KA-K1170-MG-003

1170 MAINTENANCE SERVICE GUIDE

PREPARED BY: BERNIE LEDUC
: ED SMITH
CANADIAN PRODUCT SUPPORT

REPRINT: APRIL 1988

digital equipment of canada ltd • kanata, ontario, canada

The PDP1170 Maintenance Service Guide version 3 is now available for ordering for USA and CANADA Field Service. We are unable to provide this guide for outside of North America due to customs and licensing requirements.

For those who have never seen version 1, the 1170 Maintenance Service Guide is a made in Canada, 7½" by 10" binder sized troubleshooting guide that no 1170 engineer should be without. It contains information on Diagnostics, Bootstraps, Togglein's, Adjustments, Switches, Jumpers, Indicators, and Cabling diagrams. It also has block diagrams, registers and troubleshooting info on the CPU, Cache, Memory Management, Unibus Map, RH70, MJ11, MK11 and KY11R.

Anyone who would like to get on my distribution list for future updates and corrections to this guide can send me VaxMail at ENET node:

CGFSV1::SMITH

Also if you have any questions, suggestions or corrections to Version 3 please send them to me.

You can order your copy, complete with binder for $40.00 per guide, by sending:

DECMail PAT DAVIES @KAO
ENET KAFSV1::DAVIES

Please include the following information:

PART NUMBER: KA-K1170-MG-003
DESCRIPTION: 1170 Maintenance Service Guide
REQUESTERS NAME:
COST CENTER: (to be charged to)
COST CENTER MANAGER: (approving purchase)
INTERNAL MAIL STOP:
COMPLETE MAILING ADDRESS:

DOCUMENTATIONS

DIAGNOSTICS

BOOTSTRAPS

TOGGLE-INS

TECH-TIPS/FCRS

ADJUSTMENTS

SWITCHES, JUMPERS, INDICATORS

CPU

CACHE

MEMORY MANAGEMENT

UNIBUS MAP

RH70

MJ11

MK11

KY11-R

PM'S

CABLES

TROUBLESHOOTING AIDS

1170 MAINTENANCE GUIDE

CONTENTS

CHAPTER 1
DOCUMENTATION

DOCUMENTATION

A) DOCUMENTATION LIST AND AVAILABILITY

A) MANUALS

DESCRIPTION		PART NUMBER
—KB11B	PROC MANUAL	EP-KB11-B-PM-A
—KB11C	PROC MANUAL	EK-KB11-C-TM-A
—PDP-11/70	MAINT & INST MANUAL	EK-11070-MM-A
—KY11-R	TECH MANUAL	EK-KY11R-TM-002
—FP11-C	MAINT MANUAL	EK-FP11C-MM-001
—RWP05/06	MAINT MANUAL	EK-RWP56-MM-A1
—M9301	TECH MANUAL	EK-M9301-TM-001
—M9312	TECH MANUAL	EK-M9312-TM-003
—MJ11	MEM SYS MAINT MANUAL	EK-0MJ11-MM-A
—MK11	MOS MEM TECH MANUAL	EK-0MK11-TM-001

B) PRINTS

DESCRIPTION		PART NUMBER
—PDP-1170	MAINT PRINTS (KB11B)	MP-00823-00
—PDP-1170	MAINT PRINTS (KB11C)	MP-00822-00
—KY11-R	MAINT PRINTS	MP-00348-00
—FP11-C	MAINT PRINTS	MP-00038-00
—RH70	MAINT PRINTS	MP-0RH70-00
—MJ11	MAINT PRINTS	MP-0MJ11-00
—MK11	MAINT PRINTS	MP-00523-00

C) MAINTENANCE CARDS

DESCRIPTION		PART NUMBER
—MK11	MEM MAINT CARD	EK-0MK11-MC-001
—PDP 11/70	ELECT CONSOL COMMANDS	EK-E1170-MC-001
—PDP 11/70	MAINT CARD	(THROUGH ED. SERVICES)

D) PREVENTIVE MAINT WORKSHEETS

DESCRIPTION		PART NUMBER
—1170	PM WORKSHEET	EK-11070-WS
—RH70	PM WORKSHEET	EK-0RH70-WS
—FP11-B	PM WORKSHEET	EK-FP11B-WS

E) OTHERS

DESCRIPTION		PART NUMBER
—1170	PROC IPB	EK-01170-IP
—PDP-1170	MEM SYS MAINT GUIDE	EK-11070-HP-001

CHAPTER 2

DIAGNOSTICS

DIAGNOSTICS

All of the PDP1170 diagnostics rely on the previous diagnostics verifying sections of the hardware. They should be run in the following order:

Latest Rev.	Name
EKBAD0	CPU TEST PART 1
EKBBF0	CPU TEST PART 2
EKBCD1	CACHE TEST PART 1
EKBDE0	CACHE TEST PART 2
EKBEE1	MEMORY MANAGEMENT TEST
EKBFD1	UNIBUS MAP TEST
EKBGC0	PDP1170 POWER FAIL TEST
EQKCE1	PDP1170 INSTRUCTION EXERCISER
EMJAD0	MJ11 MEMORY TEST
EMKAB0	MK11/MJ11 MEMORY TEST
ERHAE1	RH70 TEST
EFPAA0	FP11-C TEST PART 1
EFPBA1	FP11-C TEST PART 2

The following is a summary of the diagnostic setups and commands:

EKBA

Starting Address(s): 200
Switch Register: None

Notes:
**Halts on Error.*
***First pass runs with cache disabled.*

EKBB

Starting Address(s): 200
Switch Register:

BIT15	Halt on error
14	Loop on test
13	Inhibit error typeouts
12	Inhibit T-bit trapping
11	Inhibit iterations
10	Ring bell on error
09	Loop on error
08	Loop on test in <7:0>
07	No action
06	Skip BR6 tests
05	Skip BR5 tests
04	Skip BR4 tests
00	Skip operator intervention tests

Notes:
**First pass runs with Cache disabled.*
***This diagnostic will also write on any write enabled disk drives. You must spin down or write protect any diagnostic pack or customer software pack to prevent a disaster. If you wish to use this feature simply mount a SCRATCH pack on any drive and write enable it.*
****This diagnostic looks for an RH70 to do BR5 tests. If there are none in the system it will print "NO BR5 DEVICES". This is not an error message.*

EKBC/D

Starting Address(s): 200

Switch Register:

BIT15	Halt on error
14	Loop on test
13	Inhibit error typeouts
12	No action
11	Inhibit iterations
10	Ring bell on error
09	Loop on error
08	Loop on test in <6:0>
07	Skip MM tests
<6:0>	Test # to loop on

Notes:

**Use ^C to restore monitor.*

***The EKBD diagnostic will also write on any write enabled disk drives. You must spin down or write protect any diagnostic pack or customer software pack to prevent a disaster. If you wish to use this feature simply mount a SCRATCH pack on any drive and write enable it.*

****There is also a problem when trying to chain the EKBD diagnostic or typing ^C to exit to the monitor. It doesn't chain or exit properly because the monitor doesn't get fully restored. The following patch will correct the problem.*

EKBDD0	location 05162/2734 3000 54546/2734 3000
EKBDE0	location 05324/2734 3000 56110/2734 3000

EKBE/F

Starting Address(s):

200	Normal, run all tests
204	Test R/W bits in MMR's
210	Test PAR's/PDR's
214	Relocation and Adder test
220	MM aborts and trap logic tests
224	D space tests
230	A + W bit logic and dual mapping
234	MFP/MTP tests

Switch Register:

BIT15	Halt on error
14	Loop on test
13	Inhibit error typeouts
12	Inhibit trace trap every other pass
11	Inhibit iterations
10	Ring bell on error
09	Loop on error
08	Loop on test in <6:0>
07	Inhibit multiple error typeouts
<6:0>	Test # to loop on

Notes:

**To run with cache disabled halt the program and load a 14 into 17777746, then load the starting address into 17777707 and hit continue.*

EKBG

Starting Address(s):

200	Normal
220	Restart

Switch Register:

BIT15	Halt on error
14	Loop on test (section 1 only)
08	Enable system powerfail test 25
07	Disable section 1 tests (multi only)
06	Multiprocessor mode
05	UBE selected

Notes:

**See listing for multiprocessor mode.*

EMJA

Starting Address(s): 200

Switch Register:

BIT15	Halt on error
14	Loop on test
13	Inhibit error typeouts
12	Inhibit using MM
11	Inhibit iterations
10	Ring bell on error
09	Display error count in DR
08	Halt program unrelocated
06	Use 18 bit mapping only

Notes:

**Program must not be relocated when restarting.*

EMKA

Starting Address(s): 200

Switch Register:

BIT15	Halt on error
14	Loop on test
13	Inhibit error typeouts
12	Inhibit program relocation
11	Inhibit iterations
10	Bell on error
09	Loop on error
08	Halt program and flush DBE's
07	Detailed error report
06	Print configuration map
05	Limit max errors per bank
04	132 column terminal
<3:1>	Pattern (see listing)
00	Detect single bit errors

Notes:

**There must be no unibus devices addressed from 752100 to 752136. If there is remove them prior to running the diagnostic.*

***The lower 16K must have no DBE's.*

Special Functions:

^C	Flush DBE's and boot DK0:
^D	Enter modified ODT

^E	Exit ODT
^T	Tell me whats happening
^F	Field service mode

0	Exit field service mode
1	Read MK11 CSR
2	Load MK11 CSR
3	Examine memory
4	Modify memory
5	Select BANK, MARGIN,PATTERN
6	Type configuration map
7	Battery backup test
8	SOB-A-Long test
9	Super tight scope loop
10	Error summary
11	Refresh test
12	Set fill count
13	Enter KAMIKAZE mode
14	Exit KAMIKAZE mode
15	Turn off cache
16	Turn on cache
17	Run only multi-port tests
18	Resume single and multi tests
19	Test only selected banks
20	Resume testing all banks

Display register:

15	Relocated
<14:8>	Bank #
<7:5>	Margin
<4:1>	Pattern
0	Not used

Margin:	MJ11	MK11
0	Normal	Same
1	Force addrs par	Same
2	Early strobe	Early MDR load
3	Late strobe	Late refresh
4	Low current	Reserved
5	High current	Reserved

EQKC

Starting Address(s):

200	Normal
210	Display register test (see listing)
214	Micro-break reg test (see listing)
230	CPU ID register jumper cutting aid

Switch Register:

BIT15	Halt on error
14	Loop on test
13	Inhibit error typeouts
12	Inhibit UBE
11	Inhibit iterations
10	Ring bell on error
09	Loop on error
08	Relocate on disk

07	Inhibit system size typeout
06	Inhibit relocation
05	Inhibit round robin
04	Inhibit random disk address
03	Inhibit MBT
<2:0>	Specific device code

Device code:

0	RP11/RP03
1	RK05
2	Not used
3	Not used
4	RH70/RP04
5	RH70/RS03/RS04
6	Not used
7	Not used

Notes:

**The diagnostic will use the UBE, MBT and FP11-B or FP11-C if installed. See listing for details.*

***To use with RM03/5's set SWR bit 4 (inhibit random disk relocation) and select it as an RP04.*

EFPA/B

Starting Address(s): 200

Switch Register:

BIT15	Halt on error
14	Loop on test
13	Inhibit error typeouts
12	Skip MM tests
11	Inhibit iterations
10	Bell on error
09	Loop on error
08 = 1	Loop on test in <7:0>
08 = 0	Load uBreak register from <7:0>
<7:0>	Test #/ uBreak register value

Notes:

**See listing for more details.*

***If you have FP11-B use the CFP??? series of diagnostics.*

ERHA

Starting Address(s):

200	Standard RH70, PGM drives inhibited
210	Non-standard, PGM drives not inhibited
220	Standard RH70, no inhibits

Switch Register:

BIT15	Halt on error
14	Loop on test
13	Inhibit error typeouts
12	Not used
11	Inhibit iterations
10	Bell on error
09	Loop on error
08	Loop on test in <7:0>
<7:0>	Test # to loop on

Notes:

**Must have a tape loaded at BOT and/or a scratch pack mounted with heads at cylinder zero. Be careful of programmable drives!*

DIAGNOSTIC PATCHES AND SPECIAL MODIFICATIONS

EKBA** CPU Part 1

EKBAD0 No Patches

EKBB** CPU Part 2

EKBBF0 No Patches

EKBC** CACHE Part 1

EKBDC1 Memory size problem with 1920K or more of memory

Location	From	To
31420	23712	22712
31422	172356	170000

EKBD** CACHE Part 2

EKBDD0/E0 See page 5 for Monitor reload problem patch

EKBE** Memory Management Test

EKBEE1 Memory size problem with 1920K or more of memory

Location	From	To
30756	23712	22712
30760	172356	170000

EKBF** Unibus Map Test

EKBFD0/D1 Fails test 34, parity errors from Group 0 of CACHE

Rev	Location	From	To
D0	17722	40010	40100
D1	17734	140000	140010

EMKA** MK11/MJ11 Memory Diagnostic

EMKAA0/B0 You can disable Cache in EMKA with the following patch instead of using Field Service command 15.

Rev	Location	From	To
A0	2132	1	15
B0	2700	1	15

Note 1. Cache is not disabled until initialization is completed

Note 2. Halt EMKA by setting SWR bit 8 to a 1 (swr =400). This cleans up memory by clearing any errors the diagnostic may have forced into memory data locations, or the CSR's. It will also reenable ECC providing that ECC is not turned off by the switch on the memory control panel.

Note 3. Typing a ˆC cleans up memory and tries to boot RK05 drive #0. If unsuccessful, drive #1. After ˆC is typed and system halts, normal Memory System Error Register ($44) will contain 4200. The Cache Control Register ($46) will contain any of the following;14,15,1 .. All are legal diagnostic residue.

EMKAA0/B0 Peripheral Address Problem

When EMKAA0 is run on a 1170 with a peripheral at address 166000, the diagnostic will hang in a loop after printing the diagnostic header "CEMKAA0 1170 MAIN MEMORY DIAGNOSTIC" due to the address of IIST (Interprocessor Interrupt and Sanity Timer at address 166000). EMKAB0 was patched to solve this problem. The solution was to change IIST address to 777500. Therefore, with EMKAB0, if there is a device at address 177500, the same problem symptom as EMKAA0 will be evident. If you run into this problem, You can modify those two (2) locations to equal a non-existant I/O page address.

Rev	Location	From	To
A0	2162	166000	Non-existant I/O address
	2164	166002	Non-existant I/O address
B0	2736	177500	Non-existant I/O address
	2740	177502	Non-existant I/O address

EQKC** 1170 Instruction Exerciser

There is a common problem with all revisions of EQKC, the 1170 CPU Exerciser. This diagnostic supports the Massbus tester used in production. The addresses specified for the MBT in EQKC are 160100 thru 160176. Since these addresses are commonly used for communications gear (DZ11's), this may create a problem. The problem seems to be that if EQKC is halted in a disorderly manner, it may think that the system has a Massbus tester attached. If a device responds to address of 160100 a trap thru loc. 4 will result and the program must be reloaded. This has caused some techs to go on a wild goose chase. It is recommended that EQKC be halted by typing a ^ C on the console terminal.

EQKCB0/C0/D0/E0 Non-Standard I/O Addresses

If you want to allow EQKC to relocate to a device at a non-standard address you will need to patch several locations.

B0/C0	D0	E0	Contents
2142	2202	2252	176700
2144	2204	2254	176702
2146	2206	2256	176704
2150	2210	2260	176750
2152	2212	2262	176706
2154	2214	2264	176710
2156	2216	2266	176752
2160	2220	2270	176712
2162	2222	2272	176714
2164	2224	2274	176734
2166	2226	2276	176740
2170	2230	2300	176742
2172	2232	2302	176736
2174	2234	2304	176732
2176	2236	2306	000254
2200	2240	2310	000256

Note: Change above locations to reflect the new addresses and vector and note that they are not in order.

EQKCE1 Memory size problem with 1920K or more of memory

Location	From	To
62102	23712	22712
62104	172356	170000

ERHA** RH70 Diagnostic

ERHAD0/E0 In order to test unit numbers other than 0 with ERHA, the diagnostic must be patched as follows:

Rev	Location	From	To
D0	5376	0	Unit #
	5364	0	Slave # (for Tapes)
E0	7040	0	Unit #
	7022	0	Slave # (for Tapes)

ERHAE1 Fails T26 when going to TE16

Location	From	To
24046	177772	177766
24146	2300	2200

DECX11 ECC Enable Patch

DECX will disable ECC on an 1144 as it does on an 1170.

The following is a list of patches for the 1144:

	XMONB	XMONC	XMOND	Was	Should be
Loc	45120	45362	46122	52731	12731
Loc	45122	45364	46124	3	1
Loc	45134	45376	46136	42731	12731

The following is a list of patches for the 1170:

	XMONB	XMONC	XMOND	Was	Should be
Loc	32214	32436	33024	52731	12731
Loc	32216	32440	33026	3	1
Loc	3222	32444	33032	42731	12731

Keep in mind that these patches will not reset ECC if it is disabled but it will simply prevent DECX from disabling it. So if you load and run DECX before putting in the patch you will have to manually reset ECC by the CSR's then reload DECX and patch it.

DECX11 ECC/Parity Messages Modification

When running DECX11 on an 1144/1170 system with MS11/MK11 memory, we all know that paches are needed to correct DECX. Here is the way to leave DECX11 alone but cause it to print the full message as the authors intended:

Monitor	Location	From	To
XMONB0	44706	42400	42401
XMONC0	45150	42400	42401
CMOND0	45704	42400	42401

**** FOR THE 'E' MONITOR ONLY ****

DECX11 Memory Margins Patch

See Page 140 for procedure on using DECX with MJ11/MK11 memory margins.

DECX11 DECX11 as a Memory Test

This is a proven method to use DECX11 as an excellent memory exerciser when no other diagnostic seems to fail.

Preferred Method:

1. Run DXCL and build a DECX11 exerciser to include only the modules CPAG and CPBJ using monitor "E". Link it with a name of "FASTX.BIN" for ease of future reference. Now go ahead and run it. At the CMD< prompt, do the following:

   ```
   CMD<Mod CPAG0 36
   CMD<065744/7500
   CMD<Mod CPBJ0 36
   CMD<067322/12500
   CMD<
   ```

2. Copy down the physical address and the contents it returns. Now reboot and run UPD2. Under UPD2, load FASTX.BIN and modify the two (2) locations you wrote down above to contain 10 rather than what was in them. These are the iteration counts of the modules. Also modify location 1062 to contain 1000. Dump the exerciser back to disk using a different name (FAST11.BIN). Exit UPD2 and run FAST11.

3. You now have an exerciser which will relocate itself very quickly thru memory and compare itself to the previous copy (checksum). Normally when it fails, it halts and you need to look at $40-$46 to capture the error information.

Second Method:

1. Run an existing DECX11 and deselect all but the CPAG and CPBJ modules, change relative locations 36 of each to 10 and set the software SWR to 1000 using the "SWR 1000" command. Run it.

2. You will relocate the same but you are moving more code this way.

CHAPTER 3

BOOTSTRAPS

BOOTSTRAPS

M9301-YC

This bootstrap is used only for the PDP1170. It runs CPU, CACHE and main memory diagnostics if you use the normal start address of 17765000 and set the SWR. To bypass these diagnostics you can load 17773000 into location 17777707, set the SWR and hit continue. The SWR must be setup as follows:

SWR <02:00>	UNIT #
SWR <07:03>	DEVICE CODE
SWR <15:12>	32K MEMORY BANK

The following jumpers and switch settings apply for the M9301-YC on the PDP1170:

JUMPERS	**SETTING**
W1-6	IN

SWITCHES		**NORMAL SETTING**
S1-1		ON for low ROM enable
S1-2		On for powerup boot
S1-3:7	(etch F)	Device select
S1-10:7,5	(etch E)	Device select
S1-8:10	(etch F)	Unit #
S1-6,2,1	(etch E)	Unit #

		ETCH E	**S1-10**	**S1-9**	**S1-8**	**S1-7**	**S1-5**
DEVICE SELECT	**CODE**	**ETCH F**	**S1-3**	**S1-4**	**S1-5**	**S1-6**	**S1-7**
TM11/TU10	01		ON	ON	ON	ON	OFF
TC11/TU56	02		ON	ON	ON	OFF	ON
RK11/RK05	03		ON	ON	ON	OFF	OFF
RP11/RP03	04		ON	ON	OFF	ON	ON
RESERVED	05		—	—	—	—	—
RH70/TU16	06		ON	ON	OFF	OFF	ON
RH70/RP04	07		ON	ON	OFF	OFF	OFF
RX11/RX01	11		ON	OFF	ON	ON	OFF

	ETCH E	**S1-3**	**S1-4**	**S1-6**
UNIT #	**ETCH F**	**S1-10**	**S1-9**	**S1-8**
0		ON	ON	ON
1		ON	ON	OFF
2		ON	OFF	ON
3		ON	OFF	OFF
4		OFF	ON	ON
5		OFF	ON	OFF
6		OFF	OFF	ON
7		OFF	OFF	OFF

If an error is detected during the diagnostics the CPU halts and the halt address indicates what the failure was (see table 1).

If boot halts at 17773654
17773746
17773764

Press continue to force misses in CACHE.

To boot with cache off first deposit a 14 in 17777746, then deposit 173000 into 17777707, setup your SWR and hit continue.

M9301-YH

This bootstrap is used for both the PDP1160 and the PDP1170. It runs the same tests as the M9301-YC except for checking the kernal PDR's and PAR's. It also doesn't turn on memory management and the unibus map. To start the bootstrap you load address 17773000 and set the SWR. Diagnostics are run or bypassed depending on your switch settings. The SWR must be set in the following format:

SWR <02:00>	UNIT #
SWR <06:03>	DEVICE CODE

The following jumpers and switch settings apply for the M9301-YH on the PDP1170:

JUMPERS	**SETTING**
W1-6	IN

SWITCHES	**NORMAL SETTING**
S1-1	ON for low ROM enable
S1-2	ON for powerup boot
S1-3	Diagnostic select
S1-4:7	Device select
S1-8	OFF for device code from SWR, ON use S1-4:7
S1-9	Diagnostic select
S1-10	OFF for PDP1170

DIAGNOSTIC SELECT	**S1-9**	**S1-3**
No diagnostics	OFF	OFF
No diagnostics	OFF	ON
Run tests 1-24	ON	OFF
Run tests 1-20	ON	ON

DEVICE SELECT	**CODE**	**S1-4**	**S1-5**	**S1-6**	**S1-7**
TM11/TU10	01	ON	ON	ON	OFF
TC11/TU56	02	ON	ON	OFF	ON
RK11/RK05	03	ON	ON	OFF	OFF
RP11/RP03	04	ON	OFF	ON	ON
RK611/RK06	05	ON	OFF	ON	OFF
RH70/TU16	06	ON	OFF	OFF	ON
RH70/RP04	07	ON	OFF	OFF	OFF
RX11/RX01	11	OFF	ON	ON	OFF
PC11	12	OFF	ON	OFF	ON

If an error is detected during the diagnostics the CPU halts and the halt address indicates what the failure was (see table 1).

If boot halts at 17765614
17765732
17765752

Press continue to force misses in CACHE.

To boot with cache off first set switches S1-9 and S1-3 to the OFF position. Then deposit a 14 into 17777746, deposit a 173000 into 17777707, setup your SWR and hit continue.

M9312

This is a multipurpose PDP11 bootstrap. To use it on a PDP1170 you must have the PDP1160/1170 diagnostic ROM (part # 23-233F1) installed in E20 and the appropriate boot ROMs installed for the devices you have. To boot the PDP1170 you load address 17765744 and set the SWR as follows:

SWR = 00UXMM

Where:
U = Device unit # 0 - 7
X = 0 for ROM 1
2 for ROM 2
4 for ROM 3
6 for ROM 4
MM = 42 for low speed reader and TU55/56
56 for RP04/5/6 and RM02/3/5
12 for all other devices

All alternate boot procedure is to start at the device ROMs starting address as follows:

STARTING ADDRESS = 17773XNY

Where:
X = 0 for ROM 1
2 for ROM 2
4 for ROM 3
6 for ROM 4
N = 5 for RP04/5/6 and RM02/3/5
3 for low speed reader and TU55/56
0 for all other devices
Y = 4 for no diagnostics if N = 0 or 3
6 for diagnostics if N = 0 or 3
0 for no diagnostics if N = 5
2 for diagnostics if N = 5

The following jumpers and switch settings apply for the M9312 on the PDP1170:

JUMPERS	**SETTING**
W1-7	IN
W8	OUT
W9,10	IN
W11,12	OUT

SWITCHES	**NORMAL SETTING**
S1-1	OFF
S1-2	ON for powerup boot
S1-3,4	ROM socket select
S1-5:9	Device select
S1-10	ON for diagnostics

ROM SOCKET	**S1-3**	**S1-4**
1 (E35)	OFF	OFF
2 (E33)	OFF	ON
3 (E34)	ON	OFF
4 (E32)	ON	ON

PART #	DEVICE SELECT	S1-5	S1-6	S1-7	S1-8	S1-9
23-751A9	RL01/2	OFF	OFF	OFF	OFF	ON
23-752A9	RK06/7	OFF	OFF	OFF	OFF	ON
23-753A9	RX01	OFF	OFF	OFF	OFF	ON
*23-811A9	RX02	OFF	OFF	OFF	OFF	ON
23-755A9	RP02/3	OFF	OFF	OFF	OFF	ON
	RP04/5/6,RM02/3	OFF	ON	OFF	ON	OFF
23-756A9	RK03/5 (unit 0)	OFF	OFF	OFF	OFF	ON
	RK03/5 (unit 2)	ON	ON	ON	OFF	ON
	TU55/56	OFF	OFF	ON	ON	ON
23-757A9	TU16/TE16/TU45/TU77	OFF	OFF	OFF	OFF	ON
23-758A9	TU10/TE10/TS03	OFF	OFF	OFF	OFF	ON
23-759A9	RS03/4	OFF	OFF	OFF	OFF	ON
23-760A9	PC05	OFF	OFF	OFF	OFF	ON
	DL11A/W	OFF	OFF	ON	ON	ON
23-761A9	TU60	OFF	OFF	OFF	OFF	ON
23-762A9	RS11	OFF	OFF	OFF	OFF	ON
	RS64	OFF	ON	OFF	ON	OFF
23-763A9	CR11	OFF	OFF	OFF	OFF	ON
23-764A9	TS11/TU80/TS05	OFF	OFF	OFF	OFF	ON
23-765A9	TU58	OFF	OFF	OFF	OFF	ON
23-767A9	RA60/80/81/RC25	OFF	OFF	OFF	OFF	ON
23-E38A9	TK50/TU81	OFF	OFF	OFF	OFF	ON

If an error is detected during the diagnostics the CPU halts and the halt address indicates what the failure was (see table 1).

If boot halts at 17765554
17765676
17765716

Press continue to force misses in CACHE.

To boot with cache off first deposit a 14 into 17777746, then deposit a 173XN4 or 173XN0 if N = 5 (X and N refer to your boot device, see section above on the M9312) into 17777707 and hit continue. If the diagnostics detect an error the CPU halts and the halt address indicates the failure (see table 1).

**RX02 boot ROMS with date codes 7921 thru 7931 are bad*
***1104/34 console boot ROM part #* is 23-248F1
****The 1160, 1170 boot ROM 23-233F1 is replaced by 23-616F1*

TABLE 1
HALT ADDRESS DISPLAYED

M9301	M9301	M9312*	
YC	YH		INSTRUCTIONS TESTED
165004	165004	165052	BR
165020	165020	165070	CLR, BMI, BVS, BHI, BLOS, (BLT)
165036	165036	165104	DEC, BPL, BEQ, BGE, [BGT], BLE
165052	165052	165116	ROR, BVC, BHIS, [BHI], BNE
165066	165066	------	SEZ, BHI, SEN, BLT, BLOS
165076	165076	------	CLZ, BLE, BGT
165134	165126	------	MOV, REG DATA PATH, SUB, BLT, BEQ
165146	165136	------	ROL, BCC, BLT
165166	165154	165174	ADD, INC, COM, BSC, BLE, (BLT)
165204	165172	165214	ROR, BIS, ADD, INC, BLO, [BGE], (DEC)
165214	165202	------	DEC, BLOS, BLT
165222	165210	165220	COM, BLOS
165236	165224	165234	BIC, ADD, BGT, [BGE], BLE
165260	165246	165256	[ADC], (SWAB),CMP,BNE,BIT,BGT,COM, [SUB],[BEQ]
165270	165256	165266	MOVB, BPL
165312	165300	165310	SOB, CLR, INC, TST, BNE, BEQ
165346	165334	------	ASR, ASL
165360	------	------	ASH
165374	165352	------	ASH, SWAB
165450	------	------	KERNAL PARS
165474	------	------	KERNAL PDRS
165510	165376	165322	JSR
165520	165406	165332	WRONG VALUE PUSHED ON STACK
165530	165416	165342	RTS
165542	165430	165354	RTI
165550	165436	165362	JMP
NO HLT	------	------	LOAD AND TURN ON MM AND UBM
165742	165520	165460	MAIN MEMORY 1000-28K, CACHE OFF
165760	165540	165500	DATA NOT COMPLEMENT, CACHE OFF
166000	------	------	PARITY ERROR HALT
173644	165604	165544	CACHE DATA ERROR
173654	165614	165554**	NO CACHE HIT
173736	165720	165664	DATA COMPAIR, CACHE ON
173746	165732	165676**	NO CACHE HIT, R0 = ADDRESS+2
173764	165752	165716**	CACHE ERROR/PARITY ERROR

Note:

(????) = M9312 ONLY
[????] = M9301 ONLY

**For M9312 boot ROM part # 23-233F1*
***To boot with cache off, hit continue*

CHAPTER 4

TOGGLE-INS

TOGGLE-INS

There are several useful toggle-ins that are probably not very well known. They are as follows:

Memory Management Checkout

Use the following toggle-in to verify the correct operation of Memory Management Relocation.

```
        200/012737    MOV #400,@#177572 (load maint. bit in MMR0)
        202/000400
        204/177572
        206/012737    MOV #070707,@#200 (move 070707 to virtual 200)
        210/070707
        212/000200
        214/000000    HLT

        300/000300    Preset loc 300 to 300

17772300/077406       Set Kernal I PDR 0 to R/W 4K page
17772340/000001       Set Kernal I PAR 0 to (Base address 100)

Load Address 200
Start                 Display = 000216 (Halt @214)
Load Address 300
Exam                  Display = 070707 . . . Relocation works
```

Unibus Map Checkout

Use the following console operating procedure to verify the correct operation of the Unibus Map.

Load address	500	
Deposit	125252	Known data
Load address	17000500	
Examine	125252	Data path ok
Load address	17000700	
Deposit	070707	Known data
Load address	17770202	Map register 0 Hi
Deposit	000000	Relocation constant
Load address	17770200	Map register 0 Lo
Deposit	000200	Relocation constant
Load address	17772516	MMR3
Deposit	000040	Enable map
Load address	17000500	Relocates to 700
Examine	070707	. . . Relocated ok

Cache Checkout

Perform the following check-out procedure to determine if Cache correctly detects Hits and Misses.

Power-up		
Load address	17777752	Hit/Miss register
Examine	000000	Power-up cleared register ok
Load address	000200	Examine locations 200 thru 212
Examine . . .	********	Contents don't matter
Load address	17777752	Hit/Miss register
Examine	000025	50% hit rate

Load address	000200	Examine locations 200 thru 212
Examine . . .	********	Contents don't matter
Load address	17777752	Hit/Miss register
Examine	000077	100% hit rate
Load address	17777746	Cache control register
Deposit	14	Disable cache
Load address	000200	Examine locations 200 thru 212
Examine	********	Contents don't matter
Load address	17777752	Hit/Miss register
Examine	000000	No hits in Cache

Memory Clear Program

Use this toggle-in to clear all of memory from location 0 to system size.

LOCATION	CONTENTS		INSTRUCTIONS & NOTES
17772300	077406		KIPDR 0
17772316	077406		KIPDR 7 = I/O page for program
17772340	000000		KIPAR 0; Load 200 if using trap catchers.
17772356	177600		KIPAR 7 = I/O page for program
17777700	000000		R0 = Start Virtual Address
17777701	172340		R1 = KIPAR 0 Address
17777702	177572		R2 = MMR0 Address
17777703	177760		R3 = System size register Address
17777704	172516		R4 = MMR3 Address
17777705	Pattern		R5 = Desired pattern or "0"
17777706	177676		R6 = Stack pointer
17777776	000000		PSW = 0
17772240	012714		MOV #20, (R4); Enable 22-bit Mapping
17772242	000020		
17772244	005212		INC (R2); Enable Memory Management
17772246	010520	1$:	MOV R5, (R0)+; Data to memory
17772250	020027		CMP R0, #17776; Top of Page 0?
17772252	017776		
17772254	003774		BLE 1$
17772256	062711		ADD #200, (R1); Step Page
17772260	000200		
17772262	021311		CMP (R3), (R1); Top of Memory?
17772264	003402		BLE 2$
17772266	005000		CLR R0; Start at Virtual Address 0
17772270	000776		BR 1$
17772272	005312	2$	DEC (R2); Disable Relocation
17772274	000000		HLT

Load address 172240
Start

The program should halt when all of memory is cleared. If it doesn't halt examine the CPU error, Memory system error and HI/LO error address registers to determine the cause. Trap catchers can be used by depositing 200 in KIPAR 0 (17772340) instead of zero, and setting up vector locations with their address + 2 and the Vector + 2 with a 0 (halt). Note: When loading the program you must be in console physical.

If you just want to clear bad parity in 0 to 28K memory you can do so by depositing 14747 into 157776 and then loading address 157776 and start.

CHAPTER 5

TECH-TIPS/FCR SECTION

CONTENT **PAGE**

PDP1170 FIELD CHANGE ORDERS INDEX

FCO #	FICHE	DESCRIPTION
PDP1170 — R1	Green	Replace 7009540 with 7011051 power harness.
PDP1170 — S12	Green	+-15v not available for RDC panel, modify 7011051 power harness.
5411294 — S2	Blue	CPU halts when address switch turned, update revision of ETCH B to CS C.
7010329 — R2	Blue	Poor power distribution, this FCO included in PDP1170 — R1.
7010329 — H5	Blue	Changes for KB11-B to KB11-C, update revision to WL D, WT H.
7010329 — R6	Blue	Intermittant massbus errors, update revision to WL E, WT J.
7010329 — R10	Blue	Changes for DL11-W use, update revision to WL L, WT R.
G235 — S9	Blue	Memory is marginal due to high current drive, update revision of ETCH D to CS N.
M8132 — R2	Blue	For FP11-C use, update revision of ETCH A to CS B.
M8136 — S2	Blue	Unibus devices vector to wrong location on loaded systems, update revision of ETCH D to CS C.
M8136 — S4	Blue	Push button boot and downline load feature needed, update revision of ETCH B,C to CS C1,E respectively.
M8136 — S5	Blue	M9301 will not boot on MJ11 power fail, update revision of ETCH B,C to CS C1,F respectively.
M8138 — R6	Blue	Memory management aborts may go undetected, update revision of ETCH A,B to CS C2,F respectively.
M8139 — R3	Blue	Cache and main memory data errors, update revision of ETCH B to CS D.
M8140 — R2	Blue	Incorrect data from PAR's, PDR's and memory management registers, update revision of ETCH A to CS B.
M8142 — S3	Blue	Main memory timeouts, update revision of ETCH B to CS C.
M8142 — S4	Blue	Main memory parity errors when MAP transfers timeout and Cache address errors due to incorrect setup of E42, update revision of ETCH B to CS D.
M8148 — S3	Blue	Main memory parity errors, update revision of ETCH A to CS C.

M8148	— S4	Blue	Poor main memory bus termination, update revision of ETCH A to CS A1.
M8151	— S1	Blue	MXF errors occuring with WCE errors, update revision of ETCH A to CS A.
M8160	— R4	Blue	Circuit breaker on H775D trips during power fail, update revision of ETCH C,D to CS F.
M9301	— R1	Blue	M9301-YC diagnostic fails after power up, update revision of ETCH E to CS E.

PDP1170 TECHTIP INDEX

TT #	FICHE	DESCRIPTION
1	Green	1170 system serial #.
2	Green	1170 power supplies.
3	Green	M8143 wire damage.
4	Green	1170 bootstraps.
5	Green	KW11-L noise problem, cross to KW11-TT-10.
6	Green	Timeout problems with Fairchild 74123's.
7	Green	Array boards in MOS ECC memory, cross to MF11-TT-10.
8	Green	Disabling Cache on the 1170.
9	Green	PRS01 useage on KY11-R equipped 1170's, cross to KY11R-TT-3.
10	Green	1170 backplane replacement.
11	Green	BC06R cable testing, cross to BC06R-TT-1.
12	Green	Tripping circuit breakers, cross to H775-TT-4.
13	Green	Mixed MOS/CORE on 1170's.
14	Green	RL11 and RM03 configured on an 1170, cross to RL11-TT-4.
15	Green	MK11 initialization cable, cross to MK11-TT-4.
16	Green	Cabling for MK11 boxes, cross to MK11-TT-5.
17	Green	RH11's on 1170's, cross to RH11-TT-7.
18	Green	PDP1170/PDP11 software problem.
19	Green	Guidelines for optimized system cooling, cross to Cabinets-TT-10.
20	Green	DMC11 basic W/R and Microprocessor diagnostic failures, cross to DMC11-TT-11.
21	Green	MK11 upgrade, cross to MK11-TT-7.
22	Green	MK11 add-on installation, cross to MK11-TT-8.
23	Green	MJ11 troubleshooting flowchart, cross to MJ11-TT-12.
24	Green	1170 troubleshooting, RH70/Cache jumpers.
25	Green	Massbus ribbon cable orientation.
26	Green	ID resister setup.
27	SB274	MK11 voltage regulator 70-14251.
28	SB288	1170 power harness in corporate cabinets.

29	SB383	Troubleshooting 1170 MK11 write parity errors.
30	SB384	PDP1170 Bootstrap halt addresses.
31	SB441	1170 Pause Hang Troubleshooting
32	SB476	MK11 CSR's verses 1170 $40, $42, cross to MK11-TT-14
33	SB476	1170 Maintenance Service Guide
34	SB476	1170 Memory System Error Register

MK11 TECHTIP INDEX

TT #	FICHE	DESCRIPTION
1	Green	Array boards in MOS ECC memory, cross to MF11-TT-10.
2	Green	Troubleshooting MK11 systems.
3	Green	Mixed MOS over CORE, cross to 1170-TT-13.
4	Green	MK11 ECC initialization cycle.
5	Green	Cabling MK11 boxes.
6	Green	Tripping circuit breakers during battery backup, cross to H775-TT-4.
7	Green	MK11 upgrade.
8	Green	Bottom cover shorting to backplane.
9	Green	Enable boot on power-up cable.
10	Green	MK11 fatal SBE's and undetected DBE's
11	SB274	MK11 memory voltage regulator, 70-14251.
12	SB292	Adjustment of memory voltage regulators.
13	SB383	Troubleshooting 1170 MK11 write parity errors, cross to 1170-TT-29
14	SB476	MK11 CSR's verses 1170 $40, $42, cross to 1170-TT-32

MJ11 TECHTIP INDEX

TT #	FICHE	DESCRIPTION
1	Green	MJ11B configuration.
2	Green	MM11-U/UP memorys failing high current margins, cross to MM11-TT-15.
3	Green	Intermittant data parity errors.
4	Green	Memory bus cabling.
5	Green	Core memory upgrade.
6	Green	Core parity errors.
7	Green	BC06R cable testing, cross to BC06R-TT-1.
8	Green	Mixed MOS over CORE on 1170's, cross to 1170-TT-13.
9	Green	MJ11 memory maintenance philosophy.
10	Green	MJ11 cabling procedure.
11	Green	Memory cables.

12	Green	MJ11 memory troubleshooting flowchart.
13	Green	H224 cables on MJ11B.
14	Green	Bottom cover shorting to backplane, cross to MK11-TT-14.
15	Green	MJ11 strobe timing adjustment.
16	Green	Backplane contact interference from foam and tags.

SHOULD'VE BEEN FCO'S

M8136

Problem: SACK timeouts, TRAP 0's and Unibus timeouts due to improper clearing of FAIRCHILD, AMD and SIGNETICS 74123's in timeout logic.

Solution: Replace E15 and E25 with TEXAS INSTRUMENTS 74123's (DEC # 19-10436) if one of the above vendors chips present.

G116

Problem: NATIONAL SEMICONDUCTOR NS7520 sense amps occasionally generate transient spikes that will effect only other stacks on the same side (ODD/EVEN) of the box. They do not effect the stack they are in. This results in transient data parity errors in other stacks.

Solution: Replace G116's that have NS7520's if problem symptom evident. There are 20 sense amps along the bottom row towards the fingers.

H224

Problem: Over the top cables get destroyed from adjacent module insertion, removal and unreliable crimp connections result after several removals of the cables.

Solution: Replace any H224's or tape and recrimp damaged cables which have the above problems.

M8149

Problem: Mixed mos over core memory configurations require 3 etch cuts to the M8149 to prevent the unused data bit (byte 1, bit 9) from rebooting the system through the MK11 boot enable. This also prevents noise problems from the unused data bit (byte 3, bit 9) and unused data-cable BOCC.

Solution: Reference PDP1170-TT-13.

M7984

Problem: Green bypass capacitors burn up causing intermittant data parity errors.

Solution: Replace any M7984's with burnt bypass capacitors. This is a good area to check if you are having intermittant data parity error problems.

CHAPTER 6

ADJUSTMENTS

CONTENT **PAGE**

ADJUSTMENTS

11/70 CPU VOLTAGES, TOLERANCES, AND TEST POINTS

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

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

	UPPER POWER SUPPLY h7420				H7420 BULK SUPPLY
REGULATOR E	REGULATOR D	REGULATOR C	REGULATOR B	REGULATOR A	
NOT USED	+5V RH70 & SPC	+5V CENTRAL PROCESSOR & MEMORY MANAGEMENT	+5V CENTRAL PROCESSOR	+5V FLOATING POINT	+15V TO CENTRAL PROCESSOR CACHE RH70 & SPC
	+5V TO ROWS 36,37,38,39,40 41,42,43,44	+5V TO ROWS 10,11,12,13,14,15	+5V TO ROWS 1,6,7,8,9	+5V TO ROWS 2,3,4,5	+ BV TO MAINT SLOT AC LOW DC LOW 50/60 HZ SIGNAL TO CLOCK

	LOWER POWER SUPPLY h7420				H7420 BULK SUPPLY
REGULATOR L	REGULATOR K	REGULATOR J	REGULATOR H	REGULATOR F	
+5V RH70	+5V RH70	+5V CACHE CONSOLE	+5V CACHE	NOT USED	+15V TO CENTRAL PROCESSOR CACHE RH70 & SPC
+5V TO ROWS 29,30,31,32 33,34,35	+5V TO ROWS 24,25,26,27,28	+5V TO ROWS 16,17,18	+5V TO ROWS 20,21,22		AC LOW DC LOW

MK11 VOLTAGE ADJUSTMENTS, TOLERANCES AND TEST POINTS

OUTPUT	SLOTS	TEST POINT	V.D.C.	MAX RIPPLE
H7441 +5V	10-17	J21-7	4.9 → 5.3V	200MV pp
REGULATOR A				
70-14251 +5VB		J18-4	4.9 → 5.3V	200MV pp
+12VB	6-13	J18-5	11 → 13V	250MV pp
-12VB		J18-7	-10.5 → -13.5V	250MV pp
REGULATOR B				
70-14251 +5VB	15-21	J21-5	4.9 → 5.3V	200MV pp
+12VB		J21-4	11 → 13V	250MV pp
-12VB		J21-2	-10.5 → -13.5V	250MV pp
REGULATOR C				
70-14251 +5VB	22-25	J21-6	4.9 → 5.3V	200MV pp
+12VB	2-5	J21-3	11 → 13V	250MV pp
-12VB		J21-1	-10.5 → -13.5V	250MV pp

LAYOUT

REG C	REG B	AC INPUT BOX	#7441	REG A

MJ11 VOLTAGE ADJUSTMENTS, TOLERANCES AND TEST POINTS

		M8149 (TP7 GND)		
OUTPUT	SLOTS	TEST POINT	VDC	MAX RIPPLE
REGULATOR #1				
H754 +20V		TP4	+20 ± 1.0V	1.0 V pp
-5V		TP6	-5 ± .25V	250MV pp
REGULATOR #2		TP2	+5 ± .25V	200MV pp
H744 +5V				
REGULATOR #3				
H754 +20V		TP3	+20 ± 1.0V	1.0V pp
-5V		TP5	-5 ± .25V	250MV pp
REGULATOR #4				
H744 +5V		TP1	+5 ± .25V	200MV pp

LAYOUT

REG #4	REG #3	AC INPUT BOX	REG #2	REG #1

11/70 TIG BOARD (M8139) ADJUSTMENTS

Procedure for adjusting 11/70 TIG Boards.

