Chapter 8

82/83x0 and VAXBI

This Chapter now owned by Dave Bazley.....

Revision: Reorganised and updated January 1993

Acknowledgement: Originally written by Stefan Fiala with additions by Peter Griffin

8.1 Configuration - Differences

8.1.1 Who's who, 82/83x0 (Quick guide)

Scorpio Code name for whole 82/83x0 range

8200 1 x KA820 CPU, 12 slot BI backplane, BA32 box, Configuration 1

8300 2 x KA820 CPUs, 12 slot BI backplane, BA32 box, Configuration 1

8220 1 x KA82x CPU, 24 slot BI backplane, '750 type cabinet, Configuration 2

8320 2 x KA82x CPUs, 24 slot BI backplane, '750 type cabinet, Configuration 2

8250 1 x KA825 CPU, 12 slot BI backplane in BA32 box or 24 slot BI backplane in '750 cab

8350 2 x KA825 CPUs, 12 slot BI backplane in BA32 box or 24 slot BI backplane in '750 cab

Table 8–1: Supported (saleable) DEC Unibus options

DEUNA	1	DELUA	1	DMF32	2(first on bus)
DHU11	6	TU80	2	TU81	2(KLESI-U)
DMR11	2(1M)	DR11W	4	DMZ32	2
LP11	4	DUP11	3	TUK50	1

Table 8–2: Supported (work) DEC Unibus options

RX211	1	UDA50A	1	DMP11	1
DZ11	1	KW11-P		KMS11BD	2
DRE11	1	TSU05			

Table 8–3: Unsupported (Won't work)

KMC11	RL211	RK711	IEC11	DR11C	DZS11EA
KCT32	DMC11	ISB11	LPA11K	DT07	

Table 8–4: Emulex Unibus options that work

Product	Digital Service Classification	Firmware Revision	Comments
CS11/F	CAT A	H or later	J is current level
CS21/F	CAT A	G or later	H is current level
CS23	CAT C	D	
CS32/F	B USA	F or later	G is current level
CSM32/MF	CAT C	G	
CS40/F	CAT C	B	
CS41/F	CAT C	B	
CS62	CAT C	B	

The CS11/U (CAT A),CSM11/MU (CAT C), CS11/V (CAT A), CS21/U (CAT A), and CSM21/U (CAT C) are not supported on the 82/83x0 systems. CS11/U and CS21/U can be upgraded to the CS11/F and CS21/F respectively, via new firmware.

Of Significant Importance: the DEC DWBUA BI-To-Unibus Adapter must have an E2 or higher ECO revision level.

8.1.2 Configuration Rules

8.1.2.1 Generic (12-slot or 24-slot)

Cages are numbered right to left facing the front of the machine K1-K4. Slots are numbered in each cage right to left J1-J6. For configuration 2 systems then Cages K1 and K2 are powered by Power Supply #1 and cages K3 and K4 are powered by Power Supply #2. Configuration 1 systems have only one power supply.

1. Maximum number of VAXBI options is 16 as limited by VAXBI architecture.
2. All multi-module options (with one exception - CIBCI) are configured so that the module containing the BIIC goes to the right of the other modules contained within the option.
3. All modules of a multi-module option must be placed in contiguous slots without crossing cage boundaries per VAXBI architectural rules.
4. The primary CPU (KA820/KA825) must go in K1J1 to enable connection of the PCM cables and to enable system clock. K1J1 must also have node ID #2, because all attached processors have node #2 in their EEPROMs as the logical console ID.
5. DWBUA must be configured at node ID #0. The DWBUA must be set to arb fixed-high-priority by the operating system to avoid unibus slave sync timeouts. Node #0 is highest priority node. Only one DWBUA allowed since it must be at highest priority. The only exception is in the 6-slot OEM box, where the only other options on the VAXBI are one CPU and memories. In that case the DWBUA may be placed at any node ID.
6. Maximum of 6 MS820-BA/CA's allowed per power supply for +5B current limits. Maximum of 8 MS820-AA's allowed per power supply for +5B current limits.

7. Any mix of MS820-AA, MS820-BA and MS820-CA is allowed, but when memory types are mixed, then the maximum total number allowed per power supply is 6. This allows simplification the rules instead of trying to calculate the power sums for any mix of MS820-AA's , MS820-BA's and MS820-CA's.
8. The only modules allowed to the left of MS820-BA's or MS820-CA's are other MS820-BA/CA's or the side wall of a cage. This is because MS820-BA/CA's have components on the reverse side. Non-memory modules could cause mechanical clearance problems and/or thermal problems, violating the 50 watts/slot guidelines, if placed to the left of the MS820-BA/CA. MS820-AA's could conceivably be placed to the left of a MS820-BA, but to keep the rule simple, there should be no non-MS820-BA/CA modules to the left. In addition it is suggested that MS820-CA modules are placed to the left of MS820-BA modules.
9. Maximum of 1 DEBNT/DEBNA per power supply allowed in system as there is only 1 +15v cable per power supply distribution board to power the NI IOCP.
10. All DEBNT/DEBNA's must have higher node IDs than any KLESI-B. Otherwise the TU81 will not be configured by VMS on the KLESI-B. This is is fixed in VMS V4.5, from which time the rule can be ignored.
11. Maximum number of cluster connects (CIBCI/CIBCA) is 1 since that is all VMS supports.
12. The power available to VAXBI options is limited. Power totals must be checked for each power supply, either by using the data given in the Systems and Options Catalogue, the VAXBI handbook or by referencing another article.
13. The IO Panel space available to VAXBI option is limited. IO panel space should be checked (either using the data given in the Systems and Options Catalogue or the VAXBI Handbook).
14. Maximum of one H7231-L BBU per system.

8.1.2.2 Configuration 1 (12 slot) System Specific Rules

1. Maximum number of VAXBI modules is 12 as limited by VAXBI slots.
2. Maximum number of storage options (KDB50/CIBCI/CIBCA) is set to 2 to facilitate cabling in manufacturing.
3. Maximum number of KLESI-B's is set to 1 to facilitate cabling in manufacturing.
4. Maximum number of DMB32's is set to 2 to facilitate cabling in manufacturing.
5. Total sum of all DRB32 options (-M, -W, -E) is set to 2 to facilitate cabling in manufacturing.
6. Maximum of 1 DEBNT/DEBNA due to availability of +15 V connectors on power supply.

8.1.2.3 Configuration 1 (12-slot) Preferred Placement Rules

Slots should have node ID plugs as follows

Slot	Plug
K1J1	node 2
K1J2	node 0
K1J3	node 4
"	"
"	"
"	"
K2J6	node D

Table 8–5: Typical Configuration

KA820/KA825 Primary CPU	K1J1 Node 2
KA820/KA825 Attached CPU	Left justify after MS820s
DWBUA	K1J2 Node 0
MS820-CA	Populate from the left K2J6, K2J5...
MS820-BA	Populate from the left after any MS820-CA modules. K2J6, K2J5...
MS820-AA	Populate from the left after any MS820-BA/CA modules.
Dual Slot (CIBCA,CIBCI,KDB50)	Populate from the right beginning at K1J3. If a CI/KDB mix then CI should be to the right.
Single Slot (DEBNA,DMB...)	Populate from the left after MS820 and any attached CPU.

8.1.2.4 Configuration 2 (24 slot) System Specific Rules

1. Maximum number of VAXBI modules is 24 as limited by VAXBI slots.
2. PS2 must have +5v pre-load on it to allow +5v regulator to turn on. This means that at least one non-memory module must be placed in K3 or K4 that loads +5v to at least 6 amps. To keep things simple we decided that at least one storage module (CIBCI, CIBCA, KDB50) be placed in K3/K4. It is guaranteed to have at least one of these per system and they all draw enough +5v to load it.
3. If there is an H7231 BBU in the system, then the maximum number of memory modules is 6 All the memories must be placed in cages K3 and K4. If there is no H7231 BBU in the system, then the maximum memory configuration is 12 MS820's (6 per power supply). The memories can be placed in any cage.
4. VMS V4 limits the number of memory controllers to 8, limiting the maximum memory configuration on an 82x0/83x0 Configuration 2 system to 128 Mb. VMS V4.5 and below had further restrictions on the memory limits.
5. The diagnostic EBKAX requires a memory module in a configuration 2 system be at node #F. This tells EBKAX that it is a configuration 2 system which controls how it runs the BBU manual tests. 12 slot systems have no node F.
6. The Configuration 2 system has a hinged IO panel door. When the IO panel door is opened, some external cables connected to panels A10/A11, B10/B11, C10/C11 may hit the cabinet wall. To avoid this, the KA820/KA825 and DEBNT/DEBNA IO panels will be fixed in locations C10/C11, B10/B11 respectively. The external cables to these IO panels do not hit the cabinet wall.
7. Maximum number of storage options (KDB50/CIBCI/CIBCA) is arbitrarily set to 4.
8. Maximum number of KLESI-B's is arbitrarily set to 3.
9. Maximum number of DMB32's is 6. There is only enough IO panel space for 6 DMB32's.
10. Maximum number of DRB32-M's is arbitrarily set to 8.
11. Maximum total of DRB32-W/-E options is arbitrarily set to 4.
12. Total sum of all DRB32 options (-M, -W, -E) should not exceed 8 since there is insufficient +5 power.
13. maximum of 2 DEBNT/DEBNA due to availability of +15V power connectors on each power supply.

