Chapter 4

11780 and 11785

This chapter is written by Pete Griffin. Originally it mainly contained information to familiarize a 11780 engineer with the 11785. I've re-written it for two reasons.
Firstly, to comply with the current format needed by the DataDoc publishing software.
Secondly, to accompany a ground-level seminar for non-trained engineers who will be responding to calls on the offset rota.
This is not a replacement for the Black Book given out on the L1 course or the VAX Hardware Handbooks available from the sales library. It is intended to be a collection of the minimum information needed, along with some practical troubleshooting advice.

Although the 11780 was the original VAX, we do still have enough around to keep us aware of them.
The 11782 was a ASMP configuration, consisting of two 11780s each with an SBI MA780 interface to a cabinet of common memory.
The 11785 was a "mid life kicker" giving about 50% more power.
To the engineer there are subtle differences in :-

The Control Store layout and control.
G and H mechanism.
The LSI configuration.
Organisation of the files on the floppy.
Implementation of the Boot command.
Diagnostics.

The delivered 11785 was also in a different style of cabinet, this is registered as FCC compliant. Because of this a non FCC 11780 cabinet which has been upgraded to a 11785 has to be labelled 11780-5, instead of 11785.

4.1 Engineer's view of system blocks

Figure 4–1: Functional block diagram

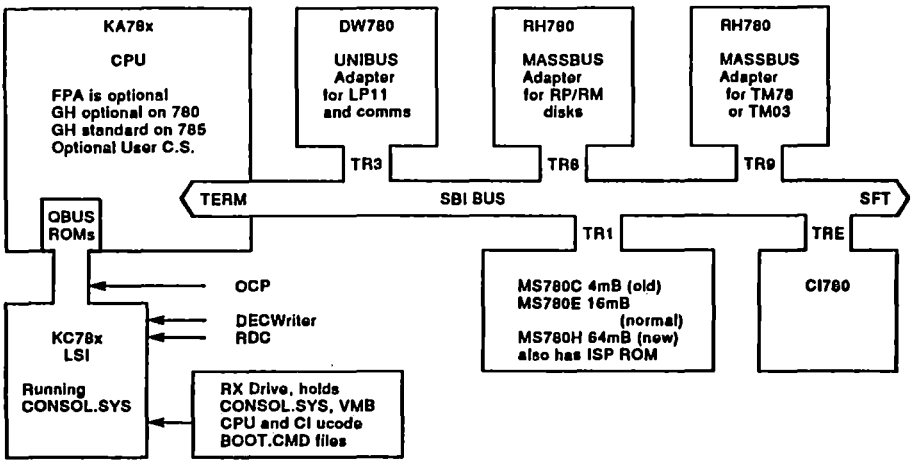
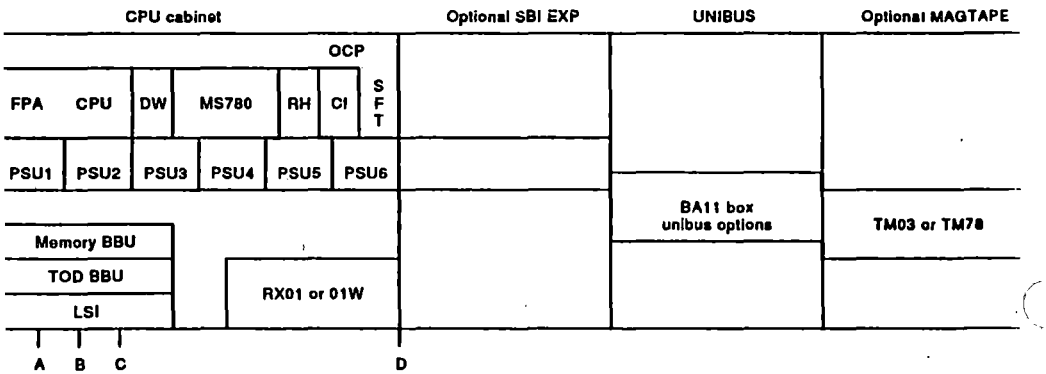


Figure 4–2: System layout



A – Round the back, Transition piece for ROUND massbus cables to disk or TM78.
B – Round the back, CI cable connections.
C – Round the back, transition pieces for Decwriter and RDV modem.
D – In cabinet side, Transition piece for UNIBUS, FLAT massbus to TM03 and odds for the optional SBI expansion cabinet.

4.1.1 Cabinet

- There are LABELS on the doors or sidewalls which show the module utilisation, and give the revision.
- The MEMORY PSU and BLOWER remain on all the time, you have to switch off the individual memory psu to replace modules. The 3-phase breaker around the back will switch it all off.
- AIRFLOW is downwards over the modules, through the power supplies and out the rear.
- The usual airflow and temperature sensors will switch off the system.
- Watch out for the soundproofing in the exhaust coming unstuck and blocking the airflow. Also the door filters must be reasonably clean.
- The exhaust box is removable, this is needed to get at the LSI brackets.
- A "delayed bus" feeds the UNIBUS power controller.

4.1.2 Power supplies

The main power supplies are "modular". We order several variants, which consist of a mother assembly fitted with some combination of internal regulators:-

PS#1,3 are H7100B's (just the basic H7100)
PS#2 is a 70-14956-01 (basic H7100 plus H7101)
MS780C uses a 70-14957-01 (basic H7100 plus H7102 and H7103)
MS780E uses a 70-18759-01 (basic H7100 plus H7103 and H7107)

The total current drain on a psu made up of any combination of regulators must not exceed 100A. the individual ratings are:-

H7100 (+5 at 100A)
H7101 (-5 at 30A)
H7102 (+12V at 10A)
H7103 (+5V at 20A, -5V at 0.2A)
H7107 (+5V at 20A, +12V at 1A)

PSU#2 feeds the SBI transcievers on ALL the NEXI as well as the System-Far-end-Terminator.
PSU#2 also feeds the DW780.
PSU#1 is only fitted on systems with an FPA.
Other supplies present in the system are the memory BBU (H7112) and the TOD BBU (H7111).

A PSU only feeds ACDCLO to the same backplanes it is feeding DC to. The ACDCLO from PSU#2 has to go down to the SFT as well as the CPU and DW.

4.1.2.1 Memory BBU description

The memory BBU is rare (and information is even rarer). The LED only comes on during the 'fast-charge' period. In the event of a power failure, the H7112 sees ACLO from the memory power supply, the H7112 will then put 36V out to the H7100. The H7100 then provides +/-5VB and -12VB to the memory during a power fail.
The 30V can be checked by measuring on pin 1 and 2 of J7, these read 30V when everything is switched on normally.
Other things that stop the batteries working are:- the switch must be ON (this has 2 poles, one for mains to charger and the second is part of an interlock loop), also J7 pin3 must be joined to pin2 by the innards of the H7100 (also part of the interlock), finally the thermal cutout (J10 or J11 pin2) must not be grounded.

4.1.2.2 869E power controller

The 869E has several signal connections which allow exceptional events to cause a SHUTDOWN, as well as Power Control from rthe OCP :-

- J09 - TOTAL SHUTDOWN, grey wire from overtemp sensor in every psu.
  Causes circuit breaker to trip out. There is also sensor in the 869E.
  p1 is ground, p2 is signal.
- J10 - UNSWITCHED SHUTDOWN, from airflow sensor for memory fan.
  Causes circuit breaker to trip out.
  p1 is ground, p2 is signal, p3 is n/c, p4 is +15V.
- J08 - SWITCHED SHUTDOWN, from airflow sensors for switched fans.
  A ground on p2 stops mains to switched outlets.
  p1 is ground, p2 is signal, p3 is +15V, p4 is n/c.
- J04, J05, J06 - POWER CONTROL BUS,
  p1 is POWER REQUEST, a ground here will put mains to the switched outlets as long as the 869E is set to REMOTE
  p2 is EMERGENCY SHUTDOWN, a ground here will over-ride p1.
  p3 is ground.
- J07 - DELAYED OUTPUT, provides a 0.5 sec delay to POWER REQUEST signal.

And several mains outlets:-

PHASE 1 SWITCHED - goes to FAN1, FAN3, PSU2, PSU5.
PHASE 2 SWITCHED - goes to PSU1, PSU3, RX fan.
PHASE 3 SWITCHED - goes to RX01, LSI
PHASE 1 UNSWITCHED - goes to PSU4, FAN2
PHASE 2 UNSWITCHED - goes to TOD BBU, Memory BBU.
PHASE 3 UNSWITCHED - not used.

4.1.2.3 Fault conditions

An ACDCLO cycle will cause the LSI to reload the consol software from the floppy, report the defaults, reload the main cpu's microcode and then carry out the action determined by the OCP.

ACDCLO timing is done on the CLK module.
A sequence is triggered by PSU 1, 2 or 3, the TOD BBU, or the LSI PSU.
Also an SBI signal can trigger the sequence, but only memories and the SFT can do this.
It is unusual to have a solid PSU fault that is not shown in the regulator LEDs, but you may have to look at the signal inputs to the cpu.

ACLO is the yellow wire, DCLO is the mauve wire. -9V is a normal level (power is o/k), 0V means asserted (power is bad).
The QBUS BDCOK is at B29A1, QBUS BPOK is at B29B1. 0V is bad, +3.5V is good.
The TOD DCLO is at A08F1. 0V is bad, +3.5V is good.
SBI FAIL is at B16T2, SBI DEAD is at B16N1. 0V is bad, +3.5V is good.

Although the sequence is generated on the CLK module, the CIB is also involved.

The worst case is obviously the intermittent one.
Most times it's the LSI psu (this is the most difficult one to replace).
I made a ACDCLO monitor, which is known by logistics as "78085SUPP".

4.1.3 SBI

Termination is needed at both ends. The System Far-end Terminator (SFT) gets its DC from PSU#2. If an SBI expansion cabinet is fitted, this cabling goes thru a special transition piece. To avoid introducing a broken line on a SBI cable, always maintain the cable's position and orientation if you are moving them around.
Also, if you get new SBI cables from Logs and you have to "form" them yourself, the word SIGNAL should be on the outside.
The SBI TR number defines the base address of a NEXUS (in I/O space) and its SBI arbitration level. Each NEXUS has jumpers and a wire wrap to set it's specific TR number. TR 0 is special.

4.1.3.1 Addressing

The system has a 30 bit address space (as normal on subsequent Vaxen). So 00000000-1FFFFFFF is used for the memory RAM space and 20000000-3FFFFFFF is used as I/O space. The I/O space is broken down thus:-

- NEXUS register spaces
- Windows to four Unibus spaces

The base address of the registers in each NEXUS is governed by its **TR number.**

Table 4–1: NEXUS registers, base addresses

TR	Base Address	standard use
TR 1	20002000	first memory, ISP ROM is at 20003000
TR 2	20004000	second memory, ISP ROM is disabled
TR 3	20006000	mandatory DW780
TR 4	20008000	optional second DW780
TR 5	2000A000	
TR 6	2000C000	
TR 7	2000E000	
TR 8	20010000	first RH780
TR 9	20012000	second RH780
TR A	20014000	
TR B	20016000	
TR C	20018000	
TR D	2001A000	
TR E	2001E000	CI780

The base address of the Unibus associated with any DW780 is governed by the **UBA NUMBER** (nothing to do with TR number) which is set up in the DW jumpers.

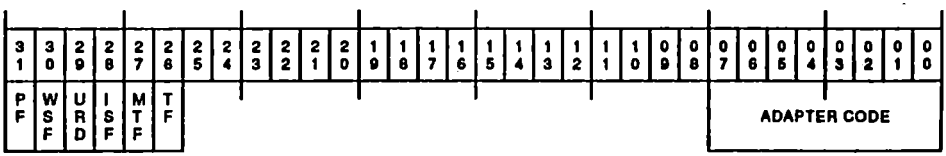
Table 4–2: Windows to Unibus, base addresses

UBA Number	Base Address
0	20100000
1	20140000
2	20180000
3	201C0000

4.1.3.2 SBI NEXUS Configuration and Status Register

The first address in each nexus is called the CSR and is almost standard. It contains SBI related error bits and a device identification code. Every Nexus monitors the SBI for 5 different types of error. If an error is detected the Nexus asserts the SBI FAULT line. The SBI FAULT line causes several things to happen to assist us. Every Nexus freezes its Configuration Register, the CPU locks the SBI silo, and the CPU enters the VMS interrupt routine which will result in an error log entry.
All the CSRs could be easily examined if required using the EXAM command,
e.g. for DW at TR3 do a >>>EXAM 20006000.
On EXAM/DEP you may get a MIC-ERR if the SBI is broken, you would then have decode the reported ID registers to analyse the problem.

Figure 4–3: Nexus SBI error and device type bits in CSR



4.1.3.2.1 D31 - SBI PARITY FAULT

Every Nexus continually monitors the SBI for even parity. So if a Nexus sees bad parity we ask ourselves things like "did they all see it?"

4.1.3.2.2 D30 - WRITE DATA SEQUENCE FAULT

A Nexus saw an SBI TAG saying "write" and its own address, BUT, this was not followed by the data. TR ACTIVE in the SBI silo will show which Nexus was doing the "write" and if it followed up with the data.

4.1.3.2.3 D29 - UNEXPECTED READ DATA FAULT

A Nexus saw an SBI TAG of 0 (which means someone pushed some data out) and the SBI ID was his (the data is for him), BUT, he don't remember asking for any data. TR ACTIVE in the SBI silo will tell us who sent this data.