1. Install M8139/09 on Extender.

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

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

Note: *refer to figure 1 for the following symmetry adjustments. (Oscillator Symmetry.)*

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

5. Phase Splitter Symmetry

   a. Monitor signal *TIG C TF L* (backpanel pin ER2 or emitter lead of Q57 or Q58.) Adjust R129 (R210 for M8109) for a symmetrical 15 ns for each half cycle.

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

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

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

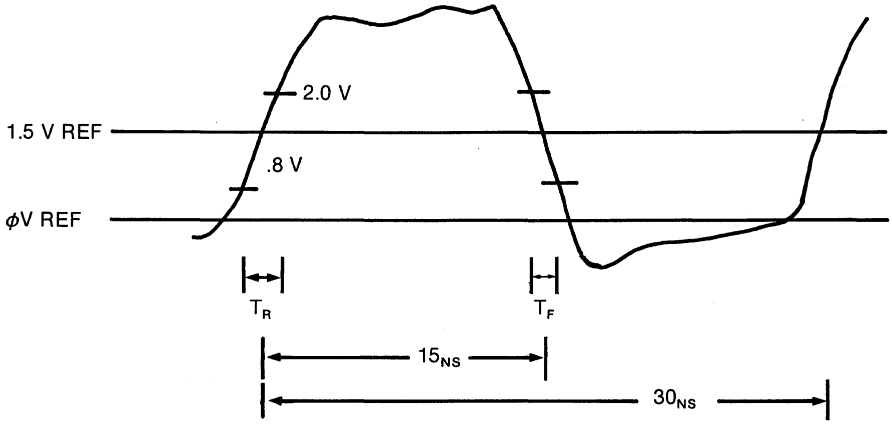
FIGURE 1 REPRESENTATIVE WAVE FORM

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

Rise time 2 NS
Fall time 2 NS

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

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

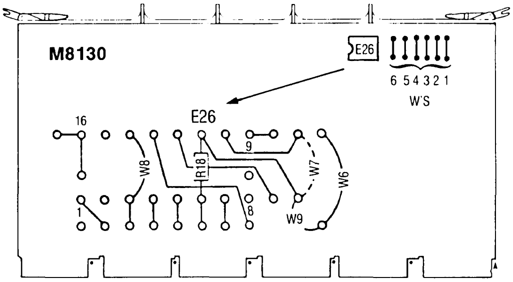


CHAPTER 7

SWITCHES, JUMPERS AND INDICATORS

SWITCHES, JUMPERS & INDICATORS

M8130 START VECTOR JUMPERS



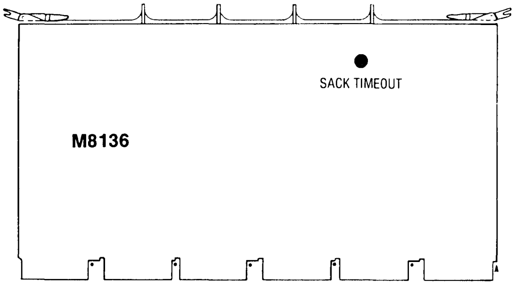
JUMPER	FUNCTION WHEN IN
W1	*START VECTOR BIT 2
W2	START VECTOR BIT 3
W3	START VECTOR BIT 4
W4	START VECTOR BIT 5
W5	START VECTOR BIT 6
W6	START VECTOR BIT 7
W7	**START VECTOR = 173000 + XXX
W8	PUSH BUTTON BOOT 173000 + XXX (W7 MUST BE OUT)
W9	PUSH BUTTON BOOT 173200 + XXX (W7 MUST BE OUT + W8 IN)

*START VECTOR BITS 0, 1 ALWAYS = 0
**XXX = OFFSET FROM W1 THRU W6.

STANDARD SETUPS: 1 = JUMPER INSTALLED

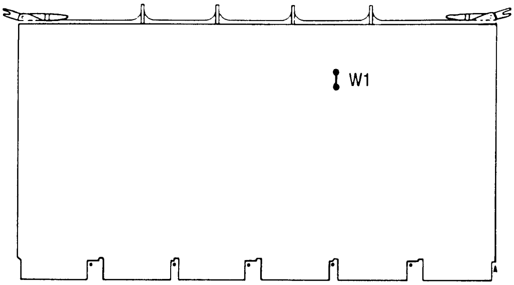
JUMPER	M9301/M9312 (NO BOOT SW)	M9301 (BOOT SW)	M9312 (BOOT SW)
W1	1	1	1
W2	0	0	0
W3	1	1	1
W4	0	0	0
W5	0	0	0
W6	0	0	0
W7	0	0	0
W8	0	1	1
W9	0	1	0

M8136 SACK TIME OUT LED



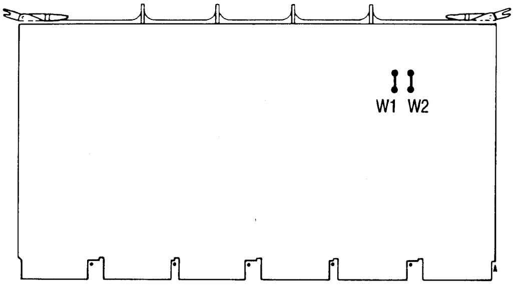
IF NO RESPONSE (SACK) IS RECEIVED FROM THE UNIBUS 10 MICROSECONDS AFTER A UNIBUS GRANT (NPG OR BG) IS ISSUED THE SACK TIMEOUT LED IS LIT. IT'S RESET BY INIT.

M8132 FLOATING POINT SELECTION JUMPER



W1 in = FP11-B
W1 out = FP11-C

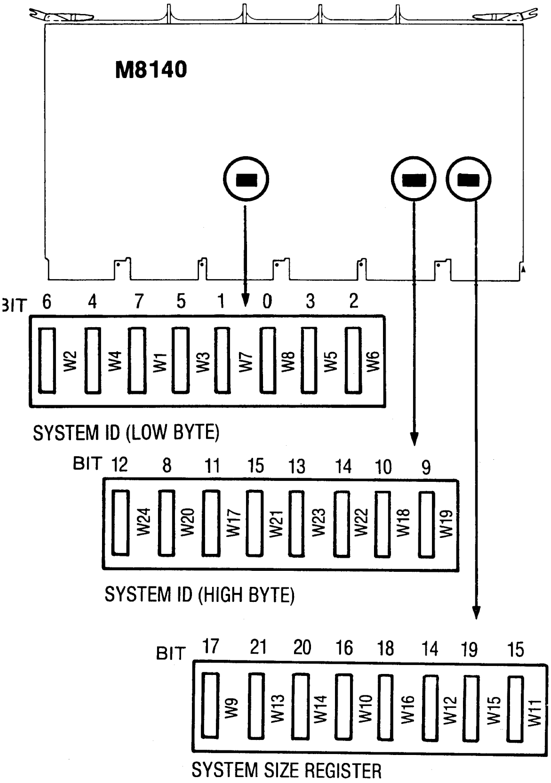
M8137 PLFE JUMPERS (PDR BIT 15)



W1 in, W2 out = Normal
*W1 out, W2 in = Reserved

*This Reserved Jumper configuration connects the extra bit in the PDR registers to bit 15. This bit is not used or supported by any operating system and should be left grounded by having W1 in and W2 out.

M8140 SYSTEM SIZE AND ID REGISTER SWITCHES


M8140
BIT 6 4 7 5 1 0 3 2
W2 W4 W1 W3 W7 W8 W5 W6
SYSTEM ID (LOW BYTE)
BIT 12 8 11 15 13 14 10 9
W24 W20 W17 W21 W23 W22 W18 W19
SYSTEM ID (HIGH BYTE)
BIT 17 21 20 16 18 14 19 15
W9 W13 W14 W10 W16 W12 W15 W11
SYSTEM SIZE REGISTER

ALL SWITCHES ON = 0 OFF = 1

SYSTEM ID REGISTER

NORMALLY NOT USED, HOWEVER IT'S CONTENTS CAN BE CHANGED BY SETTING SWITCHES IN THE HIGH/LOW BYTE SYSTEM ID SWITCH PACKS.

SYSTEM SIZE REGISTER

THIS REGISTER MUST BE SET TO REFLECT BITS 21 THRU 14 OF THE LAST PHYSICAL MEMORY ADDRESS. BITS 13 THRU 1 ARE ALWAYS 1's.

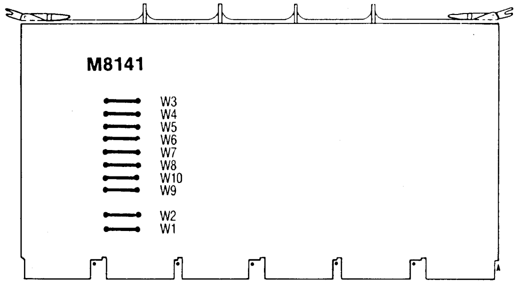
EG: 1024 K WORDS MEMORY
LAST ADDRESS = 07777776
BITS 21→14 = 0|111|111|1

SWITCH SETTING LEFT TO RIGHT IS

OFF, ON, OFF, OFF, OFF, OFF, OFF, OFF

NOTE: EMKAB0 WILL INFORM YOU IF THE SWITCHES ARE SET WRONG AND GIVE YOU THE CORRECT SETTINGS.

M8141 MAP LIMIT JUMPERS

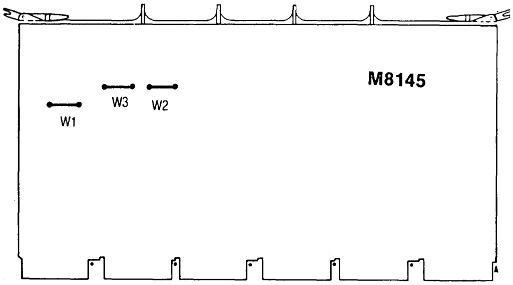


THE STANDARD SETUP FOR THE MAP JUMPERS IS W1, W3 THRU W6 OUT AND W2, W7 THRU W10 IN. THIS GIVES A MAPPING RANGE OF ADDRESSES FROM 0 TO 757776. THE JUMPERS ALLOW YOU TO CHANGE THE UPPER OR LOWER LIMIT IN 4K INCREMENTS. TO SET UP THE JUMPERS SELECT THE UPPER AND LOWER ADDRESS LIMIT YOU WANT THE MAP TO USE FROM THE RANGE OF 0 TO 760000 (0 to 124K). THEN USING BITS 17 THRU 13 SET UP THE JUMPERS USING THE FOLLOWING TABLE.

ADDRESS BIT	LOW LIMIT	HIGH LIMIT
13	W2	W1
14	W10	W6
15	W9	W5
16	W8	W4
17	W7	W3

NOTE: JUMPER REMOVED = 1
JUMPER INSTALLED = 0

M8145 MASSBUS PRIORITY JUMPERS



NOTE: TO AVOID PROBLEMS JUMPERS W1 THRU W3 SHOULD ALL BE "IN", GIVING SEQUENTIAL PRIORITY FROM RH70 "A" TO RH70 "D".

MBC SELECTION PRIORITIES (M8145)

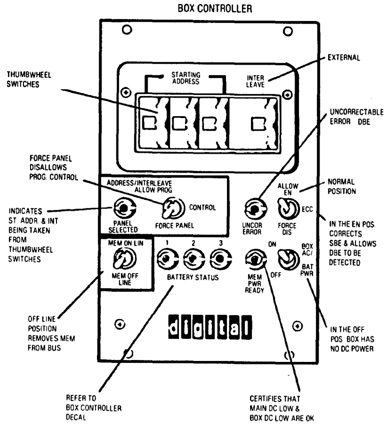
JUMPER CONFIGURATION			PRIORITY STRUCTURE*
W1	W2	W3	
OUT	OUT	OUT	(A↔B)↔(C↔D)
OUT	OUT	IN	(A→B)↔(C↔D)
OUT	IN	OUT	(A↔B)↔(C→D)
OUT	IN	IN	(A→B)↔(C→D)
IN	OUT	OUT	(A↔B)→(C↔D)
IN	OUT	IN	(A→B)→(C↔D)
IN	IN	OUT	(A↔B)→(C→D)
IN	IN	IN	(A→B)→(C→D)

*A→B indicates the expression on the left has a higher priority than that on the right (A over B).

B↔C indicates the selection alternates between the expressions on either side of the symbol.

(A→B)→(C→D) indicates group A→B is over group C→D and A is over B. A has the highest priority and D the lowest.

MK11 SWITCHES, INDICATORS AND JUMPERS


BOX CONTROLLER
THUMBWHEEL SWITCHES
STARTING ADDRESS
INTER LEAVE
EXTERNAL
UNCORRECTABLE ERROR DBE
FORCE PANEL DISALLOWS PROG. CONTROL
ADDRESS/INTERLEAVE ALLOW PROG
CONTROL
FORCE PANEL
PANEL SELECTED
ALLOW EN
ECC
UNCOR ERROR
FORCE DIS
NORMAL POSITION
INDICATES ST. ADDR. & INT BEING TAKEN FROM THUMBWHEEL SWITCHES
MEM ON LIN
MEM OFF LINE
1
2
3
BATTERY STATUS
ON
OFF
MEM PWR READY
BOX AC/ BAT PWR
IN THE EN POS. CORRECTS SBE & ALLOWS DBE TO BE DETECTED
OFF LINE POSITION REMOVES MEM FROM BUS
digital
IN THE OFF POS. BOX HAS NO DC POWER
REFER TO BOX CONTROLLER DECAL
CERTIFIES THAT MAIN DC LOW & BOX DC LOW ARE OK

MA1326

INTERLEAVE SWITCH SETTINGS ON FRONT CONTROL PANEL

		A03	A02	
	Not interleaved = 0			
	Invalid = 1			
• 2-Way:	Box 0 = 2	X	0	2-Way
	Box 1 = 3	X	I	
• 4-Way:	Box 0 = 4	0	0	4-Way
	Box 1 = 5	0	I	
	Box 2 = 6	I	0	
	Box 3 = 7	I	I	

CONTROL PANEL SETUP, INDICATORS

BATTERY STATUS	
LED'S	FUNCTION
1. SLOW FLASH	FAST CHARGE
2. ON	SLOW CHARGE
3. FAST FLASH	DISCHARGE
4. OFF	BATTERY OFF

INTERLEAVE
0 NO EXT INTERL
1 NOT USED
2 (2) WAY EXT INTERL BOX 0
3 (2) WAY EXT INTERL BOX 1
4 (4) WAY EXT INTERL BOX 0
5 (4) WAY EXT INTERL BOX 1
6 (4) WAY EXT INTERL BOX 2
7 (4) WAY EXT INTERL BOX 3

STARTING ADDRESS							
PANEL SET	K WORDS	PANEL SET	K WORDS	PANEL SET	K WORDS	PANEL SET	K WORDS
000	0	020	512	040	1024	060	1536
001	32	021	544	041	1056	061	1568
002	64	022	576	042	1088	062	1600
003	96	023	608	043	1120	063	1632
004	128	024	640	044	1152	064	1664
005	160	025	672	045	1184	065	1696
006	192	026	704	046	1216	066	1728
007	224	027	736	047	1248	067	1760
010	256	030	768	050	1280	070	1792
011	288	031	800	051	1312	071	1824
012	320	032	832	052	1344	072	1856
013	352	033	864	053	1376	073	1888
014	384	034	896	054	1408	074	1920
015	416	035	928	055	1440	075	1952
016	448	036	960	056	1472	076	1984
017	480	037	992	057	1504	077	2016

Starting Address

- Box 0 starting address = 0
- Box 1 starting address = amount of 32K memory contained in Box 0 in octal

Conditions for External Interleaving

- Number of boxes must be even.
- Boxes must have same capacity.
- Boxes must have same starting address.

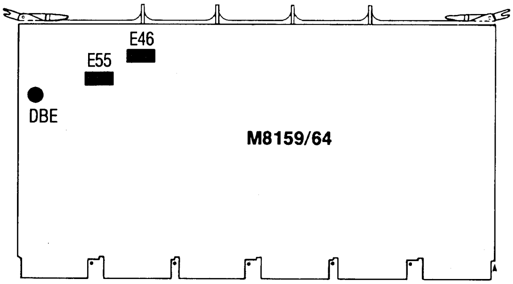
Conditions for Internal Interleaving

- Number of array cards within a box must be even.
- Each pair of array cards must be identical.

MK11 CONFIGURATION RULES

- Local/Remote switch on 861 bulk power supply must be in local.
- All core must be closest to CPU (physically).
- MOS Memory closest to controller in the following order:

  32K modules (4K chips)
  32K modules (16K chips par populated)
  128K modules (16K chips fully populated)
- No empty slots between arrays.
- SW1-8 on M8158 to Off (open) for test (allows one to access CSR directly). This switch is On (closed) for PDP-11/70.
- Terminators must be screwed in tight as the screws supply power and ground to terminator.



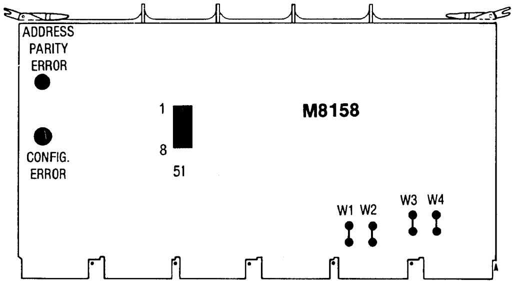
ALL SWITCHES MUST BE OFF
TO USE THE MK11 CONTROL PANEL

M8159/M8164 SWITCHES, INDICATOR

ALL 11/70's WITH MK11 SHOULD HAVE A CONTROL PANEL. IF FOR SOME REASON YOU NEED TO RUN WITHOUT THE PANEL, YOU CAN SET IT UP USING THE SWITCHES.

E46 SW	FUNCTION	E55 SW	FUNCTION
1	ADRS BIT 0	1	ADRS BIT 8
2	1	2	INTERLEAVE BIT 1
3	2	3	INTERLEAVE BIT 0
4	3	4	INTERLEAVE BIT 2
5	4	5	FORCE PANEL ADRS.
6	5	6	ECC DISABLE
7	6		
8	7		

DOUBLE ERROR = MEMORY HAS AT LEAST ONE DBE LOCATION



M8158 SWITCHES, JUMPERS, INDICATORS

CSR SWITCHES

BOX NO.	CSR ADDRESS	S1 SWITCH POSITION		
		S1-3	S1-2	S1-1
0	17772100	ON	ON	ON
1	17772104	ON	ON	OFF
2	17772110	ON	OFF	ON
3	17772114	ON	OFF	OFF

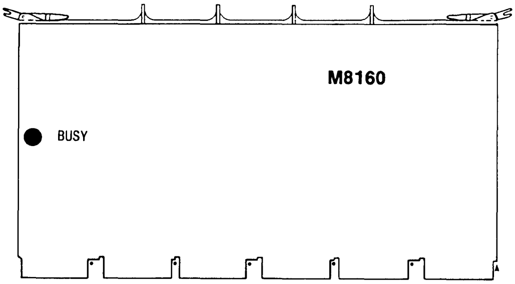
SW1-4 THROUGH SW 1-7 NOT USED.
SW1-8 MUST BE "ON" FOR THE 1170.

POWER FAIL JUMPERS

	W_1	W_2	W_3	W_4
LAST MEMORY ON BUS	OUT	IN	OUT	IN
ALL OTHER MEMORY BOXES	IN	OUT	IN	OUT

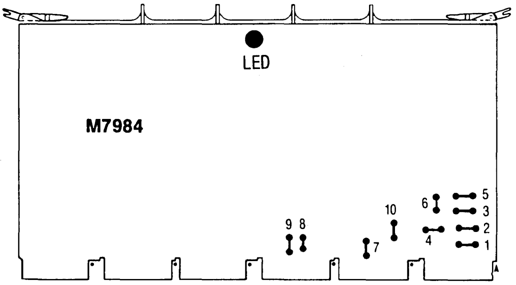
CONFIGURATION ERROR
= SPACE IN MEMORY ORDER
= MOS MEMORY IN WRONG ORDER.

ADDRESS PARITY ERROR
= INCOMING ADDRESS HAD BAD PARITY.



M8160 INDICATOR

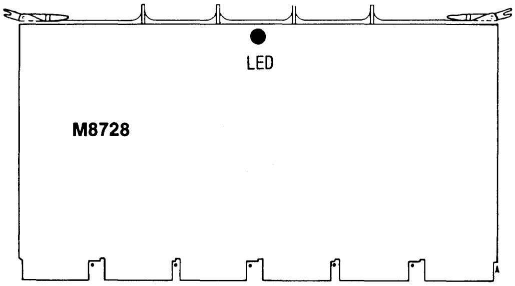
BUSY LED — MEMORY IS BUSY.



M7984 JUMPERS, INDICATOR

JUMPERS W1-W10 SELECT MEMORY SIZE AND CHIP TYPE. THEY ARE FACTORY SET AND SHOULD NOT BE ALTERED.

LED (GREEN) INDICATES MODULE IS RECEIVING +5V.



M8728 JUMPERS, INDICATOR

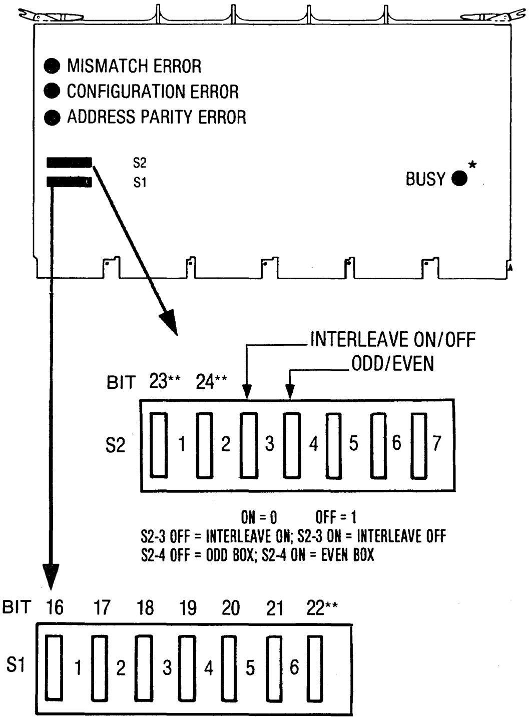
THERE ARE 23 JUMPER LOCATIONS (W1-W23) ON THE MODULE WHICH ARE FACTORY SET FOR MEMORY CHIPS AND SHOULD NOT BE ALTERED.

LED (GREEN) INDICATES MODULE IS RECEIVING +5V.

MJ11 SWITCHES, INDICATORS AND JUMPERS

M8147/8 ADDRESS/INTERLEAVE SWITCHES AND INDICATORS.

M8147/8



ON = 0 OFF = 1
S2-3 OFF = INTERLEAVE ON; S2-3 ON = INTERLEAVE OFF
S2-4 OFF = ODD BOX; S2-4 ON = EVEN BOX

*BUSY LED IS BELOW ADDRESS PARITY ERROR LED ON M8147
**BIT 22, 23, 24 NOT USED

1. Hi and Lo stacks must be same size.
2. All 16K stacks must occupy addresses lower than 32K stacks (within same box).
3. Starting address switches:
   - *Not Interleaved*
     a. Count the multiples of 32K below this box.
     b. Insert that binary number in the starting address switches.
   - *Interleaved*
     a. Same as above, but both boxes must have same starting address and amount of memory.
     b. Set the Interleave switch for INTLV on both boxes.
     c. Set the ODD/EVEN switch to EVEN on the first box and to ODD on the second box.

16K MODULES **1M MAX**		**32K MODULES** **2M MAX**
M8148	ADRS/CONTRL	M8147* ADRS/CNTRL
M8149	DATA XCVR	M8149 DATA XCVR
G114	SENSE/INH	G116
G235	DRIVE	G236
H217C	STACK	H224C
7010497	BACKPLANE	7010497 with ECO - 04

*M8147 will work with 16K and/or 32K stacks.
*Insure screws are installed tightly in terminators as they connect power and ground to the terminator.

LED'S

MISMATCH ERROR — HI AND LOW STACK PAIR NOT THE SAME SIZE.

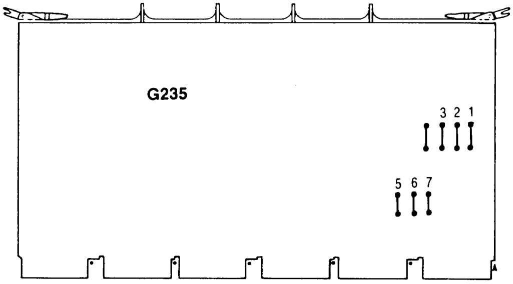
CONFIGURATION ERROR — 16K STACKS NOT BEFORE 32K STACKS.

ADDRESS ERROR — PARITY ERROR DETECTED ON INCOMING ADDRESS.

BUSY LED — MEMORY IS BUSY.

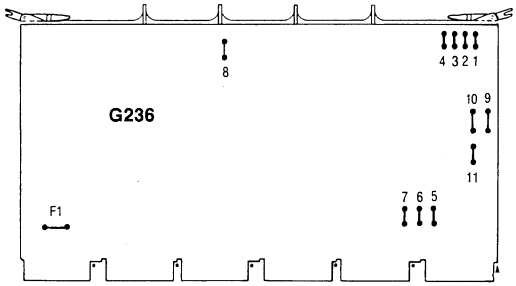
G235/G236 BIAS CURRENT, STROBE JUMPERS AND FUSES

REFER TO ADJUSTMENT PROCEDURE BEFORE ALTERING JUMPERS.



BIAS JUMPERS: (IN = 1; OUT = 0.)

W7	W6	W5	.... G236 MODULE
0	0	0	HIGHEST CURRENT
0	1	0	
1	0	0	
1	1	0	
0	0	1	
0	1	1	
1	0	1	
1	1	1	LOWEST CURRENT



STROBE JUMPERS: (IN = 1; OUT = 0.)

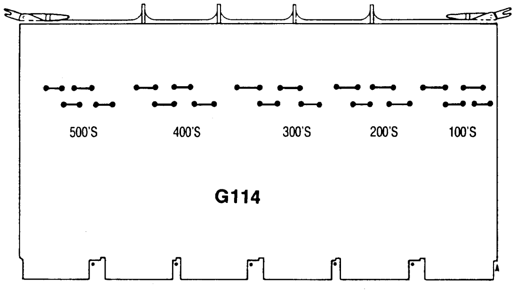
W4	W3	W2	W1	.... G236
W1	W4	W3	W2	.... G235
0	0	0	0	LATEST STROBE
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	
0	0	0	1	
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	1	
1	0	1	1	
1	1	0	1	
1	1	1	1	EARLIEST STROBE

W8-W11 MUST BE INSTALLED
W8 = IN = X READ CURRENT SOURCE
W9 = IN = Y READ CURRENT SOURCE
W10 = IN = X WRITE CURRENT SOURCE
W11 = IN = Y WRITE CURRENT SOURCE

F1 = 3/4 AMP PICO FOR +20 VOLT

G114/G116 FUSES

ALL FUSES 3/4 AMP PICO

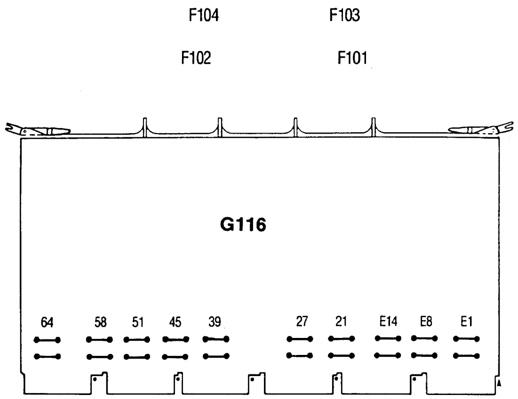


G114 FUSES

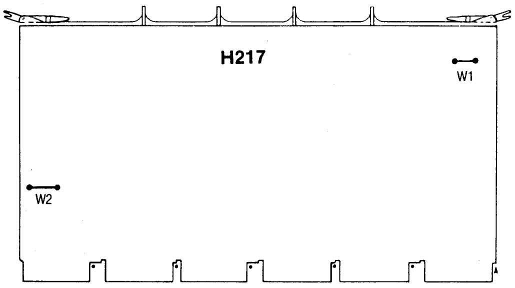
G114 FUSE	G116 FUSE	INTERNAL BUS DATA BIT	PDP 1170 BYTE - BIT
101	10	19	1-PARITY, 3-PARITY
102	9	18	NOT USED
103	12	17	NOT USED
104	11	16	0-PARITY, 2-PARITY
201	2	15	1-6, 3-6
202	1	14	1-5, 3-5
203	4	13	1-4, 3-4
204	3	12	1-3, 3-3
301	6	11	1-2, 3-2
302	5	10	1-1, 3-1
303	8	09	1-0, 3-0

FUSE LAYOUT EXAMPLE:

304	7	08	1-7, 3-7
401	14	07	0-7, 2-7
402	13	06	0-6, 2-6
403	16	05	0-5, 2-5
404	15	04	0-4, 2-4
501	18	03	0-3, 2-3
502	17	02	0-2, 2-2
503	20	01	0-1, 2-1
504	19	00	0-0, 2-0



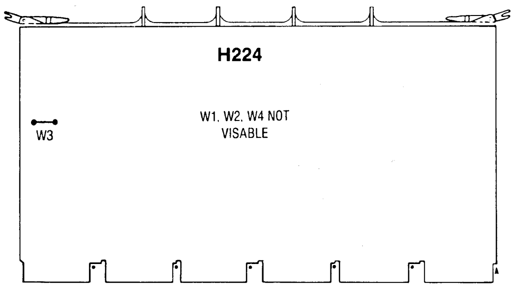
G116 FUSES



H217 JUMPERS

W1 + W2 = TEST PURPOSES ONLY

DO NOT ALTER



W1, W2, W4 NOT VISABLE

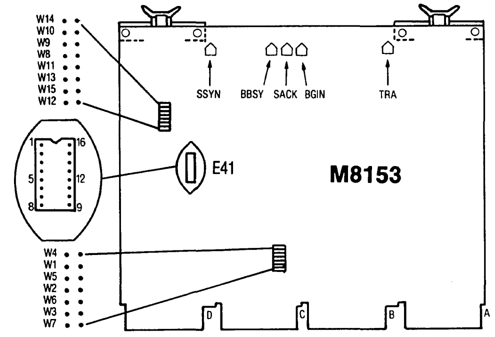
H224 JUMPERS

W1 THRU W4 = TEST PURPOSES ONLY

DO NOT ALTER

RH70 JUMPERS AND INDICATORS

M8153 JUMPERS AND INDICATORS



OF REGISTERS SELECTION

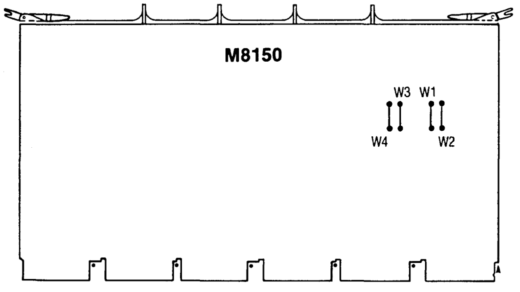
SET THE JUMPERS (E41) TO SELECT >20 OR <20 REGISTERS AND FOR THE NUMBER OF REGISTERS MINUS 2 BY THE BINARY WEIGHT.

JUMPER	>20 REGISTERS	<20 REGISTERS
1-16	OUT	IN
2-15	OUT	IN
3-14	IN	OUT
4-13	IN	OUT

JUMPER	BINARY WEIGHT
5-12	2
6-11	4
7-10	8
8-9	16

JUMPER OUT = BINARY WEIGHT

EG. RWP04 HAS 22 REGISTERS, SO JUMPERS AT E41 (FROM TOP TO BOTTOM) WOULD BE OUT, OUT, IN, IN, IN, OUT, IN, OUT TU/TE16/TU45/TU77 HAS 16 REGISTERS, SO JUMPERS AT E41 (FROM TOP TO BOTTOM) WOULD BE IN, IN, OUT, OUT, OUT, OUT, OUT, IN.



ADDRESS SELECTION

JUMPER	ADDRESS BIT
W14	12
W10	11
W9	10
W8	9
W11	8
W13	7
W15	6
W12	5

JUMPER IN = 0
JUMPER OUT = 1

VECTOR SELECTION

JUMPER	ADDRESS BIT
W7	2
W3	3
W6	4
W2	5
W5	6
W1	7
W4	8

JUMPER IN = 1
JUMPER OUT = 0

LED'S

THE FOLLOWING STATUS LED's ARE PROVIDED FOR TROUBLE-SHOOTING

— TRANSFER (TRA)
— BUS GRANT IN (BG IN)
— SELECTION ACKNOWLEDGE (SACK)
— BUS BUSY (BBSY)
— SLAVE SYNC (SSYN)

M8150 JUMPERS

JUMPERS W1 THRU W4 ARE USED TO ISOLATE PROBLEMS IN THE WRITE CHECK ERROR CIRCUIT. REFER TO PRINT SET PAGE MDPE.

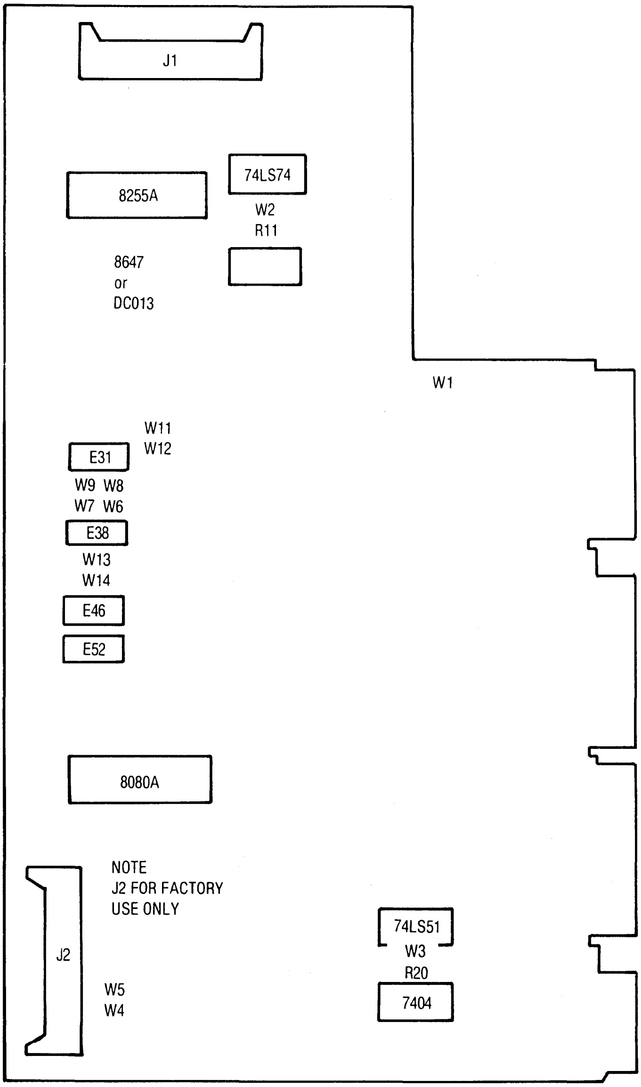
KY11-R JUMPERS AND SWITCHES

M8255 Jumper Description and Configuration

	Jumpers Installed At Factory For ROM Configuration			
Jumper	1Kx8	2Kx8	Description	IN/OUT State
W1	—	—	Factory use only	Always IN
W2	—	—	Steal Grant L	OUT on CS Rev F and earlier to disable IN on CS Rev H to enable
W3	—	—	KW11-L option	IN to disable KW11-L equivalent logic OUT to enable KW11-L equivalent logic
W4	IN	OUT		
W5	OUT	IN		
W6	IN	OUT		
W7	OUT	IN		
W8	IN	OUT		
W9	OUT	IN		
W10	—	—	There is no W10	
W11	IN	OUT		
W12	OUT	IN		
W13	IN	OUT		
W14	OUT	IN		

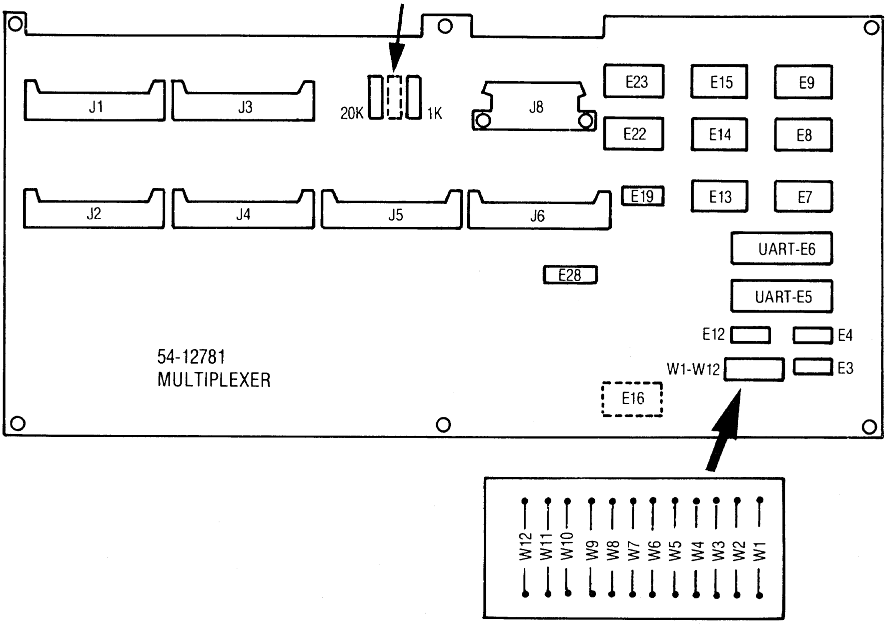
M8255 ROM/PROM Configuration

M8255	Microcode	ROM or	In ROM Location			
Variant	Version	PROM	E52	E46	E38	E31
-00	V01	1Kx8 UV EPROM (with windows)	007B7	008B7	009B7	010B7
-00	V01	2Kx8 Masked ROM (without windows)	013E2	—	014E2	—
-00	V02	2Kx8 UV EPROM or Masked ROM	175E2	176E2	177E2	178E2
-01	V02	2Kx8 UV EPROM (with windows)	075E2	077E2	076E2	121E2



NOTES: — KY11-R IS A DMA DEVICE, THEREFORE YOU MUST REMOVE BACKPLANE JUMPER CA1-CB1

— KY11-R CAN ASSERT AC LOW ON THE UNIBUS.


THIS 20K OHM RESISTOR (R26) MUST BE REMOVED FROM ALL-01 VARIANT BOARDS

54-12781 JUMPER CONFIGURATION AND LAYOUT

System Terminal Fill Character Selection
54-12781 DIP Switch E19

Switch	Setting	Function
S1	N/A	Unused.
S2	N/A	Unused.
S3	OFF	Four fills after LF (VT05).
	ON	No fills after LF.
S4	OFF	Eight fills after CR.
	ON	No fills after CR.
S5	OFF	Gate electronic console status bits into transmitter control and status register.
	ON	Suppress electronic console status bits.
S6,7,8	OFF	Reserved for future use.
	Usual switch settings for LA34, LA36, LA38, LA120: S3 = ON, S4 = ON, S5 = ON.	

Console Terminal 20 mA Loop Option Selection
54-12781 DIP Switch E28

	Transmitter Switches					Receiver Switches				
Mode	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
Active*	ON	OFF	ON	OFF	ON	ON	OFF	ON	OFF	ON
Passive	OFF	ON	OFF	ON	OFF	OFF	ON	OFF	ON	OFF

**Standard LA36 switch setting, active mode*

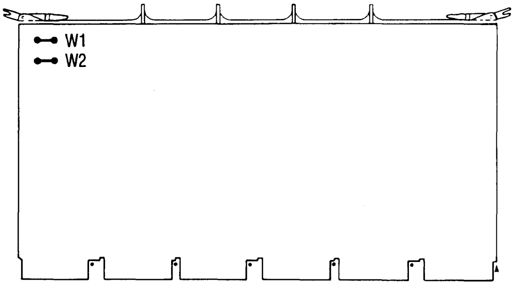
Baud Rate Selection
54-12781 DIP Switch E3

Baud	**Transmit:**	**S1**	**S2**		**S4**		**S6**		
Rate	**Receive:**			**S3**		**S5**		**S7**	**S8**
110		ON	OFF	ON	ON	OFF	ON	ON	ON
300*		OFF	ON	OFF	OFF	ON	ON	OFF	ON
600		ON	OFF	ON	OFF	OFF	ON	OFF	ON
1200		OFF	OFF	OFF	OFF	OFF	ON	OFF	ON
1800		ON	ON	ON	ON	ON	OFF	ON	OFF
2000†		OFF	ON	OFF	ON	ON	OFF	ON	OFF
2400		ON	OFF	ON	ON	OFF	OFF	ON	OFF
3600†		OFF	OFF	OFF	ON	OFF	OFF	ON	OFF
4800		ON	ON	ON	OFF	ON	OFF	OFF	OFF
7200		OFF	ON	OFF	OFF	ON	OFF	OFF	OFF
9600		ON	OFF	ON	OFF	OFF	OFF	OFF	OFF
INVALID		OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

**Standard switch settings for LA36 (300 Baud)*
†*Not compatible with LA120*

FP11-C JUMPERS

M8128 FLOATING POINT START/ATTEN DELAY JUMPERS



W1 in, W2 out = Normal
*W1 out, W2 in = Reserved

*This Reserved Jumper configuration adds 5ns to 7ns delay onto the FP START and FP ATTEN signals from the CPU (reference FRMC). This is not used and the Jumpers should always be set normal, W1 in and W2 out.