8.1.3 8250 - 8350 differences

The VAX 8250 and 8350 systems are enhanced systems giving around 20% greater performance than the standard 8200/8300 systems. This is accomplished by installing a faster CPU module, the KA825, in place of the KA820. This module is a T1001-YA and is a faster version of the T1001-00 module used for the KA820. Both modules physically look the same, except that the T1001-YA contains faster components and a faster clock crystal (see Table 8–6 below). Architecturally, the KA820 and KA825 are identical. All registers and console commands are identical for the KA825 as well. The differentiating factor between the two CPUs is bit 23 of the SID register (most significant bit in the major revision field) which is set for a KA825.

Table 8–6: 82xx CPU type differences

	KA825	KA820
CPU Module	T1001-YA	T1001-00
BI Clock	200 nanoseconds	200 nanoseconds
CPU Clock	160 nanoseconds	200 nanoseconds
Xtal	50 MHz	40 MHz
Cache/TB RAMs	35 nanoseconds	45 nanoseconds
V11 chip set	160ns parts	200ns parts
DC346	date code 8644 min	all date codes
DC347	date code 8649 min	all date codes
DC348	date code 8644 min	all date codes

Note

Attach the KA825 as a KA820. The supervisor doesn't know or care of the difference.

8.2 82/83x0 CPU information

8.2.1 Self Test

Self Test is a hardware verification procedure that performs fault detection and isolation to a VAXBI node (most consist of one or two boards) and as such it is an integral part of the service methodology. (Equivalent to microdiagnostics). All nodes will complete their self test within ten seconds, and indicate their success by lighting their yellow self test LED. Each phase of the CPU node self test is reported on the console. In addition the CPU polls the other nodes on the system, and reports their status on the console. The Self Test is invoked automatically whenever . . .

1. System power is turned on by the operator.
2. Power is restored after a power failure.
3. Console 'I' command is typed.
4. System is booted.

Note

The VAXBI Self Test does not test peripherals, i.e. RA60s and UNIBUS options, apart from the UET (Unibus Exerciser Terminator)

8.2.1.1 CPU node self test

The KA82x CPU self-test printout looks like this:-

```
#ABCDEFGHIJK.MN#
```

This decodes as follows:-

#	Self test initiation
A	Control store checksum
B	IE chip internals
C	DAL interface
D	M chip internals
E	BTB array test
F	Cache array test
G	IE - M chip interaction
H	Pcntl gate internals
I	EEprom checksum
J	Boot RAM test
K	F chip
L	Not included functionality
M	RCX50 test. Not on slave!
N	BIIC test, loop over BI
#	Self test completion

"." indicates test failed (Normal for "L" as this functionality is missing). Most tests just loop when they fail. Some continue with functionality disabled (Cache).

8.2.1.2 BI SELF TEST

The BI node self test looks like . . .

```
0 . 2 . 4 . . . . . . A B . . . .
```

If a node passed its self test then its node number is printed. If it failed then its node number is preceded by a minus (–) sign. A period (.) means node is non-existant. In the example above nodes 0, 2, 4, A, B have responded. Based on configuration standards this would mean . . .

0	DWBUA + Unibus o/k
2	KA820 primary o/k
4	Another BI node (KDB50)
A	First memory array. X Mb's
B	Second memory array. X Mb's

A "broke" node would respond prefixed by "-". For instance if the Unibus expander box was powered off.!! (This breaks the DWBUA, typically node #0).

```
-0 . 2 . 4 . . . . . . A B . . . .
```

8.2.1.2.1 Memory Size

The next printout is a calculation of memory size . . .

```
00400000
```

This means 4 Mb's. 00600000 would mean 6 Mb's.

8.2.1.2.2 Complete Auto-Boot Sequence

The complete power on printout with "auto-boot" would be . . .

```
#ABCDEFGHIJK.MN#

0 . 2 . 4 . . . . . . A B . . . .
00400000

--- The VMS BANNER would now appear here, unless ULTRIX booted !!.--
```

8.2.1.3 DWBUA Unibus Adapter self test

Self test starts at 00 and increments. If it fails examine:-

```
bb+2C = 10000000          ;where bb is base address (typically 20000000)
bb+F0 = 00xx0000          ;where xx is self test #.
```

Table 8–7: DWBUA self test codes

Test Name	Test Code	Function
29116 RAM	01	Address processor RAM check. Last 16 locn's
DWBUA BAD	02	BAD bus/gate array
BDP/MAP ram	03	IRAM march test
BDP bus latch	04	BDB bus latches
IRAM mask	05	IRAM chip select logic
BDB addr.	06	BDB address ram and 16 locn's CPU ram
IRAM addr.	07	Data-path gate-array
TB test	08	Xlation buffer
2910 CC's	09	Condition Codes branch
29116 inst.	0A	Address Processor insts
St-End addr	0B	Writes WINDOW space start+end address
BDP wrt. msk	0C	Write mask latches BDP
WMCI/read	0D	Word BI read/write
WMCI/read	0E	Word UNIBUS read/write (UET)
IRCI/UWMCI	0F	DATIP/DATOB on UNIBUS (UET)
Uni-BI	10	UNIBUS DATI DDP
DWBUA error	11	Special case Xactions
INTR/IDENT	12	Unibus - BI interrupts

8.2.1.3.1 Common DWBUA self test codes and problems

Note

Node #0 would normally respond "0" if present and o/k.
Node #0 would normally respond "-0" if present but broke.

BA11-? switched on and functioning, cables connected, G727's present, M9313 present.

```
bb+2C = 10000000
bb+F0 = 00000001
(Self test o/k, able to test M9313 UET)
```

BA11-? switched off, cables connected.

```
bb+2C = 0
bb+F0 = FFFF0000
(Self test never started. DWBUA + UNIBUS hung.)
```

Broken Unibus continuity to M9313 but all powered up o/k.

```
bb+2C = 10000000
bb+F0 = 000E0000
(Self test "E" failed. BIIC o/k but DATIP/DATOB to/from M9313 UET failed.)
Test #E is the first UET reference.
```

8.2.2 Running Load path diagnostics

8.2.2.1 Diagnostic short-list

The diagnostics shown below are only part of the required load path set. Specific disk and tape diagnostics are still required and should be of the latest available revision as certain diagnostics were modified in order to run on the Vax 8200.

Table 8–8: KA82x Processor Diagnostics

Name	Level	Description
EBSAA	-	8200 Diagnostic Supervisor
EVKAA	4	VAX Hardcore Instruction Test
EVKAB	2	VAX Basic Instruction Exerciser
EVKAC	2	VAX Floating Point Instr. Exer
EVKAE	3	VAX Privileged Arch. Instr. Test
EBKAX	3	8200 Specific Cluster Exerciser
EBKMP	3	KA820 multiprocessor Test Program
EBDAN	2R	KA820 Serial Line Unit Func Tests
EBXMA		KA820 Multiprocessor Test Program
EVCBB	3	DWBUA Functional Diagnostic
EVKAM	2R	VAX Memory User Mode Test
EBKAX	3	MS8XX memory diagnostic
EVRLF	3	UDA50/KDB50 Disk Subsystem Diag
EVRLG	3	UDA50/KDB50 Disk Subsystem Exer
EVRLJ	3	UDA/KDB Multi-drive MSCP exerciser
EVRLK	3	UDA/KDB BBR utility
EVRLL	3	UDA/KDB/RAxx Errorlog utility
EVRLB	3	VAX RA60/80/81 Formatter
EVRAE	2R	VAX Generic MSCP Disk Exerciser
EVSBA	3	VAX standalone Autosizer
EBUCA	3	VAX 8200 EEPROM Utility Program
CI Diagnostics		See Section 6.4

See Chapter 12, VAX DIAGNOSTICS for which diagnostics should be available on site and what RX50 they're on.

8.2.2.2 Driving the console to a DS> prompt

First, obviously, get the system to the >>> prompt.
Then,

- Either `B CSA1` to get EBSAA off the DIAGNOSTIC floppy.
- Or `B/R5:10 ddxn` to get EBSAA off a local disk.
- Or `B/R5:800` to get EBSAA off a HSC disk
  You will now enter `BOOT58`
  At the `BOOT58>` prompt type `@defboo.cmd`
  After the line `S 200` type `^P`
  You will now have a typeout of the boot path register settings.
  Get `BOOT58>` again by `(>>> B/R5:800)`
  Manually enter the BOOT58 commands as output previously remembering to "OR in" 10 into R5 for the DS> boot.
- On a good day the DS> banner should appear. Even on a cluster! Don't forget to "ATTACH" the disk etc. and set load as outlined in Section 6.4, CI Diagnostics .

