840 system from the Digital Service Center. Remote troubleshooting requires connecting the system to the Remote Services Console (RSC) and connecting the RSC to a telephone line by means of a DF112 modem (or equivalent). Further information on the RSC is available in the *RSC Field Service Manual* (EK–KCRSC–FS). Additional information on the DF112 modem is available in Appendix A of this manual and in the *DF112 Modem User's Guide* (EK–DF112–UG).
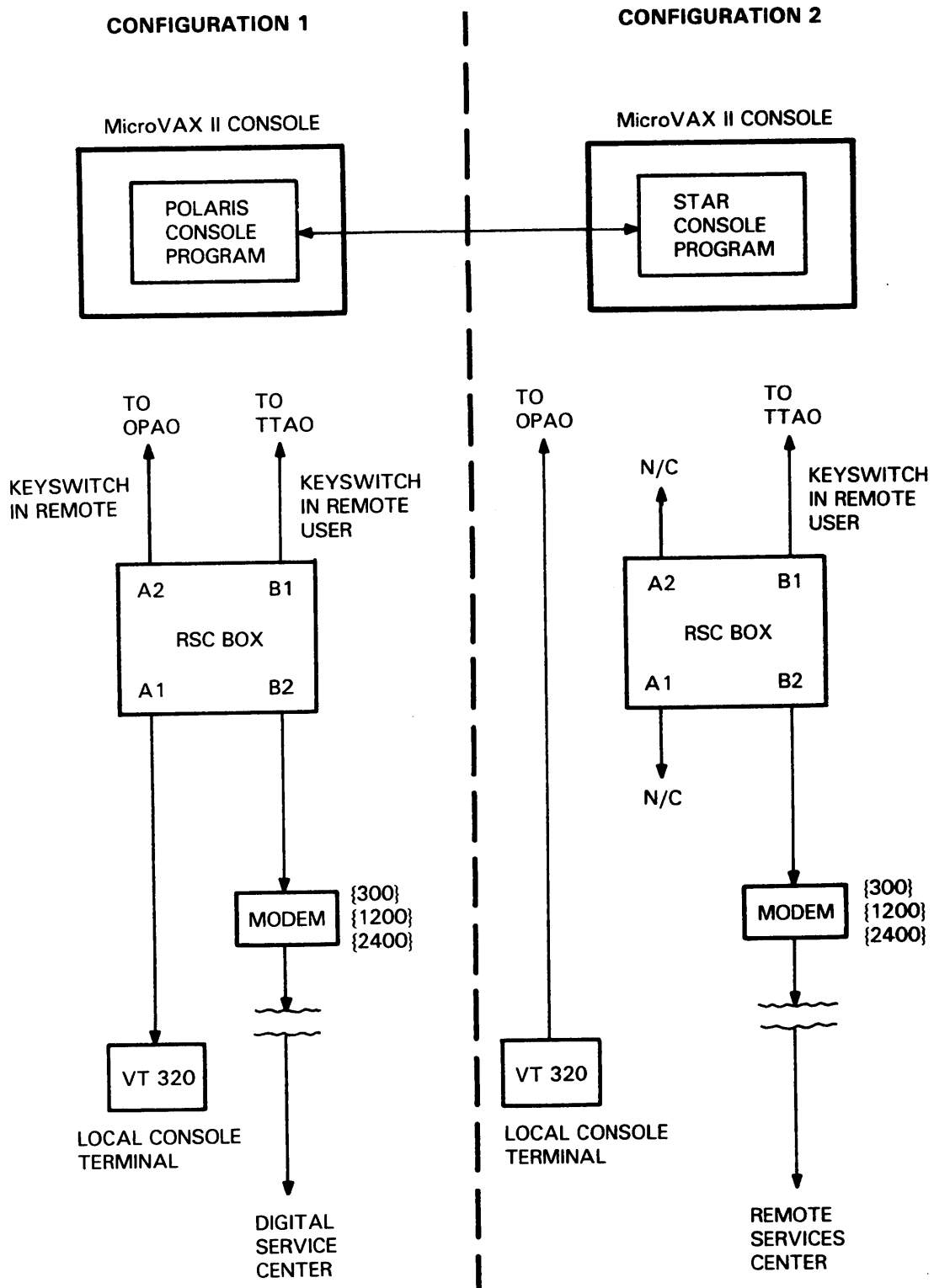
4.6.1 Remote Diagnosis Configurations

The RSC may be connected in one of two configurations. The first configuration allows the DIGITAL Service Center to have full local console control, while the second configuration allows the Service Center to have limited local console control. The first configuration is recommended as it allows the Service Center to exercise the system more fully and gain more diagnostic information.