GENERAL 11/70 JUMPERS

The following is a list of backplane jumpers useful in troubleshooting 1170 problems.

Backplane:

To disable Cache Data Memories

Group 0 C17L1 To ground
Group 1 C17J1 To ground

To remove KW11L

Jumper D1R2 to D1V2

To remove any NPR device in CPU backplane

G7273 in C and D row of affected slot or
G727 in D row and jumper C4XA1 to C4XB1

To run off RC clock on M8139 TIG module in place of crystal clock

F13J1 to ground

To clear MJ11 memory box address error LED without powering off

TP2 to ground

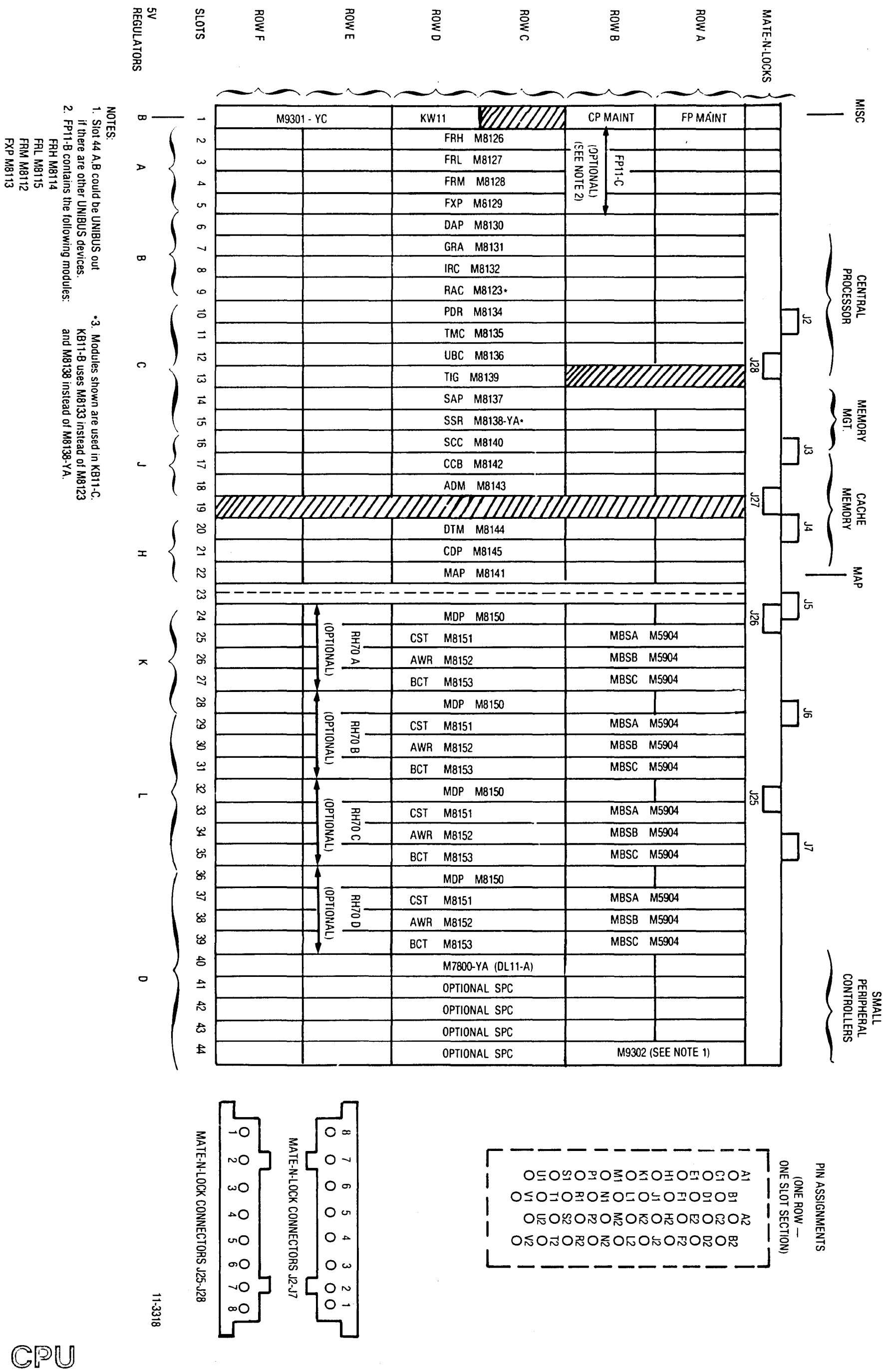
WIRE COLORS FOR PDP11 VOLTAGES

Voltage	Color
+8	white
+5	red
-15	blue
+15	grey
-5	brown
AC LO	yellow
DC LO	violet
+20	orange
LTC	brown
GND	black

1170 MODULES DC POWER USAGE

Voltage	Modules + Circuit Usage		Pin
+5	All	All TTL Chips	Any A2
+15	M8139	Clock Circuits	E13A1
	SPC's	Unique to Device	CxxU1 BxxN2
-15	M8139	Clock Circuits	E13B2
	M8142	SYNC Clock	E17B2
	M5904	Transceivers	AxxB2
	M8126	FP Clocks on FRHM	E02B2
+8	FP11-B	Maintenance card	A01B1
	KB11-B/C	Maintenance card	B01B1

CHAPTER 8
CPU



SLOTS	ROW F	ROW E	ROW D	ROW C	ROW B	ROW A
1	M9301 - YC		KW11	▨	CP MAINT	FP MAINT
2			FRH M8126		FP11-C (OPTIONAL) (SEE NOTE 2)	
3			FRL M8127			
4			FRM M8128			
5			FXP M6129			
6			DAP M8130			
7			GRA M8131			
8			IRC M8132			
9			RAC M8123*			
10			PDR M8134			
11			TMC M8135			
12			UBC M8136			
13			TIG M8139		▨	▨
14			SAP M8137			
15			SSR M8138-YA*			
16			SCC M8140			
17			CCB M8142			
18			ADM M8143			
19	▨	▨	▨	▨	▨	▨
20			DTM M8144			
21			CDP M8145			
22			MAP M8141			
23						
24		RH70 A (OPTIONAL)	MDP M8150			
25			CST M8151		MBSA M5904	
26			AWR M8152		MBSB M5904	
27			BCT M8153		MBSC M5904	
28		RH70 B (OPTIONAL)	MDP M8150			
29			CST M8151		MBSA M5904	
30			AWR M8152		MBSB M5904	
31			BCT M8153		MBSC M5904	
32		RH70 C (OPTIONAL)	MDP M8150			
33			CST M8151		MBSA M5904	
34			AWR M8152		MBSB M5904	
35			BCT M8153		MBSC M5904	
36		RH70 D (OPTIONAL)	MDP M8150			
37			CST M8151		MBSA M5904	
38			AWR M8152		MBSB M5904	
39			BCT M8153		MBSC M5904	
40			M7800-YA (DL11-A)			
41			OPTIONAL SPC			
42			OPTIONAL SPC			
43			OPTIONAL SPC			
44			OPTIONAL SPC		M9302 (SEE NOTE 1)	

MATE-N-LOCKS
5V REGULATORS
B
A
B
C
J
H
K
L
D
MISC
CENTRAL PROCESSOR
MEMORY MGT.
CACHE MEMORY
MAP
SMALL PERIPHERAL CONTROLLERS
J2
J3
J4
J5
J6
J7
J28
J27
J26
J25
PIN ASSIGNMENTS (ONE ROW — ONE SLOT SECTION)
A1 C1 E1 H1 K1 M1 P1 S1 U1
B1 D1 F1 J1 L1 N1 R1 T1 V1
A2 C2 E2 H2 K2 M2 P2 S2 U2
B2 D2 F2 J2 L2 N2 R2 T2 V2
8 7 6 5 4 3 2 1
MATE-N-LOCK CONNECTORS J2-J7
1 2 3 4 5 6 7 8
MATE-N-LOCK CONNECTORS J25-J28
11-3318

NOTES:
1. Slot 44 A,B could be UNIBUS out if there are other UNIBUS devices.
2. FP11-B contains the following modules:
FRH M8114
FRL M8115
FRM M8112
FXP M8113

*3. Modules shown are used in KB11-C. KB11-B uses M8133 instead of M8123 and M8138 instead of M8138-YA.

CPU

FIGURE 8-1

MODULE LAYOUT

11/70 REV. LEVELS

Module	**Etch**	**CS Rev.**	
M8126	C	C	FRH
M8127	C	B	FRL
M8128	C	B	FRM
M8129	C	B	FXP
M8130	A	D	DAP
M8131	C	D	GRA
M8132	B	D	IRC
M8123	C	B	RAC
M8134	C	E	PDR
M8135	A	*	TMC
M8136	C	F	UBC
M8139	B	D	TIG
M8137	B	A	SAP
M8138-YA	B	F	SSR
M8140	A	B	SCC
M8142	B	F	
M8143	B	B	ADM
M8144	A	*	DTM
M8145	B	C	CDP
M8141	A	A	MAP
70-10329	WL>M	WT>S	BACKPLANE

DAP (M8130) DATA PATHS

A “BR” Register
B “A”, “B”, “BA” MUX
C “A”, “B”, “BA” MUX
D “A”, “B”, “BA” MUX
E “KI” MUX, TRAP and START Vectors
F SHFR (0-5) PCA and PCB (0-5) ALU (0-7)
H SHFR (8-14) PCA and PCB (6-15) ALU (8-15)
J SHFR 7&15

GRA (M8131) GENERAL REG AND ALU CONTROL

A ALU Shr. Cont. and ALU sub ROM, swap byte in SHFR
B SHFT = zero detector
C Gen. Purpose Reg. Address
D GS, GD, “SR” MUX, “DR” MUX, “DR” (0-3) SR (0-5)
E GS, GD, “SR” MUX, “DR” MUX, “DR” (4-7)
F GS, GD, “SR” MUX, “DR” MUX, “DR” (8-11) SR (6-11)
H GS, GD, “SR” MUX, “DR” MUX, “DR” (12-15)
J Shifter Counter
K ALU Control ROM Chart

IRC (M8132) “IR” DECODE AND COND CODES

A “IR” Register (0-15)
B “B” Fork, IR Decode
C “C” FORK, IR Decode, Source Const.
D Instr. Traps, Destination Const.
E Condition Codes “V” Bit
F Condition Codes “C”, “N”, “Z”
H Condition Code Bits
J Inst. Decode ROM Maps

KNL CONSOLE BOARD

A,B,C,D

PDR (M8134) PROC DATA AND UNIBUS REGISTER

A “BR” MUX
B “BRA” Register, Light Register
C Program Brake Register, Stack Limit
D Prog. Interrupt, Proc. Status Register
E “D” MUX Register to Unibus
F Data Display MUX
H HI and LO Parity Bit, J1 Cable Conn.
J Bus Buffer Register

RAC (M8123) ROM and ROM CONTROL

A ROM Bits 48-63 (zap 200 generated)
B ROM Bits 32-47
C ROM Bits 16-31
D ROM Bits 00-15
E “A” Fork Logic
F “A” Fork Logic
H “A” Fork Logic
J Instruction Reg. (“A” Fork)

K Branch Conditions (ROM, ADD, MOD) (BEN +20, +40)
L ROM Address Selection

TIG (M8139) TIMING GENERATOR

TMC (M8135) Traps and Misc. Control

A Priority ARB. (Order of Sequence Chart)
B Priority ARB Trap Vectors (BQR True)
C Add Error, Odd Address Error, etc.
D CPU Reg. (Bits 2-7) Internal Stack Limit
E MSC, BCT, BSC, Decode
F "BR" MUX Selection FP Read

UBC (M8136) UNIBUS AND CONSOLE CONTROL

A MSYN, BBSY, Timeout, CP BSY
B Parity and Timing Cont. (Parity, Error, PE Abort, TIG Restart)
C Restart Const. (BUS SYNC INTRF, RESET INTR)
D BUS ARB. (External BRQ, No Sack, Proc BR4-BR7)
E PWRF Control (BUS, DCLO, Int. BUS, Start Init)
F Console Cont. (Console ACK, Stop, Start, Continue)
H Console YADR (Reg. Exam and Step, Clear, Conf)
* Timing Diagram
* Flow Chart for Unibus Hand Shaking

ADM (M8143) ADDRESS MEMORY BOARD (CACHE)

A TAG0 (add. for TAG0 Bits 14-21)
B TAG0 (add. for TAG0 Bits 13-10) and TAG0 Valid, and Par A, Par B)
C TAG1 (add. for TAG1 Bits 14-21)
D TAG1 (add. for TAG1 Bits 13-10) TAG1 valid and Par A, Par B)
E Address MUX (0-11)
F Address MUX (12-21)
H MBC Add. Buffer (MBC A00-A21) (C0-C1)
J Byte Mask, Pup Sequencer
K Equals Check, Parity Check (Group 0+1 hit, group 0+1 match)
L Main. Memory Add. Drivers (ADRS 2-21) and Memory Signals
* TAG0 and TAG1 Block Diagram

CCB (M8142) CACHE CONTROL

A Initialize and Clock Logic
B Request Arbitration
C Datip and End Cycle Control (Mem Sync, Done, UB ACKN)
D Main Mem. Control (Start, Timeout, Slow Cycle, Start Slow)
E Timing and Main Mem. Restart (Disable Req. Mem Sync Slow)
F Reg. Data Path (Reg D00-D15)
H Error ADS Reg. and Decode (ERR Add. 00-21, C0+C1)
J Cache Error Reg. (Bits 10-15)
K Cache Error Reg. (Bits 0-9)
L Cache Maint. and Hit Reg. (Hit Reg 0-5) Maint. Mode
M Valid and Write Select Logic (Valid Input, Write Sel.)
* Register Bit Breakdown Chart

CDP (M8145) CACHE DATA PATH

A BD Register (BD00-BD31)
B Data MUX (Even MUX D00-D15) (Odd MUX D16-D31)

C Main. Memory Even Word (Main Data Byte 0 [0-7] and Main Data Byte 1 [0-7]

D Main. Memory Odd Word (Main Data Byte 2 [0-7] and Main Data Byte 3 [0-7]

E Write MUX (Write MUX 00-15)

F Memory Parity Control (Main Hi Parity OK, Main LO Par OK, MUX Byte 0-3)

H MBC Request (Areg, Breg, Creg, Dreg, Block A-D)

J MBC Arbitrator (Req, Clk, Init A, Sel Data Ctrl A,B,C,D)

K MBC Data Control (Read "B", RDY CLK' Data RDY, RIP, Data RDY CNTRL A-D)

* Cache Data Path Flow Chart

DTM (M8144) DATA MEMORY

A Address Drivers (WRD 0-3 A02H and WRD0-3 A03H and WRD0-3 A04H)

B Write Pulse Drivers and Chip Select Dri. (CS0-3 and Write H and A01L)

C Word 0 Hi Byte (Even MUX D08-D15 and Group 0 Hi Par H)

D Word 0 Lo Byte (GRP0 D07-D00 and Group 0 Lo Par H)

E Word 1 Hi Byte (GRP08-15 and GRP0 Hi Par H)

F Word 1 Lo Byte (GRP0 D07-D00 and GRP0 Lo Par H)

H Word 2 Hi Byte (GRP1 D08-D15 and GRP1 Hi Par H)

J Word 2 Lo Byte (GRP1 D07-D00 and GRP1 Lo Par H)

K Word 3 Hi Byte (GRP1 D08-D15 and GRP1 Hi Par H)

L Word 3 Lo Byte (GRP1 D07-D00 and GRP1 Lo Par H)

M Data Output MUX (CDMX D00-D15 and Bad Parity H)

N Fast Memory Parity Checkers (Data Par 0 OKL and Data Par 1 OKL)

P Main Memory Data Inverters (Even MUX 00-15 and Odd MUX 15-31)

* Memory Management Block Diagram 1.

* Memory Management Block Diagram 2.

* Memory Management Registers

* Cache Address Timing

* INternal REgisters Cycle Timing

* Mem. MGMT Trap Timing

SAP (M8137) SYSTEM ADDRESS PATH

A Kernal Page Address Registers (PAF06-PAF21)

B Supervisor Page Add. Reg. (PAF06-PAF21)

C User Page Add. Reg. (PAF06-PAF21)

D Kernal Page Description Reg. (APR ADDRO-3 and WRTN Data)

E Supervisor Page Desc. Reg. (User Super, Kernal CSL)

F User Page Desc. Reg. (PLF6-0 and Addr 0-3)

H Address Buffers (VA15-01 and Ex Mem Flag H)

J Relocation Logic (16, 18, 22 Bit Map and Reloc. ALU and PA21-PA6)

K Register Select and Control (1 SPace A+B BR00-BR11)

L Abor and Trap Decode (Mem MGMT and Read Only FAult and Abort Cond.)

M PDR/PAR Read Multiplexers (APR Bit 00-APR Bit 15)

N Valid Address Check (18 Bit Overflow and not Cache Adrs)

SCC (M8140) SYSTEM DESC/CNSL CABLES

A Address Buffers (VA15-VA00 and PA06-PA15)

B Reg. Address Decoder (User Par Adrs. and MMR Adrs and Super PDR)

C Reg Address Decoder (SW Reg. L and Int. Reg. A+B and Int. Reg H)

D Reg. Add Decode (APR Reg, into Reg, Int CLR BL)

E	System Reg. add Decode (PS, SL, PIR, PB-Adrs's)
F	General Register Decode (Gen. Reg. (1) H, Console Light Driver)
H	Internal Bus Drivers (Int D15-Int D00)
J	SCC/CNSL Cables (J1, J2)
K	Console Add Display (Disp Adrs 21-06)
L	Unibus Add. Drivers and MMR3 (Bus A00L-A17L)
M	System Reg. Bits 00-07 (internal "D" Bus)
N	System Reg. Bits 08-15 (internal "D" Bus and Memory Size)

SSR (M8138-YA) SYSTEM STATUS REGISTERS

A	MMU ROM + ROM Decode
B	Address space (K, S, U + I/D) Control Logic
C	MMR0 Bits <15:13> Abort Flags
D	MMR0 Bits <12,9> Trap Flags
E	MMR0 Bits <8:0> Mode Status Bits
F	MMR1 Bits <15:0> Auto INC/DEC Information
H	MMR2 Bits <15:0> 16 Bit Virtual Address
J	Internal Data Bus MUX
K	Console Address Select Switch, Timing Logic, Init Logic
L	MMU ROM Map for M8138-YA
M	MMU ROM Map for M8138

FIGURE 8.2

1170 SYSTEM DATA PATHS

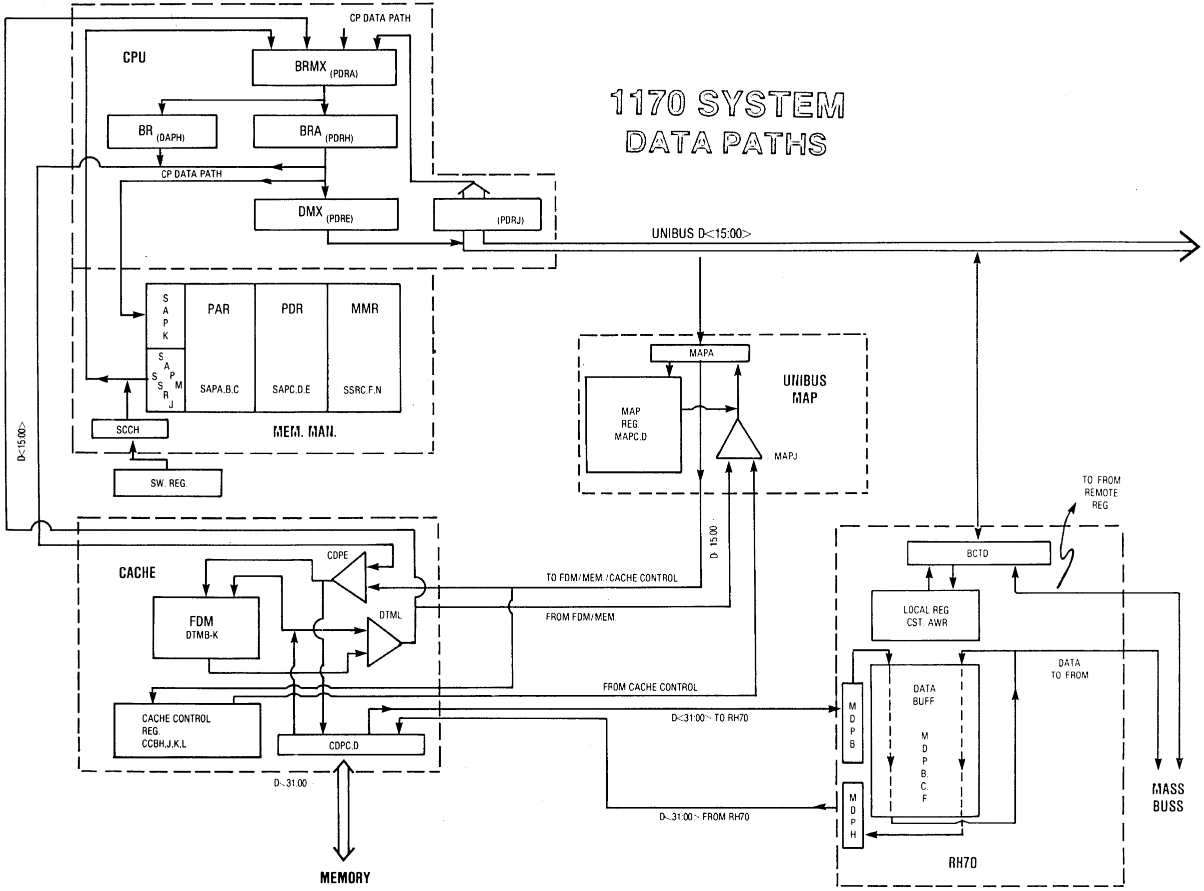


FIGURE 8.3

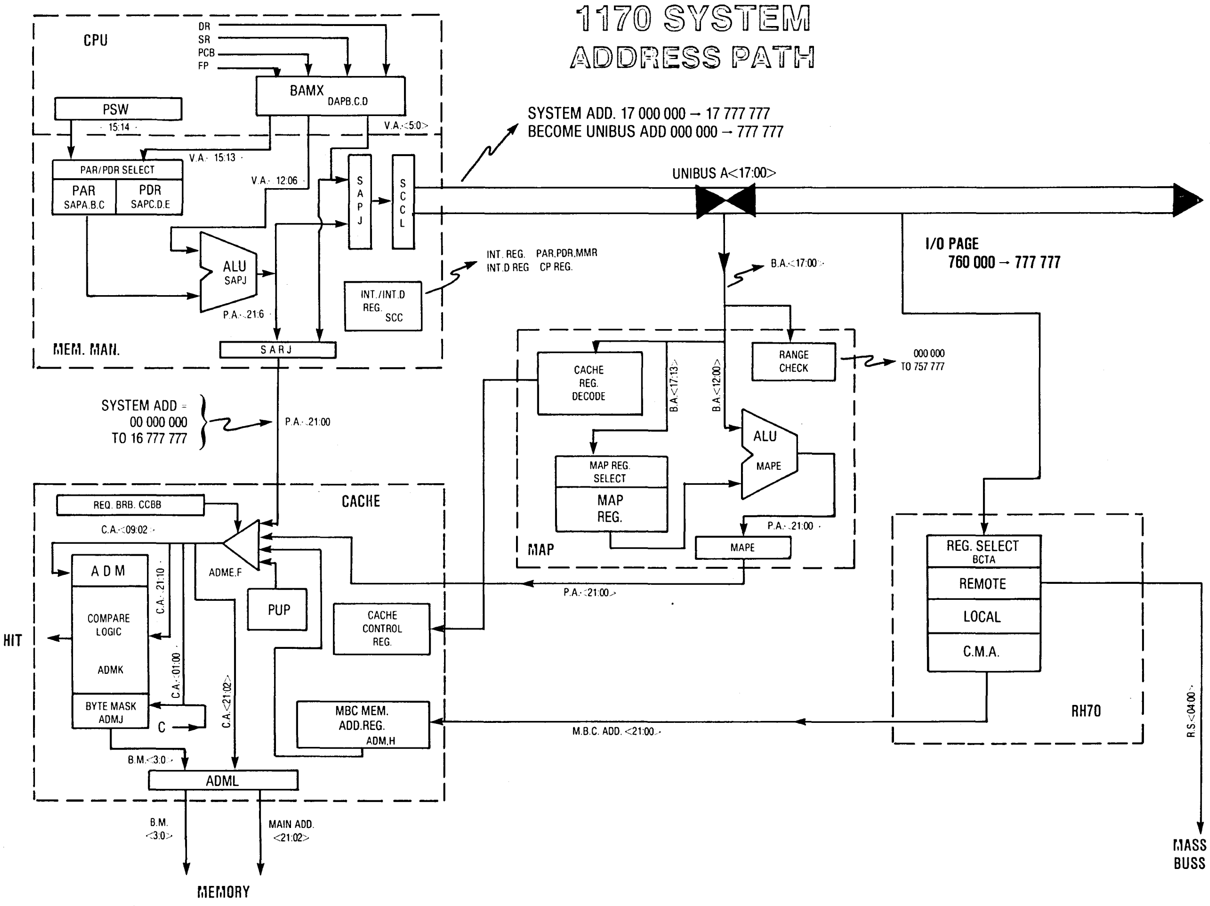

1170 SYSTEM
ADDRESS PATH
CPU
DR
SR
PCB
FP
BAMX
DAPB.C.D
PSW
15:14
V.A.<5:0>
PAR/PDR SELECT
V.A. 15:13
PAR
SAPA.B.C
PDR
SAPC.D.E
V.A. 12:06
S A P J
S C C L
ALU
SAPJ
P.A.<21:6
INT./INT.D REG.
SCC
MEM. MAN.
S A R J
SYSTEM ADD. 17 000 000 → 17 777 777
BECOME UNIBUS ADD 000 000 → 777 777
UNIBUS A<17:00>
INT. REG. PAR.PDR.MMR
INT.D REG CP REG.
B.A.<17:00>
I/O PAGE
760 000 → 777 777
RANGE CHECK
000 000
TO 757 777
CACHE REG. DECODE
B.A.<17:13>
B.A.<12:00>
ALU
MAPE
MAP REG. SELECT
MAP REG.
P.A.<21:00
MAPE
MAP
SYSTEM ADD =
00 000 000
TO 16 777 777
P.A.<21:00
CACHE
REQ. BRB. CCBB
C.A.<09:02>
ADME.F
A D M
C.A.<21:10
COMPARE LOGIC
ADMK
C.A.<01:00
C.A.<21:02>
HIT
PUP
CACHE CONTROL REG.
P.A.<21:00>
BYTE MASK
ADMJ
C
MBC MEM. ADD.REG.
ADM.H
M.B.C. ADD. <21:00>
B.M.<3:0>
ADML
B.M.
<3:0>
MAIN ADD.
<21:02>
MEMORY
REG. SELECT
BCTA
REMOTE
LOCAL
C.M.A.
RH70
R.S.<04:00>
MASS BUSS

FIGURE 8.4

1170 SYSTEM
DATA PARITY NETWORK

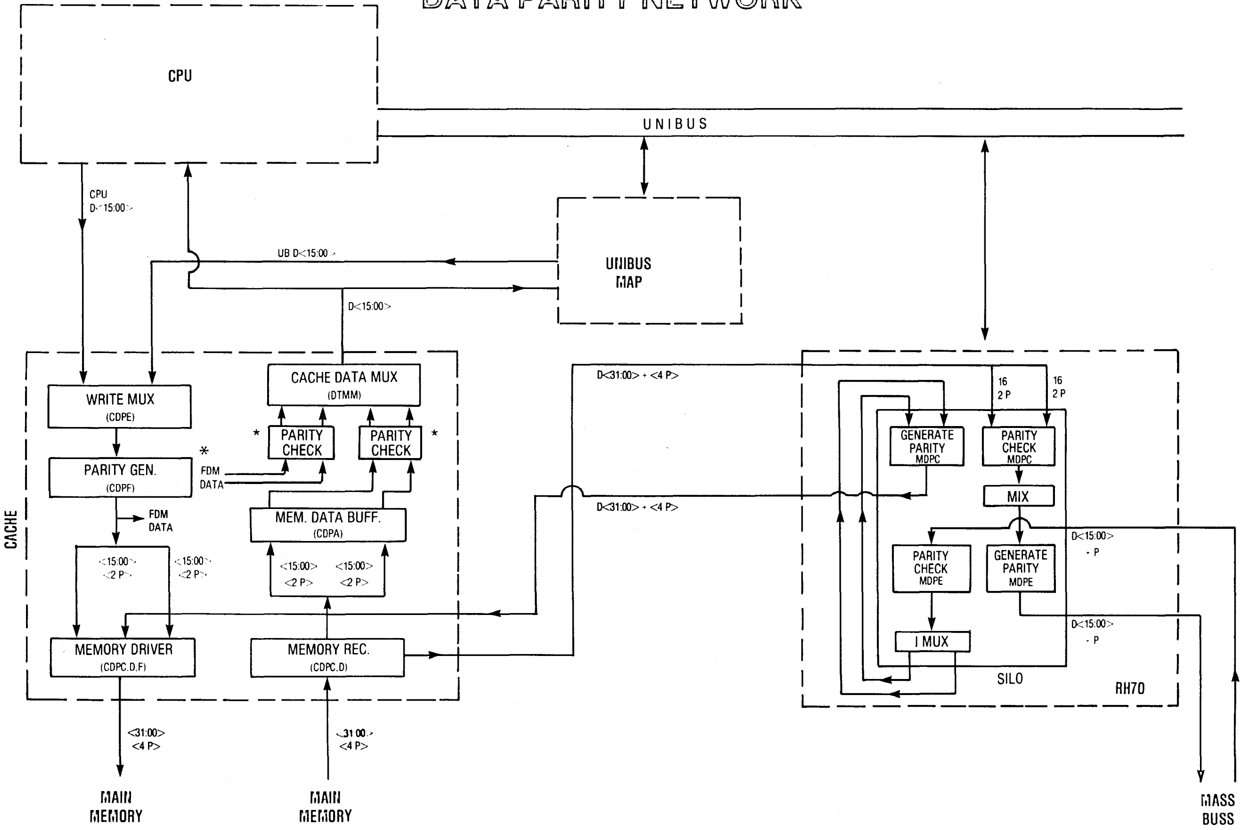
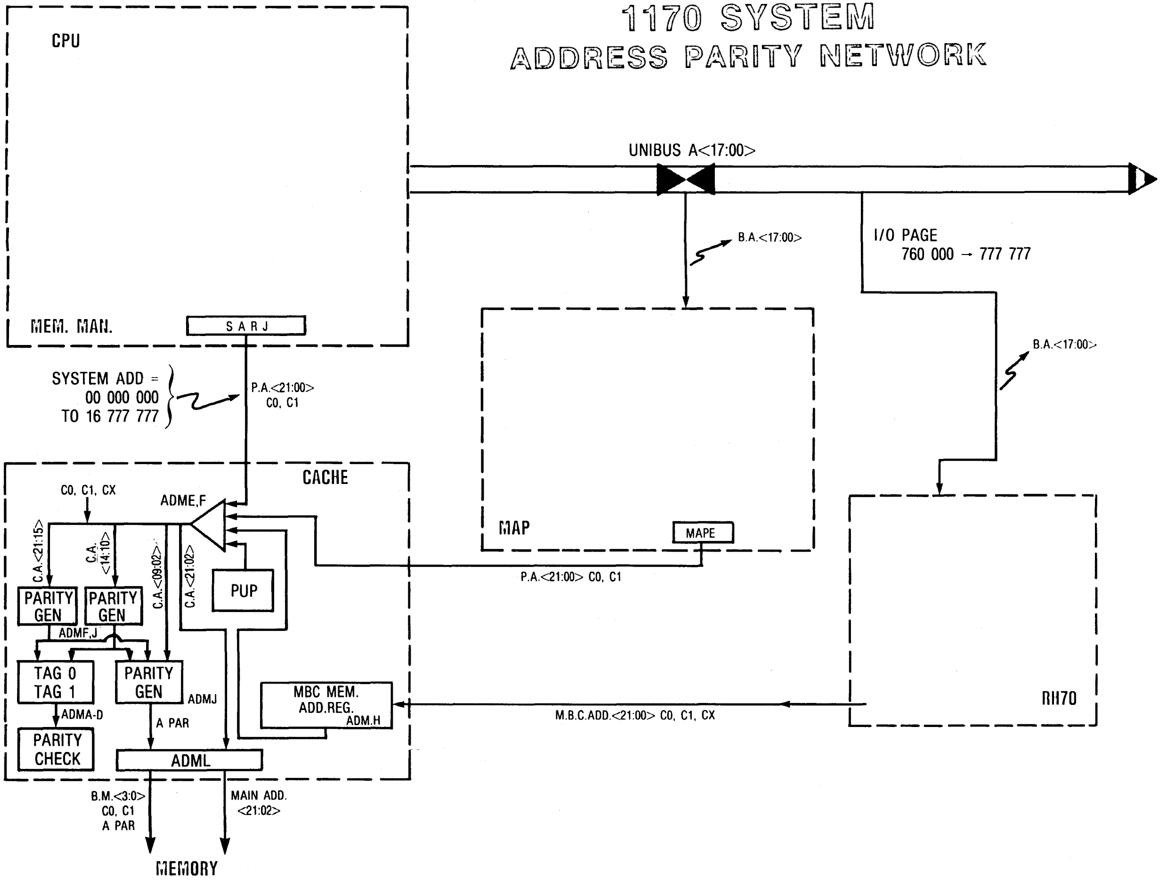


FIGURE 8.4A

1170 SYSTEM ADDRESS PARITY NETWORK



BLOCK DIAGRAM

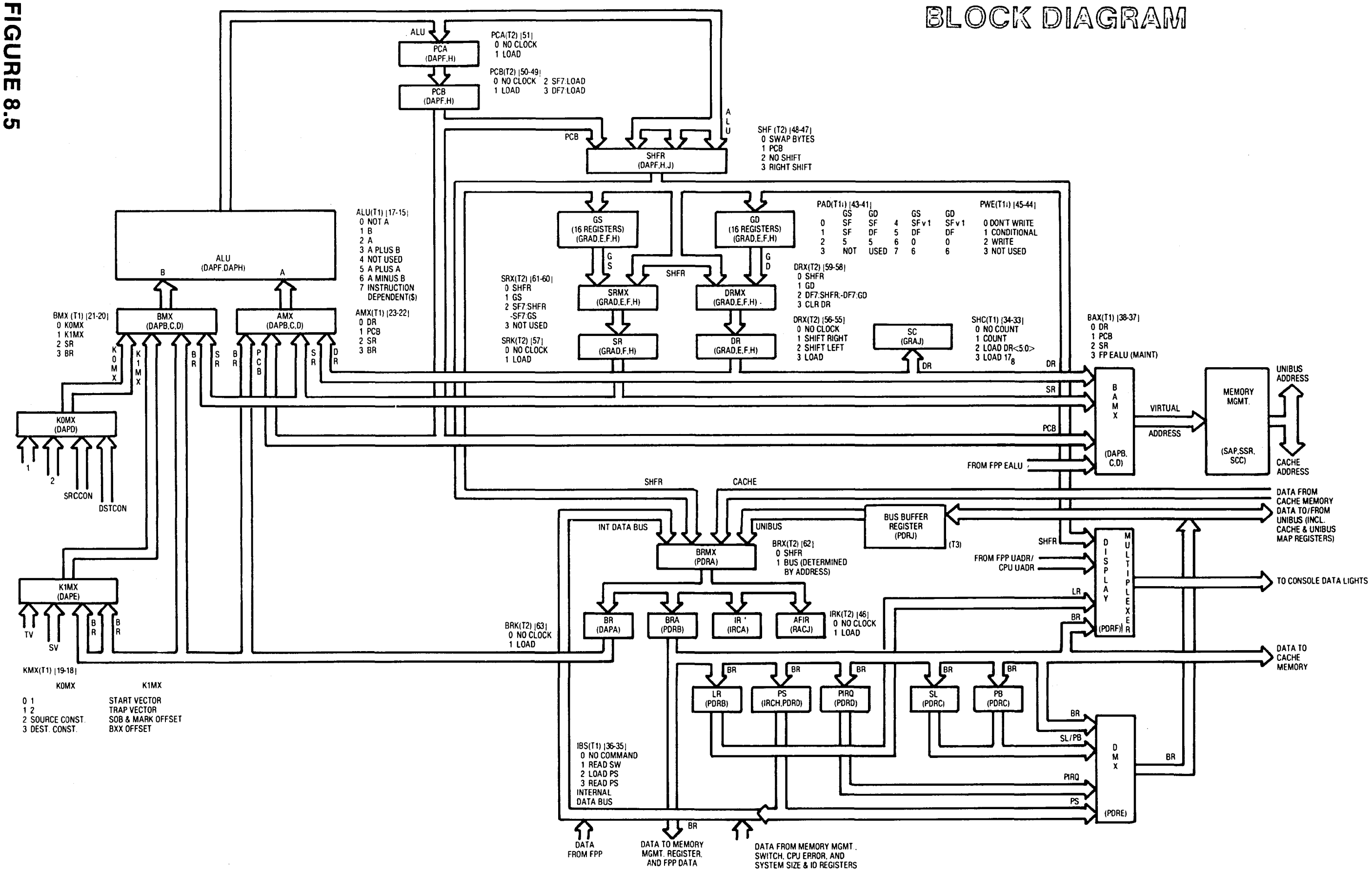
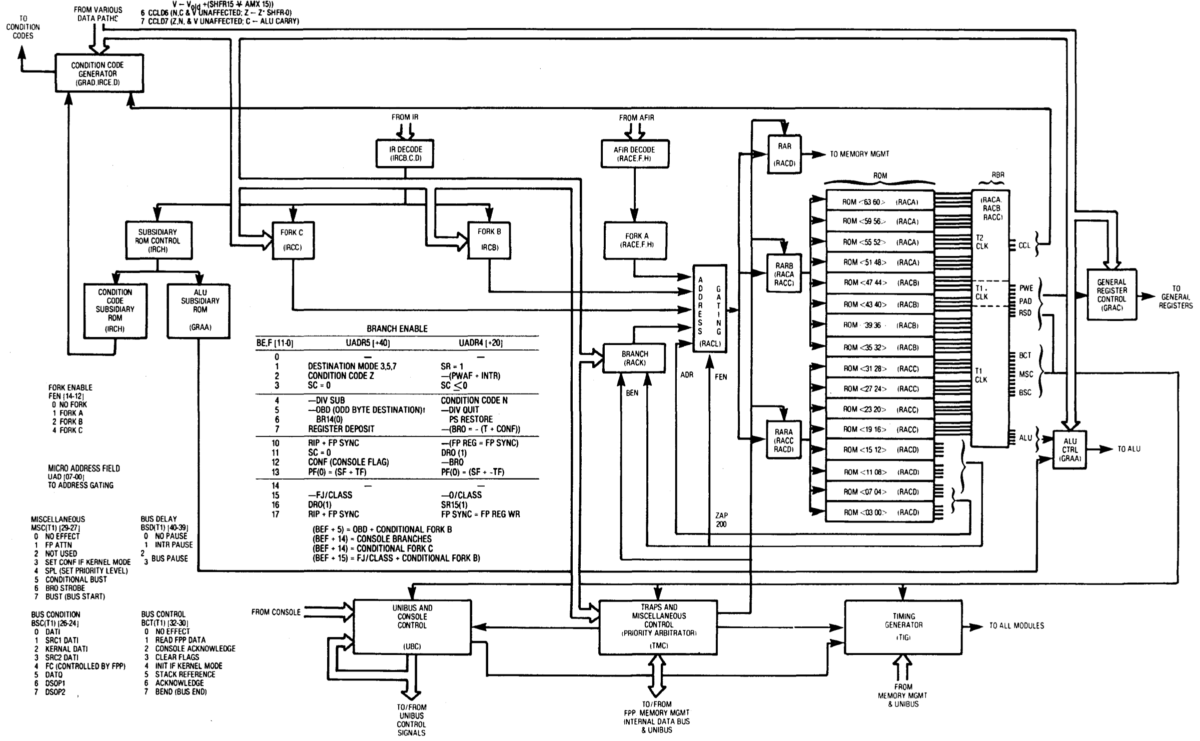


FIGURE 8.5

BLOCK DIAGRAM



CONDITION CODE LOAD
CCL (T2) [54-52]
0 NO CHANGE
1 INSTRUCTION DEPENDENT
2 SET/CLR FROM BR (CCOP)
3 LOAD FROM FPP IF ENABLED
4 CCLD4 (Z & N ACC SHFR; C & V ← 0)
5 CCLD5 (Z & N ACC SHFR; C ← AMX 15;
V ← V_{old} +(SHFR15 ⊻ AMX 15))
6 CCLD6 (N,C & V UNAFFECTED; Z ← Z· SHFR-0)
7 CCLD7 (Z,N, & V UNAFFECTED; C ← ALU CARRY)

FORK ENABLE
FEN [14-12]
0 NO FORK
1 FORK A
2 FORK B
4 FORK C

MICRO ADDRESS FIELD
UAD [07-00]
TO ADDRESS GATING

MISCELLANEOUS
MSC(T1) [29-27]
0 NO EFFECT
1 FP ATTN
2 NOT USED
3 SET CONF IF KERNEL MODE
4 SPL (SET PRIORITY LEVEL)
5 CONDITIONAL BUST
6 BRO STROBE
7 BUST (BUS START)

BUS DELAY
BSD(T1) [40-39]
0 NO PAUSE
1 INTR PAUSE
2
3 BUS PAUSE

BUS CONDITION
BSC(T1) [26-24]
0 DATI
1 SRC1 DATI
2 KERNAL DATI
3 SRC2 DATI
4 FC (CONTROLLED BY FPP)
5 DATO
6 DSOP1
7 DSOP2

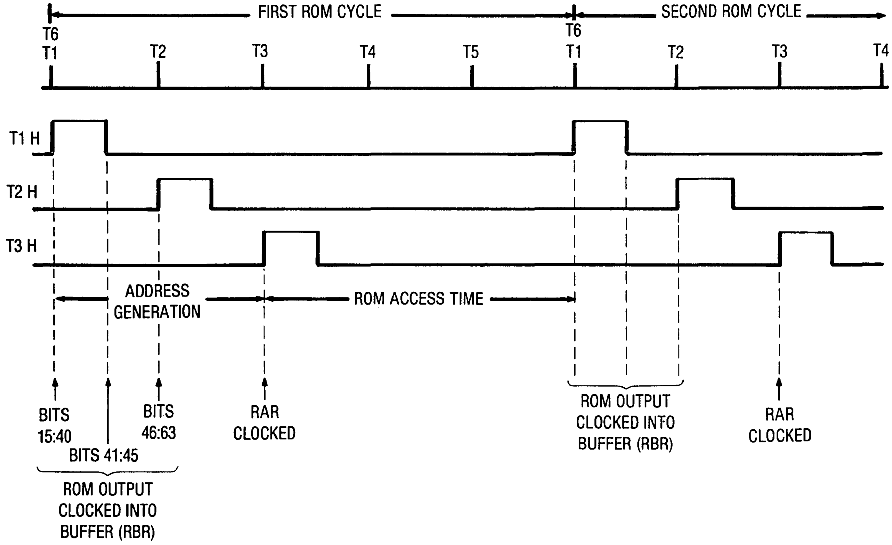
BUS CONTROL
BCT(T1) [32-30]
0 NO EFFECT
1 READ FPP DATA
2 CONSOLE ACKNOWLEDGE
3 CLEAR FLAGS
4 INIT IF KERNEL MODE
5 STACK REFERENCE
6 ACKNOWLEDGE
7 BEND (BUS END)

BRANCH ENABLE

BE,F [11-0]	UADR5 [+40]	UADR4 [+20]
0	—	—
1	DESTINATION MODE 3,5,7	SR = 1
2	CONDITION CODE Z	—(PWAF + INTR)
3	SC = 0	SC ≤ 0
4	—DIV SUB	CONDITION CODE N
5	—OBD (ODD BYTE DESTINATION)†	—DIV QUIT
6	BR14(0)	PS RESTORE
7	REGISTER DEPOSIT	—(BRO = - (T + CONF))
10	RIP + FP SYNC	—(FP REG = FP SYNC)
11	SC = 0	DRO (1)
12	CONF (CONSOLE FLAG)	—BRO
13	PF(0) = (SF + TF)	PF(0) = (SF + -TF)
14	—	—
15	—FJ/CLASS	—0/CLASS
16	DRO(1)	SR15(1)
17	RIP + FP SYNC	FP SYNC = FP REG WR

(BEF + 5) = OBD + CONDITIONAL FORK B
(BEF + 14) = CONSOLE BRANCHES
(BEF + 14) = CONDITIONAL FORK C
(BEF + 15) = FJ/CLASS + CONDITIONAL FORK B)

FIGURE 8.6



ROM TIMING

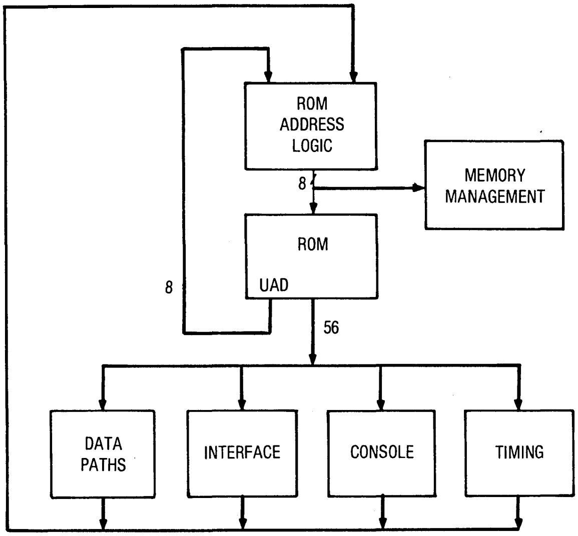


FIGURE 8.7

Microprogram Bit Usage

Bit Positions	Contents	Clocked At
	RACA	
67	FP start (UFPS)	T1
66	clear sync (UCLS)	T1
65—64	Floating Point Control (UFPC)	T1
63	bus register clock (UBRK)	T2
62	bus register multiplexer (UBRX)	T2
61—60	source register MUX (USRX)	T2
59—58	destination register MUX (UDRX)	T2
57	source register clock (USRK)	T2
56—55	destination register clock (UDRK)	T2
54—52	condition-code load (UCCL)	T2
51	program counter A CLK (UPCA)	T2
50—49	program counter B CLK (UPCB)	T2
48—47	shifter control (USHF)	T2
46	instruction register CLK (UIRK)	T2
	RACB	
45—44	pad write-enable (UPWE)	T1 +15 ns
43—41	scratchpad address (UPAD)	T1 +15 ns
40—39	bus delay (UBSD)	T1
38—37	bus address multiplexer (UBAX)	T1
36—35	internal bus (UIBS)	T1
	RACC	
34—33	shift counter (USHC)	T1
32—30	bus control (UBCT)	T1
29—27	miscellaneous control (UMSC)	T1
26—24	bus conditions (UBSC)	T1
23—22	A multiplexer (UAMX)	T1
21—20	B multiplexer (UBMX)	T1
19—18	constant multiplexers (UKMX)	T1
17—15	arithmetic logic unit cont (UALU)	T1
	RACD	
14	fork C enable (UCFEN)	not buffered
13	fork B enable (UBFEN)	not buffered
12	fork A enable (UAFEN)	not buffered
11—08	branch-enable (UBEF)	not buffered
07—00	microprogram address (UADR)	not buffered

SPECIAL uADDRESSES (as seen in the lights)

FP11

3	FP11C idle loop
76	FP11B idle loop
377	No FP11

CPU

170	Console halt loop
200	Abort entry ZAP 200
244	WAIT instr INTR PAUSE
352	Any other INTR PAUSE
275	WAIT loop (11,264)*
374	RESET loop
377	AFORK entry

*On KY11-R consoles you will see 1 of the 2 uAddresses as the KY11-R takes a snapshot of the RAR register. On the old console the lights are blurred together to make uAddress 275.