8.2.2.3 CIBOO.CMD file used by BOOT58 to boot from a HSC disk

```
!
!Boot an HSC disk
!
D/G 0 20            ;CI type device
D/G 1 'BI_node'             ;CIBCI/CIBCA node #
D/G 2 'CI_node' ;HSCxx node #
D/G 2 0x0y          ; x is non-zero. Use HSC y or x.
D/G 3 'unit'        ; Disk number.
D/G 4 0                     ; not used. (LBN)
D/G 5 r0000000  ; r is root (SYSr). DS>= r0000010
D/G E 200           ; Good memory from 0.
LOAD VMB.EXE/START:200
START 200
```

8.2.2.4 Testing second CPU on 83xx

After the Diagnostic Supervisor banner.

YOU ARE TALKING TO THE PRIMARY CPU. (The secondary is IDLE)

```
DS> BOOT <n>                    ;Tell DS> to boot the attached node <n>
DS> Prompt appears

YOU ARE TALKING TO THE ATTACHED CPU.
                        (The primary is just routing the console data)
```

You can now run diagnostics as "normal" using the secondary CPU. I believe this MAY fail if EVSBA was run previously. It is safe to run EVSBA on the secondary however.

8.2.2.5 EVKAA

Insert floppy "vax 8200 CPU pt1" in csa1: and

```
>>> B CSA1
```

8.2.2.6 EBKAX uses

It tests . . .

- Memory
- Power Fail
- SLU's
- CPU oddities

Boot the supervisor. Use EVSBA to size the system. Select KA0 and run EBKAX. You could specify just /sec:mem.

```
DS> ATTACH KA820 HUB KAn 'memsize' 'id'
DS> ATTACH SLU KAn TCp0 'EXTLP' 'rate'
```

n	KA unit #. (KA0)
memsize	(8192) if 8 MB
id	BI node of CPU
p	A or B or C. For TCA0 etc
EXTLP	external loop
rate	Baud rate. Unused

TESTS (V 1.5)

1-24	CPU
25-31	MEMORY
32	MACHINE CHECK
33-39	RCX50
40-42	SLU's
43-44	EEPROM
45-47	CACHE
47-50	WATCH CHIP
51	CONTROL STORE
52-53	INTERRUPT
54-56	MULTI CPU LOGIC
57-58	PACKET RAM

SECTIONS: MANUAL, EXERCISE, POWER FAIL, RPB, MEMORY, DEVICE (KA820,RCX50,SLU'S)

EVENT FLAGS:

1. SBE's give HARD ERROR report.
2. Use CACHE on memory test.
3. Write to L/H RX50.
4. Write to R/H RX50.

8.2.2.7 EBKAX failure on 8x50

EBKAX version 1.5 will get a "Hard error while testing KA0. More than one RX50 interrupt received." error in test 39 when run on a KA825 CPU. Version 1.3 does not contain the tests that check the RX50 therefore this version will not see the problem.

8.2.2.8 EBUCA the EEPROM Utility Program

The EEPROM Utility Program (EBUCA) can be used by the customer or Field Service to change site dependent parameters within the EEPROM such as baud rate, default boot device, or load new boot code, new microcode patches, or examine the VAXBI configuration.

This Utility runs as a Level 3 diagnostic under the Vax Diagnostic Supervisor (VDS), and should be available to the customer at all times.

Extensive help is available within the diagnostic. But lots of problems have occured due to incompatability twixt revisions of EBUCA, EBSAA and .PAT files.

8.2.2.9 Upgrading ucode patches

Patches to fix various bugs in the KA82x CPU ucode live in the EEPROM. From time to time it is necessary to upgrade these patches. This is done using EBUCA to copy a patch file, called KA00xx.PAT, from an RX50 into the EEPROM. The 'xx' tells you the revision of the .PAT file (in Decimal) and is equivalent to the patch revision field of the SID. So you can examine the SID to see which patch revision you currently have. See Section 8.2.7.

As mentioned above, using EBUCA can be tricky. If you are not familiar with it **do not** attempt a ucode patch upgrade without talking to someone who is. Two points to remember are:

1. The patches only get transfered from the EEPROM into the actual Control Store RAM when the CPU is initialised. So after an update, make sure you hit RESTART before doing a compare.

2. On 83x0 systems you have to update **both** CPUs!

The latest patch revision is KA0029.PAT. This gives a SID of 05283B14 for an 8200 and 05903B14 for an 8250. Floppies containing this and compatible versions of EBUCA etc. are kept in the Datadoc Media Cupboard at Welwyn. If you need to generate your own see Section 8.2.3.5.

8.2.3 RX50 floppies

8.2.3.1 RX50 Formating

RX50's come pre-formatted. Only a PC100 can format them from scratch. What this means is that the diskette is divided into cylinders/tracks/sectors and header codes are written at the beginning of each sector to identify them. The floppy is now blank in the sense that it has no directory structure or user data on it yet.

8.2.3.2 CONSOL floppy

The CONSOL floppy is basically RT11 with a slight variance in the bootblock area and boot flow. You use $ @SYS$UPDATE:CONSCOPY to make copies of the CONSOL floppy, and $ EXCHANGE to replace the floppy files etc.

$ MC WRITEBOOT is used to point the BOOT ROM program to the BOOT58 program on the CONSOL floppy. Use FILENAME of BOOT58, VBN of 1, and LOAD ADDRESS of C000.

8.2.3.2.1 CONSOLE floppy version

To see what version CONSOLE floppy diskette you have . . .

```
$ EXCHANGE DIR CSA1:
```

There will be a file like "FG81.VE0" this means part # BE-FG81E- is revision "E0" and that it is RX50 "FG" number 81. For a full breakdown of what diskettes hold what programs see Chapter 12, VAX DIAGNOSTICS or EVNDX on fiche.

8.2.3.3 Building a EBSAA or EVKAA bootable floppy

The DIAGNOSTIC floppies are the normal VMS (ODS2) structure. To make an RX50 bootable you must use writeboot.

```
$ INIT  CSA1: DIAGS
$ MOUNT CSA1: DIAGS
$ CREATE/DIRECTORY CSA1:[SYSMAINT]
$ COPY/CONTIG EBSAA.EXE  CSA1:[SYSMAINT]*                -- OR EVKAA
$ MC WRITEBOOT
        Target system device?  CSA1:[SYSMAINT]EBSAA.EXE -- OR EVKAA
        VBN of boot file?      2                        -- SAME
        Load address?          10000                    -- OR 200
$ DISMOUNT CSA1:
```

8.2.3.4 Building a diagnostic floppy

For example, to build an RX50 with RA diagnostics (what makes us think that Stefan wrote this?). Note that you would not need a WRITEBOOT sequence as that is only required for a bootable RX50.

```
$ SHOW DEVICE CS
```

If there are no CSxx devices . . .

```
$ SET PROC /PRIV = CMKRNL
$ MC SYSGEN
SYSGEN> Connect Console
SYSGEN> ^Z

$ INIT  CSA1: RA
$ MOUNT CSA1: RA
$ COPY  EVRL*.exe CSA1:[SYSMAINT]*
$ DISMOUNT CSA1:
```

8.2.3.5 Building a ucode patch upgrade floppy

You can copy an existing one (from the Datadoc Media Cupboard, for instance) using BACKUP/PHYSICAL. Otherwise, to build one from scratch . . .

```
$ INIT CSA1: PATCH
$ CREATE/DIR CSA1:[SYSMAINT]
$ COPY/CONTIG EBSAA.EXE CSA1:[SYSMAINT]
$ COPY/CONTIG EBUCA.EXE CSA1:[SYSMAINT]                      (Use latest versions)
$ COPY/CONTIG KA00xx.PAT CSA1:[SYSMAINT]
$ MC WRITEBOOT
        Target system device?   CSA1:[SYSMAINT]EBSAA.EXE
        VBN of boot file?       2
        Load address?           10000                        (Makes floppy bootable)
$ DISMOUNT CSA1:
```

8.2.4 Using BOOT58 to recover from loss of VMB on system disk

VMB.EXE sometimes gets moved during Purges or Software updates. It may be some time before someone reboots... the symptom is any type of hang/crash. If you suspect this as a possible no-boot reason, get BOOT58 from CSA1:, stuff up the registers and then run VMB.

- To get the BOOT58> prompt you need floppy #81 in CSA1 and >>> B/800 CSA1
- You MUST use UPPERCASE or BOOT58 gets confused (so will you !!).

COMMANDS available under BOOT58

- BOOT58> BOOT CI ;submits CIBOO.CMD
- BOOT58> DEPosit /type /size / 12345 77777 ;Put 12345 into 77777
- BOOT58> EXamine /type /size / 77777 ;examine 77777
- BOOT58> HELP
- BOOT58> LOAD 'file.ext' /STart:'location'
- BOOT58> STart 'location'

8.2.5 To check whether WRITEBOOT has been run on a system disk

First, determine the location of VMB.EXE :-
$ DUMP /HEADER VMB.EXE

Look in this output, at the the mapping information, and see what LBN the file starts at. This must then be converted from Decimal to Hexa-Decimal. Then check block 0 of the Disk.