For full console control, the RSC connects the local console terminal to OPA0, and the remote console terminal to OPA0 or TTA0. When the RSC keyswitch is at REMOTE, the Service Center (via modem) has control of OPA0. When the RSC keyswitch is at REMOTE USER, the DIGITAL Service Center (via modem) is connected to TTA0. With the RSC keyswitch in LOCKOUT, all remote activity is terminated and the local console terminal has a direct connection to the MicroVAX II.

For limited console control, the RSC connects the remote console terminal to TTA0. When the RCS keyswitch is at REMOTE USER, the Service Center (via modem) is connected to TTA0. When the RSC keyswitch is at LOCKOUT, all remote activity is terminated and the local console has a direct connection to the MicroVAX II. There is always a direct connection from the local console terminal to OPA0. In this mode, commands specific to OPA0 will not be allowed through the remote console. Commands not allowed are:

- QUIT

- ENABLE/DISABLE PRINTER

- ENABLE/DISABLE REMOTE_USER

- ENABLE/DISABLE HARDWARE

- ENABLE/DISABLE FUNCTIONAL_SIMULATION

- ENABLE/DISABLE DECSIM

- ENABLE/DISABLE DUAL

- ENABLE/DISABLE REMOTE_CONSOLE

- POWER ON/OFF/STANDBY

- EMM

CONFIGURATION 1

MicroVAX II CONSOLE

POLARIS CONSOLE PROGRAM

STAR CONSOLE PROGRAM

CONFIGURATION 2

MicroVAX II CONSOLE

TO OPAO

TO TTAO

KEYSWITCH IN REMOTE

KEYSWITCH IN REMOTE USER

TO OPAO

TO TTAO

N/C

KEYSWITCH IN REMOTE USER

A2   B1

RSC BOX

A1   B2

A2   B1

RSC BOX

A1   B2

N/C

MODEM   {300}
        {1200}
        {2400}

MODEM   {300}
        {1200}
        {2400}

VT 320

LOCAL CONSOLE TERMINAL

VT 320

LOCAL CONSOLE TERMINAL

DIGITAL SERVICE CENTER

REMOTE SERVICES CENTER

TTB-0369

Figure 4–1   Remote Console Configurations
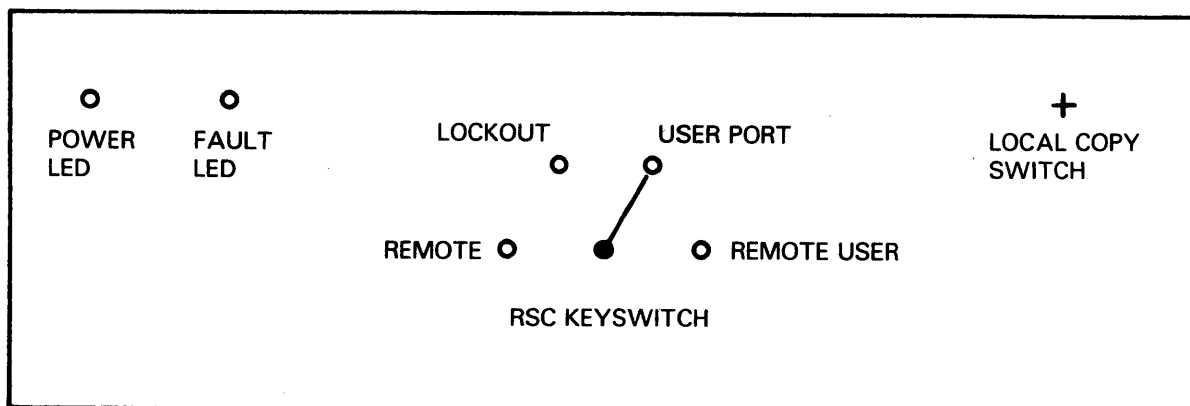
TTB-0370

Figure 4-2   Remote Services Console Front Panel Switch Locations

4.6.2 Setting Up for a Remote Diagnosis Session

VAX 8820/8830/8840 systems with a DIGITAL maintenance contract will have a remote diagnosis configuration installed. This section explains how to initiate a diagnosis session through the DIGITAL Remote Diagnosis Center (RDC) when a system performance problem is suspected.