4.1.3.2.4 D28 - INTERLOCK SEQUENCE FAULT

A Nexus received an "interlock write" command which it did not expect (it did not get an "interlock read" earlier). The SBI silo will tell us who (if anyone) sent the interlock write and if SBI INTERLOCK is set due to an earlier interlock read.

4.1.3.2.5 D27 - MULTIPLE TRANSMITTER FAULT

A Nexus has arbitrated for use of the SBI and decided it was his turn, BUT when he came to transmit he found the SBI ID lines were different to what he was sending. The CSR's will tell us if only one Nexus thought he was in control. The SBI silo will show who should have won the arbitration and what ID was on the SBI. Beware the famous RH780 multiple transmitter problem, see RH780 TT8.

4.1.3.2.6 D26 - TRANSMITTER DURING FAULT

This is not a fault as such, but it does show that this Nexus was transmitting when the SBI FAULT line was asserted. In the simple case where every other Nexus saw a parity then this shows the culprit.

4.1.3.2.7 D07:D00 - NEXUS type ID

The type of any Nexus is defined in D0-D7 of first register, the second micro-floppy will list them when it runs.

00, 08, 10, 18 = MS780C
28, 29, 2A, 2B = DW0-3
20 = RH780
30 = DR780
38 = CI780
60-7C = MS780E

4.1.3.3 SBI SILO

This gets reported as part of the errorlog printout.
It can be read standalone by >>>REPEAT E/ID 18, then stop it with a ^C but you would need the Black Book to interpret the contents.
The top line in the errorlog of the silo is the most recent SBI state.

The next line is the penultimate cycle which actually caused the FAULT LINE.

The next lines show the previous cycles.
ACTIVE TR shows a Nexus arbitrating for the SBI.
Remember the CPU is allowed in by default, no ACTIVE TR shown before things happen implies it's the cpu having a go.
TR0 is the special "hold" condition so a Nexus can hang onto the SBI for multiple cycles.
ID accompanied by a command/address or write data tag is the TR of the Nexus transmitting.
ID with read data tag is the TR of the target Nexus.

Everything transmitted on the SBI should be given some sort of confirmation on the second cycle after the transmission. ACK is best, BSY is o/k, NORESP is bad, ERR is unusual (it means the Nexus is incapable of ever doing what it was asked to do, most common is a MOVL to a unibus address)

4.1.4 CPU

To test the CPU,
see Section 4.6, Running MICROS. Lowest level CPU, SBI and memory test,
and Section 4.7, Running EVKAA and ETKAC. Next level CPU test.
and Section 4.8, Running VDS. The CPU, NEXUS and device tests.

4.1.4.1 CPU module usage and notes

SLOT	11780	11785	NOTE	Name and description
29	M8236	M7477	1❶	CIB Consol interface board
28	M8289	M7544	2❷	FCT FPA
27	M8288	M7543	2	FAD FPA
26	M8287	M7542	2	FML FPA
25	M8286	M7541	2	FMH FPA
24	M8285	M7540	2	FNM FPA
23	M8235	M7476	3❸	USC Micro Sequencer Control
22	M8234	blank	4❹	PCS PROM Control store on 780
21	empty	blank		Slot 21 must be empty on 11780, else ROMs cook
20	Mxxxx	M7475	4	Control Store
19	empty	empty		Slot 19 must be empty (else ROMS cook)
18	Mxxxx	M7475	4	Control Store
17	empty	empty		Slot 17 must be empty (else ROMs cook)
16	M8232	M7474	5❺	CLK Clock
15	M8231	M7473		ICL Interupt Control and Low bits
14	M8230	M7472		CEH Cond.codes Exception and Hi bits
13	M8229	M7471YA		DAP Data Path Control
12	M8228	M7470YA		DCP Data Path 07:00
11	M8227	M7469YA		DDP Data Path 15:08
10	M8226	M7468YA		DEP Data Path 31:16
9	M8225	M7467		DBP Data Aligner
8	M8224	M7466		IRC Instruction Decode
7	M8223	M7465		IDP Instruction Data Path
6	M8222	M7464		TBM Translation Buffer Matrix
5	M8221	M7463		CDM Cache Data Matrix
4	M8220	M7462		CAM Cache Address Matrix
3	M8219	M7461		SBH SBI interface High bits
2	M8218	M7460		SBL SBI interface Low bits
1	M8237	M7459		TRS SBI Terminator and Silo

❶ NOTE 1, the plug-in ROMs holding the PDP-11 boot-code live on here.

❷ NOTE 2, if the (optional) FPA is not installed, there must be blanks.

❸ NOTE 3, there is a set of LEDS which show the micro-PC of the running micro-code. The top LED is considered the msb of this micro-address.
On a 780, 00FF is the HALT LOOP, 0100 is the INIT address.
On a 785, 0FF0 is the HALT LOOP, 0E00 is the INIT address.

❹ NOTE 4, The CONTROL STORE is different on a 780 and a 785.

The 785 is simplest, there is a common module holding 0.5kw of ROM and 4kw of RAM. Ucode upgrades are done by issuing a new RAM image on floppy, the ROM is just enough to talk to the console.
The module is a M7475 (JCS) and must be present in **both** Slot 20 and 18.
Slot 20 is JCS#1 and Slot 18 is JCS#0; Slot 21 and 22 hold a blank.
G and H is standard (as opposed to the 780), it is just ucode on the floppy.

The 780 has a lot of ROM and a bit of RAM. Ucode upgrades are done by fitting an FPLA chip on the micro-sequencer which causes the micro-PC to go to the replacement code in the RAM. So an update consists of a new FPLA chip and a new RAM image on floppy.
Slot 22 holds the ROM and Slot 21 must be empty.
A basic 780 needs 2 kilo-word of RAM.
If KE780 (G and H) is fitted, it needs another 1 kilo-word (plus a FPLA chip on the sequencer and it utilises the tailend of the floppy ucode).
If KU780 (OCS) is fitted, it needs yet another 1 kilo-word (but no FPLA, as it gets called by the XFC instruction).
There are two types of RAM boards, the M8233 is 1 kilo-word and the M8238 is 2 kilo-word.
You can have two M8233s (in slot 18 and 20) or one M8238 (in slot 20) on a basic 780.
Slot 20 is the "first". Slot 18 is the "second". There are jumpers on the CLK module to drive the module in Slot 18 when it is present.

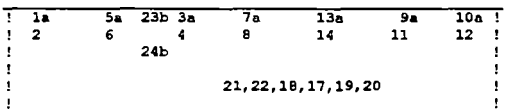
5 NOTE 5, The CLK module has some LEDs which are a blur when the clock is running. If these stop, you will not even be able to talk to the big system from the CIO prompt. There are jumpers and switches on this module which must be set the same on a replacement.

4.1.4.2 FPA

The FPA is an extension to the data paths. When it is fitted the normal micro-code hands over control to the FPA to do the relevant instructions.
The left hand PSU must be fitted for the FPA.
The CLK module has jumpers (780) or switches (785) which terminate signals if the FPA is absent, but I have removed FPA modules for fault finding without altering these and had no problems.
On a 11785 there are extra micro and macro tests for the FP785.

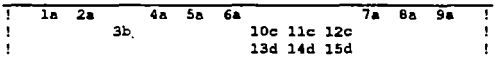
4.1.4.3 11780 CLK module jumper settings

```
a  W1 thru W14 must be IN for FP780.
b  W23 and W24 most be IN if there is a module in Slot 18.
c  W15 thru W22 must be IN
```



4.1.4.4 11785 CLK module switch settings

```
a  S1, S2, S4, S5, S6, S7, S8, S9  ON for FP785
b  S3  ON
c  S10, S11, S12  OFF
d  S13, S14, S15  ON
```



4.1.4.5 Backplane jumpers and SID

The backplane jumper pins are located in the top right-hand corner of the backplane, and are split into four groups numbered J10, J11, J12 and J13.

The pins within each group look like this:-

```
B   D   F   J   L   N   R   T   V   X   Z  BB  DD  FF  JJ  LL  NN  RR  TT  VV
A   C   E   H   K   M   P   S   U   W   Y  AA  CC  EE  HH  KK  MM  PP  SS  UU
```

and the jumper is fitted vertically, eg B to A, or DD to CC.

The SID is a set of jumpers (out for 1, in for 0) and is made up thus:-

```
J12-->  V  X  Z BB   DD FF JJ LL   NN RR TT VV
J13-->                                        B  D  F  J    L  M  R  T

       31 30 29 28 * 27 26 25 24 * 23 22 21 20 * 19 18 17 16 * 15 14 13 12
   780  I  I  I  I    I  I  I  O    I <--- REV ---> <--- rev ---> <-plant->
   785  I  I  I  I    I  I  I  O    O <---- REV -----> <--rev---> <-plant->
       <---------- CPU type --------->

 bits 11 thru 00 define the system serial number on the rest of J13.
```

You can check the SID by doing >>>E/ID 3E
013019C3 = Typical 11780 SID at Rev 6
013099C3 = Typical 11780 SID at Rev 6B
013819C3 = Typical 11780 SID at Rev 7
013899C3 = Typical 11780 SID at Rev 7B
014019C3 = Typical 11780 SID at Rev 8
014099C3 = Typical 11780 SID at Rev 8B
018C071C = Typical 11785 SID at Rev 3
018C871C = Typical 11785 SID at Rev 3B

4.1.4.5.1 11780 specific jumpers

You can select the way the CONTROL stores are addressed, the default for both sizes is:-

1kw boards (M8233). First 1kw in Slot 20, Second in slot 18.
J11-VV-IN, J11-TT-IN, J11-RR-IN, J11-NN-IN, J11-LL-OUT, J11-FF-IN, J11-DD-IN, J11-BB-IN, J11-Z-IN, J11-X-OUT.

2kw boards (M8238). First 2kw in Slot 20, Second in Slot 18.
J11-VV-IN, J11-TT-IN, J11-RR-IN, J11-NN-IN, J11-LL-OUT, J11-FF-IN, J11-DD-IN, J11-BB-OUT, J11-Z-IN, J11-X-IN.

You can select the starting address of the PCS.
Normal is J12-L-IN, J12-J-OUT, J12-F-OUT, J12-D-OUT, J12-B-OUT.

You can select the CPU TR, but it should always be 16.
J10-J-IN, J10-F-IN, J10-D-IN, J10-B-IN.

4.1.4.5.2 11785 specific jumpers

You can reverse the way the JCS modules are selected. J11-X-OUT and J11-LL-IN is normal (Slot 18 is first). J11-X-IN and J11-LL-OUT is the reverse (Slot 20 is first).

4.1.5 DW780

The first DW780 is mandatory.
The DW780 is used by the second Micro-Test floppy to check the CPU. If there is a problem (eg BA11 tripped out) the Micro-test will fail.
For testing the DW780, see Section 4.8, Running VDS. The CPU, NEXUS and device tests.

4.1.5.1 DW780 Idle state

In the idle state the bottom 2 LEDs only should be lit on the M8271. "UNJAM" should put it in this state. When stuck somewhere else the DW780 may well not even respond to EXAM/DEPOSIT commands.

4.1.5.2 Lost Devices

If unibus devices are missing on SHOW DEV, it means they were missing WHEN VMS BOOTED. Therefore there will not be any entries in the errorlog. If a device is in the floating address space, and it has gone away it will cause utter confusion to the VMS autoconfigure routine.

4.1.5.3 EXAM-DEPOSIT unibus options

For more information see Section 4.1.3.1, Addressing. To do this from the console without all the mathematics:-

```
>>> SET REL:20100000       for DW0 (20140000 for DW1)
>>> SET DEF OCTAL
>>> SET DEF WORD

   Now you can now use a normal unibus address eg >>>EX 760100 for a DZ11

IMPORTANT - you MUST reset these consol parameters before returning to normal
operation, else even DEFBOO will not work!

              SET REL:0, SET DEF HEX, SET DEF LONG
```

4.1.5.4 DW780 module usage and description

SLOT	Module	Name and description
1	M8270	USI, SBI Interface
2	M8271	UCB, Control board
3	M8272	UMD, Map and Data Paths
4	M8273	UAI, Address and Interupt control
5		blank module
6		blank module

4.1.5.5 DW0 NEXUS jumper set up

Jumpers set up the TR level (DW0 = 3), BR level (usual = 4) and the UBA number (DW0 = 0). The jumpers are numbered, from the left, W1 thru W15.
For the normal (DW0) configuration they will all be out except W4. Also a wire-wrap will be joining D01R2 to F01E2.

4.1.5.6 DW780 registers and error bits

This is not a complete list, but the registers and bits to look out for are:-

Base Address + 00, CSR.
This is covered in Section 4.1.3.2 except :-

D18 = Unibus Init asserted
D17 = Unibus Power Down
D16 = Unibus Init complete

Base address + 04, UBACR

D01 = Unibus Power Fail
D00 = Adapter Init

Base Address + 08, UBASR

D10 = Read Data Timeout,- SBI or memory problem?
D09 = Read Data Substitute, - MS780 problem?
D08 = Corrected Read Data, - MS780 semi-problem?
D07 = Command Transmit error, - SBI or memory problem?
D06 = Command Transmit timeout, - SBI or memory problem?
D05 = Data Path PE, - M8272 problem?
D04 = Invalid Map Register, - not fatal, normally software!
D03 = Map Register PE, - M8272 problem?
D02 = Lost Error Bit, - too many errors.
D01 = UBSTO see Section 4.1.5.7
D00 = UBSSYNTO see Section 4.1.5.7

Base address + 0C, DCR diagnostic stuff
Base address + 10, FMR the failing map register
Base address + 14, FUBAR the failing unibus address
Base address + 18, copy of FMR
Base address + 1C, copy of FUBAR
Base address + 20 to base address + 2C, diagnostic registers

Base address + 30 to base address + 3C, BRRVR
These 4 registers are special.
Reading one of these registers causes a unibus cycle to get the vector of any interrupting unibus device, a problem (e.g. CPUTO) reading these is more usually a hung unibus than a SBI to DW protocol error.
Experience shows a timeout to this register is a common errorlog entry on systems with Unibus problems. This can be explained by the fact that if the unibus is in trouble, one would expect the DW to interrupt eventually; when VMS goes to its interrupt routine it reads the BRRVR to identify the interrupter (D31 means a DW interrupt as opposed to a unibus device), as mentioned earlier reading this register causes a unibus cycle and if the unibus is in trouble we get a CPU timeout !