This example is for a system disk where VMB.EXE is at LBN 00052B72 hex.

```
$ SET TERM/WIDTH=80
$ DUMP /BLOCK=(START:0, END:0) 'device:'

xxxxxxxx xxxxxxxx 2B720005 xxxxxxxx  ..............  00000000
xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx  xxxxxxxxxxxxxx  00000010
etc.
```

8.2.6 Using WRITEBOOT to restore VMB pointers on the system disk

Verify the file is CONTIGUOUS using `$ DIR/FULL`, then . . .

```
$ MC WRITEBOOT  (or $ RUN SYS$SYSTEM:WRITEBOOT)
```

and answer the questions.

```
FILENAME?
VBN?
Load Address?
```

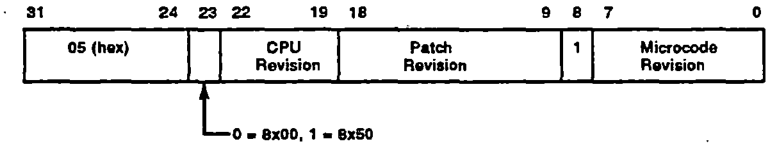
Table 8–9: WRITEBOOT parameters

Filename	VBN	Load address	Description
EBSAA.EXE	2	10000	Supervisor
BOOT58.EXE	1	C000	Boot utility
EVKAA.EXE	2	200	Hardcore
VMB.EXE	1	200	BOOT-ME program

What the WRITEBOOT utility does is to munge some words in BLOCK #0 of the device. These provide a LBN which enables the Boot Roms to find the boot file without having to understand the Directory Structure.

Bytes 0-3	Length of CONTIGUOUS file to load
Bytes 4-5	HIGH-PART of LBN
Bytes 6-7	LOW-PART of LBN
Bytes 8-B	Load Address AKA start address

8.2.7 KA820 SID



Ways of accessing the "SID" register:

1. With VMS running. (Only gives SID of the CPU your process is running on) . . .

```
$ A = F$GETSYI("SID")
```

2. With VMS running, using SDA (VMS V4). Gives both SIDs . . .

```
$ ANAL /SYSTEM
SDA> READ SYS$SYSTEM:MP_8SS.STB
SDA> EXAM EXE$GB_CPUDATA ! Primary CPU's SID
SDA> EXAM MP + MPS$GB_CPUDATA    ! Secondary CPU's SID
```

3. At Console prompt, Primary CPU . . .

```
>>> E/I 3E
```

4. At Console prompt, Non-primary CPU (via BI forward command) . . .

```
>>> Z n          ;n is the node # of the other T1001.
<esc>^P         ;wake it up.
>>> E/I 3E       ;examine the SID.
```

5. Use EBUCA. This decodes SID fields for you, so no sums are required.

Table 8–10: Example SID register values. (SID in HEX, revs in DECI)

SID	Module	CPU H/W rev	Patch rev	ucode rev
05202B14	T1001	4	21	20
05282B14	T1001	5	21	20
05882F14	T1001-YA	1	23	20
05902F14	T1001-YA	2	23	20
05283314	T1001	5	25	20
05903314	T1001-YA	2	25	20
05283B14	T1001	5	29	20
05903B14	T1001-YA	2	29	20

Note: Patch rev 25 (KA0025.pat) is minumum rev for VMS V5 SMP on 83x0 systems. Patch rev 29 (KA0029.pat) is latest (Jan 93). See Section 8.2.2.9.

8.2.8 Remote Services (RDC)

RDC connect via modem and the RSC (MDS01-AA). See DATADOC Chapter 1, VAX 6000 CALYPSO series for details of this equipment.

8.3 VAXBI Information

8.3.11 BI Modules and device codes

The device type (DTYPE) register is at bb+00: <31:16> gives the device revision. <15:00> gives the device type. <15> = 0 for DEC node, 1 for non DEC node, <08> = 0 for memory node, else = 1.

NOTE: See Section 8.5 for associated BI part numbers.

Table 8–11: BI Module List

Module	Device	Device code	Description
T1001-00	KA820	0105	82x0 CPU
T1001-YA	KA825	0105	83x0 CPU
T1002	half KDB50	010E	4 port SDI interface
T1003	half KDB50	010E	4 port SDI interface
T1008	MS820-A	0001	2 MB Memory
T1010	DWBUA	0102	BI to Unibus adapter
T1012	DMB32	0109	Serial (8x), parallel and printer port
T1014	KLESI-B	0103	STI (TU81) port
T1015	half CIBCA-A	0108	BI to CI adapter
T1017	Half CIBCI	010B	BI to CIPA (CI750 box) adapter
T1018	Half CIBCI	010B	BI to CIPA (CI750 box) adapter
T1019	MS820-B	0001	4 MB Memory
T1020	NBIB	0106	BI to NMI (88/87/85x0 system bus) adapter
T1022	DRB32-M	0101	Parallel port
T1023	half DRB32-W	0101	Used with T1022
T1024	half DRB32-E	0101	Used with T1022
T1025	half CIBCA-A	0108	BI to CI adapter
T1026	half DRB32-C	0101	+ T1022 = Super Gateway
T1030	KA800	010C	Real time accelerator
T1031	KFBTA	410D	RD5x, RX50 interface
T1032	DEBNT	010F	NI interface replaced by DEBNA
T1034	DEBNA	410F	NI + TK50 interface
T1034-YA	DEBNI	0118	NI interface only
T1035	TBK70	410B	TK70 interface
T1040	MS820-C	0001	16 MB Memory
T1043	XBIB	2107	BI to XMI (6xx0 system bus) adapter
T1044	DHB32	0109	Serial port (x16)
T1045	half CIBCA-B	0108	BI to CI adapter
T1046	half CIBCA-B	0108	BI to CI adapter
T1060	PEM	—	9xx0 Power and Environmental Module
T2050	SCM	0121	9xx0 Scan Control Module
T2051	SPM	0120	9xx0 Service Processor Module
Made by Megatape Corporation, Sold by DEC:			
2R-KZBSA-AA	KZBSA-AA	????	Differential BI to SCSI Tape interface
2R-KZBSA-CA	KZBSA-CA	????	Single ended BI to SCSI Tape interface
Made by CMD Technologies Inc., Sold by DEC. (aka CBI-1000):			
2R-KZBSA-BA	KZBSA-BA	????	Differential BI to SCSI optical disk interface
2R-KZBSA-DA	KZBSA-DA	????	Single ended BI to SCSI optical disk interface

8.3.12 Backplane replacing

Backplane replacement by level 1 engineers is not recommended.

A backplane swap needs a tool kit 29-25608–1 and 5/16 AF socket preferably 1/4 drive so it can attach to the long rod in the kit.

8.3.13 VAXBI Module Handling and packaging

VAXBI modules contain many components, specifically MOS chips, that are prone to damage from Electro-Static Discharge (ESD). In order to minimise the risk of ESD damage to these modules, the following repair tools are in place and must be utilised:

- The conductive container can be ordered via the following DEC part number: 34-24071-01
- Module bar code (including CS rev) visible through window.
- There are three coloured seals in use . . .
  Black: Repaired modules from MRC.
  Yellow: Used in fault finding, didn't fix fault. +L-07032-02
  Red: the module is defective. +L-07032-01
- The revision of the module (CS rev) is on a bar code label stuck to the component side.
- **USE ANTISTATIC STRAPS (INSIDE FRONT DOOR) WHEN WORKING NEAR MODULES.**

8.3.4 Backplane and edge connector cleaning

The backplane will not get dirty, therefore it should not need cleaning. The way we will prevent it getting dirty is by cleaning the edge connectors of any module prior to inserting it. There was trouble with the old gold-wipes leaving too much residue and it was suggested at one time to always burnish with a Kimwipe after using a gold-wipe.

This is from Peter Beddall - Product and Technology Group 07-Feb-1990

Last year there were some changes made to the cleaning practices and materials for VAXBI, NMI and XMI modules and backplanes. The OLD materials contained a lubricant like oil, which allowed contaminants to stick to any newly cleaned contact surface areas, possibly making the original problem worse.

Description	Part No.	Where used
The NEW part numbers for the cleaning materials are:-		
GoldCleans	49-01603-02 *	BI/NMI/XMI Modules
5 Segment paddle wipes	12-26321-05 *	BI/XMI Backplanes
4 Segment paddle wipes	12-26321-06 *	NMI Backplanes
Existing parts, still valid are:-		
Paddle wipe handle (long)	47-00116-02 *	BI/NMI/XMI Backplanes
Paddle wipe handle (short)	47-00116-01	BI Backplanes
Solvex Gloves	29-26403-00 *	Hands!
The OLD part numbers **that must no longer be used:-**		
Gold-wipes or Gold Electrowipes	+L-10378	
5 Segment paddle wipes	12-26321-03	
4 segment paddle wipes	12-26321-01	

* These parts are contained in Long Lane's Kit 9, "Backplane Cleaner", and Newmarket's Kit 8, "Goldwipe".