ABORTS, TRAPS, AND INTERRUPTS

A) GENERAL

ABORT = THE NON-COMPLETION OR INTERRUPTION OF A DATA CYCLE DUE TO AN ERROR. SERVICED IMMEDIATELY, PRIOR TO THE COMPLETION OF THE INSTRUCTION BEING EXECUTED.

TRAP = AN INTERRUPTION OF THE NORMAL PROGRAM FLOW BY INTERNAL MACHINE CONDITIONS. SERVICED AFTER COMPLETION OF THE INSTRUCTION BEING PROCESSED.

INTERRUPT = CAUSED BY CONDITIONS EXTERNAL TO THE MACHINE.

B) VECTORS

- ALL ABORTS, TRAPS, AND INTERRUPTS OBTAIN A VECTOR.
- FOR AN EXTERNAL INTERRUPT, THE VECTOR IS PROVIDED BY THE DEVICE CAUSING THE INTERRUPT.
- FOR ABORTS AND TRAPS, THE VECTOR IS READ FROM THE TRAP VECTOR LOGIC.
  - *THE VECTOR LOGIC IS CONTROLLED BY FUNCTIONS ON THE "TMC" MODULE AND BY FUNCTIONS ON THE "IRC" MODULE.
  - *IRC FUNCTIONS ARE TRAP INSTRUCTIONS WHICH DO NOTHING BUT GENERATE AN INTERRUPT (IOT, BPT, EMT, TRAP).
- VECTORS ARE GENERATED BY THE TRAP VECTOR LOGIC ON THE "DAP" MODULE
- THE INTERNAL VECTORS ARE:

4	=	(DEFAULT)	RED ZONE, UB TIMEOUT ODD ADD, ILLEGAL HALT
10	=		RESERVED INSTRUCTION
14	=		BPT
20	=		IOT
24	=		POWER FAIL
30	=		EMT
34	=		TRAP
114	=		PARITY ERROR
240	=		PIRQ
244	=		FP TRAP
250	=		KT TRAP

- ALL ABORTS GO TO MICRO-CODE ADDRESS 200 (ZAP)
- ALL TRAPS AND INTERRUPTS GO TO ADDRESS 240 (BRQ)
- THE FOLLOWING WILL CAUSE AN ABORT:
  - NXM
  - UB TIMEOUT
  - RED ZONE
  - KT ABORT
  - ODD ADDRESS
  - PARITY ABORT
    1) MAIN MEMORY PAR ERR ON WORD REQUESTED BY THE CPU.
    2) MAIN MEMORY TIMEOUT ON CPU CYCLES
    3) ADDRESS PAR ERR ON CP CYCLES
- SEE DIAGRAM THAT FOLLOWS. REMEMBER THAT WITH AN RDC CONSOLE, THE "IND ADRS ERR" AND "IND PAR ERR" ARE THE LOWER TWO BITS OF THE "T" COMMAND.

ABORTS/TRAPS/INTERRUPTS

	ABORTS	TRAPS	INTERRUPTS
MEM (114)	TOUT * CP APE * CP MPE * CP REQ WD	APE * UB MPE * UB MPE * CP -REQ WD TPE * CP/UB FPE * CP/UB	TOUT * UB/MBC (Dev V) APE * UB/MBC (Dev V) MPE * MBC (Dev V) (These actually cause an interrupt at BRx)
Unibus	CP->UB * PB (114)		BR4->7 (Dev V)
MMU (250)	MMU NON-RES MMU PLE MMU RO * ACF=0,1	MMU ACF = 1 * RD MMU ACF = 4 * R/W MMU ACF = 5 * WRT	
PWR (24)	POWER UP	POWER FAIL	
CPU (4)	RED ZONE UNIBUS TIMEOUT NXM ODD ADRS ERROR	YELLOW ZONE ILLEGAL HALT PIRQ 1->7 (240) TBIT PS<14> (14) CNSL FLG (No Vect)	
Instrn's		TRAP (34) EMT (30) BPT (14) IOT (20) RSVD (10)	
FP (244)		FP EXCEPTION	

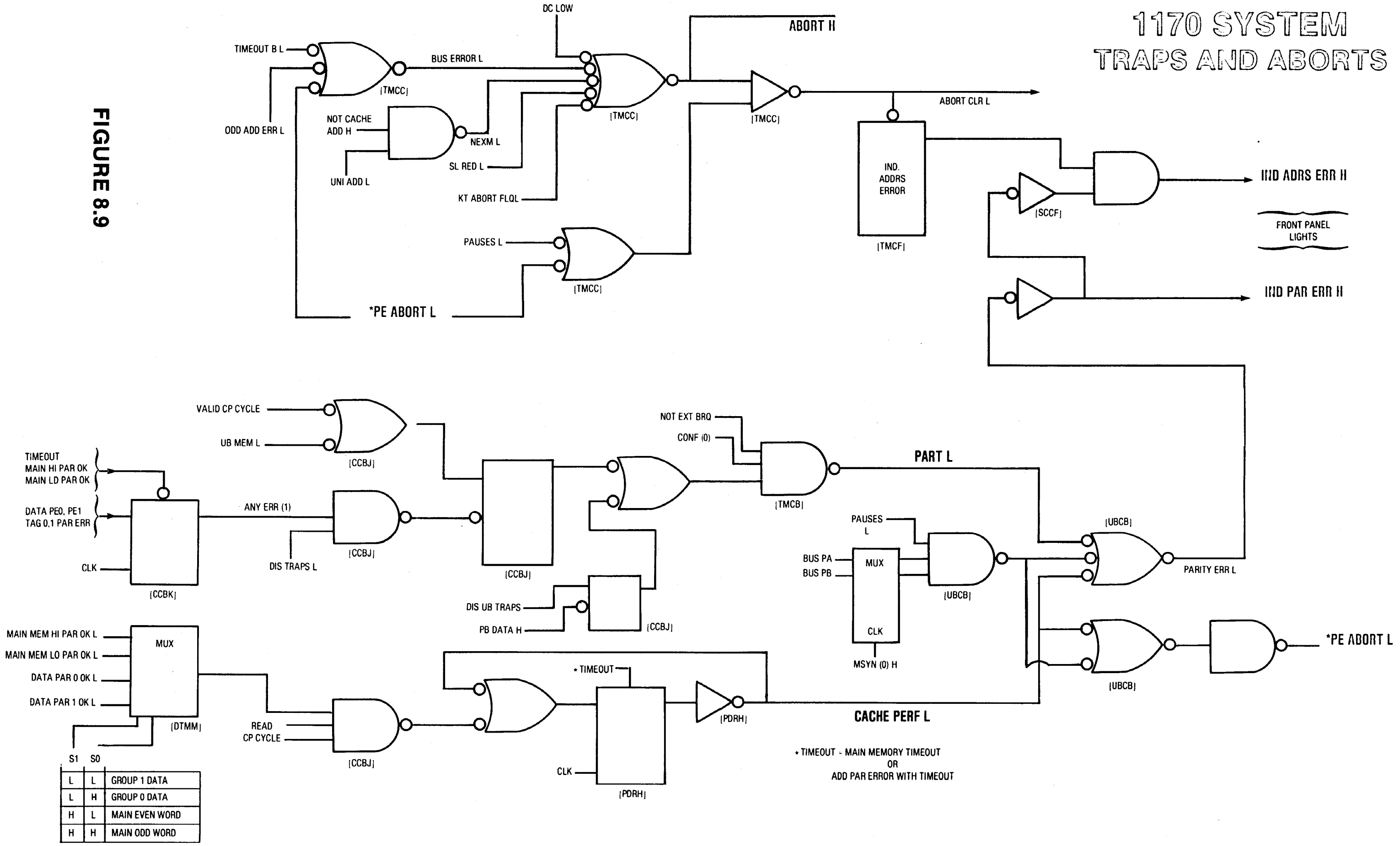
Note 1: RESET and SPL instructions execute NOP's if in USER or SUPER mode.

TERMS

TOUT	Main Memory Timeout
CP	CPU Cycle
APE	Main Memory Address Parity Error
MPE	Main Memory Data Parity Error
REQ WD	Requested Word
TPE	Cache TAG Address Parity Error
FPE	Cache Fast Data Memory Parity Error
UB	Unibus Cycle
MBC	Massbus (RH70) Cycle
PB	Unibus PB bit (Unibus Parity Error)
MMU	Memory Management
ACF	Access Control Field
NON-RES	Non-Resident Abort
PLE	Page Length Abort
RO	Read Only Abort

1170 SYSTEM TRAPS AND ABORTS

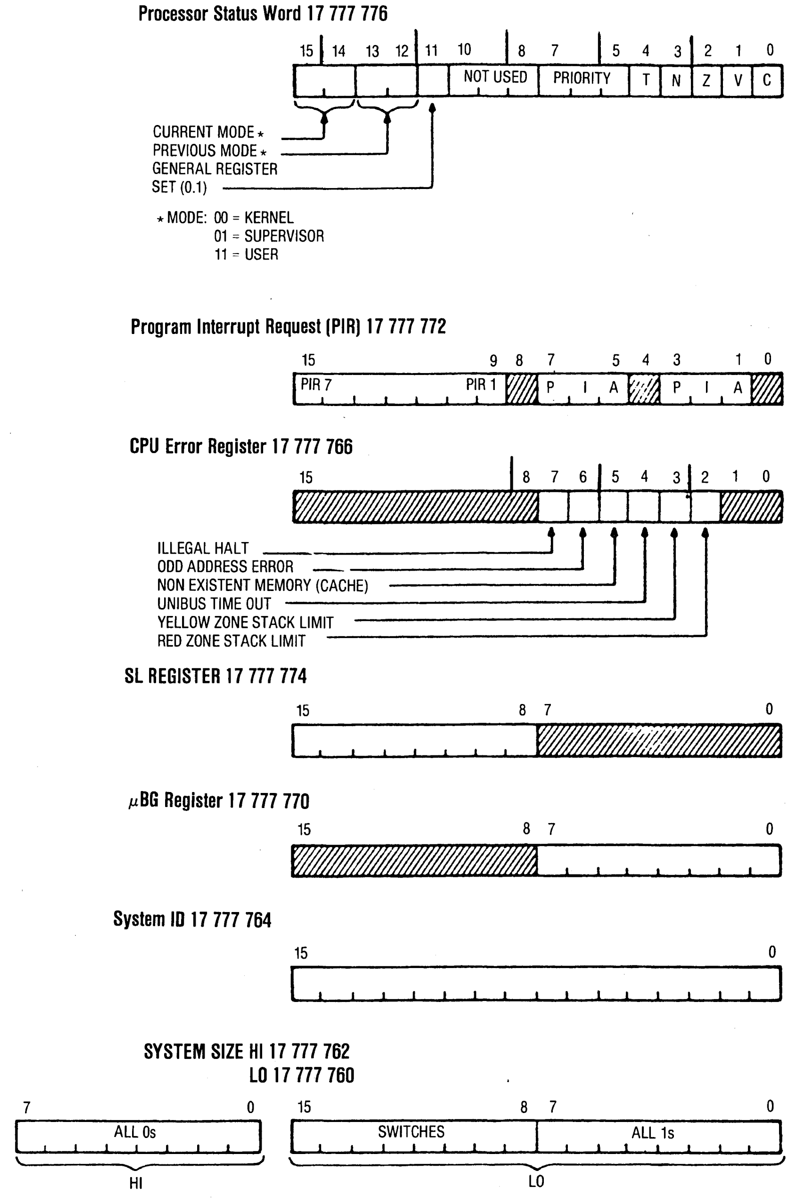
FIGURE 8.9



S1	S0	
L	L	GROUP 1 DATA
L	H	GROUP 0 DATA
H	L	MAIN EVEN WORD
H	H	MAIN ODD WORD

* TIMEOUT = MAIN MEMORY TIMEOUT
OR
ADD PAR ERROR WITH TIMEOUT

CPU REGISTERS



HI = RESERVED FOR FUTURE
SWITCHES = MAXIMUM AVAILABLE MEMORY AND REPRESENTS ADDRESS BITS 21:14

11-4552

FIGURE 8.10

CHART OF MEMORY SIZE,
$60 REGISTER CONTENTS AND SWITCH SETTINGS

Kwords	Last Adrs	$60	M8140 SW's
32K	00177776	001777	00000101
64K	00377776	003777	00010101
96K	00577776	005777	10000101
128K	00777776	007777	10010101
160K	01177776	011777	00001101
192K	01377776	013777	00011101
224K	01577776	015777	10001101
256K	01777776	017777	10011101
288K	02177776	021777	00000111
320K	02377776	023777	00010111
352K	02577776	025777	10000111
384K	02777776	027777	10010111
416K	03177776	031777	00001111
448K	03377776	033777	00011111
480K	03577776	035777	10001111
512K	03777776	037777	10011111
544K	04177776	041777	00100101
576K	04377776	043777	00110101
608K	04577776	045777	10100101
640K	04777776	047777	10110101
672K	05177776	051777	00101101
704K	05377776	053777	00111101
736K	05577776	055777	10101101
768K	05777776	057777	10111101
800K	06177776	061777	00100111
832K	06377776	063777	00110111
864K	06577776	065777	10100111
896K	06777776	067777	10110111
928K	07177776	071777	00101111
960K	07377776	073777	00111111
992K	07577776	075777	10101111
1024K	07777776	077777	10111111

CHART OF MEMORY SIZE, $60 REGISTER CONTENTS AND SWITCH SETTINGS

Kwords	Last Adrs	$60	M8140 SW's	
1056K	10177776	101777	01000101	
1088K	10377776	103777	01010101	
1120K	10577776	105777	11000101	
1152K	10777776	107777	11010101	
1184K	11177776	111777	01001101	
1216K	11377776	113777	01011101	
1284K	11577776	115777	11001101	
1280K	11777776	117777	11011101	
1312K	12177776	121777	01000111	
1344K	12377776	123777	01010111	
1376K	12577776	125777	11000111	
1408K	12777776	127777	11010111	
1440K	13177776	131777	01001111	
1472K	13377776	133777	01011111	
1504K	13577776	135777	11001111	
1536K	13777776	137777	11011111	
1568K	14177776	141777	01100101	
1600K	14377776	143777	01110101	
1632K	14577776	145777	11100101	
1664K	14777776	147777	11110101	
1696K	15177776	151777	01101101	
1728K	15377776	153777	01111101	
1760K	15577776	155777	11101101	
1792K	15777776	157777	11111101	
1824K	16177776	161777	01100111	
1856K	16377776	163777	01110111	
1888K	16577776	165777	11100111	
1920K	16777776	167777	11110111	
1952K	17177776	171777	01101111	I/O Page
1984K	17377776	173777	01111111	↓
2016K	17577776	175777	11101111	
2048K	17777776	177777	11111111	

CPU ERROR REGISTER

BIT **DESCRIPTION**

02 "RED ZONE STACK LIMIT"

- RED ZONE IS DETECTED BY THE CPU AND OCCURS WHEN IN KERNAL MODE, AND R6 IS BEING USED, AND THE R6 ADDRESS IS IN RED ZONE (000-336).
- RED ZONE ABORTS THE DATA CYCLE AND TRAPS TO 4.
- THE FOLLOWING OCCURS:
  1) SET STACK POINTER TO 4.
  2) PUT OLD PC & PSW IN LOCATION 0 & 2.
  3) TAKE NEW PC & PSW FROM LOC 4 & 6.
- MOST PROBABLE CAUSE:
  1) A BURST OF INTERRUPTS FROM AN EXTERNAL DEVICE.
  2) STACK LOGIC: M8135, M8134, M8140.
  3) A BURST OF ERRORS

03 "YELLOW ZONE"

- YELLOW ZONE ERROR TRAPS TO 4 (NO ABORT) IF IN KERNEL MODE, AND USING R6, AND THE R6 ADDRESS = 340-376.

04 "UNIBUS TIMEOUT"

- DETECTED BY THE UNIBUS CONTROL LOGIC (UBC MODULE) AFTER A UNIBUS CYCLE HAS BEEN STARTED AND THERE WAS NO RESPONSE TO MSYN WITHIN 10 MICRO-SECONDS.
- UB TIMEOUT ABORTS AND TRAPS TO 4.
- MOST PROBABLE CAUSE:
  1) BAD DEVICE ON UNIBUS.
  2) BAD UNIBUS.
  3) M8136.
  4) MMU LOGIC RELOCATED WRONG

05 "NON-EXISTENT MEMORY"

- NXM IS DETECTED IN THE MEMORY MANAGEMENT LOGIC, AFTER THE VIRTUAL ADDRESS IS FORMED INTO A PHYSICAL ADDRESS, AND THE PHYSICAL ADDRESS IS NOT DECODED AS A UNIBUS ADDRESS (I/O PAGE) OR AS A MEMORY ADDRESS (BELOW SYSTEM SIZE REG). NO DATA TRANSFER IS STARTED AND AN ABORT OCCURS FOLLOWED BY A TRAP TO 4.
- MOST PROBABLE CAUSE:
  1) SYSTEM SIZE REGISTER IS WRONG.
  2) MEMORY MANAGEMENT LOGIC.

06 "ODD ADDRESS ERROR"

- AN ODD ADDRESS IS ONLY ALLOWED IN A BYTE INSTRUCTION. AN ODD ADDRESS IS DETECTED IN THE CPU BEFORE THE VIRTUAL ADDRESS IS SENT TO MEMORY MANAGEMENT. AN ABORT OCCURS, FOLLOWED BY A TRAP TO 4.
- MOST PROBABLE CAUSE:
  1) MEMORY IS CORRUPTED.
  2) ODD ADDRESS LOGIC: M8135, M8132.
  3) SOFTWARE.

07 "ILLEGAL HALT"

- A USER WAS TRYING TO EXECUTE A HALT INSTRUCTION.
- A TRAP TO 4 OCCURS.

1170 PAUSE CYCLES

A) GENERAL

DATA TRANSFER PAUSES:

ALL CPU DATA TRANSFERS REQUIRE TWO ROM CYCLES FOR PROPER OPERATION: A "BUST" ROM CYCLE AND A "PAUSE" ROM CYCLE. IN THE "BUST" CYCLE, THE VIRTUAL ADDRESS IS SENT TO MEMORY MANAGEMENT SO IT CAN BE CONVERTED TO A PHYSICAL ADDRESS. THE "BUST" CYCLE IS ALSO USED TO INITIALIZE CACHE. THE "PAUSE" CYCLE DOES THE ACTUAL DATA TRANSFER. IN THIS CYCLE THE CPU RING COUNTER STOPS AND IS RESTARTED WHEN THE DATA TRANSFER IS COMPLETED. "ROM 40 (UBSD01)" ENABLES A DATA TRANSFER PAUSE CYCLE. THE DECODE OF THE PHYSICAL ADDRESS BY MEMORY MANAGEMENT (DONE IN THE "BUST" CYCLE), WILL DETERMINE WHAT TYPE OF DATA PAUSE WILL TAKE PLACE. THE TYPES ARE: CACHE, UNIBUS, AND INTERNAL DATA PATH.

EXTERNAL INTERRUPT PAUSE:

AN EXTERNAL INTERRUPT PAUSE IS ENABLED BY "ROM 39 (UBSD00)" AND ONLY OCCURS IN MICRO-CODE LOCATIONS 352 AND 244. SINCE NO ADDRESS IS NEEDED, THERE IS NO "BUST" CYCLE PRECEEDING IT. IN THIS CYCLE, THE CPU RING COUNTER IS STOPPED AND IS RESTARTED WHEN THE DEVICE VECTOR IS AVAILABLE TO THE CPU.

B) PAUSE DESCRIPTION

CACHE PAUSE:

SEE FIGURE 8.11.

M8139: DURING THE "BUST" CYCLE, MEMORY MANAGEMENT DECODED A CACHE ADDRESS ("CACHE ADD H"). THE NEXT ROM CYCLE IS A PAUSE CYCLE AND "ROM 40" IS ASSERTED. THE ASSERTION OF THE ABOVE TWO SIGNALS STOPS THE RING COUNTER AT T5.

M8135: WITH "ROM 40" ASSERTED (UBSD01 H), AND "CACHE ADD H" ASSERTED, "CONTROL OK H" IS SENT TO CACHE.

M8142: IN THE "BUST" CYCLE, CACHE RECEIVED "BUST H" FROM THE M8123 AND STARTED IT'S TIMING. WHEN CACHE NEXT RECEIVES "CONTROL OK", IT PERFORMS A DATA CYCLE. IF THE DATA CYCLE WAS COMPLETED WITHOUT ERROR, CACHE GENERATES "MEM SYNC H".

M8139: "MEM SYNC H" WILL ENABLE THE RING COUNTER TO CONTINUE.

UNIBUS PAUSE:

SEE FIGURE 8.12.

M8139: DURING THE "BUST" CYCLE MEMORY MANAGEMENT DECODED A UNIBUS ADDRESS WHICH WAS NOT AN INTERNAL REGISTER ADDRESS. IN THE "PAUSE" CYCLE, "ROM 40" IS ASSERTED AND STOPS THE RING COUNTER AT T2.

M8136: WITH "UNIBUS ADD H" ASSERTED, THE UBC MODULE STARTS A UNIBUS DATA TRANSFER BY ASSERTING "MSYN L" ON THE UNIBUS. THE DEVICE RESPONDS WITH "SSYN L" AND "TIG RESTART H" IS ASSERTED.

M8139: "TIG RESTART H" WILL ENABLE THE RING COUNTER TO CONTINUE.

INTERRUPT PAUSE CYCLE:

SEE FIGURE 8.13.

M8135: IN BETWEEN INSTRUCTIONS, THE CPU WILL DO A BRQ STROBE TO SEE IF ANY INTERRUPTS ARE PENDING. IF A BR4,5,6 OR 7 IS ASSERTED WHEN THE BRQ STROBE OCCURS, THEN HONOR BR4,5,6, OR 7 WILL BE ASSERTED OUT OF THE ARBATRATOR AND THE CPU WILL GO TO THE BREAK REQUEST MICRO-CODE SERVICE ROUTINE. AT MICRO-CODE LOCATIONS 352 OR 244, "ROM 39 (UBSD00)" WILL BE ASSERTED AND WILL ASSERT "INTR PAUSE H".

M8136: WITH "INTR PAUSE H" AND "HONOR BR4,5,6, OR 7" ASSERTED, "BG4,5,6, OR 7" WILL BE GENERATED ON THE UNIBUS. AT THE SAME TIME, THE RING COUNTER WILL STOP AT T2 SINCE "ROM 40" IS NEGATED, "ROM 39 (UBSD00)" IS ASSERTED, AND "EXT BRQ H" IS ASSERTED. WHEN THE INTERRUPT DIALOGUE ON THE UNIBUS IS COMPLETED, THE DEVICE WILL ASSERT "INTR L" WHICH WILL ASSERT "TIG RESTART H".

M8139: "TIG RESTART H" WILL ENABLE THE RING COUNTER TO CONTINUE.

FOR MORE DETAILS, SEE THE TROUBLESHOOTING CHAPTER.

INTERNAL DATA PAUSE:

WHEN MEMORY MANAGEMENT DECODES AN INTERNAL REGISTER ADDRESS, THE RING COUNTER IS NOT STOPPED, BUT ONLY DELAYED FOR 90 NANO-SEC BETWEEN T2 AND T3. THIS PROVIDES ENOUGH TIME FOR THE INTERNAL REGISTERS TO PUT THERE DATA ON THE INTERNAL DATA BUS.

WHICH PAUSE WAS IT?

Interrupt PAUSE	— uAddress =	352 or 244
Unibus PAUSE	— uAddress =	PAUSE ROM STATE uADDRESS
Cache PAUSE	— uAddress =	1 uADDRESS past the PAUSE ROM STATE uADDRESS.

Note: This will occur because the ROM address registers are clocked on the rising edge of T3 clock pulse. Therefore if the clock hangs in T2, the uAddress lights/register will indicate the current ROM state (the PAUSE ROM state) since it was not updated. If the clock hangs in T5, the uAddress lights/register will indicate the ROM state following the PAUSE ROM state since it was updated at T3.

FOR MORE DETAILS ON PAUSE CYCLES, SEE THE TROUBLESHOOTING CHAPTER.

PAUSE CONDITIONS

CACHE PAUSE CYCLE

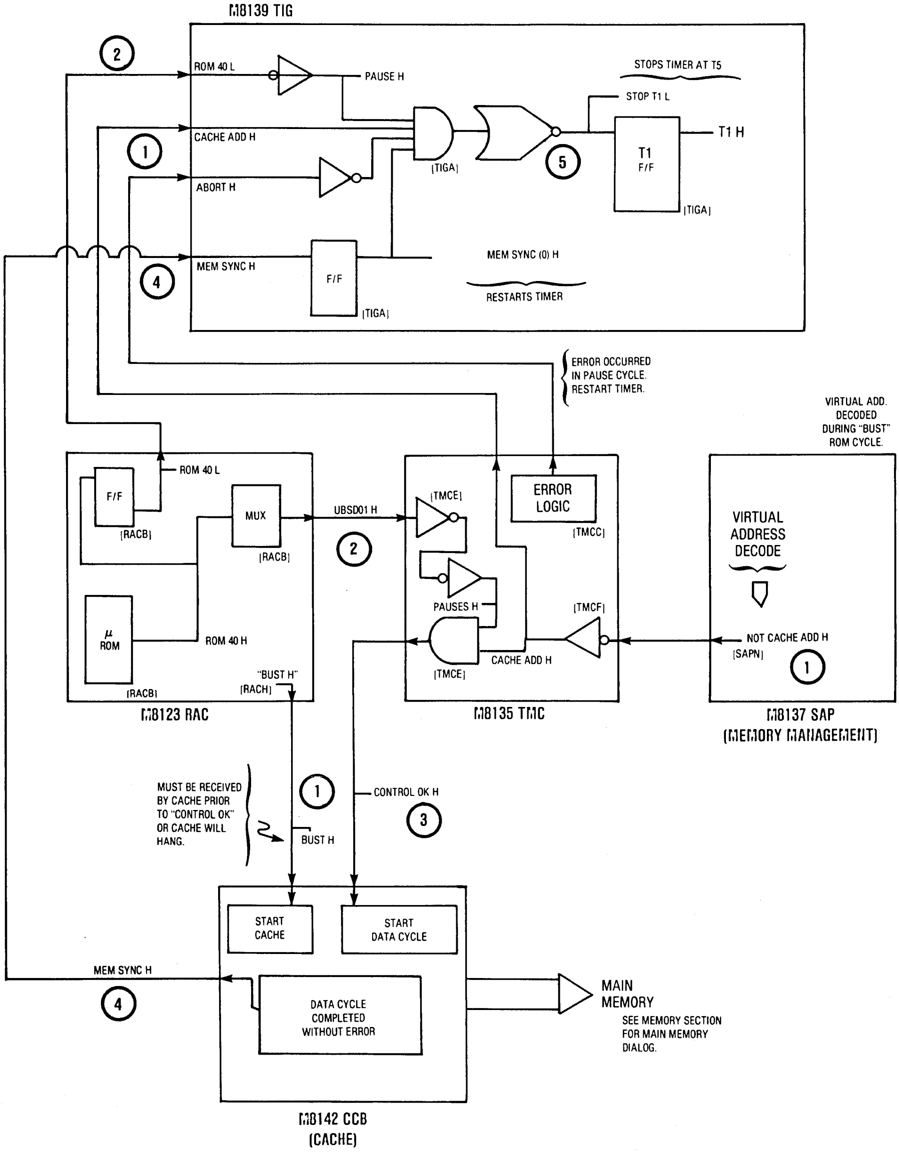


FIGURE 8.11

UNIBUS PAUSE CYCLE

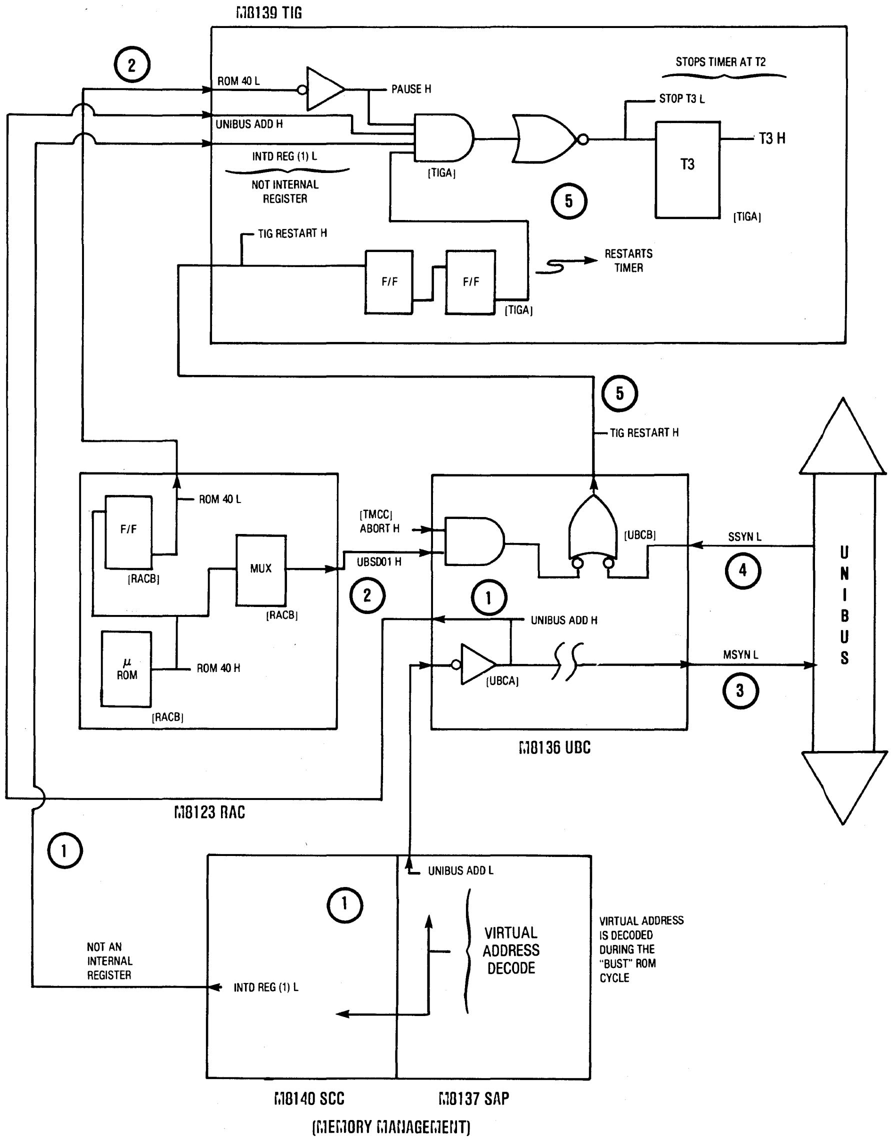


FIGURE 8.12

INTERRUPT PAUSE CYCLE

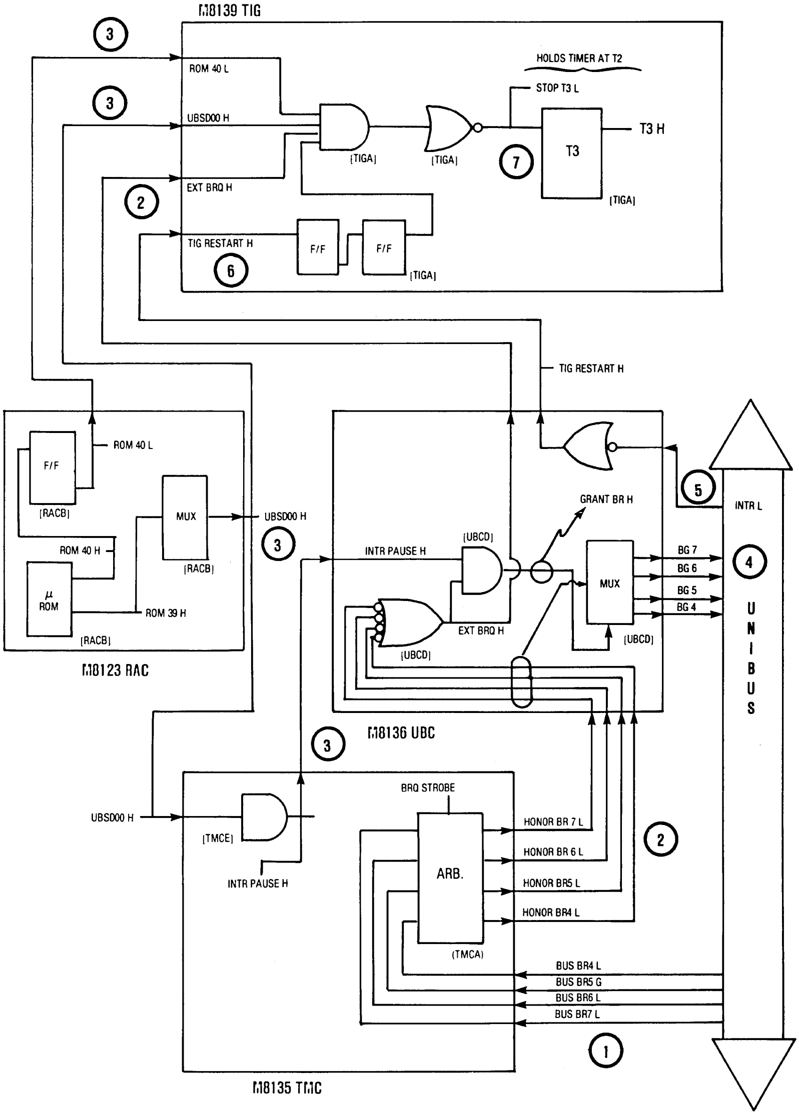


FIGURE 8.13

FIGURE 8.14

CPU TO MEMORY WRITE

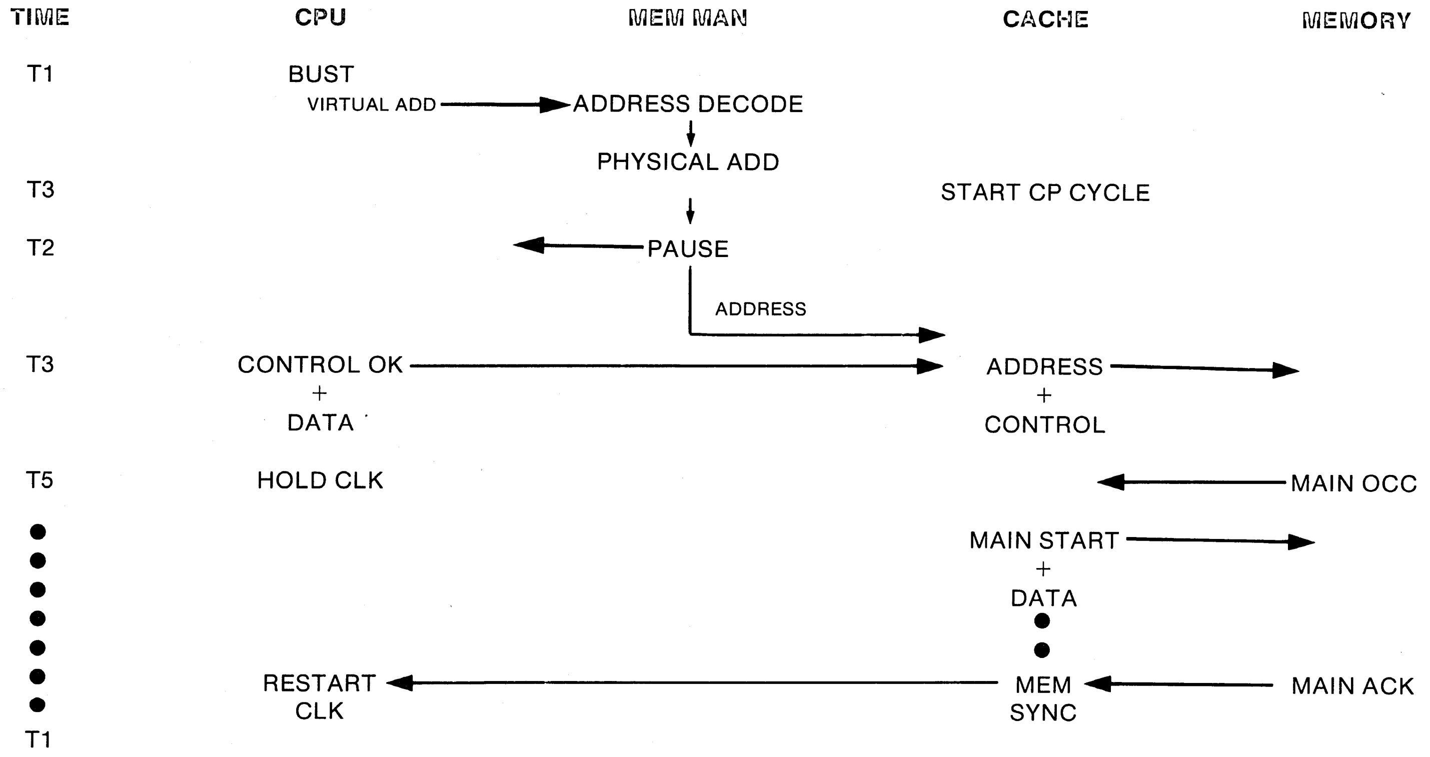


FIGURE 8.15

CPU TO MEMORY READ

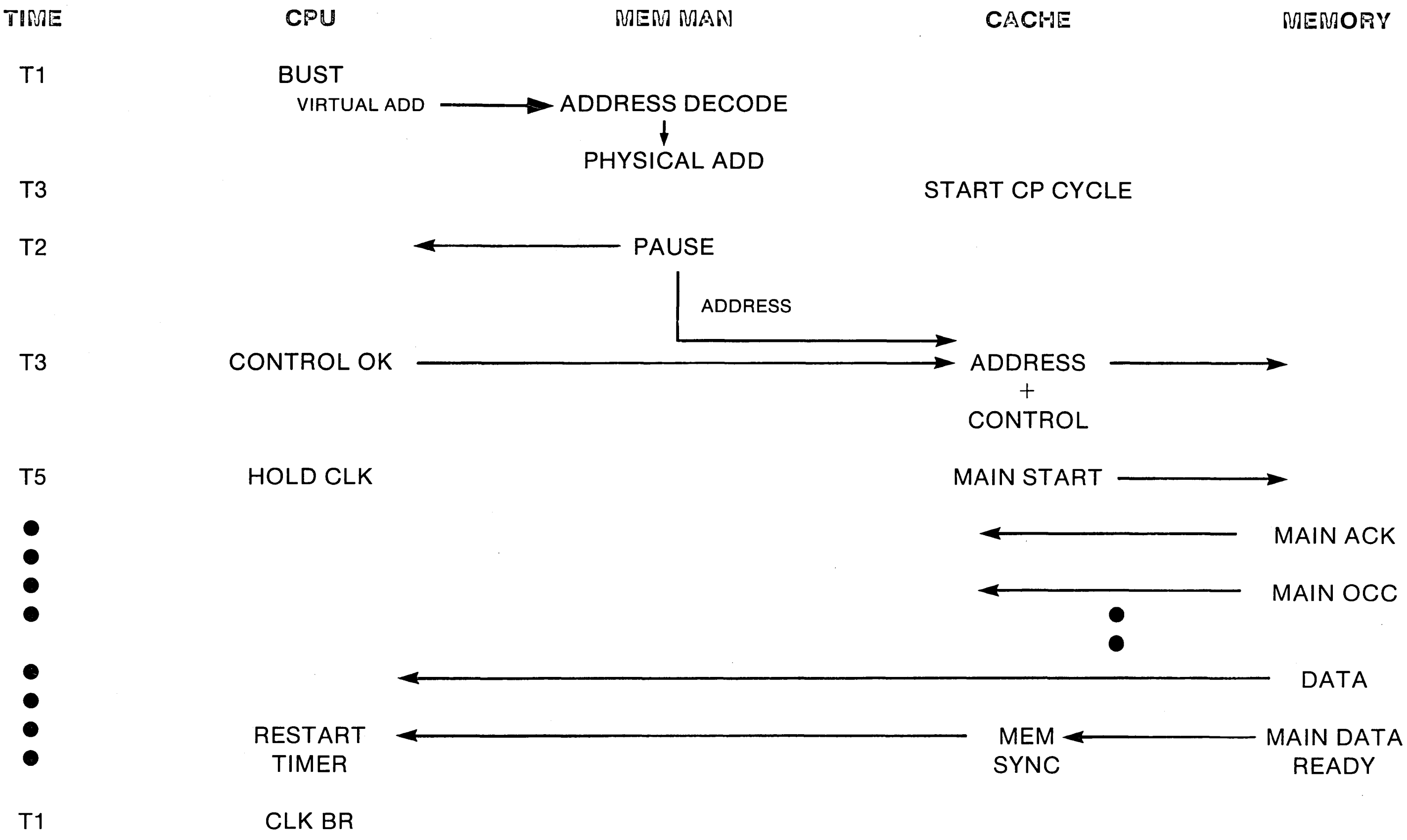
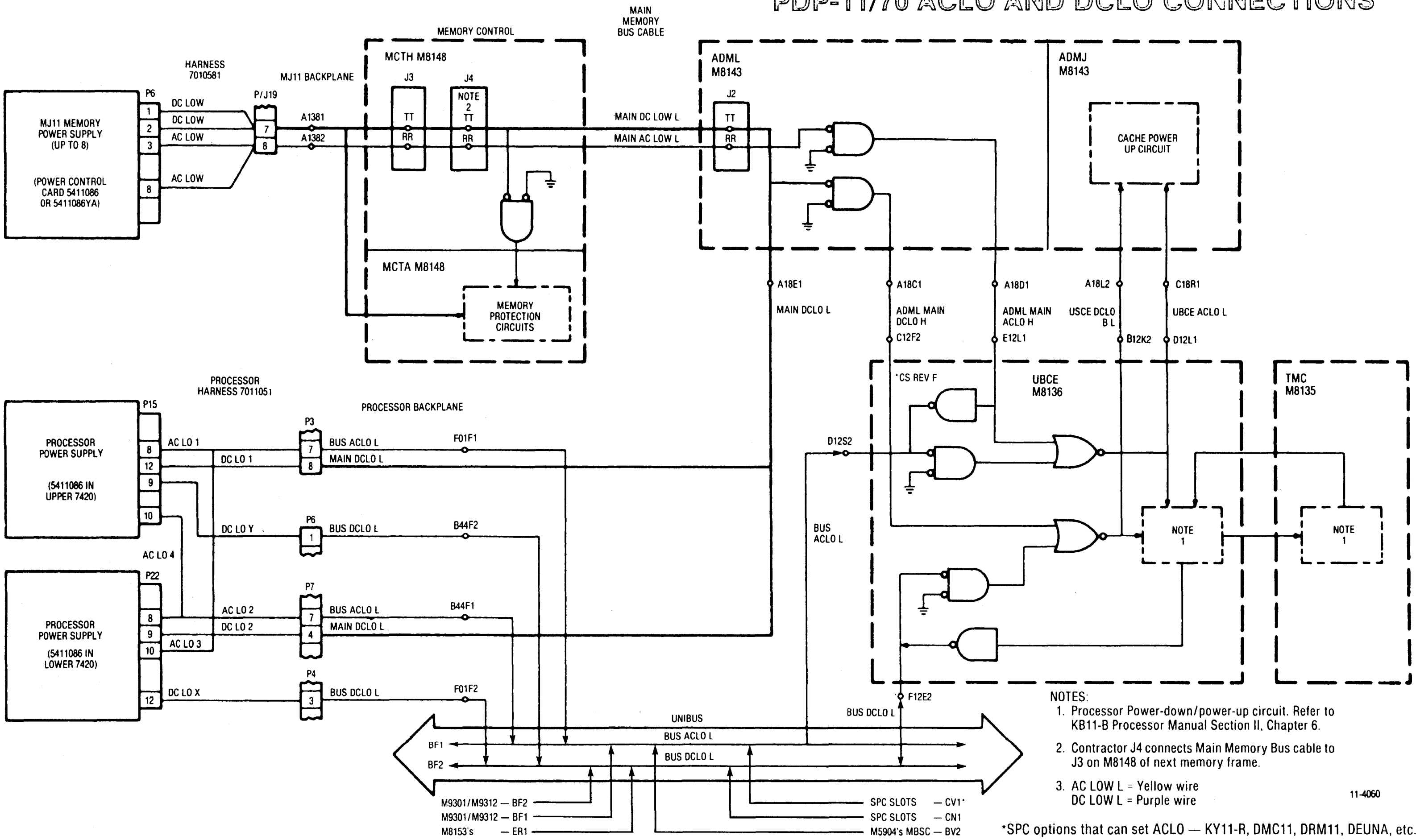