Base address + 40 to base address + 7C, Data Path registers
Base address + 800 to base address + EBC, Map registers

4.1.5.7 DW780 error bits UBSTO, UBSSYNTO, and CPU timeouts

The cpu talks to unibus devices via the DW780. To do anything to the unibus the DW must firs get control of it. It achieves this by raising NPR, the M9044 does the request arbitration an sends NPG when all the conditions are met. When the DW sees NPG it takes the unibus, "doe the bizzo" and then replies to the cpu.
UBSTO (unibus select timeout) is set if the DW780 did not get a NPG for 50uS after raisin NPR. This is equivalent to Unibus Master Error on the UDA50, and implies a hung unibus. I solid then a scope is needed to identify the problem.
UBSSYNTO (unibus slave sync timeout) is set if the DW780 did not get a SSYN for 12.8u after raising MSYN. This normally means the unibus device is not responding.
CPU TIMEOUT is set within the cpu if the cpu does not get a reply from the DW780 in 102.4uS If the problem is really a UBSTO or UBSSYNTO, the DW780 will force a dummy reply to th cpu to abort this timeout. If this bit is set, it is **either** a DW780 problem (except when readin special registers BRRVR, see earlier) **or** a unibus "clag" is forcing the DW780 to a solid loop i its microcode and it is ignoring the cpu.

4.1.5.8 Unibus problems, saga mode hit list

There are two devices available from Logs to trouble-shoot a Unibus.
First is a **Margin Tester** which we use to ensure the Unibus is "solid" even with margins termination voltages, this checks for weak drivers and poor recievers. You have to ru diagnostics with margins set at 2.93V and 7.85V using the single ended margin system. Secondl there is a **Unibus Exerciser** , called a **PMK05**, this is a dummy Unibus device which acts a a simple DMA and Read-Write register target for the DW780 Diagnostic. It allows you to tes the Unibus itself, rather than the DW780. The PMK05 has LEDs for each unibus line and ca be left on-site to allow the customer to record the state of the unibus in extreme cases.

1. The stable DW780 is at rev C1. This comprises M8270 at C or D, M8271 at F1 or F2, M827 at C or D, M8273 at B or C, M9042 at A, M9044 at B or C. The M9044 is the PCB stickin out of the back of the DW, the delay line on this PCB must be a 16-09428 NOT 16-0955 The M8272 MUST BE CS-D for DEUNA (I suspect also for UDA).
2. Any RX211 must be CS-M or later.
3. The DD11CK four slot SPC backplane has wires between F01T2-A02R2 and F04T2-A03R these have been changed to twisted pairs. Lack of twisted pairs may cause strange unibu hangs.
4. UDA50s are good unibus exercisers! They must be last NPR device, and the delay jumpe is now 0uS *unless other devices get data-lates.*
5. Check the tech-tips if you have a DR11B.
6. DZ11 must be CS-E or later.
7. TU80s must have the correct BG termination, early modules are FCO'd, later ones are o
8. Ensure +5V regulators do not have their outputs joined together due to links being prese to +5VB on multiple backplanes.

The most common saga is the Unibus hang situation, this is seen in the errorlog as a CP timeout to one of the BRRVRs.

4.1.6 The Unibus and BA11-K

4.1.6.1 Unibus rules for BI engineer

- The "unibus" is either a collection of three grey cabes mounted on paddle cards (used from DW780 backplane to first unibus backplane, a flat white cable (used to go from one BA11 box to another), a bit of flat white cable packaged as a "jumper cable" (used to connect unibus backplanes within BA11), or tracking across a unibus backplane.
- On ALL unibus backplanes (which come as 9-slot or 4-slot), the unibus cable goes IN the top right hand side, and OUT the top left hand side. The slots in-between will blow up other devices on the unibus.
- Both ends of the unibus need a terminator.
  1. On a 780/785 the start of the unibus is terminated by a M9044 which is mounted so it sticks out the back of the DW780 alongside the unibus cable paddle card. This is a "special" (not used on other unibuses) with extra logic to do unibus arbitration.
  2. The far end of the unibus is terminated with an M9302. This also is more than a simple terminator, it has bus-grant logic (see later).
- There must be an option or continuity card in every slot (see later). This is **particularly** important as the standard approach when troubleshooting a unibus is to shorten it to remove options and backplanes in one lump (or by-pass lumps with another cable) and to remove individual options to see what is causing the problem. Also remember to keep the end terminated during this troubleshooting approach.

4.1.6.2 Unibus grant explanation for BI engineer

There are four Bus Request (BR) lines which run along the unibus, passing every option. A device pulls a specific BR line when it wants to interupt.
There is also one Non-Processor Request (NPR) line which a device pulls when it wants to do a DMA.
The arbitrater prioritises the Requests (BR4, BR5, BR6, BR7 and NPR) and responds by raising an associated Grant line (BG4, BG5, BG6, BG7 or NPG).
The Grant lines are different to the rest of the unibus lines, because they go THROUGH each option. An option expecting a Grant will "block" it from passing through. This is so that, even if multiple options are making a Request, only one option sees the grant.
SO IF AN OPTION IS ABSENT, A CONTINUITY CARD IS NEEDED.
If there is no continuity, a Grant line floats high and the End Terminator sees this as an active Grant. Because this should only happen in the case of a phantom Request (the arbitrator saw a Request, but it has now dissappeared) the Terminator Grant Logic will send a Sack back to the Arbitrator to accelerate it out of its "hang". But if the Terminator is seeing a solid Request (due to the loss of continuity) it will keep Sack up and totally hang the Unibus.
BUT THERE ARE TWO TYPES OF CONTINUITY CARD.
The G727A is a small 3-inch square card and only does BG4, BG5, BG6 and BG7 and goes in SLOT D. This is because most unibus devices do not do DMA, so the idea was that we can have a cheaper jumper by hard-wiring the backplane so NPG does NOT go thru the option. If this small card is in use, you must have a link between CA1 and CB1 on the backplane of that slot to provide NPG continuity.
Replacing a DZ11 with a SMALL continuity card can even catch you out. This is because the DZ11 designer put a bit of track between these two pins, so a DZ11 will work even if the backplane jumper is missing, but it will bite you if you replace it with a small continuity card. It is very easy **to put this in upside down**, the etched links are on Side 2. If you don't know which is side 2, look at other modules - Side 1 has the components.

The second type of continuity card is a G7273, this is a double-height, full length card whic goes in SLOT C and D. This answers all our problems. It has tracking for BGs and NPG. Yo can also move it around without de-skinning your knuckles.

4.1.6.3 BA11-K power supplies for BI engineers

These can be extremly awkward to work on. They are placed across the back of the box.
In the centre (top) is a transformer which generates 30 volts to the regulators.
In the centre (bottom) is a **AC Input Box** (C/B, contacter etc), but also in here is a little modul which produces the +15V and all the ACDCLO timing.

- The -15V regulater blocks *will not come on without* +15V.
- The ACDCLO signal *is nothing to do with the state of the regulaters.*

The DC regulaters are placed either side of the transformer position.

To remove a DC regulater :-

- Slide the BA11-K forward, look out for filing cabinet interlocks.
- Tilt it vertically.
- On each side, loosen the "hinge screw" and remove the other screws holding the entire PSI assembly onto the back of the box.
  The bottom of the PSU assembly should fall thru about thirty degrees.
- Working thru the gap, remove the cabling and two phillips screws to the faulty regulater.
- Tilt the BAll back to its original position
- Round the back, remove the small phillips screw going into the faulty regulater.
- Remove the cable clamps and clear the cables out of the way across the top.
- Remove the black cover which the cables were lying on (there are screws across the top, an across the rear-top edge), you can now see the top of the regulaters.
- Pull out and replace the faulty regulater.

To remove the AC Input box :-

- Start as for a regulater, thru the gap, remove cabling to the module in this box.
- Remove the three small phillips going thru the bottom to the AC Input Box.
- Withdraw the AC Input Box thru the rear.
- You can then replace the internal module (54-11086) if you wish.

4.1.6.4 BA11-K fans for BI engineers

There are two 8-inch fans whick suck air over the modules, then blow it out over the powe supplies.
There should be a fair exhaust out of the rear of the BA11, try not to cover the slits with cable
To replace a fan, remove the lid over the modules, and remove the modules (and maybe plasti slides).
Locate and remove the two 3-inch screws going through the fan body from the module area.
The fans are part number 12-11714-00.

4.1.7 MS780 memory

To test the memory see Section 4.6, Running MICROS. Lowest level CPU, SBI and memory test and Section 4.8, Running VDS. The CPU, NEXUS and device tests.

UETP Load Paths is a good exerciser.

4.1.7.1 MS780-A and MS780-C memories

The -A was the original, the -C just has bigger arrays.
You are unlikely to come across kit this old.
There must be at least one array, the arrays must be contiguous left to right and you cannot mix different size array cards.

There are jumpers for TR number, Power-Up start address, and Allow ISP ROM.

4.1.7.1.1 MS780-A and -C, module usage and description

SLOT	Module	Name and description
20	M8214	MSB, SBI Interface
19	M8213	MCN, Memory control
18	M8212	MDT, Data Paths
17 thru 2	M821x	MAY Arrays, M8211 = 1/16 Mb, M8210 = 1/4 Mb
1		Blank module [or M9040, TRM]

4.1.7.2 MS780-E and MS780-H memories

The -E is the most common, the -H came out quite late with bigger arrays.

11780 and 11785 rev xB; was to fit updated MS780E/H controller modules (M8375s at D1, M8376 at E1).

There is a red LED on the M8376, this comes on if :-

Different number of arrays on each side
or Arrays not in consecutive slots
or Arrays of different types

4.1.7.2.1 MS780E, module usage and description

SLOT	Module	Name and description
20	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 7
19	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 6
18	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 5
17	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 4
16	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 3
15	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 2
14	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 1
13	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 0
12	M8375	Upper array controller
11	M8376	SBI Interface
10	M8375	Lower array controller
9	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 0
8	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 1
7	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 2
6	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 3
5	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 4
4	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 5
3	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 6
2	M837x	M8373 = 1Mb, M8374 = 4Mb, Array 7
1		Blank module [or M9040, TRM]

4.1.7.2.2 MS780E/H Jumpers

```
JUMPERS
J21
UU SS PP MM KK HH EE CC AA Y  W  U  S  P  M  K  H  E  C  A
VV TT RR NN LL JJ FF DD BB Z  X  V  T  R  N  L  J  F  D  B

Power Up Start Address (mB)                        TR Level
J21UU    J21SS     -E  -H            J21H J21E J21C J21A     link F11S1
 OUT      OUT       0   0          1 OUT  OUT  OUT  OUT       to F11B1
 OUT      IN       32  32          2 OUT  OUT  OUT  IN           F11E1
 IN       OUT      48  48          3 OUT  OUT  IN   OUT          F11D1
 IN       IN       96  96          4 OUT  OUT  IN   IN           F11F2

There is a confusion factor in the cold start addresses, the
book is wrong, I hope this is correct

SBI FAIL link C11R1 to C01V2 (in for all memories)

Inhibit ROM decode (J21AA) OUT=First memory, IN for others
```

4.1.7.2.3 MS780E/H Registers A thru H

1. Register A, base address + 00

D31 SBI PARITY FAULT
D30 SBI WRITE SEQUENCE FAULT
D29 0
D28 SBI INTERLOCK SEQUENCE FAULT

D27 SBI MULTIPLE TRANSMIT FAULT
D26 TRANSMITTER DURING FAULT
D25 0
D24 0

D23 POWER DOWN sequence underway
D22 POWER UP sequence underway
D21 0
D20 ERROR SUMMARY "or" of REGA D19-D15, REGC&D D28&D7

D19 CNTR 1 (upper) READ DATA BUS PAR ERR (send RDS tag with data)
D18 CNTR 0 (lower) READ DATA BUS PAR ERR (send RDS tag with data)
D17 MISCONFIGURE unequal arrays on each side
D16 CNTR 1 (upper) MISCNFG, no arrays or no controller

D15 CNTR 0 (lower) MISCNFG, no arrays or no controller
D14 1=32 }
D13 1=16 } MEMORY SIZE
D12 1=08 } number of megabytes minus one, e.g. on a MS780-E
D11 1=04 } 000000 = 1 array, 000001 = 2 arrays, 000010 = 3 arrays
D10 1=02 }
D09 1=01 }
D08 INTVL MODE WRITE ENAB (1 = allow writing to D2:0)

D07 0 }
D06 1 } ADAPTER CODE
D05 1 }
D04 } RAM TYPE, 0 = no arrays, 1 = 1Mb, 2 = 4Mb, 3=mixed

D03 } RAM TYPE
D02 } INTERLEAVE MODE see Section 4.1.7.3, Multiple memories
D01 }
D00 }

2. Register B, base address + 04

   D19 to D27 hold the starting address in 1 Mb increments.
   D14 is "start address write enable".
   D13 and D12 reflect the initialization status.
   Other bits are mainly diagnostic.