8.3.5 Accessing BI CSRs etc

NODE #	Nodespace start(bb)	Windowspace start
0	2000 0000 to 2000 1FFF	2040 0000 to 2043 FFFF
1	2000 2000 to 2000 3FFF	2044 0000 to 2047 FFFF
2	2000 4000	2048 0000
3	2000 6000	204C 0000
4	2000 8000	2050 0000
5	2000 A000	2054 0000
6	2000 C000	2058 0000
7	2000 E000	205C 0000
8	2001 0000	2060 0000
9	2001 2000	2064 0000
A	2001 4000	2068 0000
B	2001 6000	206C 0000
C	2001 8000	2070 0000
D	2001 A000	2074 0000
E	2001 C000	2078 0000
F	2001 E000	207C 0000

For instance . . .

```
>>> E/P/L 20000000
```

would tell what module was node #0 (normally DWBUA contains 0102).

8.3.6 Determination of BI module revision

The is a bar code label on all BI modules.

```
*T100100        PRD1    NI5420002 *
```

T1001 is the module type, D1 is the revision, NI5420002 is the serial #. On the etch by the label is some more information. T1001 KA820

8.3.6.1 BIIC revisions

The BIIC is the big chip with a cooling tower in the BI corner.!

```
Examine bb+04.    <31:24> are the BIIC revision (pass). Normally 04 or 05
                  <23:16> are the BIIC type. Normally 01
```

Alternatively the part number is DC324x where x is the revision.

DC324D is a rev 4 or pass 4 chip.
DC324E is a rev 5 or pass 5 chip.

8.4 Documentation

VAX 8200/8300/8x50/8x20 Installation Guide	AZ-GN5AA-TE
VAX 8200 Owners Manual	AZ-GN4AA-TE
VAX 8200 Pocket Reference	AZ-GN6AA-TE
VAXBI Technical Summary	EB-28190-46
VAXBI Options Handbook	EB-27271-46
KA820 Processor Technical Manual	EK-KA820-TM
DWBUA Unibus Adapter Technical Manual	EK-DWBUA-TM
MS820 Memory Technical Manual	EK-MS820-TM
KDB50 Disk Controller User Guide	EK-KDB50-UG
KDB50 Disk Controller Technical Description	EK-KDB50-TD
VAX 8200/8300/8x50/8x20 Field Maintenance Print Set	MP01786-01
VAX 8200/8300/8x50/8x20 Maintenance advisory	EK-ANTMA-MG
KA820 Technical Manual.	EK-KA820-TM

8.5 Parts

NOTE: See Table 8–11 for BI module numbers.

Table 8–12: Part Numbers (by option)

Part Number	Description
KDB50 Adapter cables	
70-22492-01	Sub Assy with Cables 8ft
70-22492-02	Sub Assy with Cables 15ft
17-01092-01	Under-the-bottom cable. 2inch
17-01092-02	Under-the-bottom cable. 1/2inch
17-01096-05	Cable T1003 to IOCP 8ft
17-01096-07	Cable T1003 to IOCP 15ft
12-22246-02	Transition header frame
12-22246-03	Transition assy segment
12-22246-04	Transition retaining clip
74-31369-01	Bulkhead
DWBUA Adapter cables	
17-00631-01	Cable DWBUA to M7166 ???ft
17-00631-04	Cable DWBUA to M7166 15ft
17-00632-04	Cable assy ??
CIBCI/CIBCA Adapter cables	
See Table 6–4 and Table 6–5.	
KLESI Adapter cables	
17-01098-04	Ribbon cable 5ft
17-01098-06	Ribbon cable 8ft
17-01098-05	Ribbon cable 15ft ???
DEBNA/DEBNK Adapter cables	
17-01601-02	8ft internal ethernet cable
17-01601-03	5ft internal cable
17-01601-04	15ft internal cable
17-01550-01	Internal TK50 cable + i/o connect
DMB32 adapter cables	
17-00740-xx	Ribbon cables
17-01112-01	BC19F-02 V35 cable
17-01110-01	BC19D-02 V24 cable
17-01118-01	BC19B-02 RS422 cable
17-01111-01	BC19E-02 RS423A cable

Table 8–12 (Cont.): Part Numbers (by option)

Part Number	Description
CPU/PSU/cabinet cables	
17-00632-01	KA820-KK810 CBL
17-00632-07	PRIM to CPU cable (8x20 type sys)
17-00637-01	KA820-RS232 CBL
17-00637-02	KA830-RS232 CBL
17-00681-01	FLEX CIRCUIT PSU to VAXBI
17-00682-01	POWER BUS CABLE
17-00683-01	BLOWER CABLE
17-00684-02	+15V power cable (8x20 type sys)
17-00685-01	POWER-BI CABLE (Power flex strip?)
17-00686-01	KK810-P/S CABLE
17-00691-01	KK810 CABLE
17-00691-03	PRIM to H9400 cable (8x20 type sys)
17-00714-02	CB trip cable (8x20 type sys)
17-00718-01	HARNESS DC BUS
17-00717	DC Bulk cable
17-00731-01	POWER CABLE BI-BI
17-00763-01	RCX50-KK810 CABLE
17-00763-02	PRIM to RCX50 cable (8x20 type sys)
17-00814-01	BBU SIGNAL CABLE
17-01038-01	BI INTERCAGE CBL
17-01365-01	Pwr daisy chain (8x20 type sys)
17-01378-01	DEC pwr bus cable (8x20 type sys)
17-01409-01	PSU 1 to PSU 2 Synch cable (8x20 type sys)
17-01149-01	ETHERNET boot-enable jumper
17-01637-01	Pwr flex PSU to BI #1 (8x20 type sys)
17-01638-01	Pwr flex BI to BI (8x20 type sys)
17-01639-01	Pwr flex PSU to BI #2 (8x20 type sys)
BC30A-0D	BI to BI signal cable

Table 8–13: Part Numbers (In Part Number order)

Part Number	Description
G7273	Dual height Unibus grant. (includes NPG)
G727	Single height Unibus grant. (Hand mangler)
H3033	DMB32 distribution panel
H3195	DMB32 50 pin unbalanced loopback
H3196	DMB32 50 pin balanced loopback
H3197	DMB32 asynchronous loopback
H7230-0D	BATTERY UNIT
H7231-0F	BATTERY BACK UP
H7250-00	BULK REG MODULE
H7251-00	POWER MODULE
H7253-00	POWER MODULE
H9400-AA	BI CARD CAGE
H9400-AB	BI CARD CAGE (6 slot)
M9313	UNIBUS terminator
M7166	UNIBUS paddle card
00-0RX50-AA	FLOPPY DISK DRIVE. Rev C minimum
00-00877-0B	POWER SUPPLY
12-14314-01	TWO PIN JUMPER FOR BI B/P
12-17119-01	Plastic key
12-17837-01	AIR FILTER
12-19245-01	BATTERY - NICAD
12-22246-01	TRANS CONNECTOR
12-22707-01	BLOWER CENTRIFUGAL
12-23555-01	POWER FILT.ASSY
12-23676-01	AIRFLOW SENSOR
12-23701-xx	BI NODE ID plug (01 thru 16)
12-23701-17	BI NODE ID plug (set)
12-22196-02	ETHERNET loopback
12-23676-01	Airflow sensor
20-23258-01	MOTHER BOARD
20-24486-01	TERM BI NEAR END
20-24487-01	TERM BI FAR END
30-23225-01	BA32 FILTER ASSY
30-23225-02	AC filter assy (8x20 type sys)
30-19448-02	Box of RX50 floppies
34-24701-01	ESD container
36-23595-01	Control panel label. Words
36-23595-02	Control panel label. Hieroglyphics
54-15058-00	FLOPPY CONTR. RCX50. Same as a PRO-350/380
54-16133-01	POWER DIST. BOARD. BA32 system only
54-16133-02	Power dist. board. (8x20 type sys AND BA32 system)
54-16572-01	POLARIS CONT. MOD
54-16572-04	ANTARES CONT. MOD. (8x20 type sys)
54-16574-01	SCP MODULE
54-16612-01	Polaris paddle card
70-22117-01	CTRL BEZEL ASSY
70-22128-01	DUST COVER ASSY BI
966-A	Far end terminator (what of though, Stef?)
966-B	Near end terminator
+L-13000	RDC Modem
+L-MDS01-00	MDS box

8.6 FCOs

Table 8–14: FCO summary for 82/83x0

FCO	SB#	Description
82XBX-M-001	477	New power cable prevents **Safety Hazard** due to BBU turning on during overtemp condition
82XBX-R-002	495	New Mchip (21-20850-xx) fixes a bunch of VMS problems. Takes H/W rev. to 5, Mandatory for VMS V5 SMP
83XBX-M-001	477	As 82XBX-M-001
83XBX-R-002	495	As 82XBX-R-002
Related FCOs:		
KDB50-R-001	456	New firmware fixes undetectable data corruption. Takes rev. to 15. Mandatory
KDB50-R-002	552	New firmware fixes bunch of problems. Takes rev. to 19. Essential for ULTRIX systems

8.7 Tech Tips

Table 8–15: Summary of Tech Tips for 82/83x0

Tech Tip#	SB#	Description
8200-TT-01	417	Diagnostics for 8200
8200-TT-02	454	82-8300 Front Panel Pushbutton
8200-TT-03	509	EBUCA 2.4 and diag rel. 29
8300-TT-01	454	As 8200-TT-02
8300-TT-02	509	As 8200-TT-03

8.8 Recommendations for A/R engineers

8.8.1 Preventive Maintenance

There is no PM.