FIGURE 8.16

PDP-11/70 ACLO AND DCLO CONNECTIONS



NOTES:

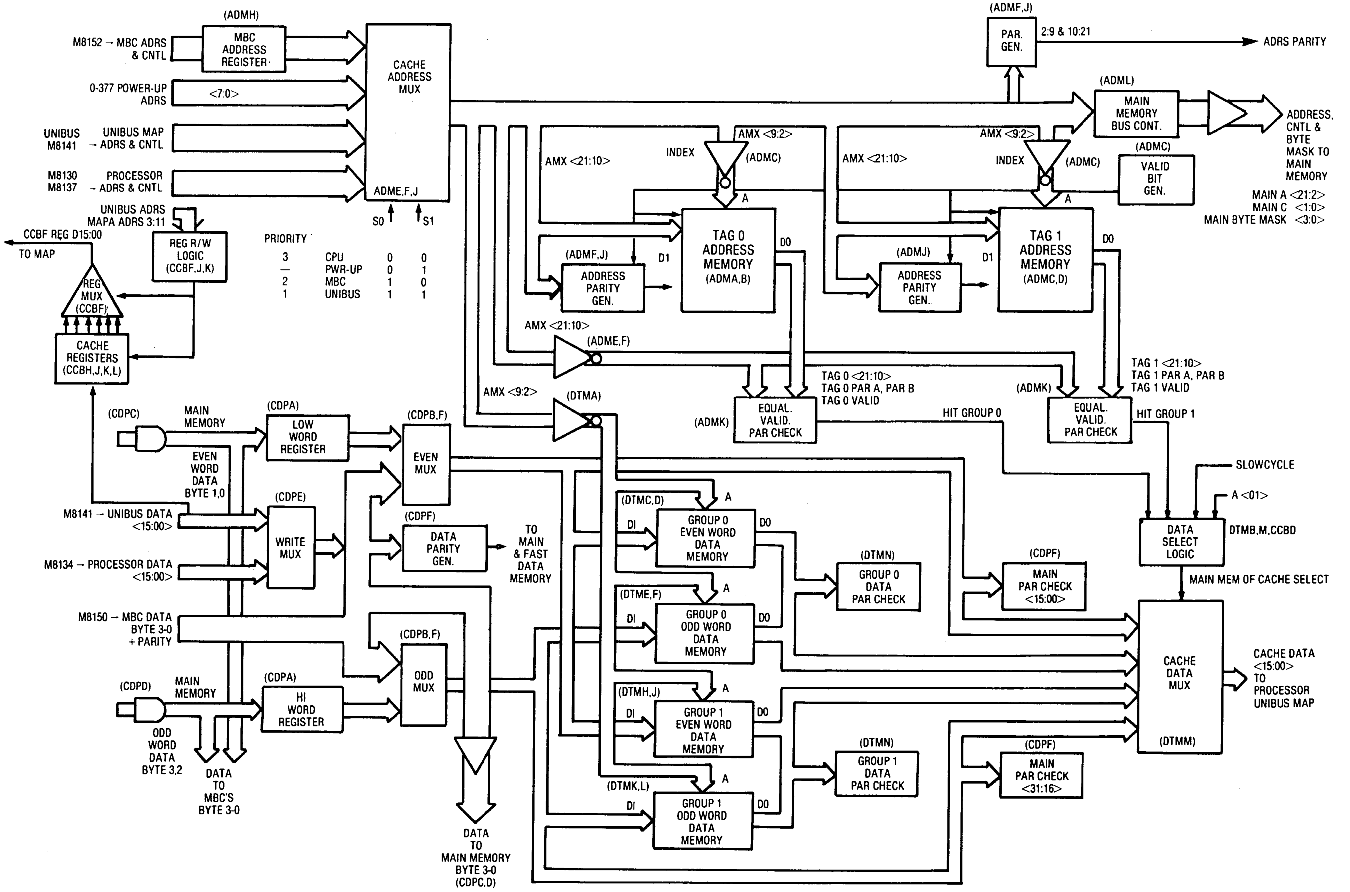
1. Processor Power-down/power-up circuit. Refer to KB11-B Processor Manual Section II, Chapter 6.
2. Contractor J4 connects Main Memory Bus cable to J3 on M8148 of next memory frame.
3. AC LOW L = Yellow wire
   DC LOW L = Purple wire

*SPC options that can set ACLO — KY11-R, DMC11, DRM11, DEUNA, etc.

CHAPTER 9

CACHE

CONTENT **PAGE #**



PDP 11/70 CACHE MEMORY

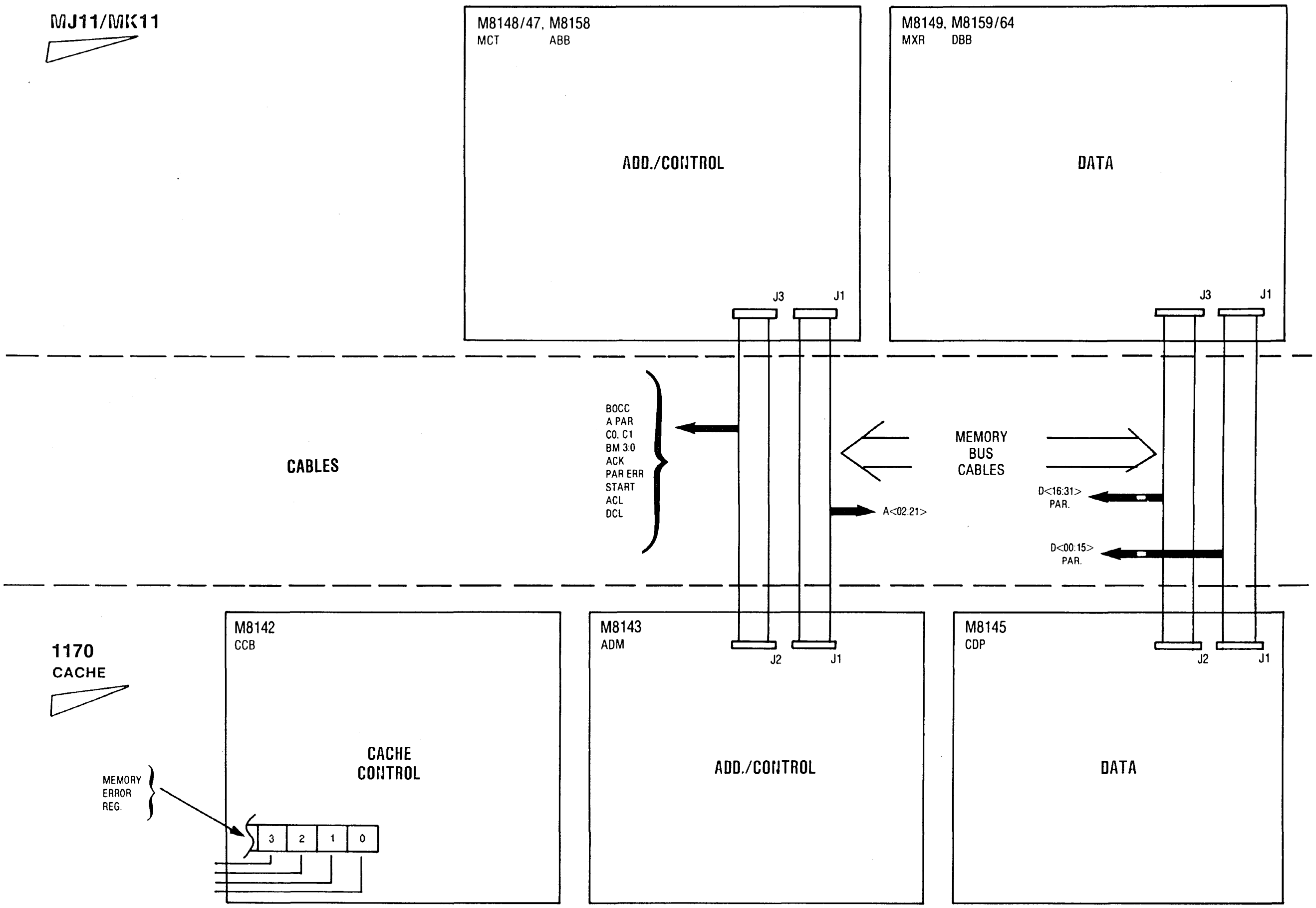
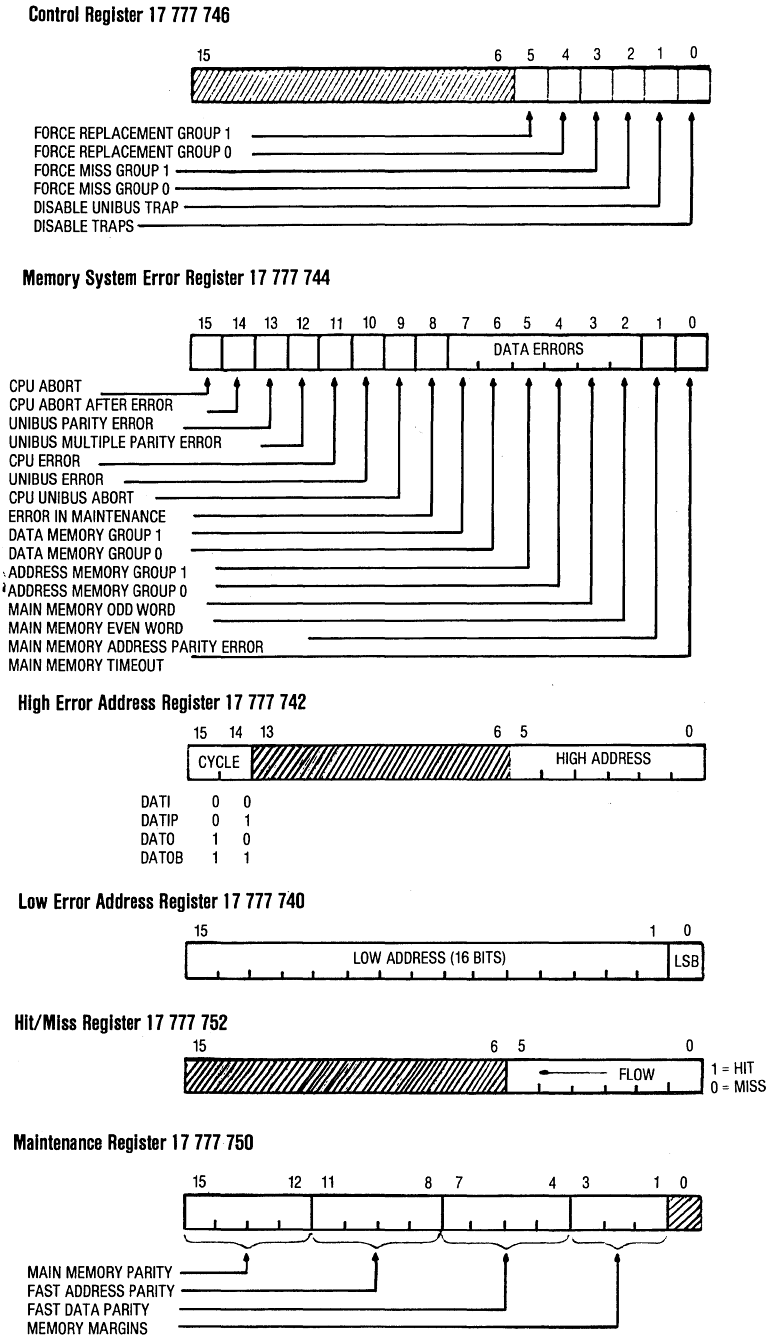

MJ11/MK11
M8148/47, M8158
MCT ABB
ADD./CONTROL
M8149, M8159/64
MXR DBB
DATA
J3
J1
J3
J1
CABLES
BOCC
A PAR
C0, C1
BM 3:0
ACK
PAR ERR
START
ACL
DCL
MEMORY
BUS
CABLES
A<02:21>
D<16:31>
PAR.
D<00:15>
PAR.
1170
CACHE
M8142
CCB
CACHE
CONTROL
MEMORY
ERROR
REG.
3
2
1
0
M8143
ADM
ADD./CONTROL
J2
J1
M8145
CDP
DATA
J2
J1

FIGURE 9.1

CACHE REGISTERS



Control Register 17 777 746

Memory System Error Register 17 777 744

High Error Address Register 17 777 742

DATI	0	0
DATIP	0	1
DATO	1	0
DATOB	1	1

Low Error Address Register 17 777 740

Hit/Miss Register 17 777 752

Maintenance Register 17 777 750

FIGURE 9.2

11-4551

MEMORY SYSTEM ERROR REGISTER

Bit	Name	Function
15	CPU Abort	Set if an error occurs which causes the Cache to abort a processor cycle.
14	CPU Abort After Error	Set if an abort occurs with the Error Address Register locked by a previous error.
13	Unibus Parity Error	Set if an error occurs which results in the Unibus Map asserting the parity error signal on the Unibus.
12	Unibus Multiple Parity Error	Set if an error occurs which causes the parity error signal to be asserted on the Unibus with the Error Address Register locked by a previous error.
11	CPU Error	Set if any memory error occurs during a Cache cycle from the processor.
10	Unibus Error	Set if any memory error occurs during a Cache cycle from the Unibus.
9	CPU Unibus Abort	Set if the processor traps to vector 114 because of a Unibus parity error on a DATI or DATIP cycle by the processor on the Unibus.
8	Error in Maintenance	Set if an error occurs when any bit in the Maintenance Register is set. The Maintenance Register will then be cleared.
7—6	Data Memory	These bits are set if a parity error is detected in the Fast Data Memory in the Cache. Bit 7 is set if there is an error in Group 1, bit 6 for Group 0.
5—4	Address Memory	These bits are set if a parity error is detected in the Address Memory in the Cache. Bit 5 is set if there is an error in Group 1, bit 4 for Group 0.
3—2	Main Memory	These bits are set if a parity error is detected on data from Main Memory. Bit 3 is set if there is an error in either byte of the odd word, bit 2 for the even word. An abort occurs if the error is in the word needed by a CPU reference. A trap occurs if the error is in the other word, or if it is a Unibus reference.
1	Main Address Parity Error	Set if there is a parity error detected on the address and control lines on the Main Memory Bus.
0	Main Memory Time-out	Set if there is no response from Main Memory. For CPU cycles, this error causes an abort. When a Unibus device requests a non-existent location, this bit will not set.

CACHE REGISTERS

A) **HIGH AND LOW ERROR ADDRESS REGISTERS**

THESE TWO REGISTERS ARE LOCKED IN WHEN ANY ERROR IS SET IN THE MEMORY ERROR REGISTER WHILE DOING A CPU OR A UNIBUS MAP MEMORY REFERENCE. THESE TWO REGISTERS CONTAIN THE 22 BIT PHYSICAL ADDRESS THAT CAUSED THE ERROR. THE REGISTERS

WILL STAY LOCKED IN UNTIL THE MEMORY ERROR REGISTER IS CLEARED. IF AN ERROR IS SET WHILE REFERENCING AN MJ11, THE LOW BITS OF THE ERROR ADDRESS REGISTER WILL POINT TO THE CORRECT SIDE (LOW OR HI) ONLY IF "CPU ABORT" IS SET IN THE MEMORY ERROR REGISTER. IF "CPU ABORT" IS NOT SET, THEN THE ERROR IS IN THE OPPOSITE WORD THEN POINTED OUT BY THE LOW BITS OF THE ERROR ADDRESS REGISTER. THE ERROR ADDRESS REGISTER ALWAYS POINTS TO THE CORRECT SIDE WHEN AN MK11 IS INVOLVED, SINCE "CPU ABORT" WILL ALWAYS BE SET FOR ANY ERROR CAUSED BY AN MK11.

B) MEMORY SYSTEM ERROR REGISTER

BIT **DESCRIPTION**

00 MAIN MEMORY TIMEOUT:

MJ11—THE MEMORY CONTROLLER (M8147/48) DID NOT RESPOND WITH "ADD ACK" 4 MICRO-SECONDS AFTER CACHE SENT "MAIN START". — SEE FIGURE 9-3.

PROBABLE CAUSES:
1) MEMORY STARTING ADDRESS IS WRONG.
2) SYSTEM SIZE REGISTER IS TOO HIGH.
3) M8147/48.
4) CABLES BETWEEN M8143 AND M8147/48.
5) M8143 AND M8142.
6) MEMORY BUS TERMINATORS ON M8147/48.
7) CPU BACKPLANE.

MK11—MEMORY CONTROLLER 0 OR 1 (M8160) DID NOT RESPOND WITH "ADD AKN" 4 MICRO SECONDS AFTER "MAIN START" FROM CACHE WAS SENT. WHEN THE ADDRESS BUFFER MODULE (M8158) IN THE MK11 RECEIVES "MAIN START" FROM CACHE, IT SENDS "GO" TO THE SELECTED CONTROLLER. THE SELECTED CONTROLLER, AFTER RECEIVING THE "GO" SIGNAL, STARTS IT'S ARRAY TIMING. AFTER THE TIMING HAS STARTED, THE SELECTED CONTROLLER SENDS "ADD AKN" TO THE ADDRESS BUFFER MODULE (M8158), WHICH IN TURN SENDS IT TO CACHE. — SEE FIGURE 9-4.

PROBABLE CAUSES:
1) MEMORY START ADDRESS WRONG.
2) SYSTEM SIZE REGISTER TOO HIGH.
3) M8160 IN BOTH CONTROLLERS.
4) M8158.
5) CABLES BETWEEN M8158 AND M8143.
6) MEMORY BUS TERMINATORS ON M8158.
7) M8143 AND M8142.
8) CPU BACKPLANE/MK11 BACKPLANE.

01 ADDRESS PARITY ERROR:

ALL MEMORY CONTROLLERS CHECK FOR PARITY ON THE ADDRESS AND CONTROL LINES SENT BY CACHE. ALL THE CONTROLLERS THAT HAVE DETECTED THE BAD PARITY WILL TURN THE PARITY ERROR ON. — SEE FIGURE 9-5.

PROBABLE CAUSES:
1) M8147/48 (MJ11), M8158 (MK11).
2) CABLES.
3) MEMORY BUS TERMINATORS.
4) M8143 AND M8142.

02/03 MAIN MEMORY EVEN/ODD WORD PARITY ERROR:

MJ11—THE MJ11 CONTROLLER WILL READ TWO WORDS FROM TWO DIFFERENT STACKS (ODD AND EVEN). THEREFORE ONLY ONE OF THESE BITS WILL MOST LIKELY BE SET AT ANY ONE TIME. FIND OUT WHICH SIDE, ODD OR EVEN, HAS CAUSED THE ERROR. SEE HI AND LO ERROR ADDRESS REGISTER DESCRIPTION ON PREVIOUS PAGE. — SEE FIGURE 9-6 AND 9-7 FOR MOST PROBABLE CAUSES.

MK11—THE MK11 WILL PULL OUT TWO WORDS FROM ONE ARRAY. THEREFORE, IF AN ARRAY IS BAD, BOTH THESE BITS WILL BE SET. TO FIND OUT WHICH SIDE IS BAD, SEE BIT #9 OF THE MK11 CSR WORD #2. IF ONLY ONE OF THESE BITS ARE SET, THEN THE ARRAYS ARE NOT AT FAULT. — SEE FIGURE 9-8.

04/05 ADDRESS MEMORY GROUP 0/1 PARITY ERROR:

THE ADDRESS MEMORY ON THE M8143 MODULE DETECTED A PARITY ERROR. THE ADDRESS MEMORY AND THE PARITY GENERATING AND CHECKING CIRCUITS ARE ALL ON THE M8143.

PROBABLE CAUSES:
1) M8143.
2) M8142.
3) CPU BACKPLANE.

06/07 DATA MEMORY GROUP 0/1 PARITY ERROR:

THE DATA MEMORY ON THE M8144 HAS DETECTED A PARITY ERROR.

PROBABLE CAUSES:
1) M8144.
2) M8142.
3) CPU BACKPLANE.

NOTE: ABLE DH/DM's below rev H7 can cause FDM,TAG and/or MAIN MEMORY DATA PARITY errors.

MEMORY SYSTEM ERRORS

MSER BIT	CP CYCLE	UB CYCLE	MBC CYCLE
.......0........ Memory timeout (No ACK in 4us)	Sets 15,11,0. Aborts.	No bits set. TOUT prevent SSYN on MAP result is Dev NXM, interupt.	No bits set. Set RH70 CSR2<11> NXM + interrupt.
.......1........ Memory APE. (Mem sends APE + no ACK so TOUT).	Sets 15,11,1. Aborts cause of resulting TOUT. (no set bit 0).	Sets 10,1. Traps, Dev NXM + interrupts.	No bits set. Set RH70 CSR3<15> APE + interrupt.
......2,3....... Main Memory PE wanted word.	Sets 15,11,2 or 3. Aborts.	Sets 13,10,2 or 3. Unibus PB + Trap.	No bits set. Set RH CSR3<14,13> MPE + interrupts.
......2,3....... Main Memory PE unwanted word.	Sets 11,2 or 3. Traps.	Sets 10,2 or 3. Traps	No bits set. Set RH CSR3<14,13> MPE + interrupts.
......4,5....... TAG Parity error.	Sets 11,4 or 5. Force miss, Trap.	Sets 10,4 or 5. Force miss, Trap.	No bits set. Doesn't check TAG parity, use match to invalidate TAG.
......6,7....... FDM Parity error.	Sets 11,6 or 7. Force slowcycle + Trap.	Sets 10,6 or 7. Force slowcycle + Trap.	No bits set. FDM not used on MBC cycles.
.......8........ Error in Maint.	Set if any bit in MSER<7:0> and any CMR<15:1> set.	Set if any bit in MSER<7:1> and any CMR<15:1> set.	Never set.
.......9........ CPU -> UB Abort	Never set.	* See Note 1.	Never set.
.......10...... Unibus Error.	Never set.	Set if any bit in MSER<7:1> set.	Never set.
.......11....... CPU error.	Set if any bit in MSER<7:0> set.	Never set.	Never set.
.......12....... Multiple UB PE.	Never set.	Set if UB PE and any MSER<7:0> set, H/LEAR was locked.	Never set.
.......13....... UB Parity error.	Never set.	Set if bits 2 or 3 wanted word set.	Never set.
.......14....... CP Abort after error.	Set if PE Abort + any MSER<7:0> set, H/LEAR was locked.	Never set.	Never set.
.......15....... CP Abort.	Set if bits 2 or 3 wanted word set or bits 0 or 1 set.	Never set.	Never set.

Note 1: UB Parity Error, bit 9 sets if the PB bit is set on the Unibus on a CPU to Unibus transfer then Aborts. (ie: accessing MK11 CSR's)

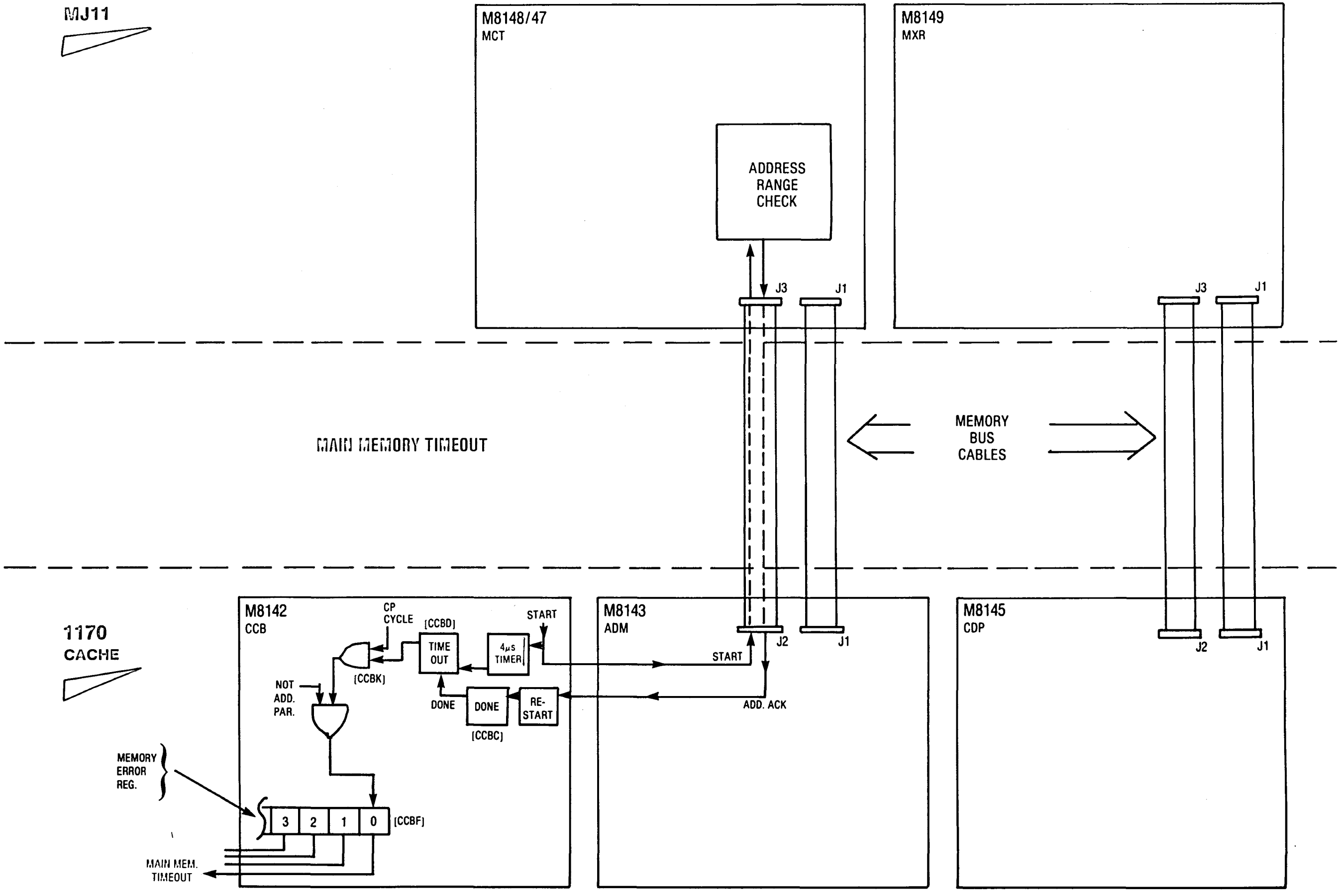

MJ11
M8148/47
MCT
ADDRESS RANGE CHECK
J3
J1
M8149
MXR
J3
J1
MAIN MEMORY TIMEOUT
MEMORY BUS CABLES
1170 CACHE
M8142
CCB
CP CYCLE
[CCBD]
START
TIME OUT
4μs TIMER
[CCBK]
NOT ADD. PAR.
DONE
DONE
RE-START
[CCBC]
MEMORY ERROR REG.
3
2
1
0
[CCBF]
MAIN MEM. TIMEOUT
M8143
ADM
J2
J1
START
ADD. ACK
M8145
CDP
J2
J1

FIGURE 9.3

FIGURE 9.4

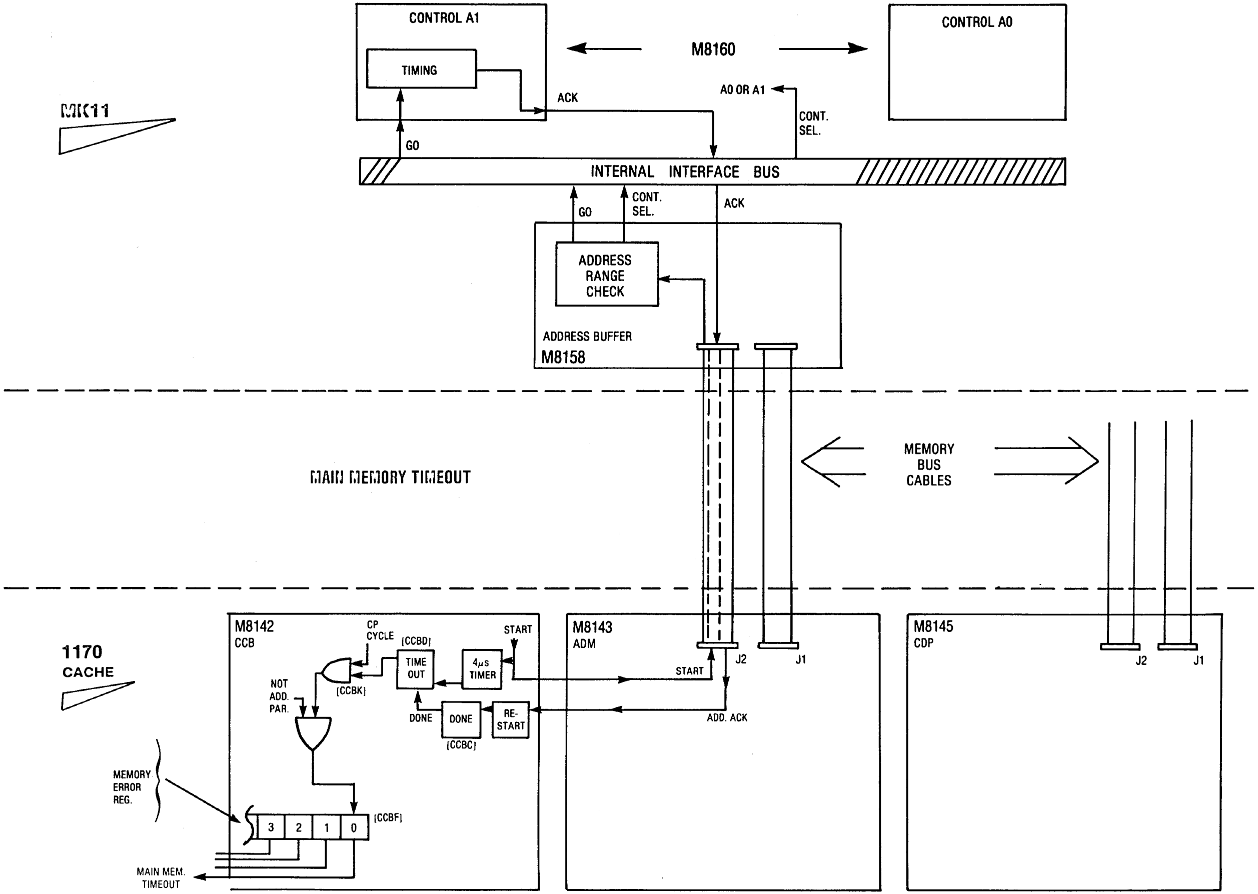
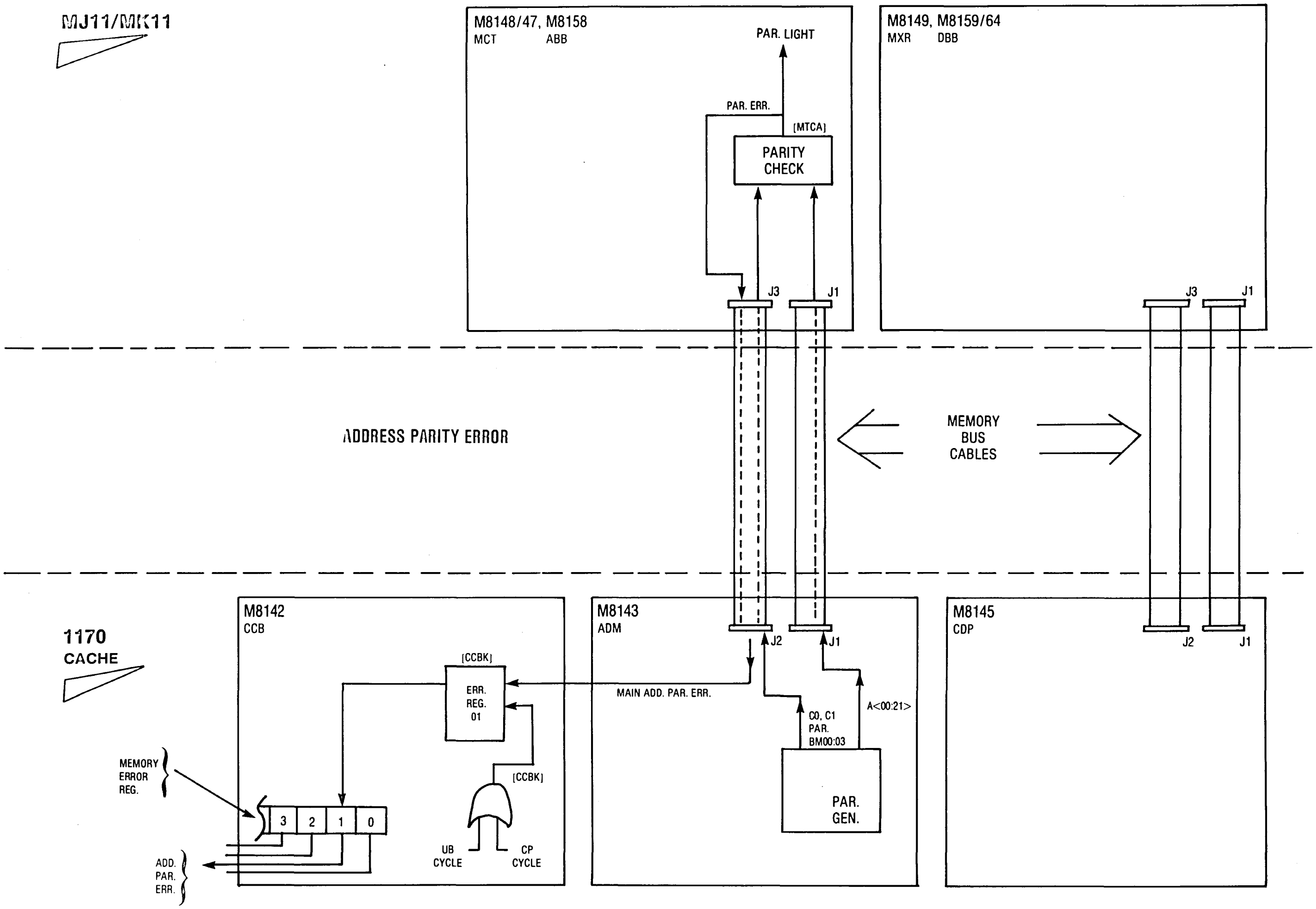
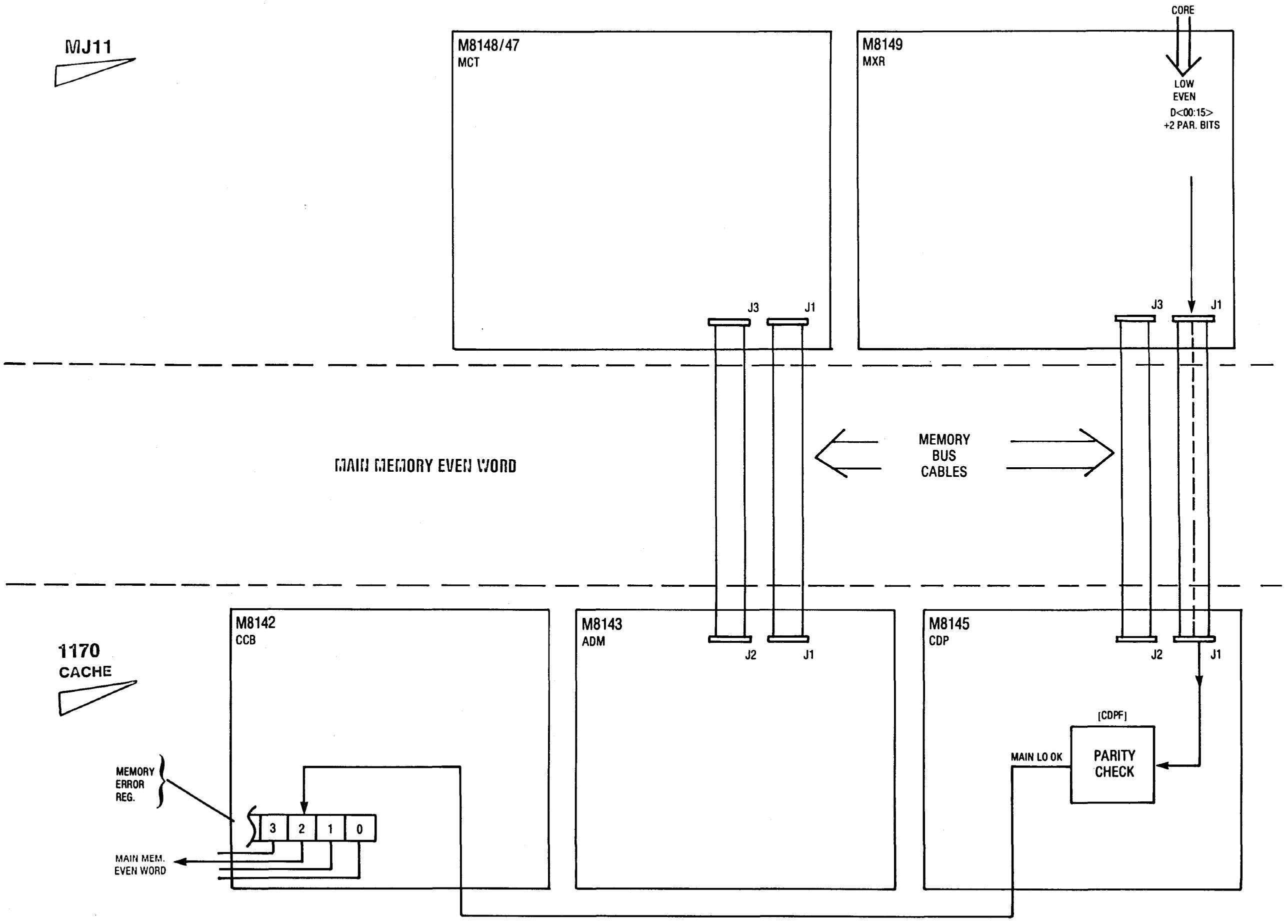