3. Register C and D, base address + 08 and + 0C, lower and upper controller status

   D31 FORCE MICRO-SEQ PARITY ERROR
   D30 INHIBIT SENDING CRD TAG WITH DATA
   D29 HIGH ERROR RATE second occurred before first cleared
   D28 ERROR LOG REQUEST

   D27 = 0 on lower controller and 1 on upper controller
   D26 }
   D25 } Failing array number.
   D24 }

   D23 thru D20 = part of failing address

   D19 thru D16 = part of failing address

   D15 thru D12 = part of failing address

   D11 = part of failing address
   D10 = RDS, multiple bit error
   D09 = CRD, single bit error
   D08 = 0

   D07 = Microsequencer PE
   D06 thru D00 = error syndrome

4. Register E and F, base address + 10 and + 14, diagnostic read-write

5. Register G and H, base address + 18 and + 1C, phantom, unused.

4.1.7.3 Multiple memories

If you have mixed memory (MS780C and MS780E), the MS780E must be TR1 because its newer ISP ROM knows about both types (the ISP ROM is called as part of the boot sequence and does some configuration checking).

The start addresses (and the interleave bits if required) must be set correctly before booting VMS, this is done by putting DEPOSIT commands in the DEFBOO.CMD file (after INIT and UNJAM). These registers have battery backup so there is no need to set them up in RESTAR.CMD, although it will do no harm.

If you have multiple memories not interleaved on the SBI, use DEPOSIT commands to set the first to a start address of 0, and the second to directly follow it.

If you have multiple memories interleaved on the SBI, use DEPOSIT commands to set the start address of both to 0 and set the interleave bits.

Jumpers are provided to set the start address on power-up. These must be set to ensure each memory space has a unique address, a gap between the end of one and the start of the next is o/k. This is because the ISP ROM might go off looking for a RPB as part of the restart sequence just after power on, if the BBU had faded then the start address registers will be wiped out and a machine check would occur.

4.1.7.4 MS780C Interleaving

Two identical MS780C's can be interleaved on the SBI by setting the start address and interleave bits in their registers. If you look at one of the .ILV files distributed on the consol floppy you will see the typical sequence to set up two MS780Cs.

4.1.7.5 MS780E/H Interleaving

"Interleaving" can be in 1 (and ONLY one) of 3 states:-

1. INTERNAL INTERLEAVE
   This state is forced by the MS780 hardware if it has two controllers with the same number of arrays on each. In this state you cannot be "external interleaved" as well. This is the state nearly all memories will be in. The upper controller holds even quadwords (byte 0-7). The lower controller holds odd quadwords (byte 8-F)
2. EXTERNAL INTERLEAVE (interleaved on the SBI)
   If you have 2 identical MS780E's and both have only 1 controller (ie not "internal interleaved") you may set both MS780E's to the "external interleave" state by writing to their registers. It is improbable that you will ever encounter this
3. NON-INTERLEAVED
   This state is forced by the MS780E hardware if you haven't got "internal interleave" requisites, and you also haven't set the "external interleave" condition by writing the register. You will see this on a small configuration (i.e. only one controller)

4.1.7.6 ISP ROM flow

MS780E's have different ISP roms to MS780C's. The general flows are the same; the ROM is entered by the floppy command file (x.CMD) at one of two entry points.

- 20003000 (booting) sets "cold flag"
- 20003004 (re-starting) sets "warm flag".

Then it does the following:-

1. Save and verify the GPR's in the DW/RH map registers specified by the floppy command file with checksum (gives DOUBLE ERROR HALT @ 20003032 if nexus not there, can give *FATAL NEXUS HAS BAD MAPS)
2. Check instruction set used by ISP (can give *FATAL CPUERR;R7 GIVES SUBTEST).
3. For up to 4 MS780's, from the bottom upwards, scan memory for:-
   - If "cold flag" look for a good area of 64kB
   - If "warm flag" look for a VMS Restart Parameter Block (RPB)

   (either can give *FATAL MIX OF ARRAYS, *FATAL MEMORY NOT INITIALIZED , *FATAL IMPROPERLY INTERLEAVED, *FATAL ADDRESSES OVERLAP)
4. Check SBI fault/status registers (can give "WARNING-FAULT ON SBI" but carries on)
5. Restore registers from DW/RH and verify checksum (can give *FATAL NEXUS HAS BAD MAPS)
6. Look at state of cold/warm flag.
   - On "cold flag",
     Did we find a good area of 64kB (can give *FATAL NO WORKING MEMORIES).
     Set SP=base address of good memory+200.
     Put F01 in TXDB (to LSI).
     HALT (200034F9 on MS780C, 2000355D[2] on MS780E).
     LSI sees F01 and continues with the .CMD file (which would have been doing a "wait done").
   - On "warm flag"
     Did we find a good Restart Parameter Block with word C=0 (ie first attempt at restart).
     - If not found or word C non zero, put F02 in TXDB.
       HALT (2000350A on MS780C, 2000356E[3] on MS780E).
       LSI sees F02, types "BOOTING" and starts DEFBOO.CMD
     - If o/k type "ATTEMPTING WARM RESTART".
       Increment word C of Restart Parameter Block (to say we've tried to restart once).
       Jump through word 4 of RPB to VMS

NOTES:-

- *FATAL error messages are followed by a loop (at 20003748 on MS780C, 200037AC[0] on MS780E).
- A "trapcatcher" can generate "*FATAL — UNEXPECTED MC CHECK" followed by the same loop.
- Numbers in [ ]'s are for a different version of the MS780E ISP ROM. In the real world, we should always find good memory a location 0 onwards, if the search for a good 64kb of memory returns anything other than G E 00000200, I would suggest you mend the memory.

4.1.8 RH780 Massbus adapter

If there are two (one for tape and one for disk), the tape one is often mounted nearest the memory.
See Section 4.8, Running VDS. The CPU, NEXUS and device tests for testing.
Old copies of EVMAC will fail if the new M8274 is fitted.
Beware that a bad bit of data during a WRITE-CHECK operation will abort the transfer immediatly, you do not get DATACHECK set!!!
Some RH780's give 'multiple transmitter fault' (really due to memory throughput) this is cured by M8275 at rev J1 or J2.
On the cabling at the paddle card; the RIBBED side should be outwards and the RED edge nearest the backplane; the top connector is "C".
On the cabling at the transition piece; the RIBBED side is on the right, and the RED edge is on the top; the right connector is "C".
The massbus itself can be made up of a round cable or three flat cables, the total allowed is 160'.
It can get confusing when you are re-configuring a mass-bus which has both flat and round cables because of "reversals" through various transition pieces.
Although the massbus goes in and out of each drive, the two connections are joined within the drive and the drive just hangs off it.
Drives have "disabling" switches which allow their massbus transceivers to be disabled, this is usefull when working on the drive.
The massbus needs a far-end terminator, the terminator gets its +5v down the cable from the RH780.

4.1.8.1 RH780 module usage and description

The M8274 was made with a bigger Silo to accomodate the faster devices (RP07, TM78), it replaced the M8277. It can also be used for slower devices.

SLOT	Module	Name and description
1	M8275	MSI, SBI Interface
2	M8276	MIR, Internal registers
3	M827x	MDP, M8277 or M8274 Data Paths
4	M8278	MCP, Massbus Control
5		blank module
6		blank module

4.1.8.2 RH0 NEXUS jumper set up

Jumpers set up the TR level (normal=8), BR level (normal=5).
The jumpers are numbered, from the left, W1 thru W15.
They should all be out except W2, W3, W4, W6. Also a wire-wrap should join F02F1 to F02M1.

4.1.8.3 RH1 NEXUS jumper set up

Jumpers set up the TR level (normal=9), BR level (normal=5).
The jumpers are numbered, from the left, W1 thru W15.
They should all be out except W1, W6. Also a wire-wrap should join F02F1 to F02N1.

4.1.8.4 SI "RHs"

These have a small switch on the paddle card which must be switched off before running Micro-Diagnostics.
They will also probably be using a patched version of VMB.EXE. So be careful if you are updating the console floppy.

4.1.8.5 RH780 registers

The registers are split into three groups.

1. The internal registers, control type registers within the RH780.
   These are from offset + 00 to + 1C.
2. The external registers, situated on the massbus drive.
   These are from offset + 400 to + 7FC.
3. Map registers, 256 registers also within RH780.
   These are from offset + 800 to + BFF

Table 4–3: RH780 Internal registers

OFF	NAME	Notes
00	CSR	As in SBI, D22 = RH power up, D23 = RH power down
04	CR	Control register, used for odds
08	SR	Status register, error conditions
0C	VAR	Virtual address, points at map register and byte offset
10	BCR	Byte count register
14	DR	Diagnostic register, to fiddle with massbus
18	SMR	Selected Map register, contains copy of map register
1C	CAR	Command-Address register, to monitor SBI

Table 4–4: RH780 external registers

RP	RM	TE	DR0	DR1	DR2	DR3
CS1	CS1	CS1	400	480	500	580
DS	DS	DS	404	484	504	584
ER1	ER1	ER	408	488	508	588
MR	MR1	MR	40C	48C	50C	58C
AS	AS	AS	410	490	510	590
DA	DA	FC	414	494	514	594
DT	DT	DT	418	498	518	598
LA	LA	CX	41C	49C	51C	59C
SN	SN	SN	420	4A0	520	5A0
OFF	OF	TC	424	4A4	524	5A4
DCA	OC		428	4A8	528	5A8
CCA	NR		42C	4AC	52C	5AC
ER2	MR2		430	4B0	530	5B0
ER3	ER2		434	4B4	534	5B4
ECCPOS	EC1		438	4B8	538	5B8
ECCPAT	EC2		43C	4BC	53C	5BC

4.1.9 CI780 Computer Interconnect

There is a lot of odds'n'ends already in Chapter 6, VAXclusters , including the register layout and module usage.

- The current (May 1990) revision of the ROMS is rev 7, the RAM ucode image is rev 8.
- During the boot sequence, VMB will load the CI ucode off the RX01 console floppy.
- See Section 4.8, Running VDS. The CPU, NEXUS and device tests for testing.
- The switch packs on the L0100 give the CI node number and must look the same.
- An FCO replaces the L0100 with a L0108 in 6 node clusters.
- Beware that the PSU is a 70-14956-01 as there is a need for -5v.
- The CI780 is at TRE and BR4, you will never see anything different.

4.1.9.1 CI780 module usage and description

SLOT	Module	Name and description
1	L0104	ISI, SBI Interface
2	L0102	IDP, Data Paths
3	L0101	IPB, Packet Buffer
4		blank module
5	L0100/8	ILI, L0100 or L0108 Link module
6		blank module

4.1.10 KC780 and KC785, the console subsystem

The console consists of an 11/03 with its software on an RX01 floppy.
The software is booted into the 11/03 (do this with the LSI's POWER switch) and takes care of initializing the main cpu (ie setting it up to run macro-code), CIO functions (eg EXAM, LOAD, BOOT etc) and acts as a PIO device (talking to VMS).

There are several versions of this console floppy, and we get bitten by them getting muddled up. The 11785 is easy, but a bit of information which may prove a life saver:- the 11780 Local CONSOL.SYS is 31 blocks and the Remote CONSOL.SYS is 34 blocks.

RX01, 11780 standard local console.
RX04, 11780 RDC console **USA only, crashes a UK system.**
RX41, 11780 RDC console, European.
RX1A, 11785 console, does local as well as RDC in the USA and UK.

Typical files and their usage:-

CONSOL.SYS, the actual console software that gets booted into LSI.
WCSxxx.PAT, the 11780 microcode, xxx gives the version.
SSUxxx.WCS, the 11785 microcode, xxx gives the version.
VMB.EXE, the primary boot program.
CI780.BIN, the CI780 microcode, VMB loads this into the CI780.
DEFBOO.CMD, the boot sequence invoked by BOOT etc.
RESTAR.CMD, the sequence invoked by a power recovery.
xxxBOO.CMD, other boot sequences on a 11780.
xxBOO.CMD, other boot sequences on a 11785.
CS1BOO.CMD, the sequence to boot Standalone Backup from floppies.

Remember that the customer will update/alter any of these files, you may well find several unlabelled ones lying around on site. Altering files is good fun, they don't test them and then wait for a genuine fault to occur to see how long it takes you to sus out they screwed it up.