Check the filter inside the front bezel during fault calls. (Filter part # 12-17837-01)

8.8.2 Air Flow

The customer should ensure that the front bezel is clear of any obstructions and is never blocked. If it is even partially blocked, the resultant decrease in air flow could cause the VAX 8200 system to shut down. Neither the top nor bottom cover can be removed for the same reason.

If you must have the box open and working, disconnect the connector to the airflow sensor. Beware the box WILL overheat. If the AIRFLOW is inadequate OR too warm on induction (exhausted by an 88xx or RAxx) the breaker on the rear of the BA32 will trip and need manually resetting. Other causes of this trip are thermal sensors in the PSU.

8.8.3 Diagnostic supervisor

On HSC sites, I recommend that A.R's create a diagnostic bootstrap command file on the console floppy. (The one normally residing in CSA1: with DEFBOO.CMD on it) This file should be a copy of DEFBOO.CMD but with '10' OR'd into R5. Its filename should be "DIAG.CMD". Use Exchange to do this. If this is done, all that is needed to get stand-alone diagnostics going is . . .

```
>>> B/R5:800
BOOT58> @DIAG.CMD
```

Local disks only need >>> B/R5:10.

The vagaries of where it is booted from is of little concern.

8.8.4 BI node map

I recommend a printout of a typical diagnostic boot and run of EVSBA be left in the SMG along with a NODE printout from EBUCA.

8.8.5 Hard-copy console

I would strongly discourage the use of a video terminal as a console for any BI based machine. This is because some of the output is irretrievable and could negatively impact fault-finding.

Take the case where a bank of memory goes "broke" only leaving, say, 2 MB left. VMS would Fatal-bugcheck and reboot and then the system would run real slow, giving a fault as "system running slow" or "hung". The fatal-bugcheck printout and the node self-test with the "bad" memory node are now scrolled off the top of the VDU. The call could take hours instead of being straightforward.

8.9 Troubleshooting

There are references in this chapter to the PSG or an assumed knowledge of it. If you don't have one GET ONE. Part # AZ-GN6AB-TE.

The level I trained engineer can replace all the FRUs associated with a failing option. After this point the fault should be escalated via the PROBLEM MANAGER. Neither Tele-support nor RDC nor District Support invoke escalation procedures. **THERE SHOULD BE A COPY OF A COMPLETE SELF-TEST RUN AND A BI NODE MAP IN THE SMG, RDC MAY ASK THE CUSTOMER TO CONSULT THESE.**

8.9.1 Power Supply

8.9.1.1 SAFETY

82XX/83XX POWER SUBSYSTEM SERVICE PRECAUTION

Be careful with the H7250.
The H7250 generates signals to drive the other regulators.
If the ILV fails the capacitors retain a LETHAL 400V DC charge.
The green LED means ILV O/K.
The H7252, H7253 are driven by the H7250.
Check the voltage on the main BUSS-BAR (across the big capacitors) before believing there is no power.
Leave three minutes between switch off (with modules present) and working on the PSU.
Switch off the AC mains and unscrew the top cover SLOWLY!!.

- **The BA32 should be powered from the SWITCHED side of a power controller.**
- **RAxx's should be powered from UNSWITCHED side of power controller.**
- **The power controller should be switched to REMOTE.**

8.9.1.2 Power supply faults

There are 3 LED's on the front right of the BA32.

Green	**normally ON**
Red	**H7251 broke**
Red	**H7253 broke**

If both RED are on then (H7250 broke).
No leds, AC mains or H7250.
H7251 +5B, -5.2, -2, +24.
H7253 +5.1, +12, -12. AND +/- 12V OR +/- 15v.
H7250 +/- 12V ILV. Has an 8A cartridge fuse.

8.9.2 BI Error Logs

VMS logs two types of errorlog for BI buss related errors. These are . . .

BI ADAPTER ERROR   This mechanism logs all BI ADAPTER registers, excluding memory
BI BUSS ERROR   This mechanism logs all BI NODES registers, including memory

The errors are detected on a per device basis and VMS is interrupted through the specific device vector into the device driver. The two types of error a device detects are Hard and Soft.

Hard is summarised by <bb+04> bit 31.
Soft is summarised by <bb+04> bit 30.
Each has an interrupt enable.
Hard is <bb+04> bit 23.
Soft is <bb+04> bit 22.
Hard errors are <bb+08> bits <30:16>.
Soft errors are <bb+08> bits <2:0>.

A node not detecting error, logs . . .

```
VAXBICSR        010108CD        NODE ID = 13.
                                ARB CONTROL = DUAL ROUND ROBIN
                                SOFT ERROR INTR ENABLE
                                HARD ERROR INTR ENABLE
                                SELF-TEST STATUS         ;passed
                                BI INTERFACE TYPE = 01
                                BI INTERFACE REV = 01

BER             00000000        ;no error.
```

A node detecting an error, logs . . .

```
VAXBICSR        04018802
                                NODE ID = 2.
                                ARB CONTROL = DUAL ROUND ROBIN
                                SELF TEST STATUS         ;passed
                                HARD ERROR SUMMARY       ;saw error
                                BI INTERFACE TYPE = 01
                                BI INTERFACE REV = 04
BER             00020000
                                NONEXISTANT ADDRESS
```

- In the above scenario there was also a M/C check logged as NODE #2 was a KA820 !!.
- The EVENT code was a NCRMC [(no ACK CONF received) (no response !!)]
- The MAR was locked consequently the MAR is valid.
- The MAR has 60002000. This translates to Physical Address 20002000 This is VAX IO_space on node #1.
- From the previous errorlog, NODE #1 wasn't mentioned therefore was nonexistant.

There may be many nodes with error bits set, by systematic examination one can pinpoint the sequence of events. Bear in mind that there can only have been ONE cause but MANY symptoms or MANY nodes having noticed. You're interested in the FIRST error. On the BI there can only be one transaction between two nodes at a time. (In normal use)

8.9.3 Machine Checks

See also the KA820 CPU tech. man. P 5-5. Or the "black-book" 1-42.

8.9.3.1 Machine check mechanism

The CPU aborts the current process and raises IPL to 1F. It then . . .

1. Sets H/W fault flag. Lights T1001 module RED LED.
2. Sets M/C check in progress flag. If already set halt the CPU and enter console mode (if enabled).
3. Push on the CURRENT stack (K or I) the Mchip registers.
4. Clears H/W fault flag .Switch OFF T1001 RED LED's.
5. Dive through SCBB + 04. (Mcheck handler).

The Mcheck handler then . . .

1. Evaluates the type of Mcheck. From Stack data.
2. Decides to recover or halt CPU based on "state of machine".

If it attempts to recover (for instance a timeout) . . .

1. Restores the machine state, from stack.
2. Writes IPR 26 to clear the machine check in progress flag.

8.9.3.2 Stack logout frame

The VMS error logger adds two longwords to the beginning. SID + Node ???.