MK11
CONTROL A1
TIMING
ACK
M8160
CONTROL A0
A0 OR A1
CONT. SEL.
GO
INTERNAL INTERFACE BUS
GO
CONT. SEL.
ACK
ADDRESS RANGE CHECK
ADDRESS BUFFER
M8158
MAIN MEMORY TIMEOUT
MEMORY BUS CABLES
1170 CACHE
M8142
CCB
CP CYCLE
[CCBD]
START
TIME OUT
4μs TIMER
[CCBK]
NOT ADD. PAR.
DONE
DONE
RE-START
[CCBC]
MEMORY ERROR REG.
3
2
1
0
[CCBF]
MAIN MEM. TIMEOUT
M8143
ADM
J2
J1
START
ADD. ACK
M8145
CDP
J2
J1

FIGURE 9.5



MJ11/MK11
M8148/47, M8158
MCT ABB
PAR. LIGHT
PAR. ERR.
[MTCA]
PARITY CHECK
J3
J1
M8149, M8159/64
MXR DBB
J3
J1
ADDRESS PARITY ERROR
MEMORY BUS CABLES
1170
CACHE
M8142
CCB
[CCBK]
ERR. REG. 01
[CCBK]
UB CYCLE
CP CYCLE
MEMORY ERROR REG.
3
2
1
0
ADD. PAR. ERR.
M8143
ADM
J2
J1
MAIN ADD. PAR. ERR.
C0, C1 PAR. BM00:03
A<00:21>
PAR. GEN.
M8145
CDP
J2
J1

FIGURE 9.6


MJ11
M8148/47
MCT
M8149
MXR
CORE
LOW
EVEN
D<00:15>
+2 PAR. BITS
J3
J1
J3
J1
MAIN MEMORY EVEN WORD
MEMORY
BUS
CABLES
1170
CACHE
M8142
CCB
M8143
ADM
M8145
CDP
J2
J1
J2
J1
[CDPF]
PARITY
CHECK
MAIN LO OK
MEMORY
ERROR
REG.
3
2
1
0
MAIN MEM.
EVEN WORD

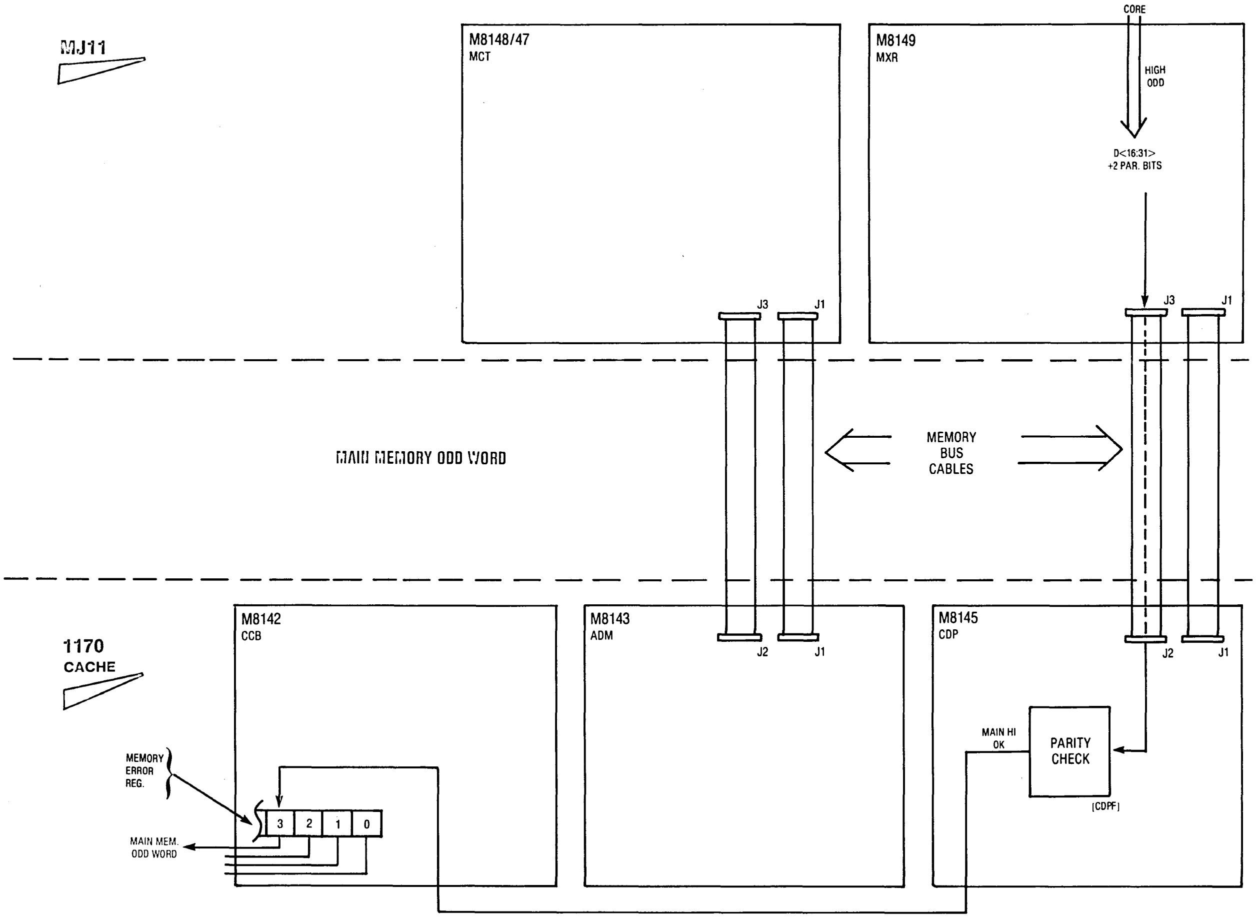
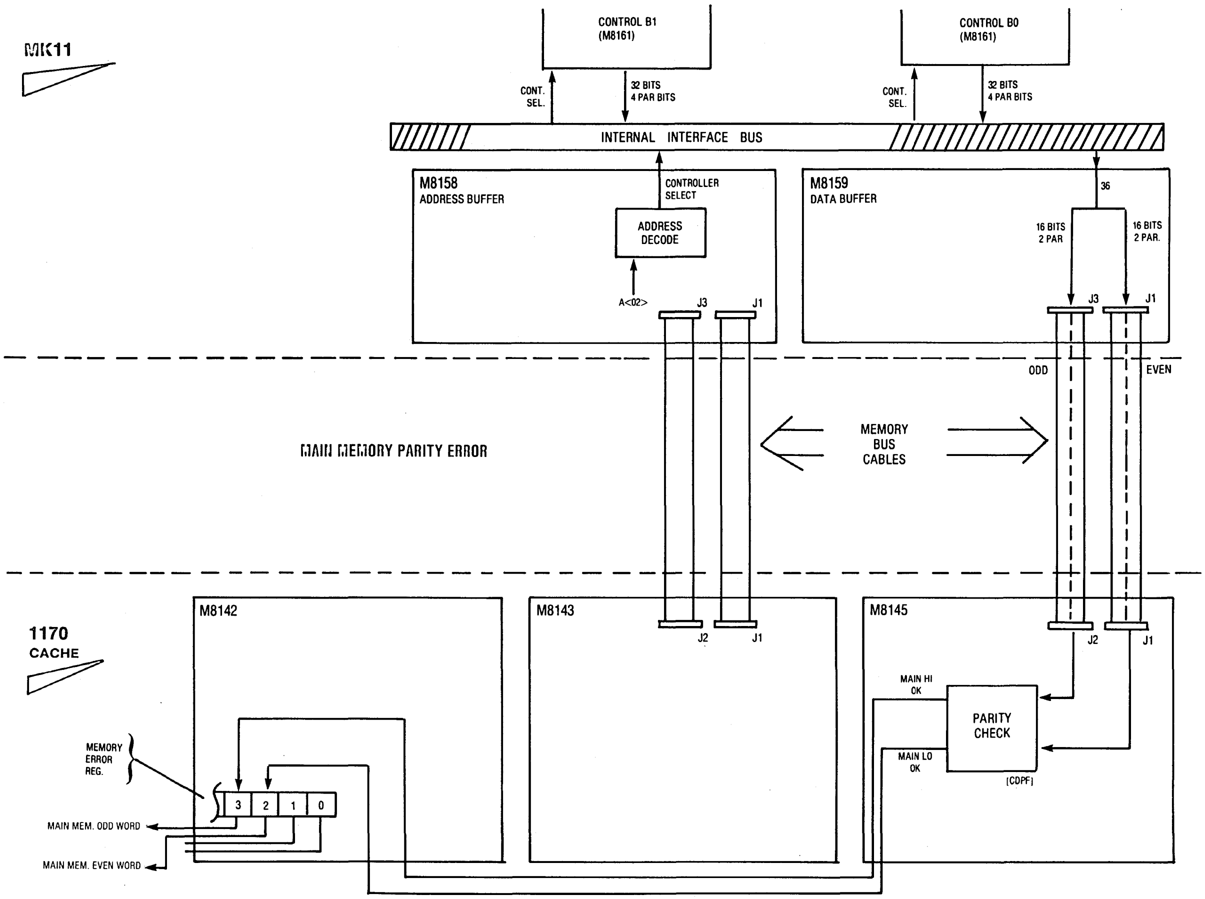

MJ11
M8148/47
MCT
M8149
MXR
CORE
HIGH
ODD
D<16:31>
+2 PAR. BITS
J3
J1
J3
J1
MAIN MEMORY ODD WORD
MEMORY
BUS
CABLES
1170
CACHE
M8142
CCB
M8143
ADM
M8145
CDP
J2
J1
J2
J1
MEMORY
ERROR
REG.
3
2
1
0
MAIN MEM.
ODD WORD
MAIN HI
OK
PARITY
CHECK
(CDPF)

FIGURE 9.7

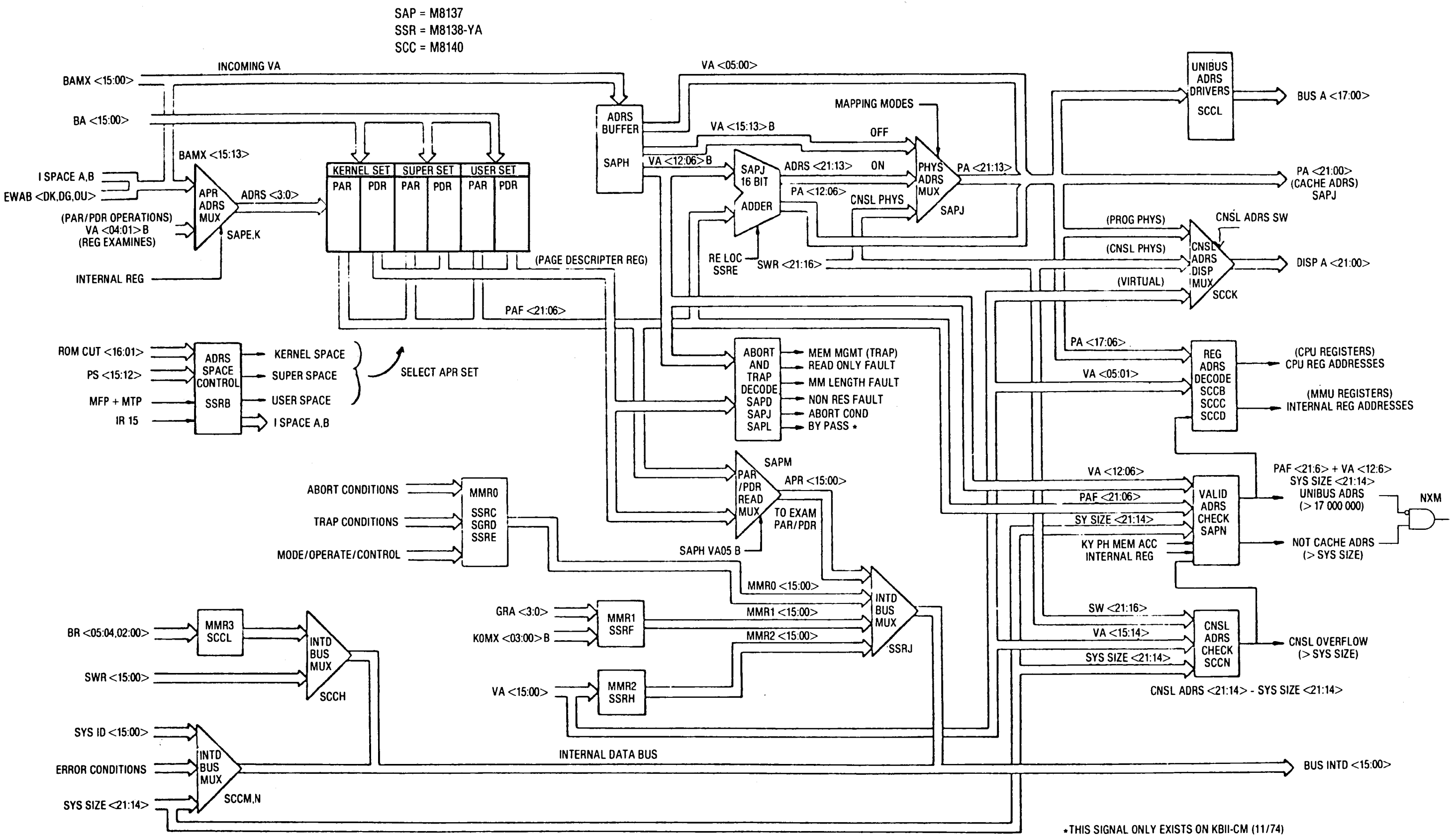
FIGURE 9.8


MK11
CONTROL B1 (M8161)
CONTROL B0 (M8161)
CONT. SEL.
32 BITS 4 PAR BITS
CONT. SEL.
32 BITS 4 PAR BITS
INTERNAL INTERFACE BUS
M8158 ADDRESS BUFFER
CONTROLLER SELECT
ADDRESS DECODE
A<02>
J3
J1
M8159 DATA BUFFER
36
16 BITS 2 PAR
16 BITS 2 PAR.
J3
J1
ODD
EVEN
MAIN MEMORY PARITY ERROR
MEMORY BUS CABLES
1170 CACHE
M8142
M8143
M8145
J2
J1
J2
J1
MAIN HI OK
MAIN LO OK
PARITY CHECK
[CDPF]
MEMORY ERROR REG.
3
2
1
0
MAIN MEM. ODD WORD
MAIN MEM. EVEN WORD

CHAPTER 10

MEMORY MANAGEMENT

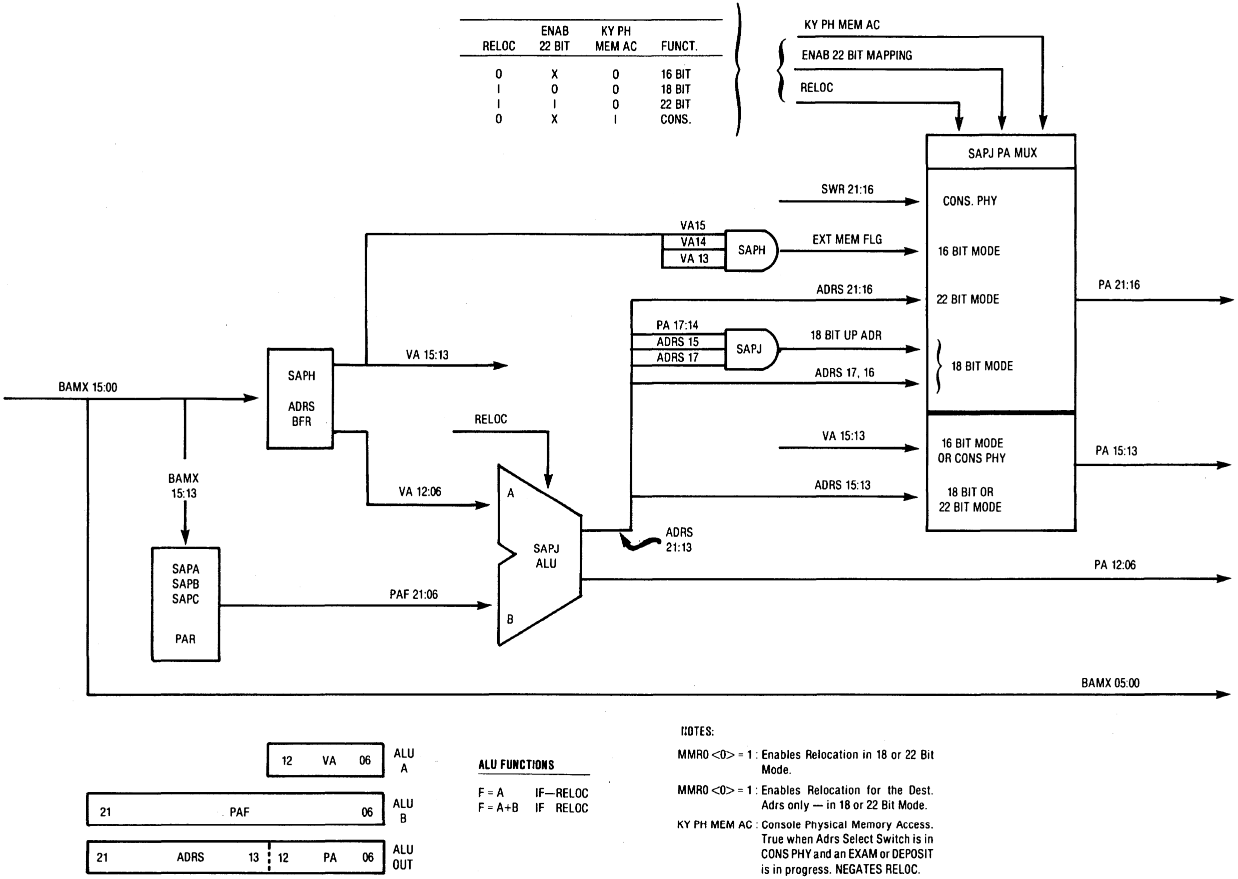
FIGURE 10.1 MEMORY MANAGEMENT BLOCK DIAGRAM



*THIS SIGNAL ONLY EXISTS ON KBII-CM (11/74)

FIGURE 10.2

PHYSICAL ADDRESS CONSTRUCTION



RELOC	ENAB 22 BIT	KY PH MEM AC	FUNCT.
0	X	0	16 BIT
I	0	0	18 BIT
I	I	0	22 BIT
0	X	I	CONS.

ALU FUNCTIONS

F = A	IF—RELOC
F = A+B	IF RELOC

NOTES:

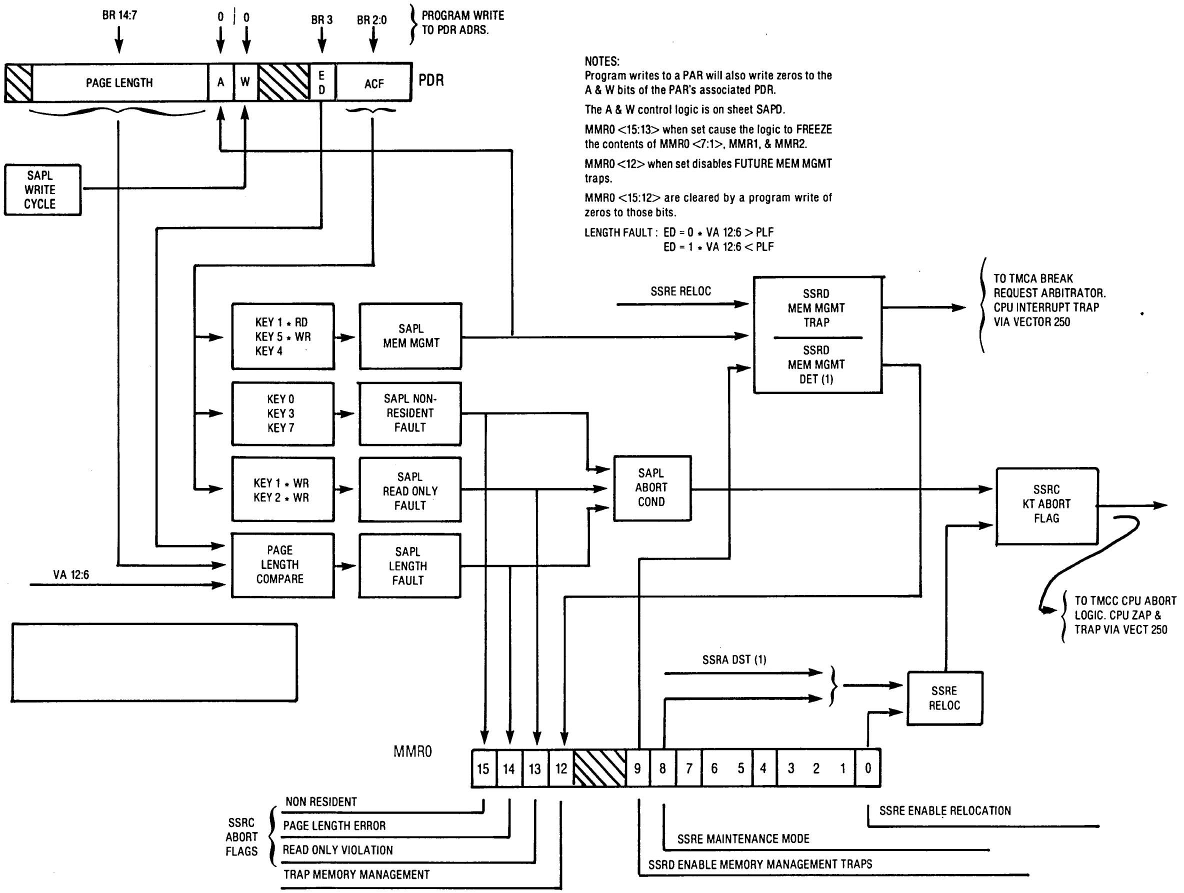
MMR0 <0> = 1 : Enables Relocation in 18 or 22 Bit Mode.

MMR0 <0> = 1 : Enables Relocation for the Dest. Adrs only — in 18 or 22 Bit Mode.

KY PH MEM AC : Console Physical Memory Access. True when Adrs Select Switch is in CONS PHY and an EXAM or DEPOSIT is in progress. NEGATES RELOC.

FIGURE 10.3

MEMORY MANAGEMENT TRAPS & ABORTS



NOTES:

Program writes to a PAR will also write zeros to the A & W bits of the PAR's associated PDR.

The A & W control logic is on sheet SAPD.

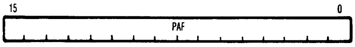
MMR0 <15:13> when set cause the logic to FREEZE the contents of MMR0 <7:1>, MMR1, & MMR2.

MMR0 <12> when set disables FUTURE MEM MGMT traps.

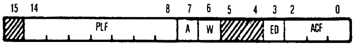
MMR0 <15:12> are cleared by a program write of zeros to those bits.

LENGTH FAULT : ED = 0 * VA 12:6 > PLF
ED = 1 * VA 12:6 < PLF

MEMORY MANAGEMENT REGISTERS

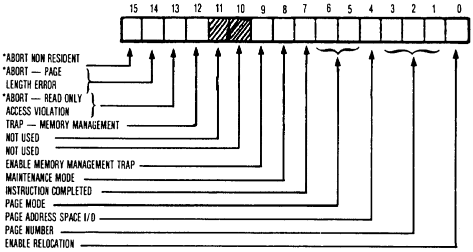


Page Address Register

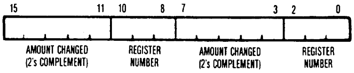


Page Description Register

Format of Memory Management Register #0 (MMR0) 17 777 572

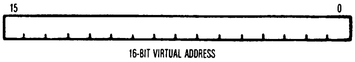


Format of Memory Management Register #1 (MMR1) 17 777574

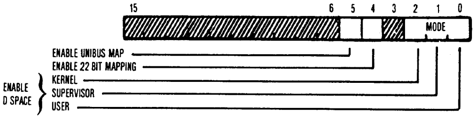


15
11
10
8
7
3
2
0
AMOUNT CHANGED (2's COMPLEMENT)
REGISTER NUMBER
AMOUNT CHANGED (2's COMPLEMENT)
REGISTER NUMBER

Format of Memory Management Register #2 (MMR2) 17 777576



Format of Memory Management Register #3 (MMR3) 17 772516



CONTENTS USED FOR 18 OR 22 BIT ADDRESS RELOCATION

ACF	MEANING
0	NON RESIDENT, ABORT ALL ACCESSES
1	READ ONLY, ABORT ON WRITE, TRAP ON READ
2	READ ONLY, ABORT ON WRITE
3	NOT USED, ABORT ALL ACCESSES
4	READ/WRITE, NO ABORTS, TRAP ON READ/WRITE
5	READ/WRITE, NO ABORTS, TRAP ON WRITE
6	READ/WRITE, NO ABORTS, NO TRAPS
7	UNUSED, ABORT ALL ACCESSES

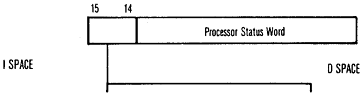
ED	MEANING
0	UPWARD EXPANDABLE PAGE
1	DOWNWARD EXPANDABLE PAGE

W	=	PAGE WAS WRITTEN INTO
A	=	ACTIVE PAGE
PLF	=	PAGE LENGTH FIELD

*SETS ADDRESS ERROR LITE ON FRONT PANEL.

NOTE: MMR0 BITS 15 AND 14 WILL BOTH BE SET IF AN ILLEGAL CPU MODE IS USED, IE: PS<15:14> = 10

ACTIVE PAGE REGISTERS

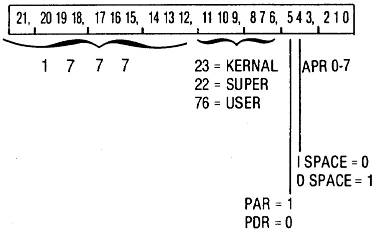


	KERNEL (00)			KERNEL (00)	
APR0	17772340	17772300	APR0	17772360	17772320
1	17772342	17772302	1	17772362	17772322
2	17772344	17772304	2	17772364	17772324
3	17772346	17772306	3	17772366	17772326
4	17772350	17772310	4	17772370	17772330
5	17772352	17772312	5	17772372	17772332
6	17772354	17772314	6	17772374	17772334
7	17772356	17772316	7	17772376	17772336
	PAR	PDR		PAR	PDR

	SUPERVISOR (01)			SUPERVISOR (01)	
APR0	17772240	17772200	APR0	17772260	17772220
1	17772242	17772202	1	17772262	17772222
2	17772244	17772204	2	17772264	17772224
3	17772246	17772206	3	17772266	17772226
4	17772250	17772210	4	17772270	17772230
5	17772252	17772212	5	17772272	17772232
6	17772254	17772214	6	17772274	17772234
7	17772256	17772216	7	17772276	17772236
	PAR	PDR		PAR	PDR

	USER (11)			USER (11)	
APR0	17777640	17777600	APR0	17777660	17777620
1	17777642	17777602	1	17777662	17777622
2	17777644	17777604	2	17777664	17777624
3	17777646	17777606	3	17777666	17777626
4	17777650	17777610	4	17777670	17777630
5	17777652	17777612	5	17777672	17777632
6	17777654	17777614	6	17777674	17777634
7	17777656	17777616	7	17777676	17777636
	PAR	PDR		PAR	PDR

APR ADDRESS SELECTION



ADDRESS MODES/RELOCATION

The 11/70 Addressing Modes are selected by single bits in two Memory Management Registers (MMR0, and MMR3).

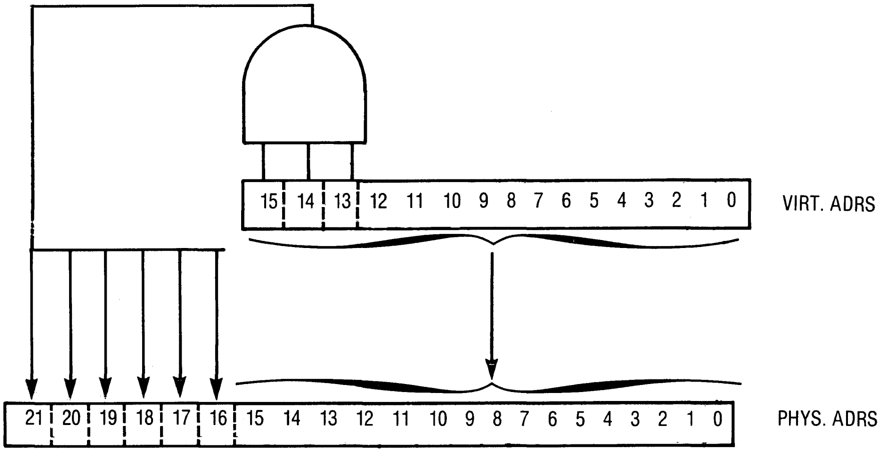
MMR0 BIT0 — ENABLE RELOCATION. When cleared Memory Management is Passive (16 Bit Mode). When set (18 or 22 bit Mode) all Virtual Addresses are relocated by Memory Management.

MMR3 BIT4 — ENABLE 22 BIT MAPPING. Meaningful only when MMR0 Bit 0 is set.

MMR0 <0>	**MMR3 <4>**	**ADRS MODE**
CLEARED	X	16 BIT
SET	CLEARED	18 BIT
SET	SET	22 BIT

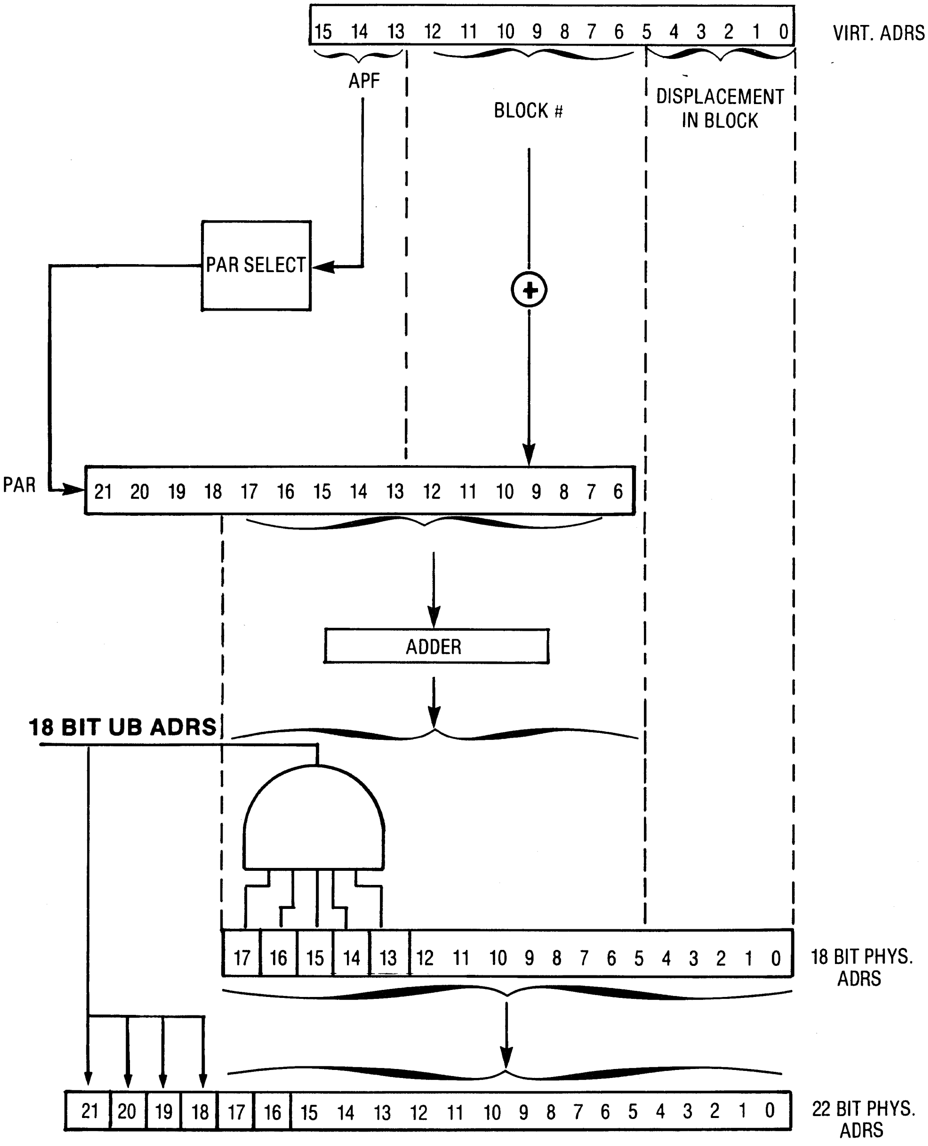
NOTE: MMR0 and MMR3 are cleared by Power Up, Console START, and the RESET Instruction.

UNIBUS ADDRESS H



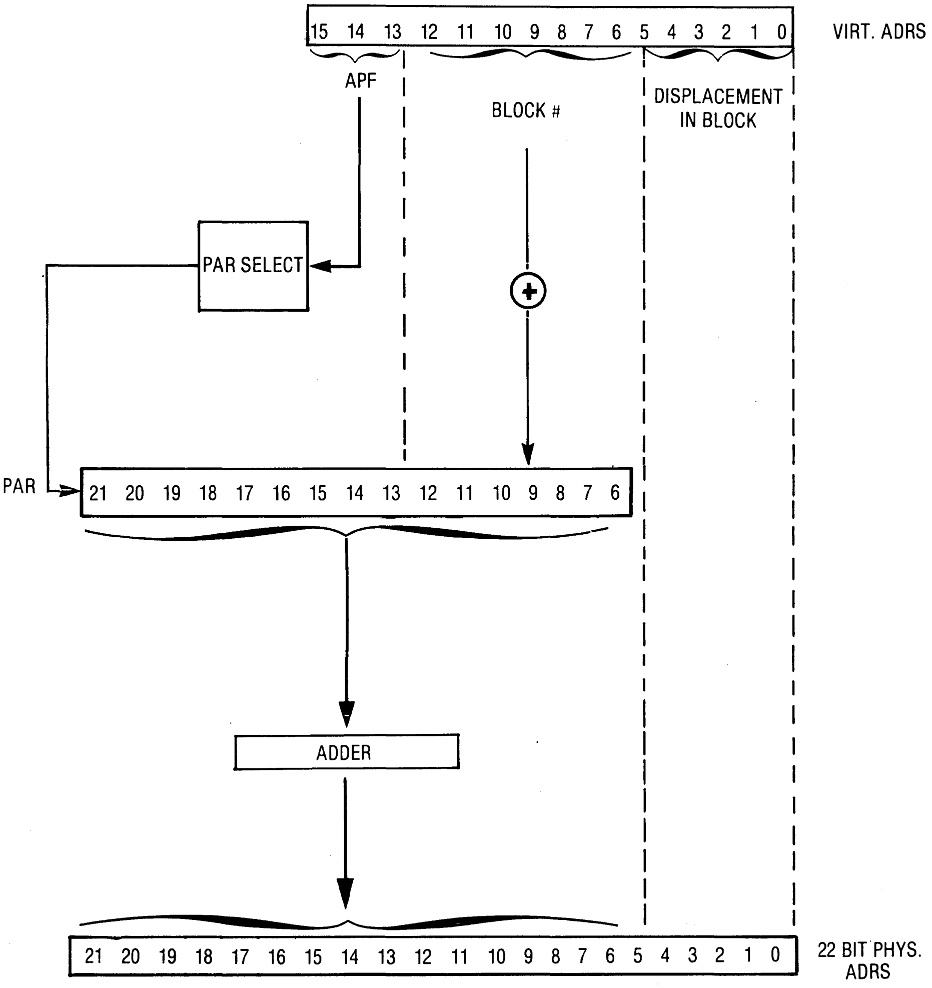
BITS <21:16> = l's IF UNIBUS ADDRESS
BITS <21:16> = 0's IF NOT UNIBUS ADDRESS