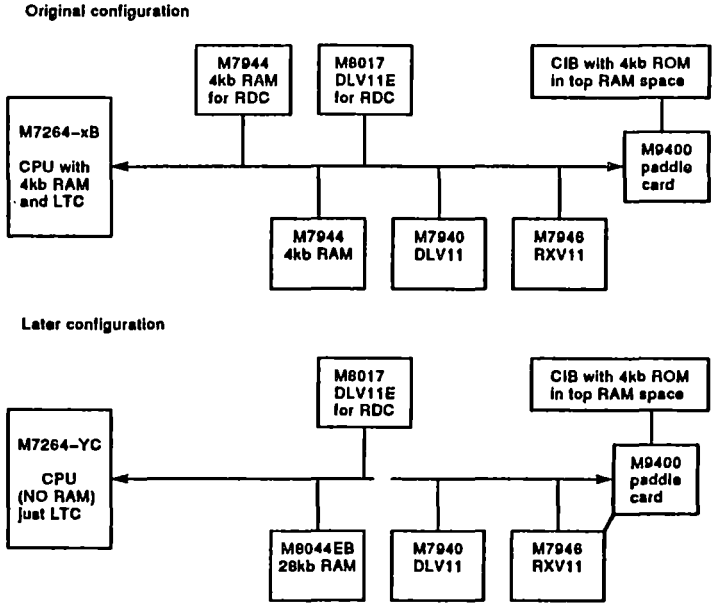
- The LSI modules have to be in the correct sequence, see Section 4.1.10.1, Upgrading the original configuration.
- The LSI [RUN] light is ON if the LSI is running PDP11 macro-code.
- The LSI [POWER] switch is the normal way of rebooting it.
- The LSI [HALT] switch will stop macro-code and got to the ODT @ prompt.
- The LSI [LTC] switch must disable the LTC for PDP11 diags and enable the LTC for RDC configurations.
- You need to remove the 4 screws holding the front cover on to get at the modules.
- Use a tie wrap to check the fan is going.
- You need to remove the whole black box to get at the PSU or fan. First remove the front cover, then the cabinet exhaust box, then the screws and brackets in the rear corners. Then the innards slide out the back of the box.
- The door of the RX opens "upwards", get the diskette the right way round!

There are 2 main variants of the LSI configuration, each with a sub-variant if RDC is fitted.

- Original configuration, using 4kb chunks of memory on individual modules and the cpu, found on old 780s and 785s which were upgraded from 780s.
- Later configurations, using memory on a 28kb memory module only, these are found on newer 780s and shipped 785s.

- The swap kit assumes the later version.
  If this is untrue, you will have to upgrade to the later configuration if you want to swap a memory or cpu board.

Figure 4–4: LSI block diagrams



4.1.10.1 Upgrading the original configuration

The new memory module is a M8044-EB. When this module is used NO OTHER memory is allowed in the LSI. This means you must remove all the 4kb memory modules and the 4kb on the CPU module.

Obviously, when you normally replace modules in the LSI, you can copy the jumpers from the one you are removing. Just as obviously, you can't do that if you are "upgrading".

So you will need to know:-

1. How the new memory (M8044-EB) needs to be set up.
2. How to disable the 4kb on the CPU module (M7264-BB or -EB (the first digit just denotes which vendor RAMS are used)) if you are keeping this module.
3. How to set up a new (M7264-YC) cpu, which comes without RAM
4. How to "re-fill" the LSI backplane to maintain QBUS continuity.

So, here you go:-

Figure 4–5: New memory setup (M8044-EB)

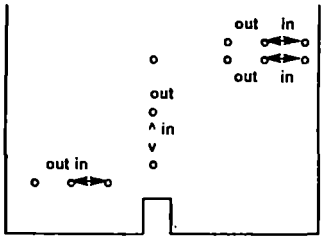
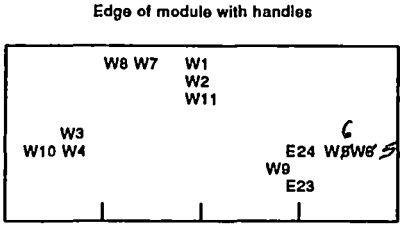


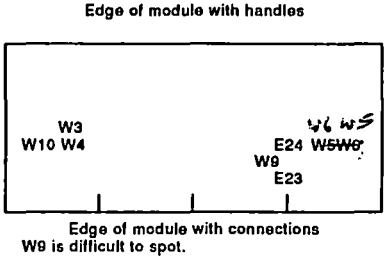
Figure 4–6: Old CPU setup (M7264-BB or -EB)



4kb on-board memory

ENABLE	DISABLE	
out	out	W1
in	out	W2
out	out	W3
out	in	W4
out	out	W5
in	in	W6
factory set		W7 don't touch
factory set		W8 don't touch
out	in	W9
in	out	W10
in	out	W11

Figure 4–7: New CPU setup (M7264-YC)



4kb on-board memory missingII (and jumpers missing)

W3	out
W4	in
W5	out
W6	in
W9	In
W10	in

For QBUS continuity, the modules have to be in the correct sequence in the backplane.
The top row of four slots is row 1, the rest are labelled 2, 3 and 4.
The left slot in each row is slot A, the rest are labelled B, C and D.
The CPU is a 4-slot option and lives in row 1 (the top row).
The remaining 2-slot options must then fill the other slots in a snake-like sequence.
Slot 2C&D, Slot 2A&B, Slot 3A&B, Slot 3C&D, Slot 4C&D, Slot 4A/B.
The QBUS paddle card must be the last one inserted, and remaining slots are left empty.

4.1.10.2 LSI RDC port

There is a new DLV11 module in use which is not described in the black book.
It is a M8017-AA.
The factory setting is not the same as needed by RDC.
There are 7 blocks of jumpers. In the figure *'s represent wire wrap posts, I= jumper in, O= jumper out. The module is shewn component side up, handle at top, set for 1200 Baud.

```
------------- BERG PLUG --------------

*
O < Please check these are not shorted by a track on side 2 of module,
*   there was a problem on some boards.
I
*
                                                          * I *
                                                          * O *

                          * I *
Insert to make 300 Baud>* O *
                          * O *    *
                                   O
                          * O *    *
Insert to make 300 Baud>* O *
                          * O *
                          * I *
                          * O *          * I *
                          * I *          * O *
                                         * O *
                                         * O *
                                         * O *
                                         * O *
                                         * I *
                                         * O *
            * O *                        * O *
            * I *                        * I *
            * O *
            * I *
            * O *
            * O *
            * I *
            * I *
            * I *                                * O *
            * I *          * *                   * O *
                                                 * I *
                          I O                    * O *
                                                 * O *
                          * *                    * I *
                                                 * I *
                                                 * I *
                                                 * O *
                                                 * O *
                                                 * O *
```

4.1.10.3 Isolating faults to the LSI subsystem

Certain faults (especially the CLK and CIB module) can make you think the LSI is bust, when it aint.

One clue is the state of the KA LEDs.
If you power up the system, without a floppy in the drive, the KA should go to its Console Halt Loop. The CLK module LEDs will be a blur, and the USC will be 0FF0 (11785) or 00FF (11780).

If you are not sure whether the problem is in the LSI or the "big machine", you can remove the QBUS paddle card and insert a M9400 QBUS teminater/bootstrap (I've arranged for one of these to be kept in the KC780/785 kit). When you power this up, you will get the @ prompt, do a 173000G and you will run the M9400 code and this will give you a $ prompt (you would have to be really tired to confuse this with VMS).
At the $ prompt you can type XC to test the LSI CPU, or XM to test the LSI memory.
You can also BOOT a XXDP floppy by typing DX, then you could R ZKMA?? to run the LSI memory test.
You must Disable the LTC to run XXDP, else it keeps Halting at 104
Unfortunatly you cannot boot up the console software when the M9400 is used instead of the CIB, this is because the console software uses routines within the CIB ROM.

4.1.10.4 Working on the RX disk drive

To extend the life of the motor, it is controlled by the console software using a SSR. It gets switched off when not in use. The SSR lives in a compartment bolted on the side of the drive assembly; if the SSR blows up, you can bypass it by pulling the mains IN and OUT cables from this compartment and mating them together.

You can eyeball the belt thru a crack in the assembly to check that it is indeed rotating.

There is a [Circuit Breaker] under the drive assembly, it can easy get caught on loose cabling and get knocked [OFF].

There is a small fan under the drive to keep the it cool.

The RX comes in 2 flavours (well it would wouldn't it?), a label near the C/B will tell you if it is a RX01W. The original RX01 used a M7726 and M7727. The later RX01W (an RX02 operating in single density mode) uses a M7745YA and M7744YC. The M7744YC needs S1-1 ON and S1-2 OFF to be in single density mode. The drive itself has a small resistor in the head cable plug which is in for an RX01W and out for an RX01 (it is best to remove it "nicely" and not to just clip it out).

It is a good idea to use a scope on the RX01 psu. RED is +24, GRAY/RED is +5, BLUE is -5, ORANGE is +9.

You can "hot-wire" a spare drive in as DX1, the M9400 will accept the DX1 command at the $ prompt, and proceed to boot, say, an XXDP floppy.

The Drive has been known to blow the R/W module, replacing either will just cause the replacement to blow. Sometimes you may have better results by replacing both at once!

4.1.10.5 The Operators Control Panel (OCP)

This is considered to be part of the console.

- The [AUTO RESTART] switch.
  The console software looks at this after a power-up or pathalogical halt.
  If it is [ON], the console will obey the RESTAR.CMD sequence.
  If it is [OFF], the console will halt in CIO mode.
- The [BOOT] push button.
  If the console sees this pushed whilst in CIO mode it will do DEFBOO.CMD.
- The [KEYSWITCH] allows a combination of RDC or LOCAL terminal, and whether CIO mode is to be ENABLED or DISABLED.
- The green [RUN] light means the CPU is running macro-code.
- The red [ATTN] light means the system is halted.

4.2 A few console commands

Once the console software is running in the LSI, it has two modes.

1. CIO, Consol Input Output Mode.
   The consol software accepts any terminal input as a command to itself.
2. PIO, Program Input Output Mode.
   The console software passes all terminal input to VMS (except ^P).

To get from PIO to CIO, type ^P.
To get from CIO to PIO, do **SET TERMINAL PROGRAM**.

>>> B, will activate the DEFBOO sequence.

>>> T, will activate the MICRO-DIAGNOSTIC sequence.

>>> H, will halt the main system, putting its microcode into the consol halt loop.

>>> DIR, (11785 ONLY) wil give a floppy directory.

>>> @DEFBOO.CMD, will activate the named command file (DEFBOO in this case), giving a printout of each line as it is executed.
Very usefull to check what you are trying to boot, or to see a command syntax.

>>> BOOT XYZ,

On a 780 this obeys the file XYZBOO.CMD, this has the disadvantage of needing a file for every circumstance.
On a 785 the console software will deposit Z into R3 and then obey XYBOO.CMD. However if Z is missing then zero is assumed, also Z must be octal else it reverts to obeying XYZBOO.CMD.

4.3 Fixing problems getting to >>>

Basically, at power up, the LSI goes direct to ROM code on the CIB, the ROM runs some tests then loads CONSOL.SYS off the RX01. This console code starts running and reports its defaults before issuing the >>> prompt.

Make sure you are aware of all the issues in Section 4.1.10, KC780 and KC785, the console subsystem, especially the bit that isolates a fault to the LSI.

The Black-Book gives some good troubleshooting flows which break down into three areas.

1. Will not print @, if powered up in HALT mode.
2. From @, you can't EXAM/DEP ROM or RAM.
3. CIB code runs, but will not run CONSOL.SYS.

If you do not have a Black-Book, this is the general approach:-

On a 11785, booting the LSI should cause the following printout:-

```
CPU HALTED, SOMM CLEAR, STEP=NONE, CLOCK=NORM
RAD=HEX, ADD=PHYS, DAT=LONG, FILL=00, REL=00000000
INIT SEQ DONE
HALTED AT 00000000

RELOADING WCS
LOAD DONE, 2000 MICROWORDS LOADED
VER: PCS=04-04 WCS=01-E CON=V06-00-C
>>>
```

A 11780 is almost the same except it only loads 800 micro words, (a 11785 takes about 2 minutes).
A 11780 at Rev 7 will report WCS 0E-10, FPLA 0E
A 11780 at Rev 8 will report WCS 0F-11, FPLA 0F

Before troubleshooting any further, check the LSI has DC, the LSI [HALT ENABLE] switch is [UP] and the [AUTO-RESTART] switch is [OFF].

Your first reaction will be to try another floppy, lets hope someone convinced the customer to keep a backup! Remember that **poor system management can easily render the floppy media useless.**
Beware floppies which give different symptoms. This is a common scenario, and generally means they are not true copies.
Also do not think that a bad media will necessarily act like it. Often the code just does not seem to run, but we still don't get a "bad media" type message.

If you got all the defaults reported, but no >>>, it is very unusual, I would guess at the LSI memory or the CIB.

Usually the LSI will take a dive before printing anything sensible:-

1. **It prints a PC, followed by the @ prompt.**

   At the @ prompt, you now type M, look at the last digit of its response to tell you why it has entered ODT.

   0 or 4, means HALT instruction or switch.
   1 or 5, means BUS ERROR during interupt cycle.
   2 or 6, means BUS ERROR during memory refresh.
   3 means DOUBLE BUS ERROR, normally due to an invalid SP.
   4 means RESERVED INSTRUCTION TRAP
   7 means a combination of 1, 2 and 4.

   Sometimes the PC gives you a clue,

   PC = 141236 or 141262, memory test has failed.

PC = 140332 or 141076, CPU test failed.
PC = 173000 either can't get to CIB module, or in HALT mode.

Otherwise, if the PC is in RAM, it is likely that you have a bad CPU, Memory, duff code off the floppy (maybe the floppy did not have the boot section written).

There is no parity checking in the LSI memory or QBUS, so most errors cause a crash entry to ODT.
You can do some EXAMs and DEPOSITs to check out a few locations. Just type in the address you are interested in followed by /, it will print out the location's contents; to alter the contents just type in the new stuff and hit CR; a LF instead of CR will automatically move on to the next address.
Location 0 always gets overwritten with 000240, if the code has come in properly off the RX boot block.

2. **It complains about the floppy.** If you are going to work on the RX, read the section on the console sub-system first.
The worst possible case, is when the drive is actually corrupting the media. Sometimes, when you are at the end of your tether, taking the media to a different site to check it can be an eye-opener!
Floppy error messages are normally self-explanatory, but sometimes it gives an Error Code.
   Error Code 1, Hardware error (CRC, PE etc)
   Error Code 2, File Not Found
   Error Code 3, Driver Queue Overflow
   Error Code 4, Program requested illegal sector number.