Table 8–16: 82xx Machine Check Stack Logout

Stack offset	Meaning	Mchip register	Double Error Halt
(SP)	byte count (20†)	—	-
+4	Type code	MtempA	E/M A
+8	Parameter 1	MtempB	E/M B
+C	VA register	Mtemp13	E/M 13
+10	VA prime	Mtemp.psl.temp	—
+14	MA register	Mtemp9	E/M 9
+18	Status word	MtempC	E/M C
+1C	PC @ fault	MtempF	E/M F
+20	microPC @fault	Mtemp10	E/M 10
+24	Current PC	—	E/G F
+28	Current PSL	—	E P

†The byte count (20) does not include the PC + PSL pair

Table 8–17: Machine Check Type Codes

Bit	Meaning
8	Unused; I think this is a PORT CONTROL TIMEOUT (SF)
7	Prefetch error
6	Cache tag parity error †
5	BTB tag parity error †
4	BI error
3	Data parity error
2	MIB parity error
1	ucode error
0	IPL bad

† Cache tag parity and BTB tag parity are mutually exclusive. (Can't both be set)

Table 8–18: Machine Check Status Word definition

Bit	Meaning
31	reserved
30	Vax cant retry. The instruction already changed something somewhere
29 to 23	reserved for use by VMS
26	Use the Primary to service
25	Cache (turned?) off
24	Attempted an REI (To restart original instruction)
23	NOT KERNEL mode
22	VAXBI error. See <21:16>
21	Write memory. <22> and <21> =1 = write memory error
20 to 16	BI event code. See chart. Only if <22> = 1
15	reserved (unused)
14	BTB/CACHE DATA parity error. No other data available
13	reserved (unused)
12	Pcntl timeout. BI xaction took longer than 12.8 Ms
11:05	reserved (unused)
04	BTB tag parity error
03	MTB miss. Uses BTB instead. No interest??
02	CACHE tag parity error
01	Pcntl detected error. See <22:16> and <14> as from Pcntl CSR
00	Memory address register lock

Table 8–19: VAXBI event codes

0	No event
1	Master port xaction complete
2	ACK received for slave read data
3	Bus timeout. VAXBI error bit (Status word <22>)
4	Self test passed
5	Retry cnf received for master port command
6	Internal register written
7	Advanced retry cnf received
8	No ack or illegal cnf received for INTR
9	No ack or illegal cnf received ipintr/stop
A	ACK CNF received for error vector
B	IDENT ARB
C	External vector [4]
D	External vector [5]
E	External vector [6]
F	External vector [7]
10	Stall timeout on slave transaction
11	Bad parity received during Slave Transaction
12	Illegal cnf received for slave data
13	Bus Busy error
14	ACK CNF received for non-error level [4]
15	ACK CNF received for non-error level [5]
16	ACK CNF received for non-error level [6]
17	ACK CNF received for non-error level [7]
18	RDS
19	Illegal CNF for master port command
1A	No ACK CNF
1B	Bad parity reserved
1C	Illegal CNF received by master port for data cycle
1D	Retry Timeout. Retried 4096 times without ACK
1E	Bad parity received during master port xaction
1F	Master Transmit check error. CPU was only bus driver and data received didn't match data sent

8.9.3.2.1 Mcheck example 1

This machine check is caused by Halting the Diagnostic Supervisor, depositing -1 (%XFFFFFFFF) in the PC and continuing . . .

```
DS> ^P
?02             PC = 0002887
>>> D/G F FFFFFFFF
>>> C

?? Machine check exception through SCB vector: 04(X)
MACHINE CHECK LOGOUT:
COUNT:                     00000020(X)
MACHINE CHECK TYPECODE:    00000010(X)     ;BI ERR ❶
PARAMETER 1:               00000000(X)
VIRTUAL ADDRESS:           000417F8(X)
VIRTUAL ADDRESS PRIME:     00041800(X)
MAR:                       7FFFFFFC(X)
STATUS:                    405A0003(X)     ;VCR, BI_ERR, BI_EVENT=NCRMC ❷
                                           ;PC ERR ❸, MAR LCK❹
PC AT FAILURE:             FFFFFFFF(X)❺
UPC AT FAILURE:            00000199(X)
PC at error:               00027CFF(X)
PSL at error:              00000001(1)
```

❶ BI ERR means a BI xaction failed.

❷ NCRMC means No Ack Confirmation Received.

❸ PC ERR means Port Controller saw error (Implies bit 22 - BI ERROR and/or bit 14 - CACHE DATA PARITY ERROR is valid)

❹ MAR LCK means the MAR (memory address register is locked, i.e. valid)

❺ The PC at failure was -1 (%xFFFFFFFF) which is what we deposited there! (This corresponds to the value in the Memory Address Register)

8.9.3.2.2 Mcheck example 2

This was extracted from a Forced-crash solicited due to an apparently Hung CPU. I later (Well Peter Beddall!) noticed the Stack pointer was pointing to a Mcheck stack frame at the time of the "hang".

```
SDA> exam 804cfb5c;40

80002F88 00000000 00000010 00000020  ................/....   804CFB5C
8000CD3A 015D0003 C0ADD188 80002F8C  ./.........À..].:î..    804CFB6C
80208351 041F0000 8000CD35 0000176D  ..m...5î......Q.....    804CFB7C
00000002 00000000 804CFBF0 80029200  .........ΩL.........    804CFB8C
```

If we separate the longwords and lay them out in the normal way, we get . . .

00000020	Count
00000010	Type ;NEX
00000000	Parameter 1
80002F88	VA ;Virtual Address
80002F8C	VA prime
C0ADD188	MAR ;BI address sent, PA. <31:30>=type 11=Octa. 01=long
015D0003	Status ;BI event=1D =Read_Timeout
8000CD3A	PC @ fault
0000176D	upc @ fault
8000CD35	PC @ error
041F0000	PSL @ error. IPL 1F indicates Mcheck in progress

VA=80002F88 translated using a PTE=F40056E8. Gives a PA=00ADD188 (The translation was done by a DCL program I have as I keep forgetting the finer points of VAX memory management!!).

8.9.3.2.3 Retry Timeout

If a node accesses a memory location and gets no response 4096 times it informs the KA820 via a BI event code 1D. The KA820 builds a stack frame and informs VMS via SCBB + 04, VMS normally restarts the faulting instruction. This could continue indefinitely if a retry timeout happened all the time. Alternatively you could get an Interrupt Stack Not Valid.

One situation where the memory could create the above situation is if a peripheral (or the KA820) had issued an IRCI BI-xaction and never followed it with an UWMCI BI-xaction. There is one memory "lock" per module, the "lock" is only cleared on an UWMCI or a COMPLETE POWER FAIL. This power fail will probably report a "software state not saved" and cause another re-boot. A >>> T or >>> B command will not disturb the "lock" bit. On two occasions I have seen this caused by a faulty CI port.

The memory lock bit is MS820 CSR1 <13>. (bb+100)

The symptom exhibits itself as bb+08 <20> being set in a node getting the "no response".

8.9.4 Pathological errors

A pathological halt occurs if the hardware is unable to continue processing instructions. No crash dump is written, nothing is pushed on the stack, the system simply stops executing instructions and prints out the PC on the console.

8.9.4.1 ?05. Double Error Halt

If the hardware is in the processing of logging a machine check and a second machine check occurs then a double error halt occurs. Information from the first error will be in the Mchip temporary registers. Information from the second error will be in the Mchip registers. If the front panel switch is set to Halt then you can use the following console commands to determine the cause of the double error halt . . .

```
>>> E/M <M chip address> ; SEVERAL needed. See Table 8-16
>>> E/G F                ;To get the PC.
>>> E P                  ;To get the PSL.
>>> E/I 0                ;Kernel stack pointer.
>>> E/I 4                ;Interrupt stack pointer
```

If the front panel switch is not set to Halt then you will LOSE all of this information. To troubleshoot this problem you can either set the switch to the Halt position and wait for another error or you can edit the DEFBOO.CMD so that it will carry out the EXAMINE commands before rebooting the system.

8.9.4.2 ?04 Interrupt stack not valid

This error occurs if an interrupt occurs and the hardware cannot use the interrupt stack because the Interrupt Stack Pointer does not point to a valid Virtual Address.

This typically occurs when too many interrupts were encountered before they were able to be handled. The interrupts are normally generated due to some kind of error. The possibility that the Interrupt Stack Pointer got corrupted somehow is very unlikely (never seen one!!). The only thing you can do is . . .

1. Set the panel switch to Halt not Reboot.

2. Wait.

3. When ?04 happens. Examine the Interrupt Stack pointer.

   ```
   >>> E/I 4
           I 00000004        xxxxxxxx
   ```

   xxxxxxxx Should be within Physical memory (not 8xxxxxxx, 7xxxxxxx).

4. Subtract 200 from xxxxxxxx giving xxxxxyyy

5. Examine the stack

```
>>> E/P/L xxxxxyyy
>>> E                         ;repeat for 200 !!!.
```

6. Scan the stack for what looks like a machine-check.

```
20
small number
...
...
```

7. If you see a machine check, analyse it. If not then contact Support, you're in trouble.

Reminder - An interrupt causes . . .

```
MOVL PC,  @-(ISP)
MOVL PSL, @-(ISP)
```

The stack looks like

```
        -          ---PC---         ;hi address
            ISP>>> --PSL---         ;lo address
                   -unused-         ;next push goes here
```

Reminder - The interrupt stack occupies some number of pages and either end there is a guard page. This guard page is Virtual addresses that do not translate to valid physical addresses. These are bad addresses.

8.9.5 KDB50 related

8.9.5.1 DU bootstrap problem

If the EEPROM utility EBUCA displays the revision of DUBOOKA (the KDB bootstrap) as "100" (revision 1.00) and you have a T1002 (KDB50) module with a pass 5 BIIC (see Section 8.3.6.1) you will be unable to boot a DU type device. The fix is to run EBUCA and update the DUBOOKA bootstrap with "101" (rev. 1.01) from VAXPAX 24+. Bear in mind you'll have to redefine the default boot device if this is done. For help on EBUCA see the 8200 users manual or run it and ask for help (its copious).