16-BIT ADDRESSING



BITS <21:18> = l's IF UNIBUS ADDRESS
BITS <21:18> = 0's IF NOT UNIBUS ADDRESS

18-BIT ADDRESSING



UNIBUS ADDRESS IF BITS <21:18> = 1

22-BIT ADDRESSING

CHAPTER 11

UNIBUS MAP

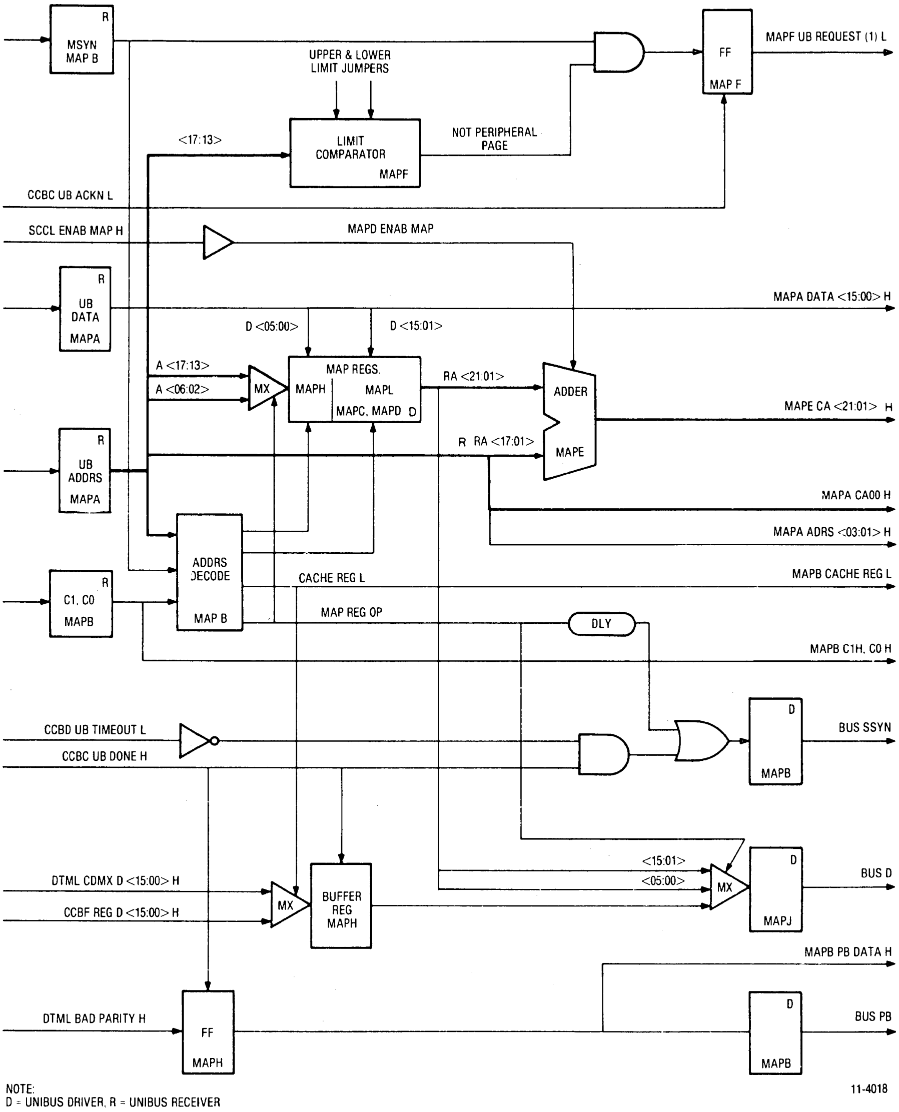


FIGURE 11.1

UNIBUS MAP BLOCK DIAGRAM

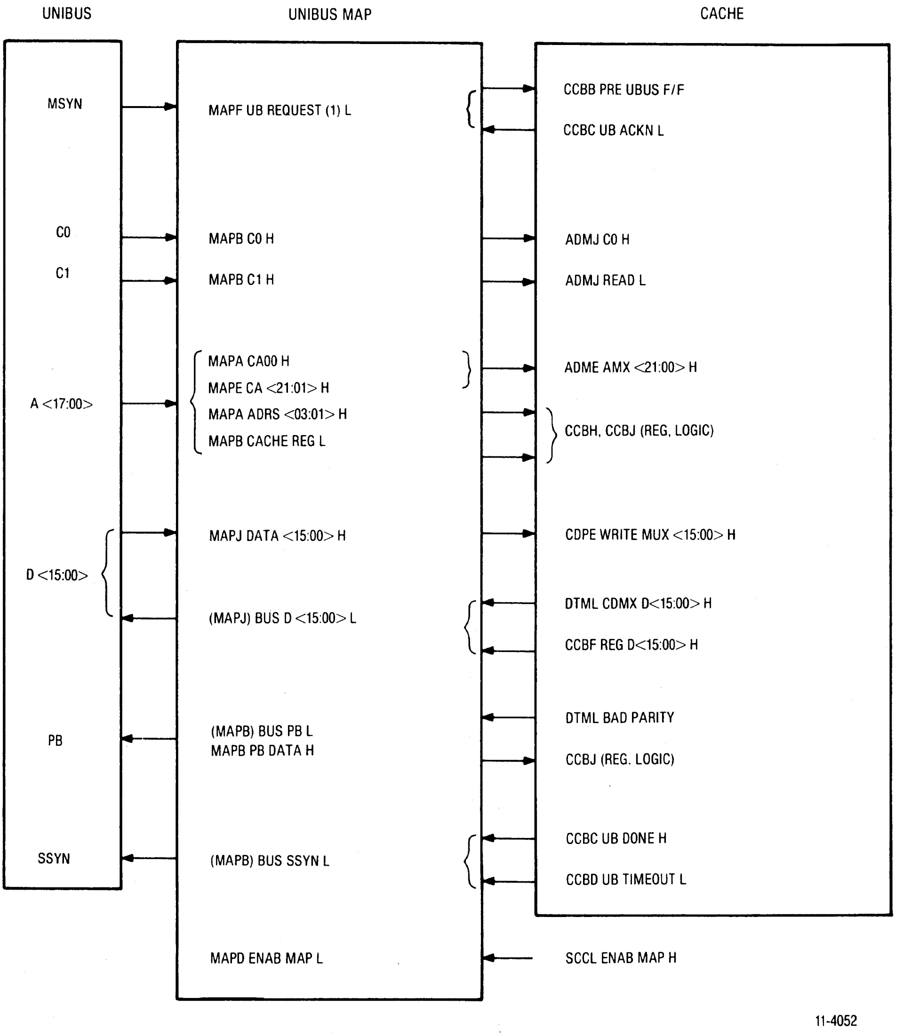


FIGURE 11.2

UNIBUS MAP INTERFACE

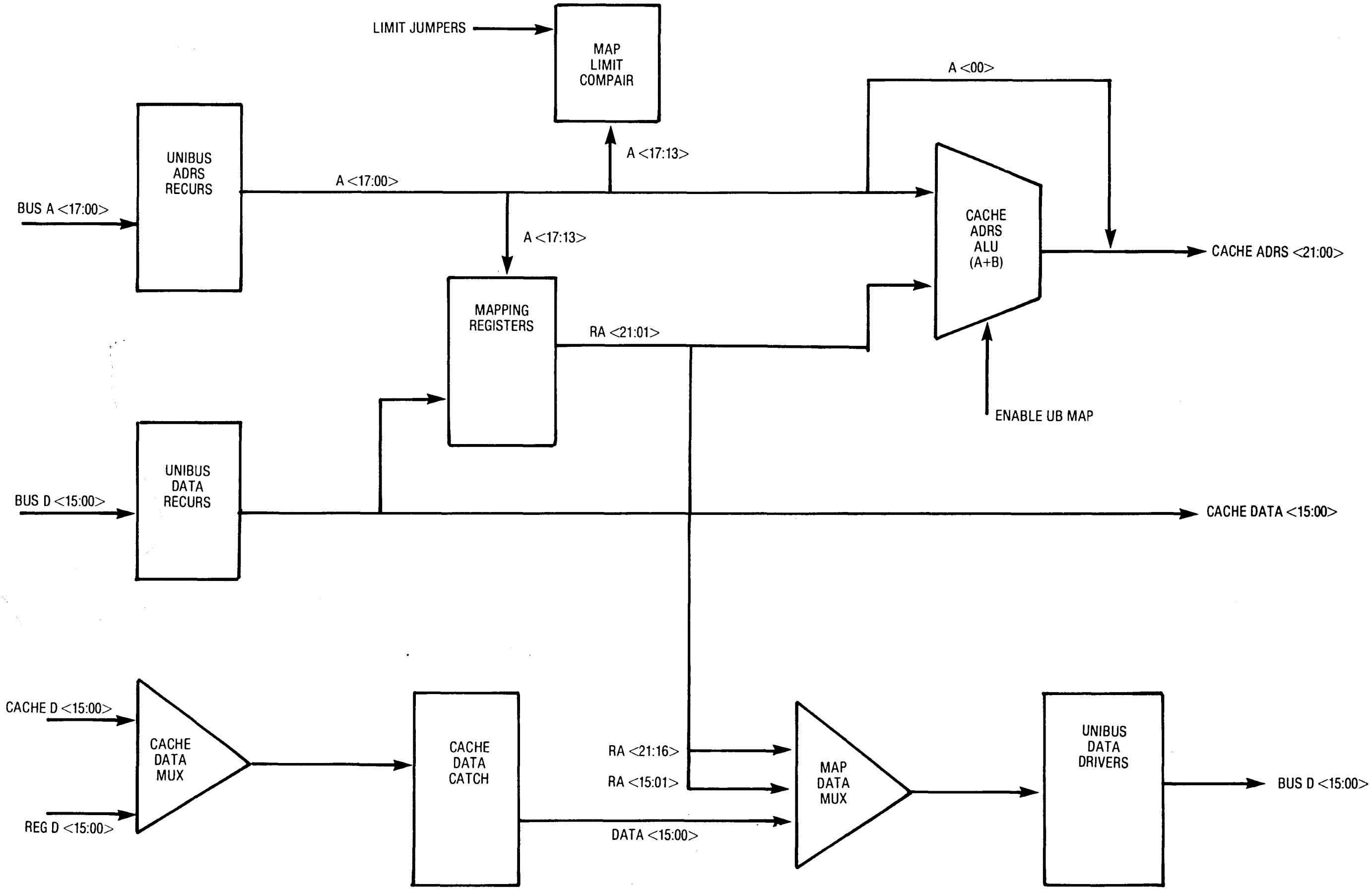


FIGURE 11.3

UNIBUS MAP BLOCK DIAGRAM

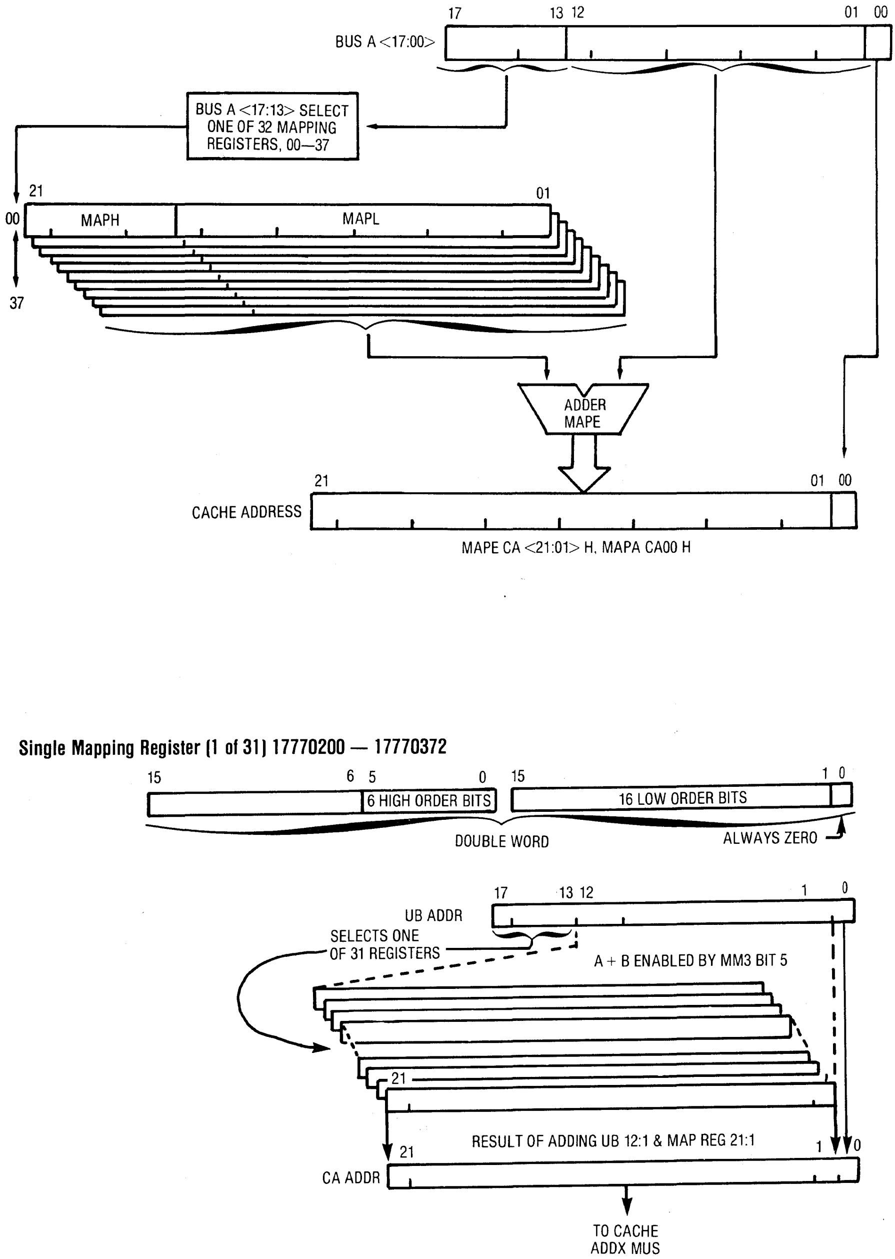

17
13 12
01 00
BUS A <17:00>
BUS A <17:13> SELECT ONE OF 32 MAPPING REGISTERS, 00—37
21
01
00
MAPH
MAPL
37
ADDER MAPE
21
01 00
CACHE ADDRESS
MAPE CA <21:01> H, MAPA CA00 H
Single Mapping Register (1 of 31) 17770200 — 17770372
15
6 5
0 15
1 0
6 HIGH ORDER BITS
16 LOW ORDER BITS
DOUBLE WORD
ALWAYS ZERO
17
13 12
1
0
UB ADDR
SELECTS ONE OF 31 REGISTERS
A + B ENABLED BY MM3 BIT 5
21
RESULT OF ADDING UB 12:1 & MAP REG 21:1
21
1
0
CA ADDR
TO CACHE ADDX MUS

FIGURE 11.4

CONSTRUCTION OF THE PA

ACCESS TO UNIBUS MAP REGISTERS

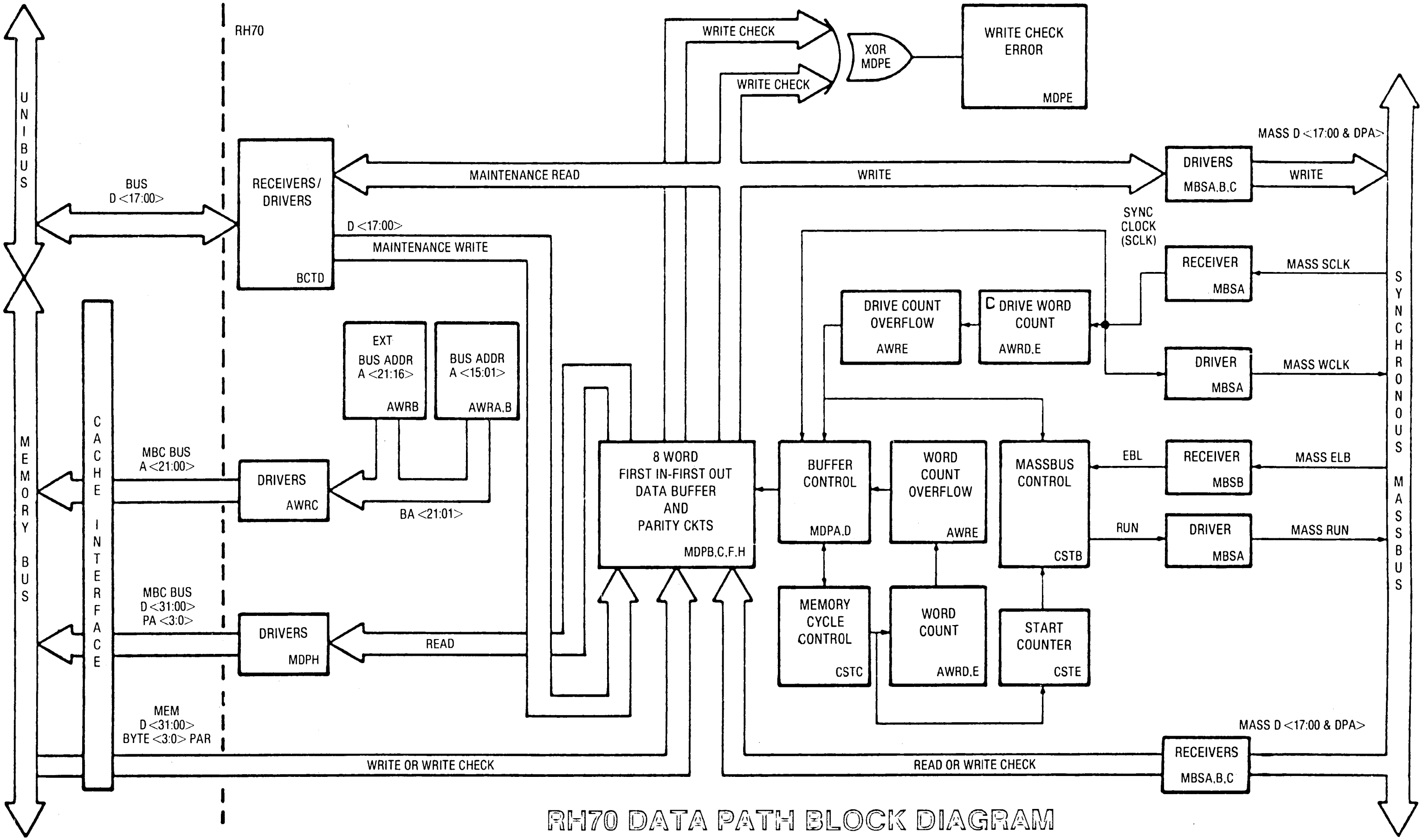
	Unibus Address Read or Write		
Register No.	MAPL	MAPH	Unibus Address for Memory Reference
0	17 770 200,	02	17 000 000 — 17 017 777
1	17 770 204,	06	17 020 000 — 17 037 777
2	17 770 210,	12	17 040 000 — 17 057 777
3	17 770 214,	16	17 060 000 — 17 077 777
4	17 770 220,	22	17 100 000 — 17 117 777
5	17 770 224,	26	17 120 000 — 17 137 777
6	17 770 230,	32	17 140 000 — 17 157 777
7	17 770 234,	36	17 160 000 — 17 177 777
10	17 770 240,	42	17 200 000 — 17 217 777
11	17 770 244,	46	17 220 000 — 17 237 777
12	17 770 250,	52	17 240 000 — 17 257 777
13	17 770 254,	56	17 260 000 — 17 277 777
14	17 770 260,	62	17 300 000 — 17 317 777
15	17 770 264,	66	17 320 000 — 17 337 777
16	17 770 270,	72	17 340 000 — 17 357 777
17	17 770 274,	76	17 360 000 — 17 377 777
20	17 770 300,	02	17 400 000 — 17 417 777
21	17 770 304,	06	17 420 000 — 17 437 777
22	17 770 310,	12	17 440 000 — 17 457 777
23	17 770 314,	16	17 460 000 — 17 477 777
24	17 770 320,	22	17 500 000 — 17 517 777
25	17 770 324,	26	17 520 000 — 17 537 777
26	17 770 330,	32	17 540 000 — 17 557 777
27	17 770 334,	36	17 560 000 — 17 577 777
30	17 770 340,	42	17 600 000 — 17 617 777
31	17 770 344,	46	17 620 000 — 17 637 777
32	17 770 350,	52	17 640 000 — 17 657 777
33	17 770 354,	56	17 600 000 — 17 677 777
34	17 770 360,	62	17 700 000 — 17 717 777
35	17 770 364,	66	17 720 000 — 17 737 777
36	17 770 370,	72	17 740 000 — 17 757 777
*37	17 770 374,	76	17 760 000 — 17 777 777

**Note:* *Can be read or written into, but not used for mapping.*

CHAPTER 12

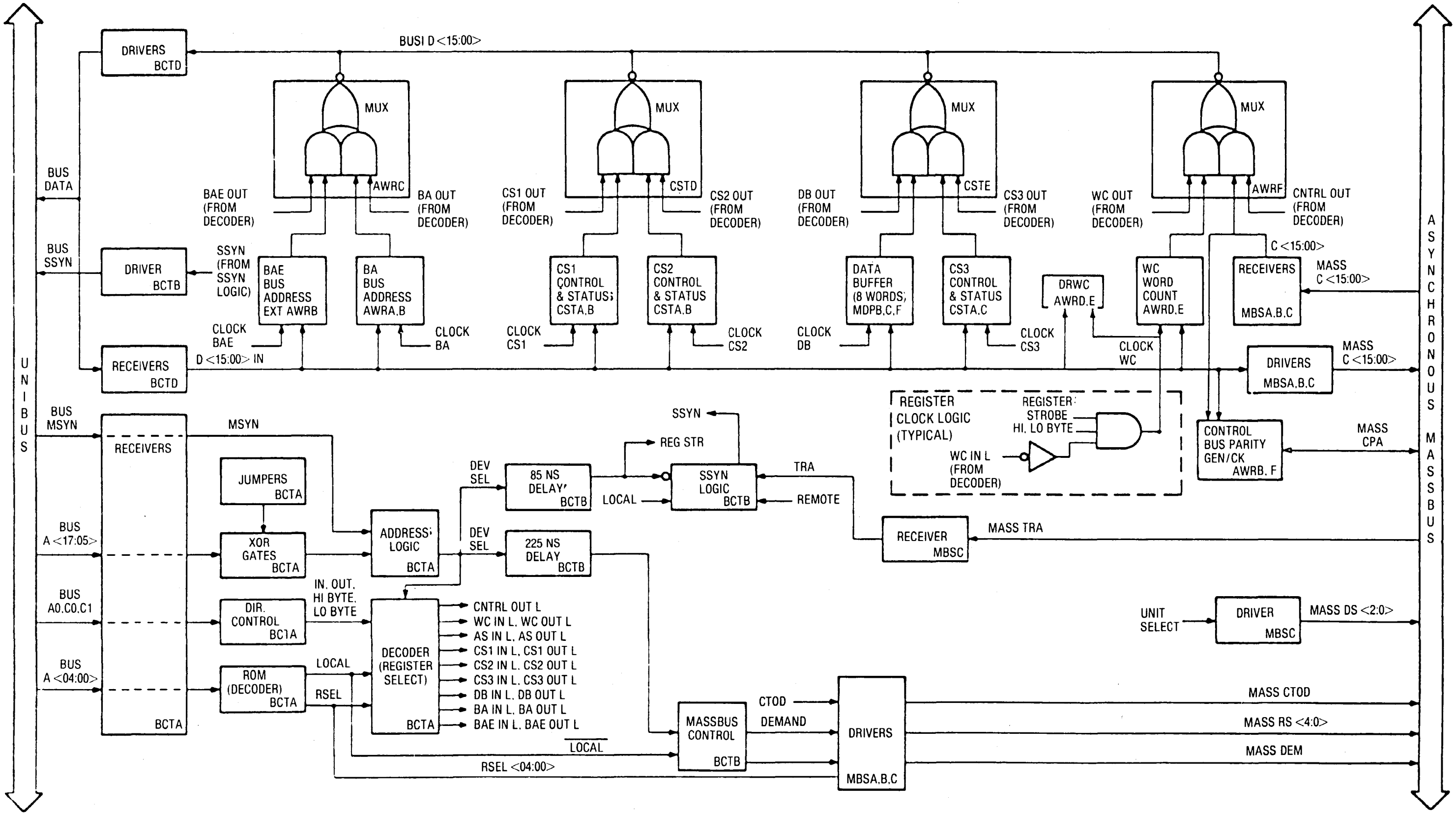
RH70

FIGURE 12.1



RH70 DATA PATH BLOCK DIAGRAM

FIGURE 12.2



RH70 REGISTER CONTROL PATH

ADDITIONAL REGISTERS FOR RH70

The RH70 has the same registers as the RH11 plus two more, the Bus Address Extension Register and Control and Status Register 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	A21	A20	A19	A18	A17	A16

Bus Address Extension Register

BITS <5:0> — USED FOR 22 BIT ADDRESSING ON THE 1170

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APE	DPE OW	DPE EW	WCE OW	WCE EW	DBL	0	0	0	IE	0	0	IPCK 3	IPCK 2	IPCK 1	IPCK 0

Control and Status Register 3

BIT	MEANING WHEN SET
15	ADDRESS PARITY ERROR DETECTED IN MEMORY.
14	DATA PARITY ERROR ODD WORD FROM MEMORY.
13	DATA PARITY ERROR EVEN WORD FROM MEMORY.
12	WRITE CHECK ERROR ON ODD WORD.
11	WRITE CHECK ERROR ON EVEN WORD.
10	DOUBLE WORD WAS THE LAST TRANSFER.
6	INTERRUPT ENABLE (SAME AS BIT 6 OF CS1).
3	INVERSE PARITY CHECK (DIAGNOSTIC PURPOSES)
2	
1	
0	

CHAPTER 13

MJ11

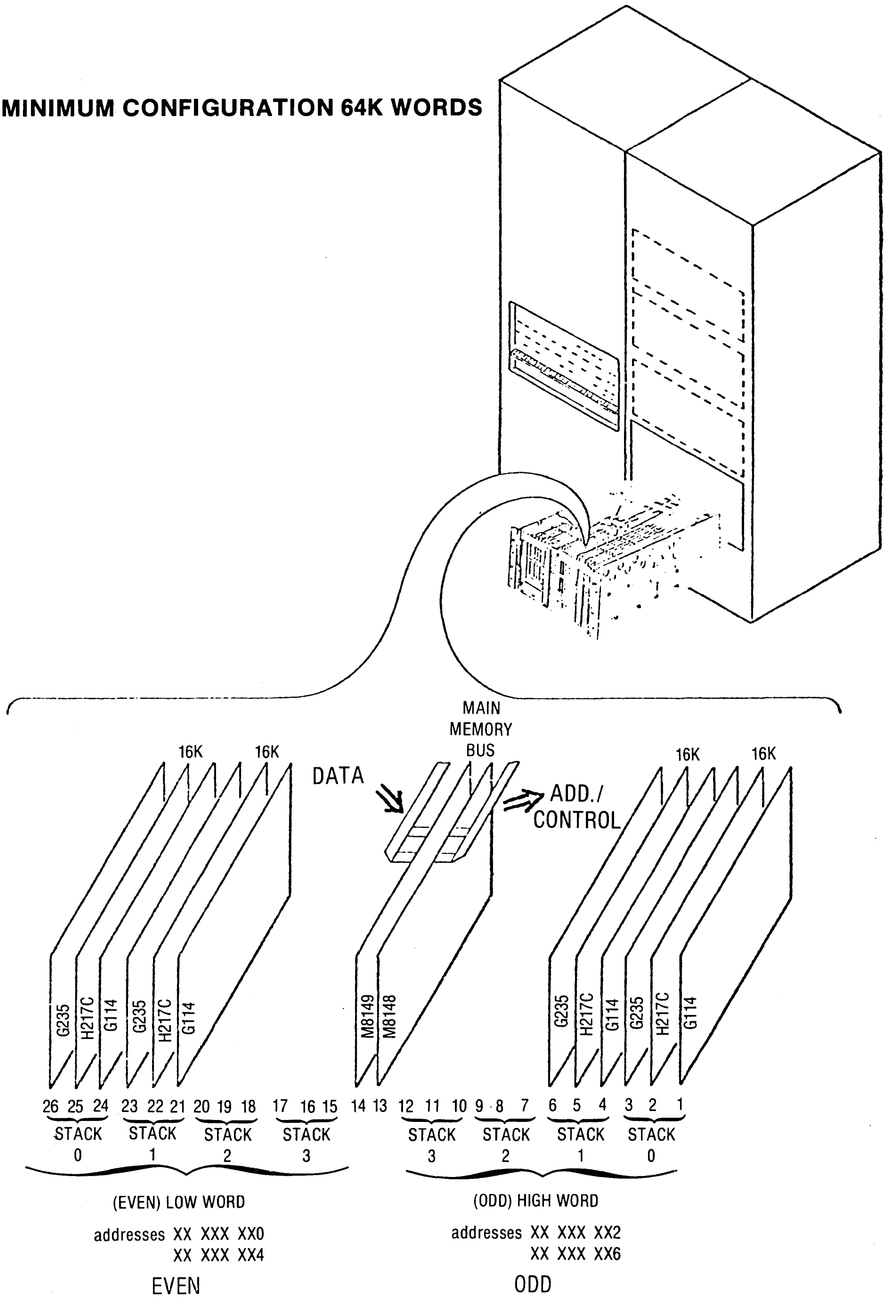
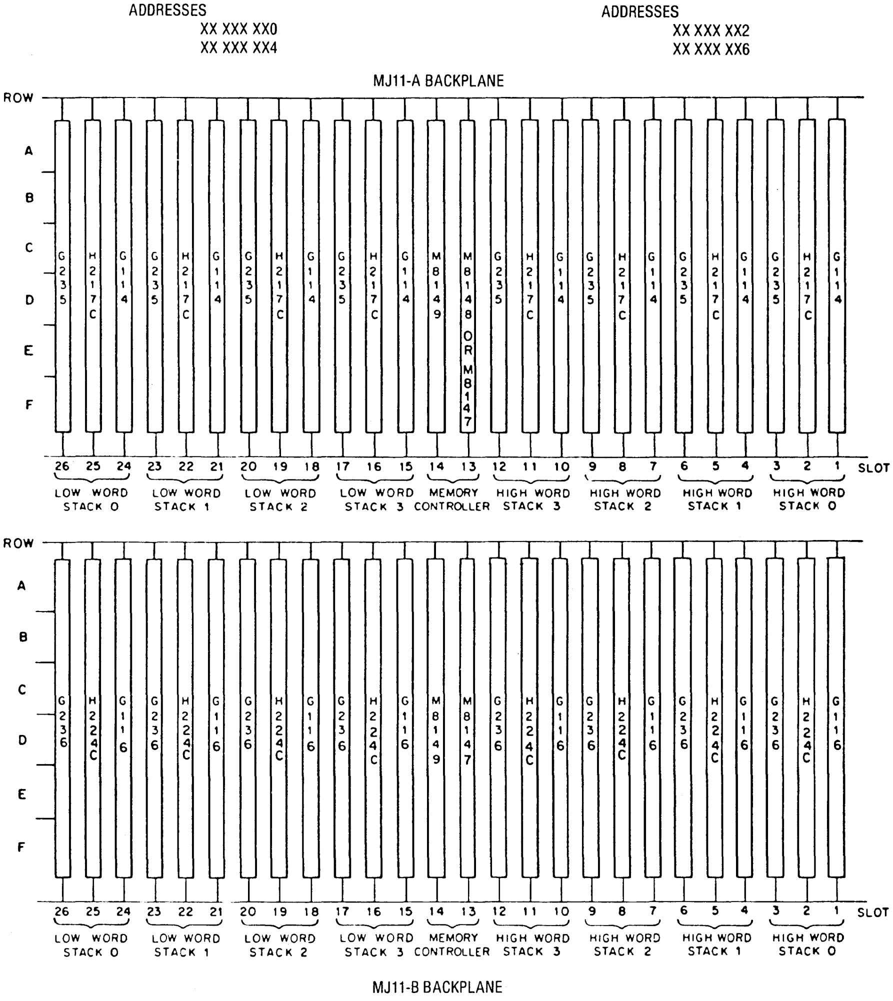


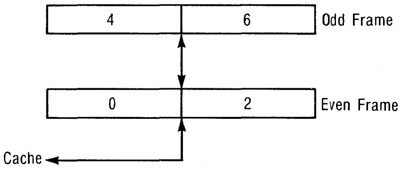
FIGURE 13.1

MJ11 MEMORY



NOTE:

1. This figure illustrates a view as seen from the pin side (in a maintenance position with the power supply below the modules).
2. Stack 0 consists of two 16K word stacks (MJ11-A) or two 32K word stock (MJ11-B). Likewise for stack 1, stack 2, & stack 3.



MJ11 Interleaving

FIGURE 13.2

MJ11 MODULE UTILIZATION

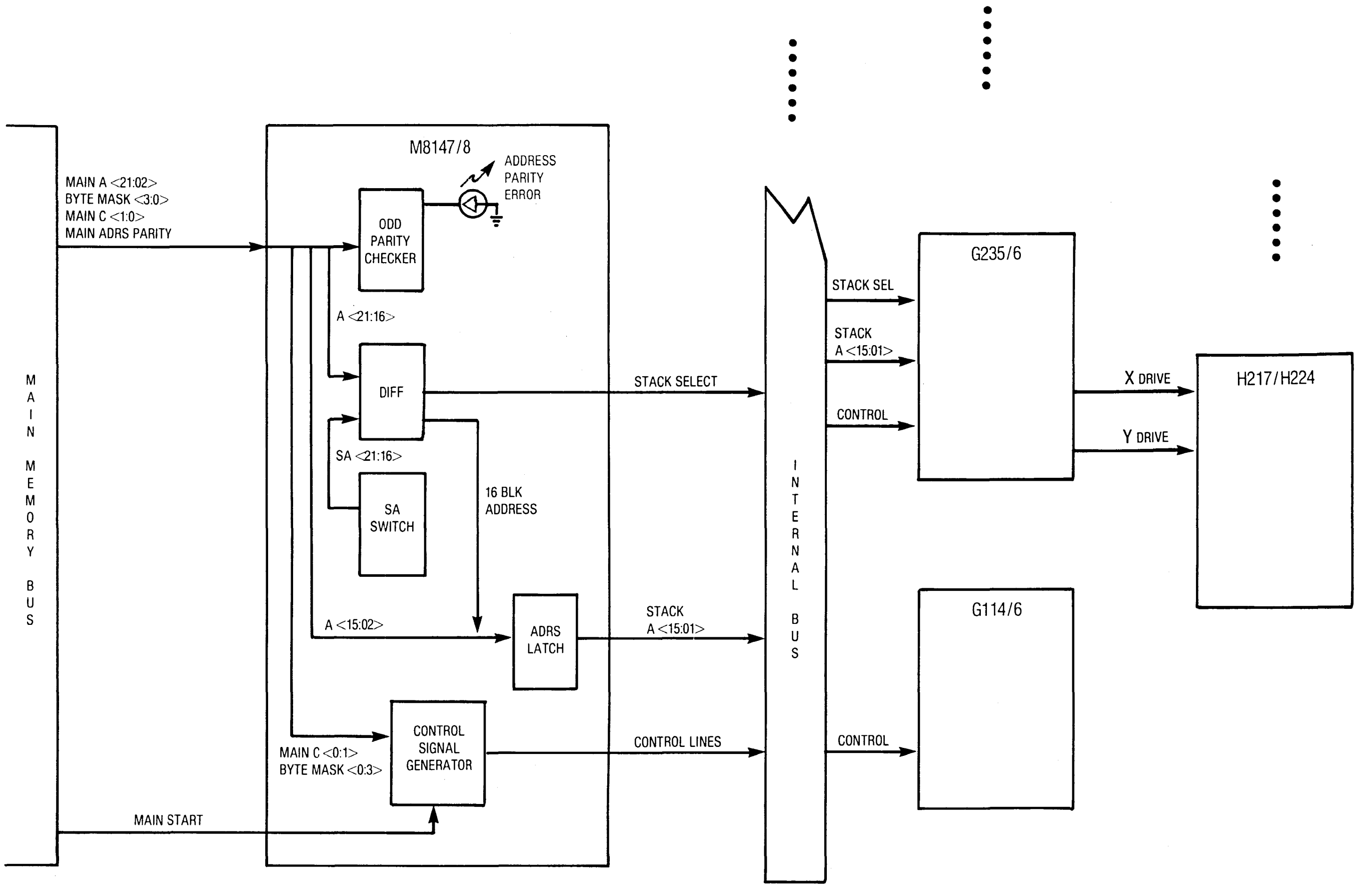
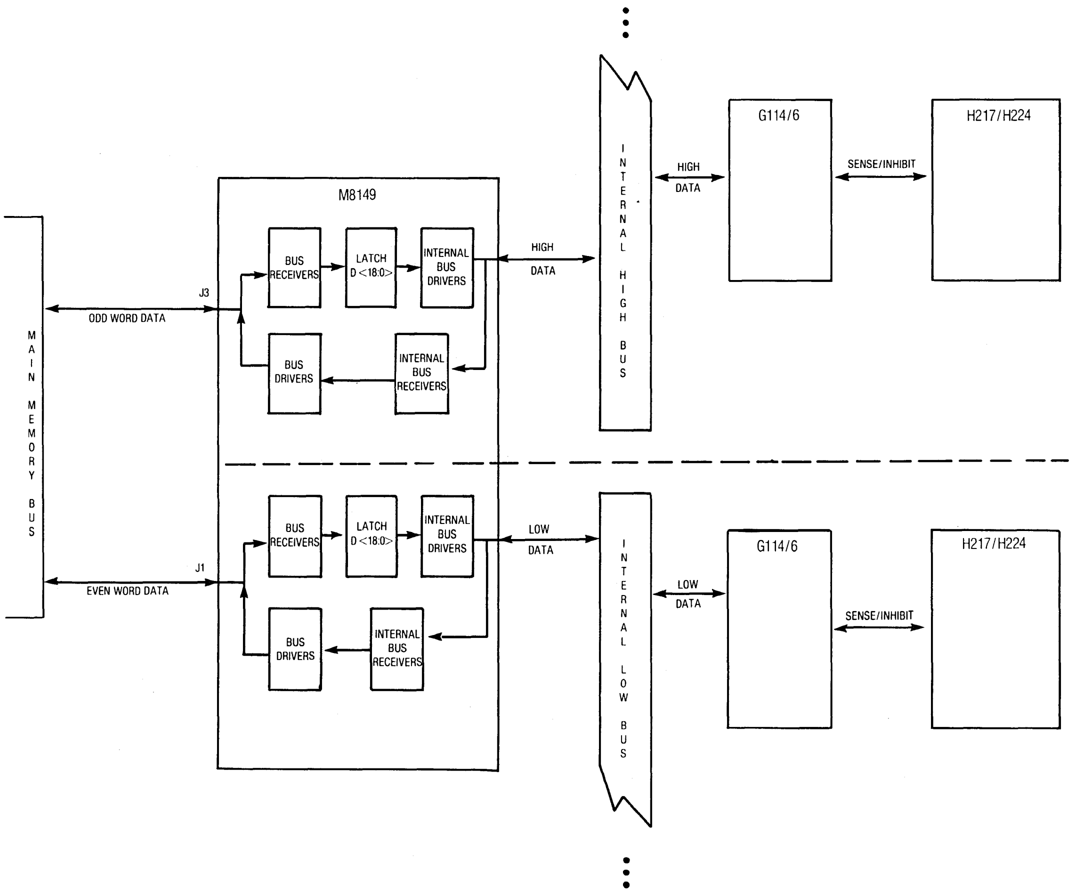


FIGURE 13.3

MJ11 ADDRESS/CONTROL PATH

FIGURE 13.4
MJ11 DATA PATH



MJ11-A 16KW STACKS
BOX MAX = 128KW

MJ11 MEMORY SYSTEM

CONTROLLER STARTING ADDRESS (OCTAL) ____________ 00000

INTERLEAVED □ YES □ NO □ EVEN □ ODD

MEMORY SYSTEM SERIAL NO. DATE INSTALLED

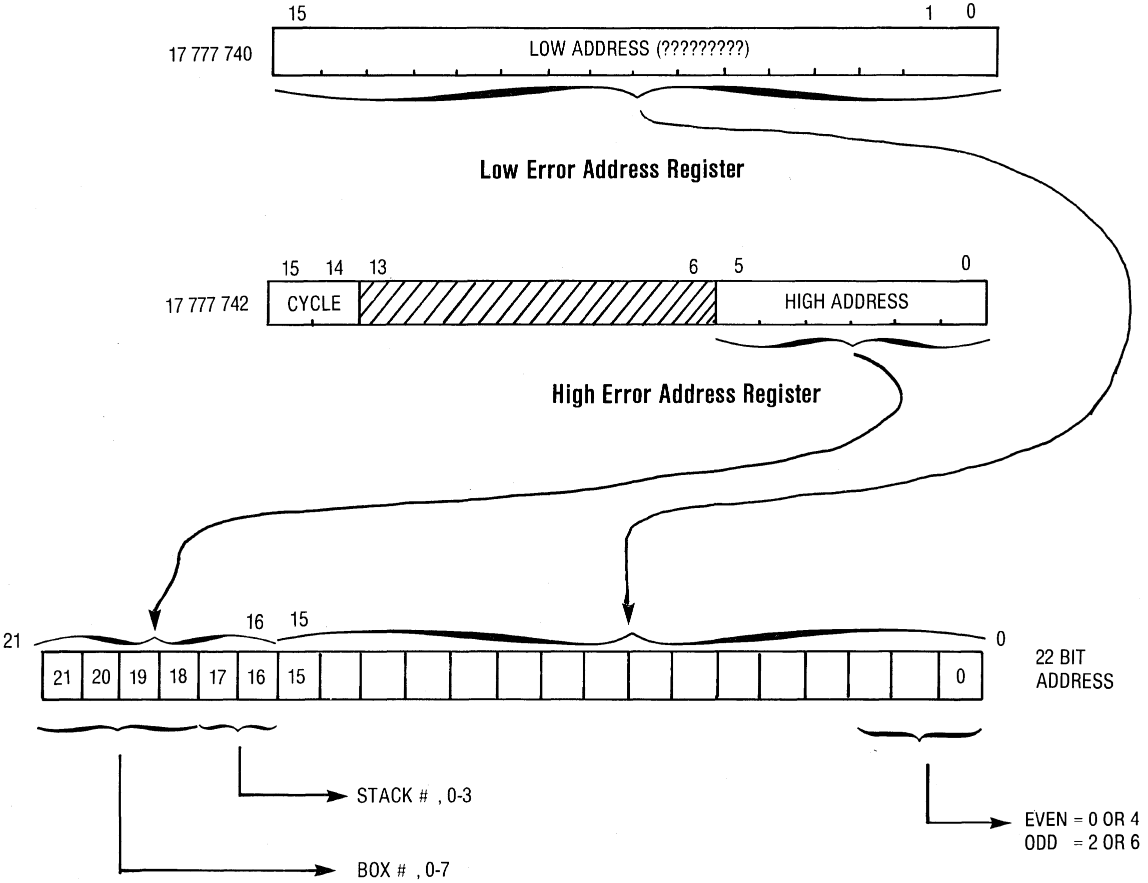
SLOT	MODULE TYPE		ADDRESSES	MODULE (GROUP) FUNCTION
01	G114	SIN	000002 → 177776	2 STACK 0 HIGH WORD/ODD 6
02	H217—C	STK		
03	G235	DRV		
04	G114	SIN	200002 → 377776	2 STACK 1 HIGH WORD/ODD 6
05	H217—C	STK		
06	G235	DRV		
07	G114	SIN	400002 → 577776	2 STACK 2 HIGH WORD/ODD 6
08	H217—C	STK		
09	G235	DRV		
10	G114	SIN	600002 → 777776	2 STACK 3 HIGH WORD/ODD 6
11	H217—C	STK		
12	G235	DRV		
13	M8147*	MCT		MEMORY CONTROL & TIMING
14	M8149	MXR		MEMORY TRANSCEIVER
15	G114	SIN	600000 → 777774	0 STACK 3 LOW WORD/EVEN 4
16	H217—C	STK		
17	G235	DRV		
18	G114	SIN	400000 → 577774	0 STACK 2 LOW WORD/EVEN 4
19	H217—C	STK		
20	G235	DRV		
21	G114	SIN	200000 → 377774	0 STACK 1 LOW WORD/EVEN 4
22	H217—C	STK		
23	G235	DRV		
24	G114	SIN	000000 → 177774	0 STACK 0 LOW WORD/EVEN 4
25	H217—C	STK		
26	G235	DRV		

*NOTE M8147 CAN BE USED FOR BOTH 16K AND 32K M8148 FOR 16K ONLY.