3. **It doesn't type anything on the console.** This can be the worse, it could be real complicated, or real simple.
The I/O cable coming from under the Decwriter is vulnerable.
Power fail the Decwriter (especially if it hasn't been used since the last RHM session).
Try putting the LSI into HALT mode and then powering it up. You should get to the @ prompt. This is about the simplest thing the LSI can do.
You may have a duff Decwriter (or even a Baud rate problem). On an EIA decwriter, look at the DLV11 Berg pin F for data to the terminal, and Berg pin M is data from the keyboard.
Check the LSI power with a scope.

4.4 Fixing problems getting to VMS banner

The sequence here is that >>> B invokes the command sequence DEFBOO.CMD. This sets up some registers, runs the ISP ROM code to check basic KA sanity and find working memory, then load and run VMB.EXE. VMB will load CI microcode (if the CI is present), bring in SYSBOOT off the big disk, and fire up VMS.
Assuming these files are all available and uncorrupted.

You will probably run diagnostics starting with the micros, and working thru to the complete load path.
But, to stair-step it a bit, do **@DEFBOO.CMD** to the >>> prompt.

You get a trace of each line as it is executed.

- First check that all the register set ups make sense. See the register definitions for VMB in Chapter 7, VMS. Particularly, ensure it is the **correct Unit number, CI and HSC node number.** Also ensure the **HSC ONLINE button, and the correct DRIVE PORT button is pressed.**
- If it fails **before START 20003000.**
  You have a very basic problem, run micros.
- If it fails **after START 20003000 but before G E 00000200**
  The KA has failed its sanity check or finding working memory.
  - If it is some message (apart from Machine Check) Peruse Section 4.1.7.6, ISP ROM flow to understand what you are dealing with.
  - If it is a Machine-Check or Double-Error-Halt, look at the Stack Frame or Internal Registers. See Section 4.5, Machine-Checks and Double-Error-Halts.
    But remember that the wrong register set-up can cause this.
  - If it loops, do a **^P HALT**, to get the PC.
    Peruse Section 4.1.7.6, ISP ROM flow to understand what you are dealing with.
- If it returns something other than **G E 00000200** at the end of the ISP ROM code, you have bad memory. Did you just run Micros? Try power-cycling the memory; if problrm persists, fix the memory.
- If it fails **after START @.**
  This means VMB has encountered a problem.
  - If it is a Machine-Check or Double-Error-Halt, look at the Stack Frame or Internal Registers. R9 will hold a pointer to the Stack Frame. See Section 4.5, Machine-Checks and Double-Error-Halts.
  - If it loops or hangs or "fails to initialize device", you could have a problem in the load-path.
    Look at lights on the drives and controllers.
    Run VTDPY on the HSC to monitor VCs being formed.
  - Try booting the diagnostics or Standalone backup, from a big disk.

4.5 Machine-Checks and Double-Error-Halts

When the KA micro code detects a machine check, it gathers the contents of certain registers it thinks will be usefull to the maintenance humanoid.
It transfers the contents of these chosen registers to a set of temporary ID Registers.
Then it transfers these ID registers onto the Stack in memory.
At this point one of two things can happen.

1. If this part of the sequence fails (eg can't get to memory) the microcode will enact a **Double-Error-Halt**.
   Fortunatly for the humanoid, the copy of the registers for the first error are still in the temporary ID registers.
   We need to examine the contents interpret them to identify the problem.

   To recover the copy of these registers from the temporaries :-

   E/ID/N:9 30
   Will respond with :-

   ID 30 xxxxxxxx - this is the Summary Parameter
   ID 31 xxxxxxxx - this is the CPU error register
   ID 32 xxxxxxxx - this is the trapped UPC
   ID 33 xxxxxxxx - this is the VA/VIBA
   ID 34 xxxxxxxx - this is the D register
   ID 35 xxxxxxxx - this is the TB ERR 0
   ID 36 xxxxxxxx - this is the TB ERR 1
   ID 37 xxxxxxxx - this is the timeout address
   ID 38 xxxxxxxx - this is the Parity register
   ID 39 xxxxxxxx - this is the SBI error register

   You now have the information relating to the **first machine check** and can proceed to Section 4.5.1, Decoding Machine-Checks and Double Error Halts.

2. After putting the registers to the stack, the microcode bungs a 28 onto the stack, to tell the software how big this stack-frame is. (The 28 is useful for us at other times, as it enables us to "spot" machine-check stack-frames on the stack and subsequently decode them if necessary.)
   The microcode then calls the **machine-check** handler by interupting through Vector 04.
   Depending what software is running, different things happen.
   We will consider the cases of the **ISP ROM** software, the **VMB** software or **VMS**.

   - **VMS** - this would have put the entry into the erorlog and you print out the machine check entries in your favoured manner.
     Then you simply read off the formatted entries to arrive at the cause (eg Control Store Parity Error) and proceed through the specific section in Section 4.5.1, Decoding Machine-Checks and Double Error Halts.
     Just be careful that the Cache PE bits are set for o/k, and the errorlog formatter reports them as "o/k", you have to look to see if an "o/k" comment is missing!

   - **ISP ROM** - this will have looped during the boot flow, with a FATAL — UNEXPECTED MACHINE CHECK message.
     To look at the machine check stack frame on the ISP stack :-

     ^P
     H
     E SP
     E/N:A @

   - **VMB** - this will have halted during the boot flow, with a %BOOT - Unexpected Machine Check message.
     To look at the machine check stack frame on the VMB stack :-

     E/G 9
     E/N:A @

Whether you were running ISP or VMB, you are now looking at the **machine check stack frame** :-

(SP+00) xxxxxx28 - this acts as a verification.
(SP+04) xxxxxxxx - this is the Summary Parameter
(SP+08) xxxxxxxx - this is the CPU error register
(SP+0C) xxxxxxxx - this is the trapped UPC
(SP+10) xxxxxxxx - this is the VA/VIBA
(SP+14) xxxxxxxx - this is the D register
(SP+18) xxxxxxxx - this is the TB ERR 0
(SP+1C) xxxxxxxx - this is the TB ERR 1
(SP+20) xxxxxxxx - this is the timeout address
(SP+24) xxxxxxxx - this is the Parity register
(SP+28) xxxxxxxx - this is the SBI error register

You now have the information relating to the **unexpected machine check** and can proceed to Section 4.5.1, Decoding Machine-Checks and Double Error Halts.

4.5.1 Decoding Machine-Checks and Double Error Halts

The approach to identifying the cause of a **machine-check** or **double-error-halt** is the same, once you have found the contents of the error registers at the time of failure.
You will be looking at the **Stack or Errorlog for Machine-Checks**
You will be looking at the **ID registers for double-error-halts**.

The following is provided as a guide for "seminared" engineers, it does not give the whole truth and nothing but the truth, it should be good enough for most instances.
Look at the least significant byte of the **Summary Parameter**, either (SP) + 4, or ID 30.

- **00 or 0D or F0, Read Timeout or Error Confirmation**

  Look in (SP) + 20, or ID 37 for the timeout address.
  It is in a "funny" format. Ignore the top nibble, then shift the rest by 2 binary places (ie convert <27:00> to binary, then add 2 zeroes) this gives a physical address.

  Look in (SP) + 10, or ID 33 for the Virtual Address.

- **02 or 0A or F2, Translation Buffer Parity Error**
  Look in (SP) + 1C, or ID 36, for the T.Buff Error Reg 1.
  If any bits <09:14> are set, you have a TB Address Tag P.E.
  If any bits <15:20> are set, you have a TB Data P.E.
  - see Section 4.5.2, Troubleshooting CPU Parity Errors

- **03 or 0F or F3, Cache Parity Error**
  Look in (SP) + 24, or ID 38, for the Cache Parity Error Register.
  If any bits <00:05> are CLEAR, you have a Cache Address Tag PE.
  If any bits <06:13> are CLEAR, you have a Cache Data PE.
  - see Section 4.5.2, Troubleshooting CPU Parity Errors

- **05 or 0C or F5, Read Data Substitute**
  - you need to look at the memory registers, someone had a double bit error.

- **F1, Control Store Parity Error**
  Look in (SP) + 0C, or ID 32, for the Trapped Microcode Address Register.
  - see Section 4.5.2, Troubleshooting CPU Parity Errors

- **F6, Microcode "not suppossed to get here"**
  This is awkward, generally means there is some "transitionary" fault.
  For instance a signal sent us down a microcode flow but then disappeared.
  I have only ever seen this at the same time as other symptoms.
  Hopefully, the other symptoms will give a clue, else it is a case of looking at the microcode listing to see what happened.

4.5.2 Troubleshooting CPU Parity Errors

The following sections are to help you isolate the cause of any of the cpu parity errors. There are sections for **Translation Buffer, Cache, and Control Store** parity errors.
The numbered lines within each section list :-

1. The most probable cause (the module the error detection logic was built to monitor).
2. The module which has the error register bits.
3. The module which generates the parity bits to store with data.
4. The module which checks the parity of the data from the store.
5. The module giving data to the parity bit generator.
6. The module addressing the store.

4.5.2.1 CONTROL STORE PARITY ERROR

1. Probable cause PCS or WCS (11780), JCS (11785)
   On 11780 PCS in slot 22 if address 0000-0FFF, else WCS in slot 18 or 20.
   On 11785 Slot 18 if address 0000-0FFF, Slot 20 if address 1000-1FFF.
2. Error register is on ICL
3. Parity bits generated on WCS when writing
4. Parity checked on PCS - this checks the parity of the CS bus which, though driven by the CS modules, goes all over the cpu
5. Data to WCS over ID bus from CIB
6. Control Store is addressed by USC from CSbus,IRC,CEH,CIB

4.5.2.2 CACHE ADDRESS TAG PARITY ERROR

1. Probable cause CAM
2. Error register is on SBL
3. Parity bits generated on CAM
4. Parity checked on CAM
5. Data to generator over PAbus from TBM,SBH,SBL
6. Address tag store addressed over PAbus from TBM,SBH,SBL

4.5.2.3 CACHE DATA PARITY ERROR

1. Probable cause CDM or MDbus
2. Error register bits on SBL
3. Parity bits for byte <0:1> generated on SBL on read memory
   Parity bits for byte <2:3> generated on SBH on read memory
   Parity bits for byte <0:3> generated on DBP on write memory
4. Parity checked on CDM
5. Data to generators on SBL/SBH comes from SBI
   Data to generator on DBP is from data path block
6. Data store addressed over PAbus from TBM,SBH,SBL

4.5.2.4 TB ADDRESS TAG PARITY ERROR

1. Probable cause CAM
2. Error register is on TBM
3. Parity bits generated on CAM
4. Parity checked on CAM
5. Data to generators from DDP,DEP
6. Address tag store addressed by DDP,DEP

4.5.2.5 TB DATA PARITY ERROR

1. Probable cause TBM or IDbus
2. Error register is on TBM
3. Parity bits generated on DBP
4. Parity checked on TBM
5. Data to generators on DBP is from data path block
6. Data store addressed by DDP,DEP

4.5.3 Example of Double Error Halt situation

Actual case, SDRC Hitchin, May 1990.

```
>>>B

      CPU HALTED
      INIT SEQ DONE
      ?CPU DBLE-ERR HLT
      HALTED AT 20003798
```

So we have blown away whilst running the ISP ROM software.

```
>>>E/ID/N:9 30

   ID 00000030  000000F1
   ID 00000031  00000002
   ID 00000032  000002A8
   ID 00000033  00010000
   ID 00000034  00000010
   ID 00000035  00007C00
   ID 00000036  00000040
   ID 00000037  08000C34
   ID 00000038  00004000
   ID 00000039  00000002
```

From this, a Summary Code of F1 shows it is a CONTROL STORE PARITY ERROR, and the TRAPPED UPC (000002A8) is in the first JCS.

```
>>>TE
Z2-ETKAB-V2.1, VAX-11/785 MICRO DIAGNOSTIC MONITOR

ETKAD -- V1.0, VAX-11/785 MICRO DIAGNOSTIC #1 (HARDCORE)
01,02,03,04,05,06,

ETKAD -- V1.0, VAX-11/785 MICRO DIAGNOSTIC #1 (HARDCORE)
ERRORPC: 000162  TEST: 001F  SUBTEST: 0001
SECTION: 06

DATA: 1000
      1001
      0229

TRACE:
FAILING MODULES: JCS(M7475)

MIC>
```

Notice the Micros called out the JCS module but did not specify which one!

4.6 Running MICROS. Lowest level CPU, SBI and memory test

- If you have a Systems Industries RH look-a-like see Section 4.1.8.4, SI "RHs".
- The second Micro does an initial check of memory assuming the arrays are good, this can fail calling out a controller, even though it is really a faulty array. If you have tried the controller and things are still bad, reduce the arrays to a minimun and then try new ones.
- Sometimes an old Micro will call out an array using the old number (eg M8211), just interpret it as "the array card".
- If you have a Micro failure, look at the **VAX Diagnostic Chapter** for known reasons (particularly if you have already replaced the module being called out!).
- After running the Micros, it is a good idea to power cycle the memory in case they have been left in a "funny state". Typical symptoms of this is VMS thinks there is only half as much memory as is really available, or solid errors get left in memory.
- You don't need the CPU microcode to be loaded to run micros - so you can do a ^C once you see the (RELOADING WCS) message. This gives you an acceptable short-cut to the >>> prompt.
  It will give you some messages about WCS which you can ignore.
- You start the micros from the >>> prompt by taking out the console floppy and putting in the micros floppy and typing **TE**.
  Once the first one has run and returned to the MIC> prompt, put the next floppy in and just type **DI**. However, you can go straight to the second floppy if you want.
- If the tests fail a long way in, you can restart at a specific section during troubleshooting (Sections are "traced" as the micros run).
  Instead of using **>>> TE**, use **>>> TE/C**.
  Then **MIC> DI/SEC:xx/CON** will start at section **xx** and carry on to completion.
- If a micro fails, calling out a module, you can always **MIC> CON** to see what it calls out next along the line.