The system manager (or her/his agent) should use the following steps to initiate a remote diagnosis session when the system does not appear to be performing properly:

1. Prepare for the call by gathering information as suggested in section 4.5.

2. If the system is set for limited remote console control, enter these two commands on the local console:

   - ENABLE REMOTE USER

   - ENABLE REMOTE CONSOLE

3. Make sure that the console is in local console mode.

4. Place the Remote Services Console front panel switch in the REMOTE position.

5. Call the Remote Diagnosis Center (RDC) and tell them that you want them to perform a remote diagnosis session on your system.
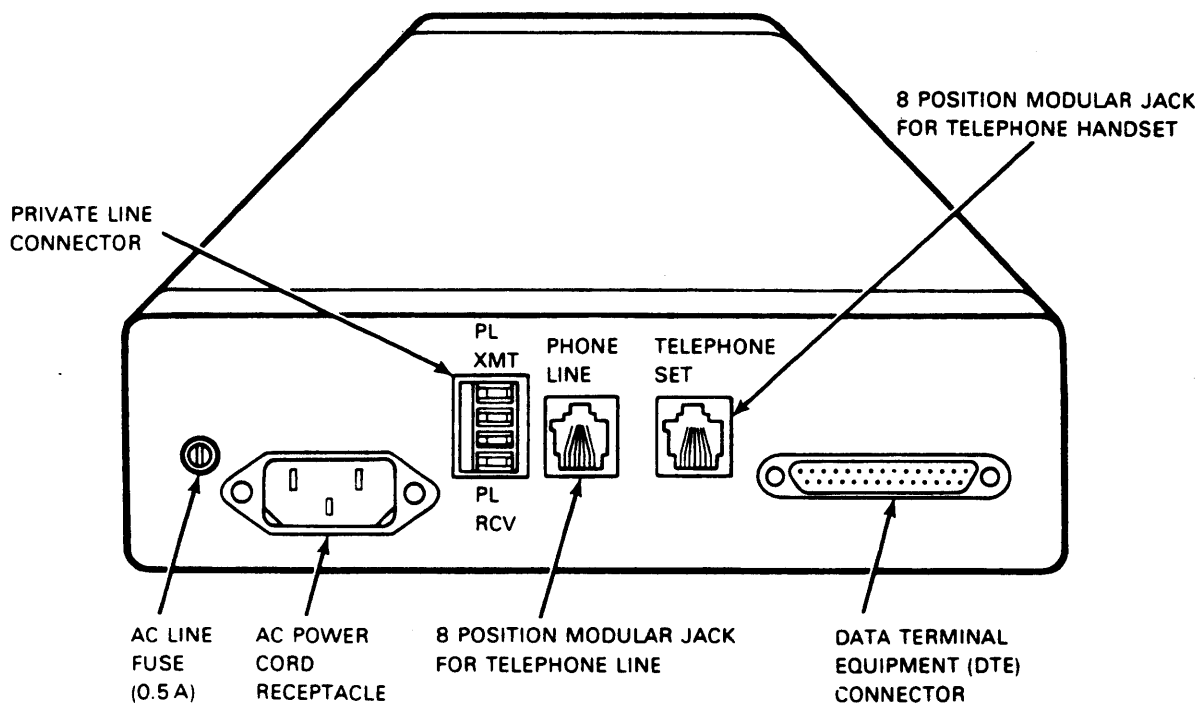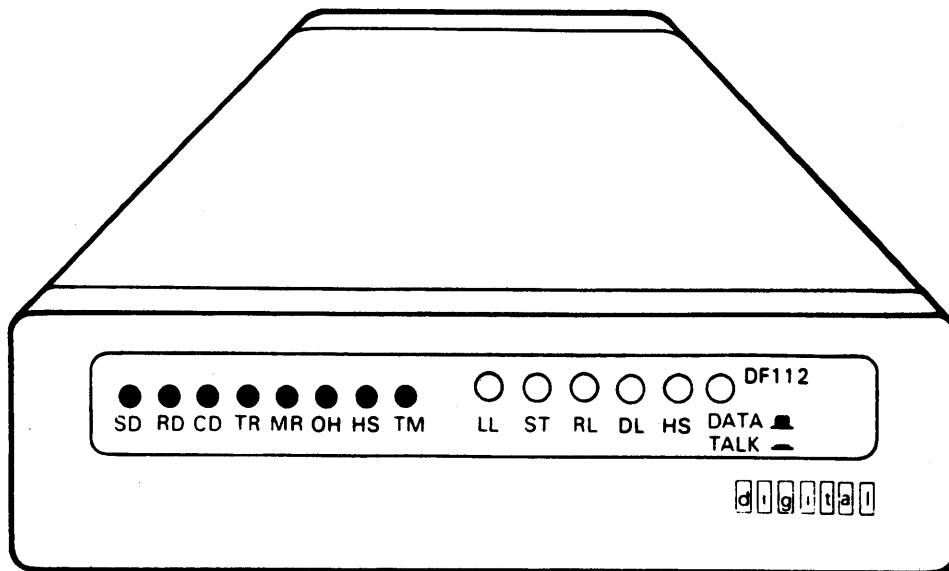
APPENDIX A
DF112 MODEM INFORMATION

A.1 INSTALLING THE DF112 MODEM.