8.9.5.2 KDB50 revision levels

Examine the DTYPE register at bb+00

31 24	23 20	19 16	15 0
Microcode Revision	SDI module Revision	Proc module Revision	10E = KDB50

%x 0F1B010E would indicate.

<31:24>	ucode revision (Proms)	%x0F = 15
<23:20>	SDI module revision (T1003 jumpers)	%x1 = A
<19:16>	Processor module revision (T1002 jumpers)	%xB = K
<0:15>	010E = KDB50	

Note

As a matter of interest (do not alter!!) the T1002 jumpers are adjacent to the BCAI chip above the BI corner. The T1003 jumpers are below the four large ROMs in the middle of the module.

8.9.6 VMS V4 and 83xx's

8.9.6.1 Is the attached there/running?? V4 VMS only

Use the VMS command

```
$ SET PROC /PRIV = CMKRNL
$ SHOW CPU,
```

"Device is offline" means the secondary does NOT exist, was BROKE, or didn't exist at boot time.

"Attached is in the Exec State" means the secondary is there and functional.

To check on what the secondary is doing use the VMS command `$ MONITOR MODE`, if you receive a split screen display showing PRIMARY and SECONDARY all is well. If the screen is not split then try `$ SHOW DEVICE KA` You should see KA0 and KA1. Try using `SYSGEN> SHOW /CONFIG` and see if another KA820 appears and verify all the other nodes are present by BI node #.

During the AUTOCONFIGURE on power up SYSGEN should see the other CPU and add it into the database. All that remains is to check that a `$ DEFINE /EXEC MP MP_8SS.EXE` has been done followed by a `$ START/CPU` Don't worry that the secondary is IDLE a lot. The secondary ONLY runs processes that run in USER mode ONLY it can NOT do I/O to devices OR switch to KERNEL mode, if it tries the process is given back to the primary. To make the secondary 100% busy is quite simple, create a DCL procedure that consists of the following . . .

```
$ LOOP:
$ GOTO LOOP       !Do absolutely nothing at all...
```

And submit it as normal (`$ @FILE.COM`). `$ MONITOR MODE` from another terminal should now indicate the secondary 100% busy and the primary as normal.

8.9.6.2 Secondary machine checks and VMS V4

It is possible for VMS to log a machine check for the secondary and let it cease to function with only the loss of one process. The secondary can log NO other errors. Anyone coming across a machine check logged where the machine appeared to continue should look CLOSELY at the machine check entry for the following . . .

1. The SID register agrees with Primary or Secondary they, unfortunately, may be the same.
2. The BI node number that the error was logged against is probably correct.
3. Whether the CPU was running in KERNEL or other mode (from the PSL). Kernel MUST be the PRIMARY other could be either.
4. Consult support if you are not sure.
5. If you find what looks like a SECONDARY machine check, I'd like to see it, as the last one I saw was VERY confusing. (Stefan F.)

Note

VMS V5 appends more data to the error log which uniquely identifies which CPU was at fault.

8.9.7 Hints'n'tips

8.9.7.1 SCBB and IPL levels see PSG

8.9.7.2 Accessing the secondary CPU from >>>

To talk to the attached CPU for the purposes of troubleshooting etc. use the BI-forward command.

```
>>> Z <node>      ;where <node> is the node number for the attached.
```

All subsequent text goes to the <node> except ^P and <esc>. To send ^P or <esc> prefix them with an <esc>, for instance . . .

```
>>> Z 4
<esc>^P
?02
        PC = 00000700
>>> T
#ABCDEFGHIJK..N#
```

Note that on the attached CPU the RCX50 test is disabled because it has no RCX50. To disable the RX50 tests see the 8200 PSG. Use EBUCA.

8.9.7.3 Module Disable by software

To disable a node without touching it . . .

```
>>> D/P/L <bb+04> 00000000
```

This survives a self-test and a >>> B command. It removes everything from the BIIC into the module, all that is left are the "idle" BI BIIC transceivers.

For memories the self-test finds what is remaining and writes a contiguous starting + ending address into them. The net result being one array missing.

8.9.7.4 CPU Red LED

This LED is On if . . .

1. In Console mode >>>
2. In the process of handling a Machine-Check.

8.9.7.5 BI nodes as NEXUS

Each BI node is classed as a NEXUS and consequently should be visible under SYSGEN with the SHOW /NEXUS command which obtains its information from the database built by AUTOCONFIGURE/ALL/LOG from the data supplied in the BIIC DTYPE register at <bb+00>.

8.9.7.6 BROKE BI nodes

Any node that has the BROKE bit set (and the RED led on), does not exist on the bus. If you suspect this check the console output on the last boot. (See Section 8.2.1).

8.9.7.7 Manual use of TEST or BROKE

Deposit into bb+4 the following . . .

1. 1000 To force the node to SELF-TEST. Only that node will test itself.

   Note

   Check the BROKE bit or the RED LED for the status.

2. C00. To BROKE the node, the response from the console will be ?4C (Non-exist register). Further examines also give ?4C. To get the node back use the front-panel "INIT" or instigate a power-fail.

   Note

   VMS will be unable to un-broke the node consequently it will AUTOCONFIGURE without it. Also the CPU self-test (>>> T) Will NOT clear it.

8.9.7.8 Self Test Failure

82/83x0 systems have four terminal ports connected to the primary processor module in slot K1J1. One is dedicated to the console. If a terminal or a modem is connected to one of the other three ports, the self-test of the module can intermittently fail thus

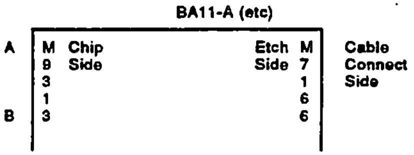
#ABC

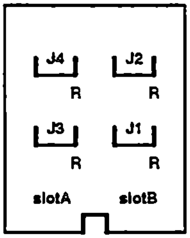
?4C

This is due to test D, the MChip test, doing a loopback test on all 4 UART's. The use of these 3 extra terminal ports The use of these 3 extra terminal ports should be discouraged.

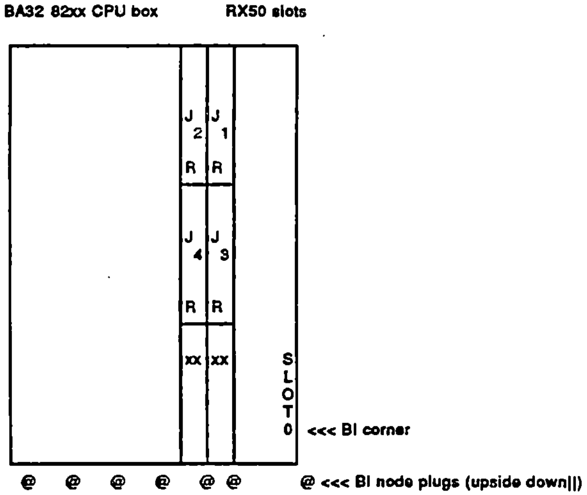
8.9.7.9 DWBUA Cabling

M7166 Unibus A-B slot paddle card.





Red stripes to the right.
Looking at the "1" side (normal chip side)



R = Red stripe. xx = Unused.

The cable loom supplied will only go one way round. The connectors are keyed and marked.

8.9.8 Checklist for engineer in trouble

1. EBSAA must be 10.0 for CIBCI.
2. KDB50 with T1002 rev L (pass 5 BIIC) needs DUBOOKA.SYS rev 101 and 55 block VMB.EXE.
3. DECNET BOUNCE?. DWBUA must be rev F. See Comms chapter
4. Clean new modules before installation. Follow procedure.
5. BA11-A's can trip prematurely. Fix is rev E1 H7204-BB.
6. Don't force the module lock cam.
7. There are some modules with a bad T slot, but how do you know?
8. CPU self test can fail "K" due to a bad console terminal.
9. Set process/cpu=noattached stops process from running on attached.
10. Bugchecks?. Check the M-chip FCO has been done.
11. Use breaker for power off, not the keyswitch.
12. TU81 AUTOload causes KLESI-B to fail selftest.
13. TU81+ dead PSU breaks KLESI-B, pull cables out to check.
14. DMB32 needs distribution panel connected to pass.
15. DMB32, SIA0 only appears with correct cable installation.
16. DMB32, LIA0 only appears with correct cable installation.

    To connect an LP25/LP26 to a DMB32 you need the following configuration.

    DMB32
    Bulkhead connector
    BC27A (30ft maximum)
    29-23396 (DP 257340-004) Short Line Harness
    29-23438 (DP 257345-002) Short line interface module

17. KDB50 fails self test if no -5.2V.
18. Intermittent -5.2V cause SDI funnies.
19. Primary CPU resides in first slot, if not present nothing works.
20. The console is hard-connected to the Primary CPU.
21. If the Primary won't converse with the console but auto-boots VMS o/k its probably the cables between the front panel/PCM/CPU.
22. MS820-C 16MB need KA0023.PAT and self test timeout @20 seconds.
23. Always use the latest revision of diagnostics or CONSOLE software see Chapter 12, VAX DIAGNOSTICS.