Table 4–5: Micro diagnostic floppies

FLOPPY	TITLE	Usage
RX02	11780 MICRO 1	Part 1 of micro tests
RX03	11780 MICRO 2	Part 2, for systems with MS780C
RX08	11780 MICRO 3	Part 2, for systems with MS780E/H
RX22	11780 MICRO	Part 3, for systems with MA780
RX42	11780 MICRO 1	Part 1, for systems on RDC
RX43	11780 MICRO 2	Part 2, for systems on RDC
RX02A	11785 MICRO 1	Part 1 of micro tests
RX03A	11785 MICRO 2	Part 2, for systems with MS780C
RX05	11785 MICRO 4	Part 4, for systems with FP785
RX08A	11785 MICRO 3	Part 2, for systems with MS780E/H
RX22A	11785 MICRO	Part 3, for systems with MA780

The Micro-Diagnostic structure consists of a "MONITOR" (ESKAB on a 780).
First, this calls in a "HARDCORE" control program (ESKAC on a 780), this uses a test stream (ESKAD on a 780) of SECTIONS 01 to 1F to checkout that it can poke at the KA over the visability bus, and it can reliably use the KA control store.
Second, it calls another control program (ESKAE on a 780), this uses a test stream (ESKAH on a 780) of SECTIONS 20 to 3D to run little routines within the KA control store to checkout the KA cpu.

Thirdly, you manually (using DI) make it call another test stream for the external world (ESKAR on a 780).

It prints out the section number before it runs it

Each section can "bomb out" if some TRAP condition occurs, so you will not get a module call out then.
But you do get a printout similar to a machine check stack frame.

The reported TRAP CODE being similar to the summary parameter.
0=System Init, 1=unaligned data, 2=Page error,
3=M bit, 4=Access Violation, 6=TB Miss,
7=TB parity, 8=Cache parity, 9=Reserved FP oper.,
C=RDS, D=Timeout, F=Control Store Parity, 10=Odd Address Error.

4.6.1 Typical micro diagnostic session

You have to run the sequence of micro-diagnostic floppies from Table 4-9, depending on the system configuration you are faced with.

In this case we are looking at a 11785, with MS780H and FP785.

```
CPU HALTED, SOMM CLEAR, STEP=NONE, CLOCK=NORM
RAD=HEX, ADD=PHYS, DAT=LONG, FILL=00, REL=00000000
INIT SEQ DONE
HALTED AT 00000000

RELOADING WCS ^C
LOAD DONE, 0015 MICROWORDS LOADED
VER: PCS=04 CON=V06-00-C
?WARNING - WCS LOAD INCOMPLETE
?FATAL ERROR: WCS NOT LOADED
```

[Here you remove the console floppy and insert RX02A - 11785 Micro1]

```
>>>TE

ZZ-ETKAB-V2.1, VAX-11/785 MICRO DIAGNOSTIC MONITOR

ETKAD—V1.0, VAX-11/785 MICRO DIAGNOSTIC #1 (HARDCORE)
01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,14,15,16,17,
18,19,1A,1B,1C,1D,1E,1F,
ETKAM—V1.0, VAX-11/785 MICRO DIAGNOSTIC #1 (CPU GO CHAIN)
20,
SYSTEM ID REGISTER = 018C91E2
21,22,23,24,25,26,27,28,29,2A,2B,2C,2D,2E,
2F,30,31,32,33,34,35,36,37,38,39,3A,3B,3C
END PASS 0001
```

[Here you remove RX02A and insert RX08A - 11785 Micro3]

```
MIC> DI

ETKAR—V3.0, VAX-11/785 MICRO DIAGNOSTIC #3 (MS780-E,F GO CHAIN)
3D,3E
CPU TR = 00000010
MS780H 256K CHIP AT TR 00000001
LOWER CONTROLLER MAX ADDRESS+1= 01000000
UPPER CONTROLLER MAX ADDRESS+1= 01000000
DW780 AT TR 00000003
DW780 AT TR 00000004
```

```
RH780 AT TR 00000008
RH780 AT TR 00000009
RH780 AT TR 0000000A
3F,40,41,42,43,44,45,46,47,48,49,4A,4B,4C,4D,4E,4F,50,51,52,53,54,55,56,
MS780-E/H IO BASE ADDRESS = 20002000
LOWER CONTROLLER MAX ADDR + 1 = 01000000
BOARD NUMBER = 00000000
NUMBER OF CRD ERRORS = 000000000
BOARD NUMBER = 00000001
NUMBER OF CRD ERRORS = 000000000
BOARD NUMBER = 00000002
NUMBER OF CRD ERRORS = 000000000
BOARD NUMBER = 00000003
NUMBER OF CRD ERRORS = 000000000

MS780-E/H IO BASE ADDRESS = 20002000
UPPER CONTROLLER MAX ADDR + 1 = 01000000
BOARD NUMBER = 00000000
NUMBER OF CRD ERRORS = 000000000
BOARD NUMBER = 00000001
NUMBER OF CRD ERRORS = 000000000
BOARD NUMBER = 00000002
NUMBER OF CRD ERRORS = 000000000
BOARD NUMBER = 00000003
NUMBER OF CRD ERRORS = 000000000

MS780-H 256K CHIP AT TR 00000001
M8376 ROMS OK
57,58,59
END PASS 0001
```

[Here you remove RX08A and insert RX05 - 11785 Micro4]

```
MIC> DI

ETKAU-V3.0, VAX-11/785 MICRO DIAGNOSTIC #4 (FP785 GO CHAIN)
01,02,03,04,05,06,07,08,09,0A,0B,0C,0D
END PASS 0001

MIC>
```

The 11780 is basically the same, except the FP780 is tested in the second part of the micro diagnostics, so there is no Micro4.

4.7 Running EVKAA and ETKAC. Next level CPU test.

Remember to look at Chapter 14, VAX DIAGNOSTICS.

After the micros, one would feel confident that the VDS should function, however, **occasionally micros work but VDS crashes.** Remember that, even after micros, we still have not run any macro-code in the KA.

In these circumstances, there are two diagnostics (loadable from the RX) which are in VAX macro-code and run basic tests on the KA.
For these you must have the KA microcode loaded correctly.

The first of these intermediate diagnostics is **EVKAA, Basic Instruction exerciser.**
It is found on RX62, VAX INST 1.
To run it:-

Get to the >>> prompt, without any error messages.
Remove the consol floppy and put RX62 into the drive.
Type **LOAD EVKAA.EXE/START:0**
Type **START 200**
You should get EVKAA DONE! messages every few seconds.

The second of these intermediate diagnostics is **ETKAC, A standalone version of EGYPTD**
I can't make out what number this floppy is, (like the XXDP one, its from a different part of our organisation).
But it should be labelled something like AS-U112A-AE, VAX 785 FLOATING POINT RX01.
To run it:-

Get to the >>> prompt, without any error messages.
Remove the consol floppy and put the floppy into the drive.
Type **LOAD ETKAC.EXE/START:200**
Type **START 200**
You get a bit of speil, telling you how to alter the pass count.
The test terminates with a "Everything is O/K" message.

The FPA is a good thing to pull out, if micros are o/k but VDS crashes. The FPA can corrupt the data paths, and it doesn't just do floating instructions!

4.8 Running VDS. The CPU, NEXUS and device tests

Remember to look at Chapter 14, VAX DIAGNOSTICS. All the floppies listed in EVNDX can be built using DUCT, including the MICROS but you must be in possesion of VAXPAX 36 or earlier, as the distribution has been discontinued.

4.8.1 Loading the diagnostic supervisor

ESSAA is the Supervisor for both 11780 and 11785, and can be got from the floppy or a big-disk.

- It takes a bit of time, but it is easy to get the Vax Diagnostic Supervisor in from a floppy (use RX7).
  VMB is not used, a straightforward LOAD command does the trick.
  To use RX7, the Diagnostic Supervisor floppy. The factory have been kind enough to make our lives easier, just insert RX7 and type B. They have provided a DEFBOO.CMD with the LOAD and START commands in it. [for interest it is LOAD ESSAA.EXE/START:FE00 then START 10000].

- More commonly, we use the copy on the System Disk.
  Do **@DEFBOO.CMD**
  When it prints **LOAD VMB.EXE/START:@** hit **^C**.
  Now **D/G 5 x0000010** where x is zero or the same as was being deposited in R5 anyway.
  Manually type in the **LOAD VMB.EXE/START:200** command.
  Start VMB, **START 200**
  This will get DIAGBOOT then ESSAA in for you.

4.8.2 Attaching the CPU and devices

Normally you will just run the VDS Autosizer (EVSBA), found on RX61.
The only problem is that it will make a mistake attaching the CI780 if the ucode is not loaded, it calls it PAA0 but gets the node number wrong. If this happens you do a manual attach as PAA1, using the correct node number, and then select and test PAA1 instead of PAA0.

- Attaching and testing the CPU
  It wants to know whether you have G, H, size of WCS, type of FPA.
  The size of WCS seems to be irrelevant.
  The FPA is 0 if missing or 1 if fitted.

  A 11780 will look like this:-
  ATTACH KA780 HUB KA0 NO NO 07FF 0 or 1

  The 785 has G and H floating point as standard so use:-
  ATTACH KA785 HUB KA0 YES YES 1FFF 0 or 1
  You will then run

  EVKAB off RX62,
  EVKAC, EVKAD, and EVKAE off RX25

- Attaching and testing the DW780.
  This just needs to know the TR and BR levels of the DW780.
  ATTACH DW780 HUB DW0 3 4
  You will then run EVCBA off RX61 (on very old floppies this will be called ESCBA).

  ESCBA will run on the DW even when the unibus cable paddle card has been disconnected. It will also drive the unibus exerciser (UBE-PMK05) without any manual intervention. Driving the PMK05 can test the unibus for you without any other options fitted. Also the UDA test is a good unibus buster. RUN EVRLA/TEST:1:1/PASS=5 and then rattling all the options on the unibus has proved worthwhile.

- Attaching and testing the RH780.
  This just needs to know the TR and BR levels of the RH780.
  ATTACH RH780 HUB RH0 8 5
  You will then run EVCAA off RX61 (on very old floppies this will be called ESCAA).

This will report all the massbus drives it can see.

- Attaching and testing the CI780.
  This just needs to know the TR and BR levels and Node number of the CI780.
  ATTACH CI780 HUB PAA0 14 4 n

  You will then run EVCGA, EVCGB, EVCGC, EVCGD repair level diags off RX63.
  These diagnostics do not use the "real" CI microcode (ie CI780.BIN), they just load little noddy routines into the CI780 and step thru them.

  Follow up with (or go direct to) EVGAA and EVGAB the functional diagnostics off RX72.
  These need the real CI microcode to be available or loaded.
  As it will not normally be loaded (either you got ESSAA off the RX and did not run VMB to load it, or it got overwritten by the Repair diags) there is a switch you can set before starting the diagnostic to force it to load up the CI780 file for you.
  So put RX72 in and **LOAD EVGAA**; then **SET EV 1**; remove RX72 and put the console floppy back in; then **START**.

- Attaching and testing devices.
  First the controller (DW780 or RH780) has to have been attached/selected, then you just attach the device controller (eg UDA50) to it.
  e.g. ATTACH DZ11 DW0 TTB 760110 310 5 EIA or ATTACH RP06 RH0 DBA0

- Diagnostics restricted to working under VMS.
  EVKAM is a on-line memory diagnostic.
  EVDAC, EVDAF and EVDAH are the on-line DMF32, DMZ32 and DHU11 tests.
  For any particular device use the VDS HELP facility, as a general confidence test the **UETP Load Users Phase** is about the best in my experience.

4.9 Guessing a failing FRU from errorlog entries

If you are looking for an intermittent type of problem, you would normally run off the errorlog in search of clues.

First, get an overview of potential problems by doing
$ ANALYZE/ERROR/NOFULL/SUM
You can then see what areas are worth investigating.

- **Machine Checks**, see Section 4.5, Machine-Checks and Double-Error-Halts.
- **Memory errors**, these are self explanatory.
  The only problem encountered so far is in the array numbering on MS780E/H.
  The errorlogger numbers the arrays on the Lower Controller (slots 09 thru 02) as array 00 to 07, and those on the Upper Controller (slots 13 thru 20) as array 10 thru 17.
- **SBI errors**, a reasonable understanding of the SBI is needed to be able to progress this kind of problem. See Section 4.1.3.2, SBI NEXUS Configuration and Status Register for possible causes.
- **DW780 errors**, peruse Section 4.1.5, DW780.
  There is also a "softy" problem which causes Invalid Map problems if a soft error occurs during DMR11/DMC11 transfers.
- **DW780 RH780 CI780 errors**, generally an error (typically something like an internally detected PE) can be tied to a module using a block diagram or just knowing the module functionality as per the utilization chart I've listed in each section.

4.10 Running standalone backup

Backup can be run from console media or from a "big disk".