The following procedures are used to unpack and install the DF112 Modem. Refer to the *DF112 Modem User's Guide*, (EK–DF112–UG) for additional information.

1. Unpack, inspect, and set up the DF112 modem. (Install the stand–alone modules if required.)

2. Connect the modem to the public telephone network using either the handset connection or the direct telephone line connection on the rear of the DF112 modem (see Figure A–1).

3. Connect one end of a null–modem cable to the Data Terminal Equipment (DTE) connector on the rear of the DF112 modem (refer to Figure A–1); connect the other end of the cable to the chosen user port on the MicroVAX II console.

4. Set the DF112 front panel controls to the desired operating positions (refer to the *DF112 Modem User's Guide*).

SD RD CD TR MR OH HS TM   LL ST RL DL HS   DATA
                                                TALK

DF112

digital

8 POSITION MODULAR JACK
FOR TELEPHONE HANDSET

PRIVATE LINE
CONNECTOR

PL
XMT   PHONE   TELEPHONE
           LINE   SET

PL
RCV

AC LINE   AC POWER   8 POSITION MODULAR JACK   DATA TERMINAL
FUSE   CORD   FOR TELEPHONE LINE   EQUIPMENT (DTE)
(0.5 A)   RECEPTACLE   CONNECTOR

TTB-0376

Figure A–1   DF112 Modem

A.1.1 DF112 Modem Characteristics

The DF112 modem is the recommended modem to be used with the remote port of the VAX 8820/8830/8840 system (Table A–1). Equivalent modems may be substituted as required.

A.1.1.1 DF112 Modem General Characteristics

The DF112 is a 300/1200 bits/s full–duplex, aysynchronous/synchronous modem that allows terminals, personal computers, and processors to communicate over dial–up or private/leased telephone lines. The DF112 modem is compatible with Bell 103J/212A™ modems.

The modem can operate at data rates between 0 and 300 bits/s or 1200 bits/s for asynchronous communications and 1200 bits/s for synchronous applications.

A.1.1.2 Major Features

Major feature of the DF112 modem include:

* Microprocessor–Controlled Autodialer: Stores up to five 26–digit telephone numbers; compatible with both rotary and tone dialing.

* Extensive Built–In Diagnostics: Help to locate network problems and reduce downtime.

* DATA/TALK Switch: Allows alternate voice/data capability with a standard telephone set.

Table A–1   DF112 Specifications

Characteristic	Parameter
Functional Compatibility	Bell 212A; Bell 103J
Data Format	Serial, binary, asynchronous/synchronous data
Operating Mode	Auto originate, auto answer, manual originate, manual answer
Communication Mode	Full–duplex at all data rates
Data Rate	Low–speed: 0–300 bits/s asynchronous High–speed: 1200 bits/s asynchronous/synchronous
Telephone Line Requirement	Dialup (PSTN) RJ11C (US)/CA11A (Canada) RJ41S (US)/CA41A (Canada) RJ45S (US)/CA45A (Canada) Leased line (P/LTN) 625 connection block RJ14 to RJ11 line connection
Certified for FCC Part 15 (Subpart J)	For use in home or office environment
Interface Compatibility	EIA RS–232–C/CCITT V.24 and V.28 compatible voltages
Power Requirements	120 V ac, 60 Hz 0.15 A at 5 V dc 0.015 A at 12 V dc 0.045 A at 12 V dc

---

™ Bell 103J and 212A are registered trademarks of Western Electric

Table A–1 (Cont.)   DF112 Specifications

Characteristic	Parameter
Power Requirements	120 V ac, 60 Hz
	0.15 A at 5 V dc
	0.015 A at 12 V dc
	0.045 A at 12 V dc
Operating Environment	
Temperature	10–40°C (50–104°F)
Relative Humidity	0–90% noncondensing, noncaustic
Physical Characteristics	
Height	25.4 mm (1.0 in)
Width	203 mm (8.0 in)
Depth	264 mm (10.4 in)
Weight	0.68 kg (1.5 lb)