From floppies, you need a kit of about 5 floppies.
First, keep the normal consol floppy in and type **>>> B CS1**
The sequence CS1BOO.CMD will set up the registers and invoke VMB.
VMB will prompt you to start feeding in the floppy kit.

From "big disk", this assumes the customer has built a copy some time previously.
The standard is to have built it in Root E, and made a command file called STBBOO.CMD to boot it.

So, on a good day, just need **>>> B STB**.

But if they forgot to build STBBOO.CMD,
Do **@DEFBOO.CMD**, as soon as the line **LOAD VMB.EXE/START:@** is reported do a **^C**.
Now do **>>> D/G 5 E0000000**.
Then do **>>> LOAD VMB.EXE/START:200**
And then **>>> START 200**
VMB will drag standalone backup in from the specified device.

4.11 Altering files on the console floppy

You (A/R?) may need to update files on the console or diagnostic floppies.
Typically CPU or CI microcode, and DEFBOO.CMD are the files you are interested in.

There have been lots of pitfalls in this area over the years (media full, directory corruption, incorrect file types), most have been rectified.

What follows is **foolproof** when operating on a console floppy.
NEVER overwrite the one currently in use.
Try to understand what you are trying to achieve. For instance, a recent FCO sent out a USA floppy and told engineers to copy the site's DEFBOO.CMD to it - some people followed these instructions and caused problems later (machine crashed at the next RHM session).

So:-

- Save the existing floppy to the system disk.
  $ @SYS$UPDATE:CONSCOPY
- Invoke the RT11 media handling utility.
  $ EXCHANGE
- Pretend the system disk copy (really just a file) is a floppy drive.
  EXCHANGE> MOUNT/VIRTUAL FSE: CONSOLE.DSK

  Either, copy new microcode to the virtual disk from a FCO distribution floppy.
  EXCHANGE> COPY CSA1:CI780.BIN/TRAN=BLOCK FSE:*.*
  Or, copy a prepared boot sequence file to the virtual disk from the system disk.
  EXCHANGE> COPY STBBOO.CMD FSE:*.*
- Use the /TRAN=BLOCK switch on binary files, leave it off for ascii ones.
- Finish modifying the virtual disk.
  EXCHANGE> EXIT
- Put a SCRATCH floppy in and Restore to it.
  @SYS$UPDATE:CONSCOPY
  This procedure will write the boot area on the floppy so the LSI loads CONSOL.SYS when it powers up.
  This is the bit that will hiccup on a diagnostic floppy update, because there is no CONSOL.SYS, but the resultant floppy is still usable.

4.12 FCO status

This is just a reminder, all systems should have been done ages ago. I doubt that anyone could get EQ kits, but this information might be useful, especially if a system has been "away". The revision is reflected in the SID; if the minor field (rev) is 1, this is interpreted as "B".

- 11780s with MS780C should be rev 8.
- 11780s with MS780E/H should be rev 8B.
- 11785s with MS780C should be rev 3.
- 11785s with MS780E/H should be rev 3B.

4.12.1 11780

11780 rev 8x: was to fit WCS126.PAT and a USC at CS N1 or N2.
Originally it was WCS125.PAT, but there were problems, and it was put on hold, and then they re-released it with WCS126.PAT.
If you follow the instructions with this FCO, you end up without RDC software!
There was a "microcode patch" floating around for WCS124.PAT, make sure your DEFBOO does not have any DEP/ID 22 nnnn commands in it.

11780 rev 7x: was to fit WCS124.PAT; USC at CS H3 or M; IRC at E; 2 kw WCS. (G and H now a possibility).

11780 rev 6: was to make M8227 rev B1 or D; M8228 at rev B1 or D.

11780 rev 5: was to the FP780, made M8289 rev E or C1. Fixed POLYD problem.

4.12.2 11785

11785 rev 3x: is to fit SSU01E.WCS; M7460 at CS D1; M7472 at C1 or C2; Data Paths become -YAs; and there is a backplane re-work.
This fixes a chip-tolerance problem; before the FCO, we used the work around of putting a SET CLOCK SLOW command in DEFBOO.CMD.
So EDIT this out if you stumble across it!

11785 rev 2: was to fit SSU01C.WCS; this fixed a REMxI micro-code bug.
There was a "microcode patch" floating around for SSU01A and SSU01B, *make sure your DEFBOO does not have any DEP/ID 22 nnnn commands in it.*

11785 rev 1: was to fit new M7460

4.12.3 FP785

At the same time as the KA was made rev 3x, the M7540 should be D1 or D2, the M7541 should be E1, the M7542 should be D1, the M7543 should be A1 and the M7544 should be D1.
This fixes an accuracy problem, as well as failures occuring on clock margins!

4.12.4 MS780 E/H

There were problems with the MS780E/H, doing the FCO bought the SID to sub-revision level B. The action was to fit a new 3-board MS780E/H controller (M8375s at D1, M8376 at E1)

4.12.5 MS780 C

The M8212 was made up to D. And a thicker power wire was inserted on the backplane.

4.12.6 DW780

The M8271 was made up to D1 or D2.
The M8273 was made up to D.

4.12.7 RH780

The M8276 was made E, the M8277 was made H.
Two backplane wires were also added, E02J1 to E03C1 and E02C1 to F03E2.

The M8277 was replaced by the M8274 (bigger silo).

The M8275 was made J1, to stop Multiple Transmitter Faults.

TT30 - Rev 7 introduction
TT31 - Arrays stop the PSU coming up.
TT32 - RX01 power cord
TT33 - Getting registers at a crash
TT34 - LSI trap thru 4
TT35 - SBI cables sensitive
TT36 - MS780 printset error
TT37 - CI780 causes Micro2 error
TT38 - Metering SBI cables and terminator
TT39 - deskidding fcc cabs
TT40 - use Micro3 on MS780-E
TT41 - error in RDC DLV11E installation manual
TT42 - mixing MS780 C and E
TT43 - duct
TT44 - MS power fail jumpers
TT45 - RH FCO
TT46 - CI attenuator pads
TT47 - CI and grounding issues
TT48 - CIBOO.CMD
TT49 - drive detected errors on HSC50
TT50 - mscp codes
TT51 - RA81 boot file
TT52 - TM03 noddy
TT53 - explains mauve fiche
TT54 - RX01 sharp edges
TT55 - Tool for MS780C FCO
TT56 - cab trim
TT57/58 - 11785 upgrade, module return
TT59 - new LSI configuration
TT60 - L0100, rev really B2 or D1
TT61 - System and Option Catalogue
TT62 - CIB printset error
TT63 - TM78 diagnostic problem
TT64 - Blower wires worn thru to housing **safety alert**
TT65 - 869-D, loose connections in mains plug

Consol rev 10.0 may not respond to ^P
FP780 M8289 FCO, etch not cut at E28p8 side 2

4.14.3 VAX 11785

TT1 - RX01 sharp edges
TT2 - 11780 pre-req revision for upgrade
TT3 - LSI memory
TT4/5 upgrade return modules
TT6 - LSI configuration
TT7 - slow chip, fault symptoms
TT8 - M8017-AA, R2 out for RDC (book wrong)
TT9 - TM78 diagnostic problem
TT10 - Blower wires worn thru to housing **safety alert**
TT11 - 869-D, loose connections in mains plug

The M7540 has some vulnerable resistor banks, if they get pressed together, the backplane can get fused.

4.14.4 KC780 and KC785

TT1 - floppy fan should SUCK
TT2 - CIB ROM checking
TT3 - DLV11E jumpers
TT4 - floppy upgrade, M7744-YC switches.
TT5 - RDC jumpers
TT6 (785 TT1) - new LSI memory
TT7 (785 TT2) - LSI hangs

4.14.5 Power Supplies

TT1 - "no load" testing procedure
TT2 - noisy aclo on power up
TT3 - false aclo
TT4 - order whole power supply

TOD register getting cleared on power up due to sensitive 54-12763

4.14.6 MS780

TT1 - multiple MS configuration
TT2 - interleaving
TT3 - interpreting reg C
TT4 - UNIX parity message
TT5 - M8210 caps shorting
TT6 - mixing -C and -E
TT7 - Micro3 for -E
TT8 - backplane wire FCO for -C
TT9 - power fail jumpers
TT10 - M8210 caps

MS780E/H TT1 - EMCC array problems

See speedy 293 MAST for installation manual
Listing of MS780C ISP ROM is ES0AD

4.14.7 DW780

TT1 - DR11Bs
TT2/3 - backplane jumpers
TT4 - new diagnostic names
TT5 - diagnostic failures
DMR11 driver causes Invalid Map Register on soft error.

4.14.8 RH780

TT1 - M8275, bad ECO
TT2 - see TU77 TT8, correctable error
TT3 - FCO for fast RP07
TT4 - mu sticker
TT5 - more on fast RP07
TT6/7 - diagnostic failures
TT8 - multiple transmitter failures, good tech tip.
MASSFAIL jumpers?

TM03, cut w1 on H870 on M5903.
RP/RM, cut W2 on 70-09938 (also make M7684 rev P)

WRITECHECK errors cause massbus exception, then EBL, then ABORT; so no DCK logged, because ECC is not done!
AWRE Micro2 timeout in Test 147 due to PSU!

4.14.9 CI780

TT1 - documentation
TT2 - CI780.BIN revision
TT3 - cable alert
TT4 - L0101 rev level (H1=ROM3, J1/K1=ROM7)
TT5 - CI termination
TT6/7 - diagnostic failures
TT8 - QC problem on boards with "HPc" LOGO
TT9 - L0100 rev marker wrong
TT10 - getting revisions
TT11 - updating CI780.BIN
TT12 - new diagnostic names
TT13 - diagnostic notes
TT14 - getting revisions of binary file, after rev 6
TT15 - CI780.BIN revision 7

L0102 at D1 to fix PEs
All 12 LEDS on means "reset" state.

DO YOU HAVE

ANYTHING

TO ADD TO

Data Doc ?

4.13 Part Numbers

11780 backplane 70-13628
11785 backplane 70-20579-02 (or -01 if pre-REVISION 3)
DW780 backplane 70-13626
RH780 backplane 70-13627
MS780C backplane 70-13625
MS780E backplane 70-19729
CI780 backplane 70-17654

PS#1,3 are H7100B's (normal just +5V)
PS#2 is a 70-14956-01 (H7100, H7101)
MS780C uses a 70-14957-01 (H7100, H7102, H7103)
MS780E uses a 70-18759-01 (H7100, H7103, H7107)
CI780 uses a 70-14956-01 (H7100, H7101)
The memory BBU is a H7112
The TOD BBU is a H7111

TOD charger 54-12763
TOD battery 12-12499 (also used in memory BBU)

MS780C array M8210 - 0.25mB
MS780E array M8373 - 1.0 mB
MS780H array M8374 - 4.0 mB
INTEL array +L10007 - 1.0 mB (I believe this is for a MS780C backplane)

SBI CABLES 17-00087-00 (-01=for SFT, -03=18inch, -08=24inch, -04=36inch)

3 PHASE BLOWER 12-13893
AIRFLOW SENSOR 70-14320-0E
869E CONTACTOR 12-12266

RX01 PEEWEE fan 70-14767
RX01 solid state relay 12-14417
LSI power supply H780-B
LSI fan 12-10719
Wire for putting jumpers on LSI modules, 29-10123

MASSBUS Transition piece 70-09861 (or 12-11591-19 plus 3 times 12-11591-23)
MASSBUS Terminator 70-09938

BA11-K, Fan, 12-11714-00
BA11-K, 9-slot Unibus backplane, DD11-DK.
BA11-K, POWER LINE MONITOR / +15V REG, 54-11086.
DZ11 cable protection sheet, 12-15278.

Documentation 11780=70-19611-01, 11785=70-20580-01

4.14 Tech-Tips

4.14.1 KA780-KA785-KU

KU-1, Installation manual errors.
780-1, INTSTKINV on warm restart
780-2, TOD cable causing warm restarts
780-3, micros leave MS in funny state
780-4, SYE reports SBIREG wrong, also SBIALERT on MA errors.
780-5, warm restart problems

785-1, SID
785-x, Tight wrap on F11P2-F12P2 at F12N1, micros call M7451.

Floppy timeouts caused by "connect OPA1" (8600 command).
OPA0 errors caused by /NOHANGUP default.

4.14.2 11780

NETT1 - Installation notes
NETT2 - computer room layout
NETT3 - dynamic bad blocks
NETT4 - DSC
NETT5 - backplane alignment
NETT6 - gunge on power connections
NETT7 - battery life is 2 years
NETT8 - see RP07 NETT4
NETT9 - unable to power down
NETT10 - BBU failures
NETT11 - WCS errorlog interpretation

TT1 - extenders
TT2 - door lock
TT3 - exhaust foam
TT4 - printset errors
TT5 - pcs/wcs/fpla revisions
TT6 - microcode revisions
TT7 - TU77 data correctable errors
TT8 - H7100 causes ACDCLO
TT9 - unibus addressing
TT10 - micro fast clock fails
TT11 - sbi length calculation
TT12 - sbi silo interpretation
TT13 - memory add-ons
TT14 - extender problems
TT15 - interleaving memory
TT16 - M8232 jumpers
TT17 - sysgen, bugreboot, auto-restart
TT18 - EVSBA the VDS autosizer
TT19 - updating consol floppy
TT20 - fast clock causes crashes
TT21 - unable to power down
TT22 - MS780 configuration register interpretation
TT23 - configuring VES memory
TT24 - PSU part numbers
TT25 - VDS bugs
TT26 - vaxpax 9
TT27 - microprogramming tools
TT28 - Black Book part number
TT29 - KE780 introduction