FIGURE 13.5 MJ11-A 16K STACKS

FIGURE 13.6

MJ11 WITH 16K STACKS



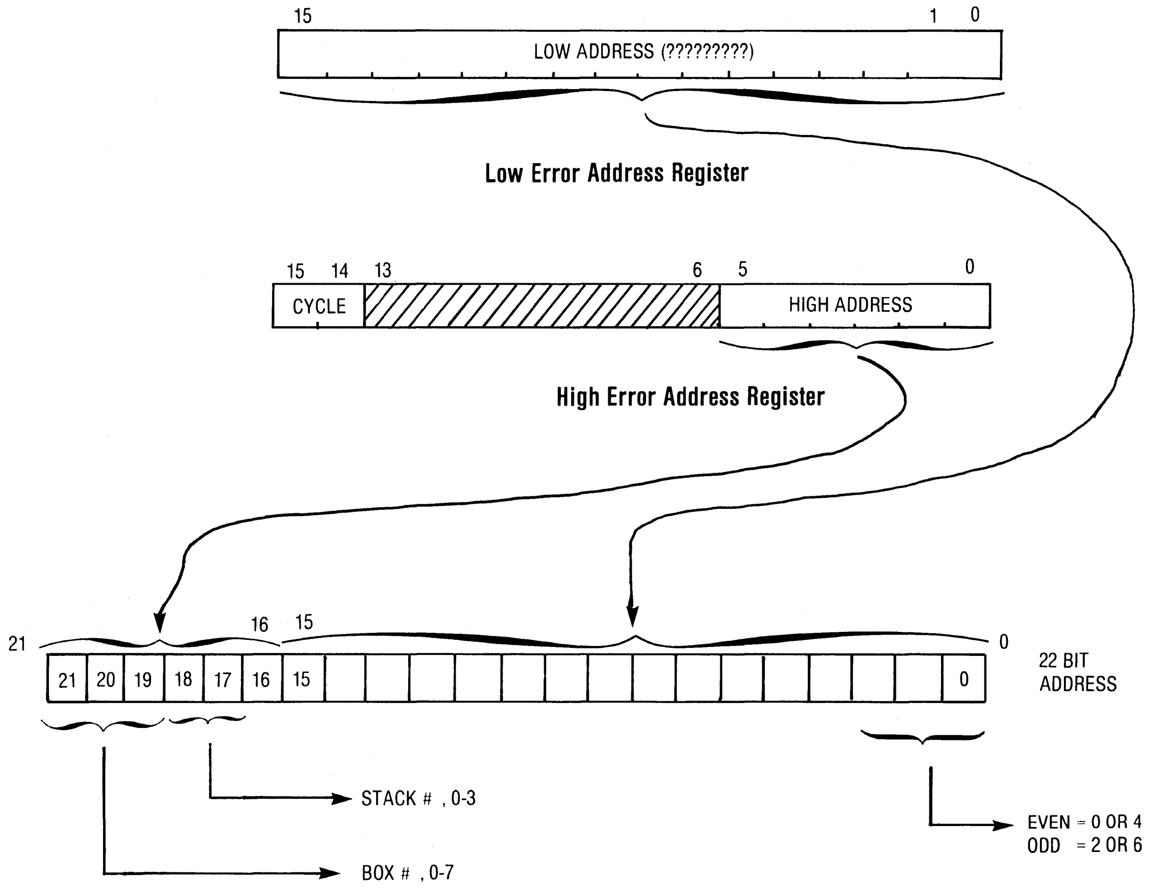
MJ11-B 32KW STACKS
BOX MAX = 256KW

MJ11 MEMORY SYSTEM	
CONTROLLER STARTING ADDRESS (OCTAL) ____________ 00000	
INTERLEAVED □ YES □ NO	□ EVEN □ ODD
MEMORY SYSTEM SERIAL NO.	DATE INSTALLED

SLOT	MODULE TYPE	ADDRESSES	MODULE (GROUP) FUNCTION
01	G116 SIN	000002 → 0377776	2 STACK 0 HIGH WORD/ODD 6
02	H224—C STK		
03	G236 DRV		
04	G116 SIN	0400002 → 0777776	2 STACK 1 HIGH WORD/ODD 6
05	H224—C STK		
06	G236 DRV		
07	G116 SIN	1000002 → 1377776	2 STACK 2 HIGH WORD/ODD 6
08	H224—C STK		
09	G236 DRV		
10	G116 SIN	1400002 → 1777776	2 STACK 3 HIGH WORD/ODD 6
11	H224—C STK		
12	G236 DRV		
13	M8147 MCT		MEMORY CONTROL & TIMING
14	M8149 MXR		MEMORY TRANSCEIVER
15	G116 SIN	1400000 → 1777774	0 STACK 3 LOW WORD/EVEN 4
16	H224—C STK		
17	G236 DRV		
18	G116 SIN	1000000 → 1377774	0 STACK 2 LOW WORD/EVEN 4
19	H224—C STK		
20	G236 DRV		
21	G116 SIN	0400000 → 0777774	0 STACK 1 LOW WORD/EVEN 4
22	H224—C STK		
23	G236 DRV		
24	G116 SIN	000000 → 0377774	0 STACK 0 LOW WORD/EVEN 4
25	H224—C STK		
26	G236 DRV		

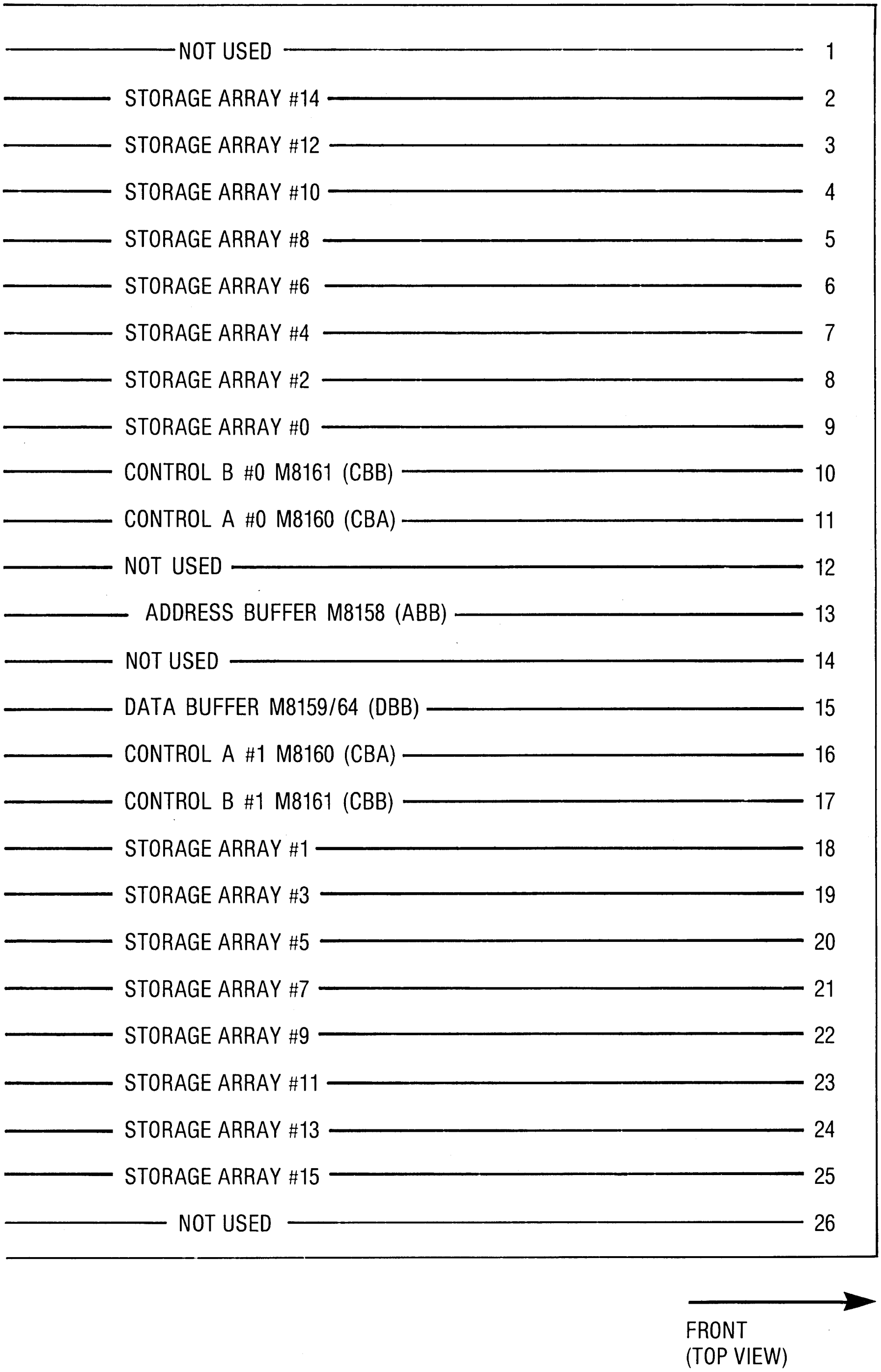
FIGURE 13.7 MJ11-B 32K STACKS

FIGURE 13.8
MJ11 WITH 32K STACKS



CHAPTER 14

MK11



NOT USED	1
STORAGE ARRAY #14	2
STORAGE ARRAY #12	3
STORAGE ARRAY #10	4
STORAGE ARRAY #8	5
STORAGE ARRAY #6	6
STORAGE ARRAY #4	7
STORAGE ARRAY #2	8
STORAGE ARRAY #0	9
CONTROL B #0 M8161 (CBB)	10
CONTROL A #0 M8160 (CBA)	11
NOT USED	12
ADDRESS BUFFER M8158 (ABB)	13
NOT USED	14
DATA BUFFER M8159/64 (DBB)	15
CONTROL A #1 M8160 (CBA)	16
CONTROL B #1 M8161 (CBB)	17
STORAGE ARRAY #1	18
STORAGE ARRAY #3	19
STORAGE ARRAY #5	20
STORAGE ARRAY #7	21
STORAGE ARRAY #9	22
STORAGE ARRAY #11	23
STORAGE ARRAY #13	24
STORAGE ARRAY #15	25
NOT USED	26

FIGURE 14.1

BACKPLANE SLOT ALLOCATION

ADDRESS RANGES FOR INTERNAL INTERLEAVE*

STORAGE ARRAYS	M7984 ARRAYS** ADDRESS RANGE	M8728 ARRAYS** ADDRESS RANGE
0,1	00000000 — 00377776	00000000 — 01777776
2,3	00400000 — 00777776	02000000 — 03777776
4,5	01000000 — 01377776	04000000 — 05777776
6,7	01400000 — 01777776	06000000 — 07777776
8,9	02000000 — 02377776	10000000 — 11777776
10,11	02400000 — 02777776	12000000 — 13777776
12,13	03000000 — 03377776	14000000 — 15777776
14,15	03400000 — 03777776	16000000 — 16777776

ADDRESSES ENDING IN 4,6 ARE ODD # ARRAYS (eg 1,3,5,7 ...)
ADDRESSES ENDING IN 0,2 ARE EVEN # ARRAYS (eg 0,2,4,6 ...)

*IF YOU HAVE AN EVEN # OF ARRAYS YOU ARE AUTOMATICALLY INTERNALLY INTERLEAVED. IF YOU HAVE AN ODD NUMBER OF ARRAYS ALL ADDRESSES FROM ARRAY 0 ARE USED THEN ARRAY 1, ARRAY 2 ETC.

**IF YOU HAVE A MIX OF M7984 AND M8728 FIGURE THE ADDRESS OUT BY COUNTING 400000 LOCATIONS FOR THE M7984 PAIRS AND 2000000 LOCATIONS FOR THE M8728 PAIRS.

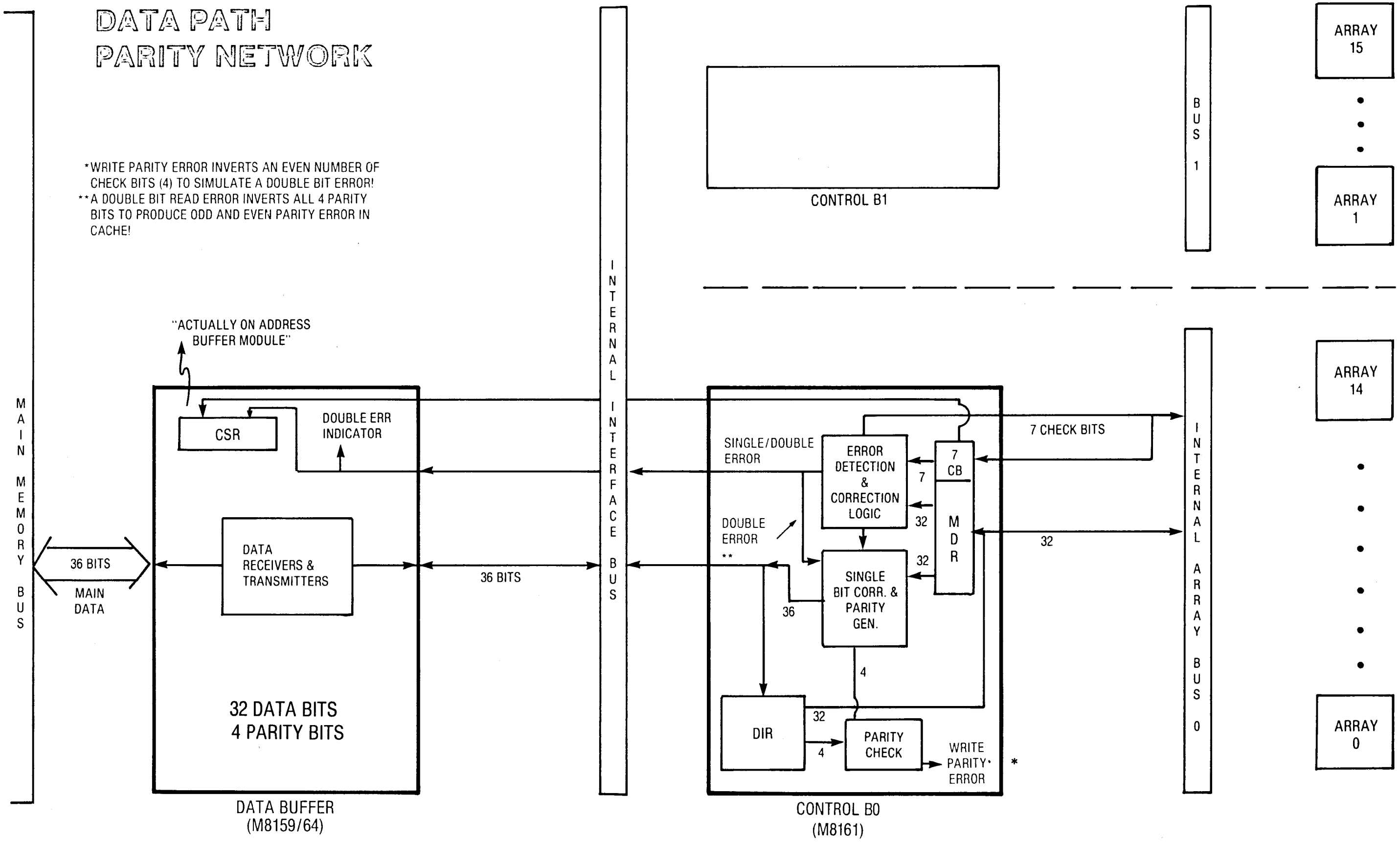
NOTE: CSR array # (CSR1<7:5>) and array side (CSR2<9>) will point to the failing array for the most recent SBE or DBE, with DBE's having priority over SBE's.

Example:

Condition in CSR	Condition occuring	CSR Points To?
No Error	SBE #1	SBE #1
No Error	DBE #1	DBE #1
SBE #	SBE #2	SBE #2
SBE #	DBE #1	DBE #1
DBE #	SBE #1	DBE #1
DBE #	DBE #2	DBE #2

This differs from the Cache Error Address registers ($40, $42) which are locked on the first DBE. Therefore if the CSR's array callout doesn't match the Cache Error Address register its possible the problem is a controller or write parity error problem.

MK11 DATA PATH PARITY NETWORK



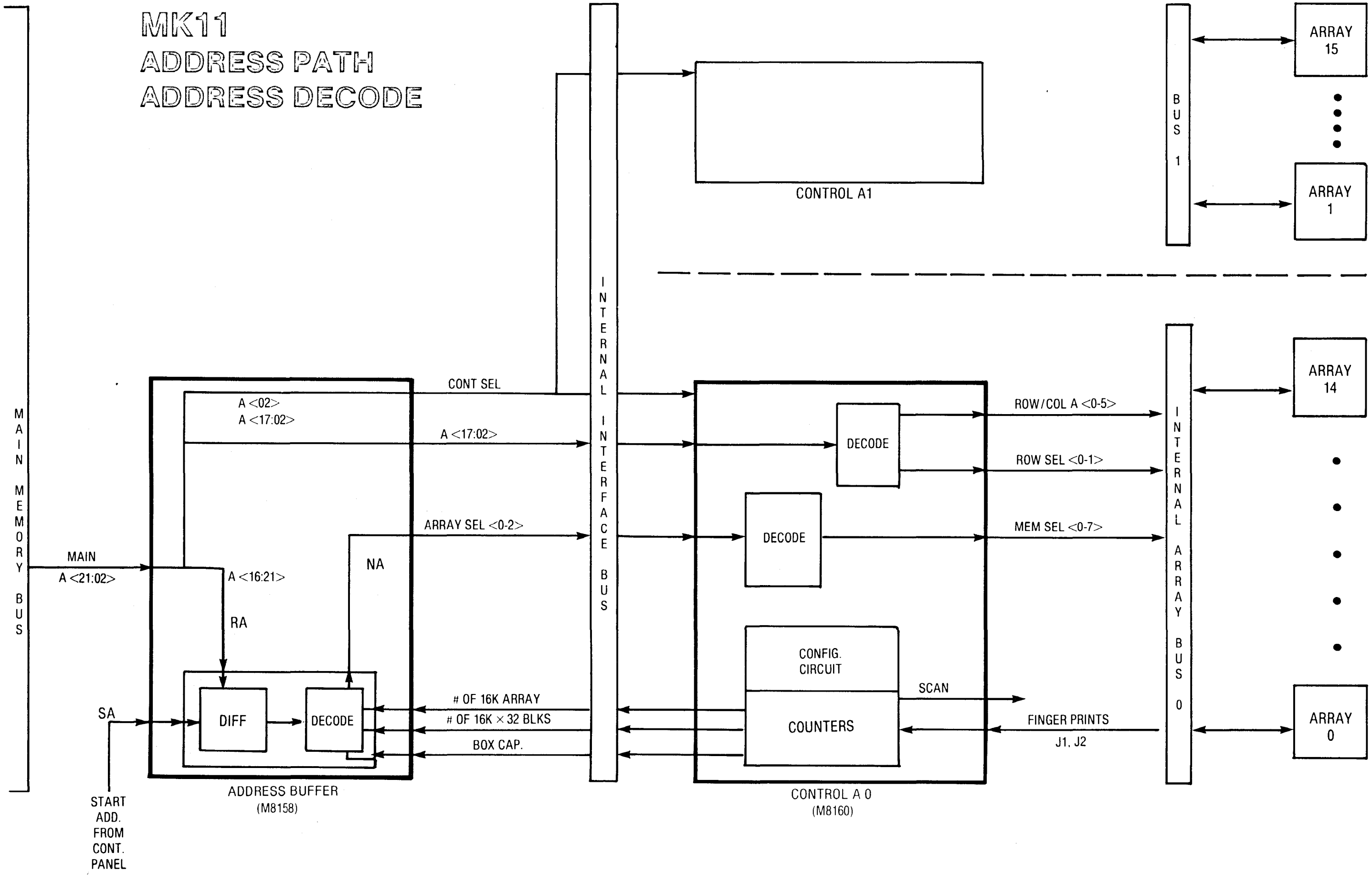
*WRITE PARITY ERROR INVERTS AN EVEN NUMBER OF CHECK BITS (4) TO SIMULATE A DOUBLE BIT ERROR!

**A DOUBLE BIT READ ERROR INVERTS ALL 4 PARITY BITS TO PRODUCE ODD AND EVEN PARITY ERROR IN CACHE!

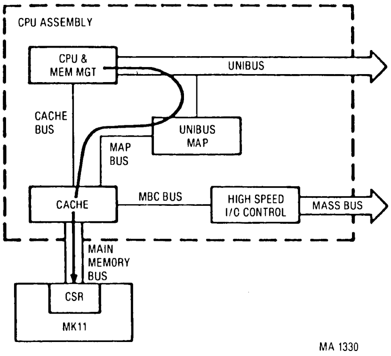
FIGURE 14.2

FIGURE 14.3

MK11 ADDRESS PATH ADDRESS DECODE



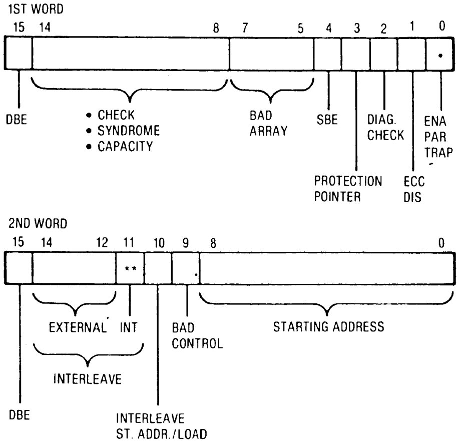
ACCESSING CSR'S



- Halt all NPR activity in system
- Turn off cache 17777746 ← 14
- Set up map 0 17770200* ← 170000
  17770202 ← 77
- Turn on Unibus map 17772516 ← 40
- Turn display switch to console physical
- Examine/deposit
  17002100 through 17002136**

*Sets up map to contain relocation constant.
**Unibus address from switch register to Unibus map which will be added to 17,770,000. This will produce I/O page address of CSR (17,772,100).

MK11 CSR'S



*NORMALLY A "1" ON POWER UP
**NORMALLY A "1" ON POWER UP WITH AN EVEN # OF CARDS

*NOTE: DBE DOESN'T NECESSARILY MEAN YOU HAVE A BAD ARRAY. MK11 DOESN'T TELL YOU WHEN IT RECEIVES BAD PARITY AND FORCES BAD ECC TO BE WRITTEN INTO A LOCATION. THIS RESULTS IN A DBE ON A SUBSEQUENT READ OF THAT LOCATION. BE CAREFUL CHANGING ARRAYS FOR INTERMITTANT DBE'S WHICH MAY HAVE BEEN CAUSED BY A PROBLEM IN THE DATA PATH TO THE MK11.

First Word

Bit 0* (Ena. Par. Trap) gets set on power up

- When set:
  Asserts bad parity on DBE
  Asserts bad parity on SBE with ECC DIS
- When cleared:
  Keeps processor from aborting on SBE/DBE

Bit 1* (ECC Disable)

- Allows SBE to be detected as uncorrectable.

Second Word

Bits <8:0>* (Starting Address on Box Controller Thumbwheel Switches)

Bit 9** (Control Select)

- Identifies array group error was found in
  0 = on right
  1 = on left

Bit 10* (Starting Address Controller)

- 0 = Source of starting address and ext. int. is box controller/sw. on data buffer mod.

Bit 2* (In diagnostic mode when = 1)

- On a read check bits of addressed DW → CSR <14:8>
- On a write <14:8> of CSR → CK bits of addressed DW.

Bit 3* (Protection Pointer Overrides Bits 1 and 2)

- 0 protects first 16K bank of memory
- 1 protects second 16K bank of memory

Bit 4* (Single Bit Error Indication)

Bits <7:5>** (Bad Array)

- When bit 9 of second word = 0 bad array on right
- When bit 9 of second word = 1 bad array on left

Bits <14:8>* (Check/Syndrome/Box Capacity)

- Bits <3:1> equal 010 = stores check bits
- Bits <3:1> equal 100 = box capacity is stored
- Bits <3:1> equal 000 = syndrome bit storage

Bit 15* (Double Bit Error)

*Can read or write this bit(s)
**Can only read this bit(s)

- 1 = Program control is initiated which allows a write to:
  <14:12> ext. interleaving
  <8:0> starting address
  <11> internal interleaving

Bit 11** (Internal Interleaving)

- Set on power up when number of array cards = even
- To uninterleave internally
  Set bit 10
  Clear bit 11
  must be in force panel mode
- If number of array cards are odd cannot be set

Bits <14:12> (External Interleaving)

- Bit 12 = two-way
- Bit 13 = four-way
- Bit 14 = ext. A02
- When bit 10 is 0 source of interleaving is control panel
- When bit 10 is 1 source of interleaving is under program control

Bit 15* (Double Bit Error)

*Can read or write this bit(s)
**Can only read this bit(s)

MK11 MAXIMUM MEMORY NOTES

Due to the I/O space taking up the top 128KW of the 2MEG total address space on the 1170 memory bus, you can not have more than 1920KW of physical memory in any number of MJ11/MK11 boxes that may exist on the bus. The reason is that any address in the 1920K to 2048K range will conflict with the addresses of the MK11 CSR's. Problems have arisen when attempts have been made to install a full set of 16, 128K arrays in a MK11 box. Adding the 16th array brings you to 2048K causing dual adressing between the last array and the MK11 CSR's in that box.

In these cases you must reduce memory size one of three (3) ways.

1. Pull one array going to a total of 15 arrays. This will kill the internal interleaving and give the max memory size of 1920KW.
2. Pull two (2) arrays (down to 14 total) reducing size to 1792K and allowing internal interleaving.
3. The best if possible. Go with two (2) 32KW arrays in arrays 0 and 1. Arrays 2-15 should be 128KW arrays. This give you 1856KW of total memory and again keeps internal interleaving in place.

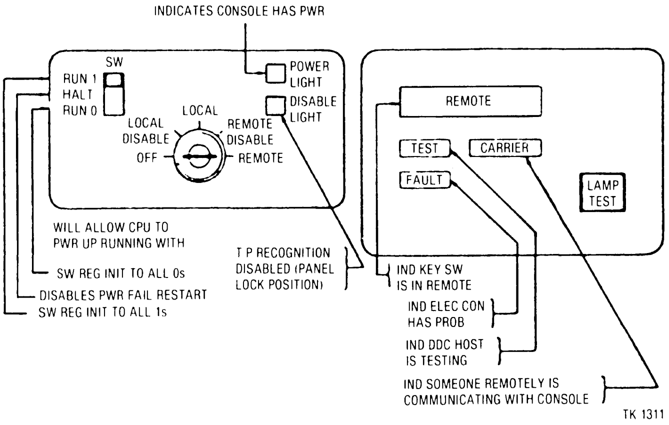
You can see that you can have the maximum of 1920KW but to do that you sacrifice the added speed of internal interleaving. By using STEP #3 above you only loose 64KW of memory but gain the speed of interleaving. If the customer must have the maximum allowed go to STEP #1.

SECTION 15

KY11-R

CONTENTS

PDP-11/70
ELECTRONIC CONSOLE COMMANDS



KEYSWITCH POSITION DESCRIPTION

OFF — Turns CPU and electronic console off.

LOCAL DISABLE — ^P recognition disabled, disables all console functions.

LOCAL — ^P recognition enabled, enables all console functions at local terminal.

REMOTE DISABLE — ^P recognition disabled, disables all console functions, forces local copy.

REMOTE — ^P recognition enabled, enables all console functions at remote terminal.

THREE SEPARATE STATES

Program I/O State	Command **Z**	Allows communication with program.
Console State	Command **^P**	Console communicates with CPU interface (accepts commands)
Talk State	Command **^L**	Allows two-way communication between local site and DDC (Digital Diagnostic Center) operator

NOTE

Console (V02-00 or greater) reverts to program I/O state when the following conditions occur:

- 20 seconds have elapsed without using console, and
- Program is running, and
- Character Ready bit is set, and
- Key switch in LOCAL.

LEGEND:
All RED symbols typed by human. All other symbols typed by CPU.

^P — Means typing the control key and the letter P at the same time to perform the command.

$ — Means typing $ to perform the command.

n — Is used to indicate that a numeric variable is REQUIRED.

[n] — Is used to indicate that a numeric variable is OPTIONAL.

m — Is content of console switch register.

<NL> — Means that a <CR> and <LF> were generated by the CPU.

CONSOLE CONTROL COMMANDS

^E — Prints ASCII text that identifies CPU and console code revision; **Example:** 11/70 V01-00.

^L — Sets talk state; ignores all characters except **P**.

^P — Sets console state; puts address display multiplexer to console physical; puts data display multiplexer to data paths.

[n] **R** — Will load and read control register; if data precedes **R** it will load it; if no data precedes **R** it will read it.

CONTROL REGISTER

5	4	3	2	1	0
*SUPRESSES REMOTE TYPEOUT	SPECIAL MODE	EXPAND PRINTOUT	*TALK ECHO	*LOCAL CONTROL	*LOCAL COPY
	SUPPRESSES • SYNTAX ERRORS • RUNNING ERRORS	ADDS AN "M" COMMAND AFTER EACH "N" COM	ENABLES LOCAL COPY IN TALK MODE	ENABLES LOCAL CONTROL IN REMOTE MODE	OUTPUTS LOCAL COPY IN REMOTE MODE

*THESE BITS CAN NOT BE SET FROM LOCAL TERMINAL.

TK-1224

^U Clears the data typed in the temporary data register.

$ Sets a flag to indicate a register versus a memory location;
Example: $n/xxxxxx
will examine contents of general register n.

, Data separator before a start or go command;
Example: 30,17765000G.

V Verifies console logic; will print V000377 if no fault is detected.

Z Sets serial line multiplexer to program I/O state.

<RO> Rub out; deletes the rightmost octal character from the temporary register. DEL KEY causes "rubout" action to occur.
Example: 1234\4\3\2\1\0\0
(For V01-00 consoles)
1234\4321\
(For V02-00 consoles)*

*On V02-00 the backslash only occurs on the first and last character being deleted from the buffer.

CPU CONTROL ODT-11 COMMANDS

n^D Dumps successive memory locations continuously, until a character is typed, starting at address n.

[n] <LF> Deposits any data typed and opens the next sequential location.

[n] <CR> Deposits any data typed and closes the location to further modification.

" Changes to hex data format from octal; data is displayed in hex while address display does not change.*

' Changes to octal data format from hex.*
Example: ^P
CON = 0/123456"
00000000/"A72E'
00000000/123456

n/ Performs load address and examine; effectively opens word location n.

: Reserved for future expansion.

[n] @ Deposits any data typed and uses contents of currently open location as address of location to open next.

B Reserved for future expansion.

[m], **nG** Go command begins at address n with switch register set to m; places console in program I/O.
Example: 30,17765000G

P Proceed command places console in program I/O. It continues program from address in program counter.

n Performs load address and examine; effectively opens byte location n.
Example:*
```
^P
CON = 0/XXXXXX 123456 <CR>
CON = 0\056 <LF>
CON = 1\247 <CR>
CON =
```

*Command is available on consoles with an ID of V02-00 or greater.

CONSOLE MESSAGES

"? SYN ER" Syntax error; command is given out of sequence or with missing character.

"? RUN ER" Illegal when CPU running; certain commands are illegal with CPU running.
Example: Start, load address, and initialize.

"? ER/TXXXXX" Memory reference error; address/data parity error detected during a memory reference.

Fault Indicator Console logic fault;

	console logic has detected an error within itself.
"+"	Serial line error; identifies framing or overrun errors with a "+".
"#"	CPU response time out; # is printed if CPU did not respond to a console command.
"*HXXXXXXXX/TXXXXX"	Programmed halt; console will print "halt notification" and HALT ADDR/ status.
"? CAR ER"	Carrier lost.

CPU CONTROL BASIC COMMANDS

[n] **A** Prints address display in octal as received from one of eight inputs to address multiplexer.
n = 0 Program Physical
n = 1 Kernel Data
n = 2 Kernel Inst.
n = 3 Console Physical
n = 4 Supervisor Data
n = 5 Supervisor Inst.
n = 6 User Data
n = 7 User Inst.

NOTE

∧**P** or "Power On" set position 3; if octal # precedes **A**, new position is set.

C Continues program from address stored in program counter.

[n] **D** Deposit; sequential deposits are possible (location must be open).

E Examine; sequential examines are possible.

H Halt; address counter and CPU status will be printed.

I Initialize CPU.

J Set single bus cycle.

K Reset single bus cycle.

nL Load address n.

[n] **M** Prints data registers:
n = 0 Bus Register
n = 1 Data Paths (shifter)
n = 2 Display Register
n = 3 CPU μADRS

NOTE

Position 1 is set by ∧**P, E** or "Power on." To change data path's position, type new position n first then type **M**

N Execute next instruction/bus cycle.

R Read switch register.

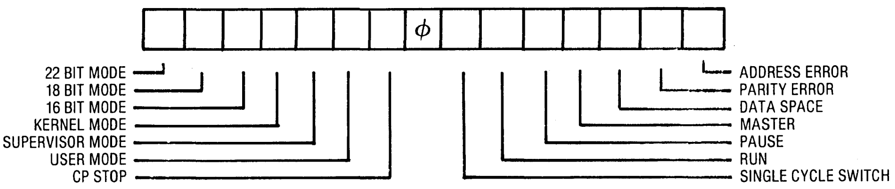
[m,] **nS** Start the CPU at the address n with switch register set to **m**.

T Read CPU status.

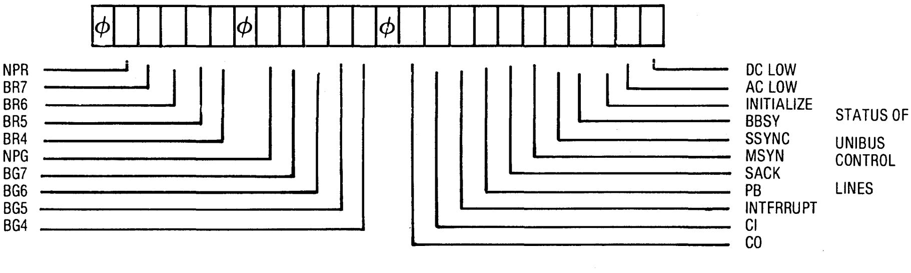
U Read UNIBUS status.

nW Write **n** into switch register.

CPU STATUS



UNIBUS STATUS



EXAMPLES

```
     BASIC COMMANDS                 WITH ODT

START PROGRAM
       ΛP <NL>                        ΛP <NL>
 CON = 200S <NL>               CON = 200G
 CON = Z
CONTINUE PROGRAM
       ΛP                             ΛP
 CON = C Z <NL>                CON = P <NL>
READ ROM STATE
       ΛP <NL>
 CON = 3MXXXXXXXX
DISPLAY ADRS IN CON PHY
       ΛP <NL>
 CON = AXXXXXXXX
TO EXAMINE
       ΛP <NL>                        ΛP <NL>
 CON = 4L E 051531             CON = 4/051531
TO DEPOSIT
       ΛP <NL>                        ΛP <NL>
 CON = 4L E 051531 77D         CON = 4/051531 77 <CR>
TO HALT
       ΛP <NL>
 CON = H 01731240/T14410
TO CK FOR RUNNING CPU
       ΛP
CON = T XX0XX CPU IS RUNNING
```

An example of how to single step a program using the electronic console is shown below.

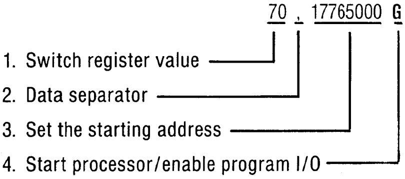
```
ΛP
CON = 1000/XXXXXX   12737<LF>     This program
00001002/XXXXXX     101<LF>       will print the
00001004/XXXXXX     177566<LF>    letter A
00001006/XXXXXX     0<CR>
CON = 1000L I N00001006_A
                            ↑
                            means program moved
character to terminal print buffer not a human
```

```
     BASIC COMMANDS                 WITH ODT

HEXADECIMAL DEPOSIT
 ONLY POSSIBLE
 UNDER ODT                     ΛP
                               CON = 0/XXXXXX"
                               0000000/"XXXX 2AF1<LF>
                               0000002/"XXXX 2AF2<CR>
HEXADECIMAL DISPLAY
 ΛP                            ΛP
 CON = "0L E"XXXX<CR>          CON = 0/XXXXXX"
 CON = E"XXXX'<CR>             00000000/"XXXX<LF>
 CON = EXXXXXX                 00000002/"XXXX<CR>
HEXADECIMAL CONTINUOUS EXAMINES
 ΛP
 CON = " 0 ΛD
 00000000/"XXXX XXXX . . . . .
OPEN BYTE LOCATION IN HEX
 ΛP
 CON = 0\XXX"
 00000000\"XX<LF>
 00000002\"XX
```

```
OPEN BYTE LOCATION AND DEPOSIT
 ΛP
 CON = 0\XXX<LF>
 00000001\XXX<LF>
 00000002\XXX 24<LF>
 00000003\XXX
OPEN BYTE LOCATION IN A REGISTER
 ΛP
 CON = $0\XXX<LF>
 177777700H\XXX<LF>
 177777701L\XXX
```

TO BOOT THE SYSTEM

Bootstrap Example with initial switch value for an RP04 (M9301-YC)



70 , 17765000 G

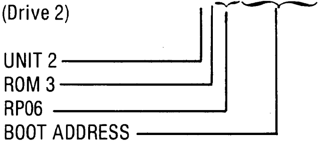
1. Switch register value
2. Data separator
3. Set the starting address
4. Start processor/enable program I/O

M9301-YC BOOTSTRAP (Type **P H** before attempting to boot.)

10,17765000G	TM11
20,17765000G	TC11
30,17765000G	RK05
40,17765000G	RP03/RP05/RP06
50,17765000G	RK06/RK07
60,17765000G	TU16
70,17765000G	RP04
100,17765000G	RS04
110,17765000G	RX01

Bootstrap **Example** with an initial switch value for an RP (M9312)

Conditions	**Using Basic Commands**
1. • Diagnostic ROM in E20 • RP06 ROM in E35 socket (ROM 1) • Unit 0 (Drive 0)	56,17765744G
2. Same as in #1 except for RP06 ROM in E34 socket (ROM 3)	456,17765744G
3. Same as in #2 except for Unit 2 (Drive 2)	2456,17765744G



UNIT 2
ROM 3
RP06
BOOT ADDRESS

KY11R Troubleshooting Procedure (Local)

Symptom: Turn key off to Local or Local Disable, All lights except Carrier come on and stay on
Causes: M8255 console board, cable from J1 of M8255 to J4 of 54-12781
Remedy: Replace M8255, Replace 54-12781 MUX board, Check cable hookup and connector seating

Symptom: Fault lite stays on after power up and all other lites go out
Causes: M8255 console board, Console cables
Remedy: Replace M8255, Replace MUX, Check cable in connector J4

Symptom: Console prints nothing, Fault lite is not on
Causes: Console cable at J5
Remedy: Check cables from J5 of MUX to the terminal, Check baud rate switches on Mux

Symptom: Console prints V, but then prints con error
Causes: M8255 board, Cables
Remedy: Replace M8255, Check J1 of M8255 board to J4 of MUX

Symptom: Console prints V000377 but still gives a Fault
Causes: LTC on M8255, No LTC slot
Remedy: Replace M8255, Check for LTC from H7420

Symptom: Console prints out to many characters (i.e. V0000003777)
Causes: M8255
Remedy: Replace M8255

Symptom: Console prints out wrong characters (i.e. V000375 or V001377), Fault lite is not on
Causes: Console board, M8255
Remedy: Replace 54-12781 (MUX), M8255

Symptom: Console prints strings of V000377
Causes: M8255
Remedy: Replace M8255

Symptom: Characters print out on terminal correctly, but there is not activity from keyboard
Causes: MUX board
Remedy: Replace MUX (54-12781)

Symptom: Console will not accept some or any console commands, although they echo back properly
Causes: M8255, +5 Volts
Remedy: Replace M8255, +5 Volts on M8255

KY11R Troubleshooting Procedure (Local)

Symptom: A "#" symbom printed on console when executing any command to PCU (i.e. 200G#, H#, 1000L#)
Causes: Cable loose or pins broken
Remedy: Check J1, J2, J3 on MUX and CPU boards (M8140, M8134), Replace M8140, Replace M8134

Symptom: Fault lite comes on after a few minutes of operation
Causes: MUX board
Remedy: MUX board

Symptom: Garbled character printed on console when typing on keyboard, Receives characters correctly
Causes: MUX, +15 Volts, -15 Volts
Remedy: Replace MUX

Symptom: Fails Run 1-Halt-Run 0 power up
Causes: MUX board, M8255
Remedy: Replace MUX, Replace M8255

Symptom: Prints H after typing N for single-step operation
Causes: CPU, MUX board
Remedy: Replace M8140, Replace MUX, Replace M8255

Symptom: Occasional *H printed on console terminal, CPU in hung, no address printed after H
Causes: Static eliminator, Grounding, M8255
Remedy: Bad or no static filter, Check for equipment (rack to rack) and earth grounds, Replace M8255

KY11R Troubleshooting Procedure (Remote)

Symptom: DAA does not answer when dialed in test mode
Causes: +12, -12 Volts, to DAA
Remedy: Replace MUX, Check +15m -15 Volts, Check DAA connections, Bad DAA

Symptom: DAA busy in test mode
Causes: Called once, DAA, Phone line
Remedy: Turn switch to Local and to Test, Have DAA checked, Have phone line checked

Symptom: DAA answers to test, but not online
Causes: Keyswitch, Modem cable, Phone line
Remedy: Put keyswitch in Remote or Remote Disable, Check and reseat cables, Replace modem (54-12498)

Symptom: DAA answers in test mode, but goes busy when online
Causes: Wrong DAA cables to DAA, Modem cable, LA36 modem
Remedy: Check for 1001d label, Check connections to DAA, Check cable seating, Replace modem (54-12498)

Symptom: DAA answers online, but no carrier
Causes: LA36 modem, Modem cable
Remedy: Replace modem, Check modem cable

Symoptom: DAA unable to connect, Carrier not received from remote console
Causes: DAA, MUX, Modem, Voltages
Remedy: Check if in test mode DA, Replace MUX, Replace modem, Check +-12v

Symptom: DDC connects but unable to establish protocol, Console has Carrier lite on for a short time
Causes: LA36 modem, Cable, M8255
Remedy: Replace modem, Check seating of J6, Replace M8255

Symptom: DCC connects but unable to establish protocol, Console does have the Carrier lite on
Causes: LA36 modem, Cable
Remedy: Replace modem, Check J6 seating

Symptom: No local control of console, Talk between DDC and remote terminal does not function
Causes: Console Uart, Console E17
Remedy: Replace MUX

Symptom: Unable to down-line load any program to remote console
Causes: M8255, CPU
Remedy: Check M8255, NPR circuitry

CHAPTER 16

PM PROCEDURES

DEVICE: PDP 1170 CPU and Memory Subsystem

Page 1 of 5

†	OPERATION	TEST POINT	NOMINAL											
A	1. Check CPU Regulators — Regulator A	F02A2	+ 5V ± 0.1 200 MV PP											
	— Regulator B	F09A2	Same											
	— Regulator C	F15A2	Same											
	— Regulator D	F44A2	Same											
	— Regulator H	F22A2	Same											
	— Regulator J	F18A2	Same											
	— Regulator K	F28A2	Same											
	— Regulator L	F35A2	Same											
	— Upper H7420	B01B1	+ 8V ± 1.2 240 MV PP											

Page 2 of 5

DEVICE: PDP 1170 CPU and Memory Subsystem

†	OPERATION	TEST POINT	NOMINAL											
	— Upper H7420	E13A1	+15V ± 1.5 450 MV PP											
	— Lower H7420	E13B2	- 15 ± 1.5 450 MV PP											
	2. Check MK11 Regulators — Regulator A	J18-4	+ 5 VBB 4.9 to 5.3											
		J18-5	+ 12 VBB 11 to 13											
		J18-7	- 12 VBB -10.5 to -13.5											
	— Regulator B	J21-5	+ 5 VBB 4.9 to 5.3											
		J21-4	+ 12 VBB 11 to 13											
		J21-2	- 12 VBB -10.5 to -13.5											
	— Regulator C	J21-6	+ 5 VBB 4.9 to 5.3											

DEVICE: PDP 1170 CPU and Memory Subsystem

Page 3 of 5

†	OPERATION	TEST POINT	NOMINAL											
		J21-3	+ 12V BB 11 to 13											
		J21-1	- 12 VBB -10.5 to -13.5											
	H7441 Regulator	J21-7	+ 5 V 4.9 to 5.3											
	3. Check MJ11 Regulators Regulator #	M8149-TP1	+ 5V ± .25 200 MV PP											
	Regulator #2	-TP2	+ 5V ± .25 200 MV PP											
	Regulator #3	-TP3	+ 20V ± 1.0											
	Regulator #1	-TP4	+ 20V ± 1.0											
	Regulator #3	-TP5	- 5V ± .25											
	Regulator #1	-TP6	- 5V ± .25											

Page 4 of 5

DEVICE: PDP 1170 CPU and Memory Subsystem

†	OPERATION	TEST POINT	NOMINAL											
A	4. Check All Fans													
A	5. Check MK11 Battery Backup units for proper operation.													
	6. Perform 1170 timing margin check; a) Connect Jumper	F13J1 to E21T1												
	b) Set R162 on M8139 for slow margin	D13T2	250 ns											
	c) Run 1170 instruction exerciser EQKC		one pass											
	d) Set R162 for Fast margin	D13T2	140 ns											
	e) Rerun EQKC		one pass											
	f) Set R162 to normal	D13T2	150 ns											
	g) Disconnect Jumper from F13J1 to E21T1													

DEVICE: PDP 1170 CPU and Memory Subsystem

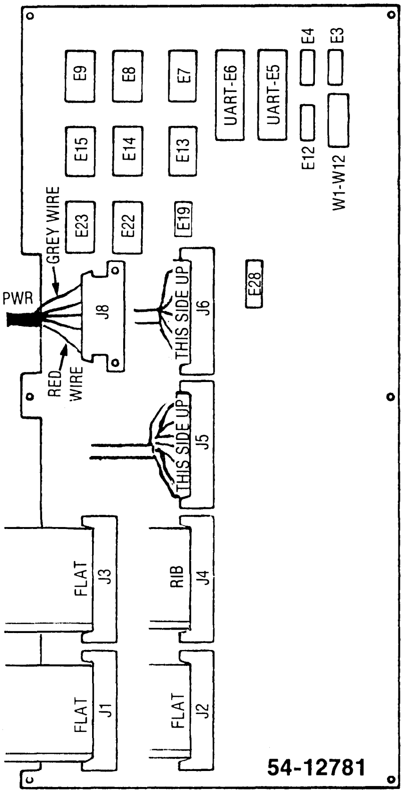
Page 5 of 5

†	OPERATION	TEST POINT	NOMINAL											
A	7. Run EMKA (halt by setting SWR bit 8)		2 passes											
A	8. Power MK11 off/on + return system to customer.													

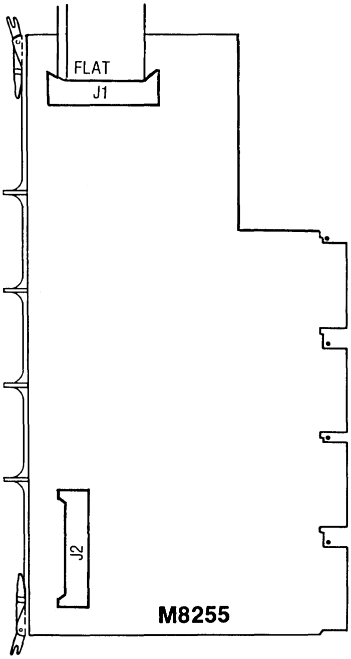
CHAPTER 17

CABLING

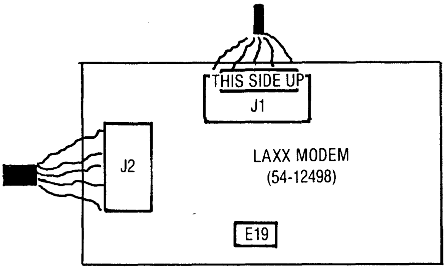
KY11-R CABLING



54-12781-J1	TO	M8134-J1
54-12781-J2	TO	M8140-J1
54-12781-J3	TO	M8140-J2
54-12781-J4	TO	M8255-J1
54-12781-J5	TO	LOCAL TERMINAL
54-12781-J6	TO	54-12498-J1 (or EXTERNAL MODEM)
54-12781-J8	TO	P1 POWER

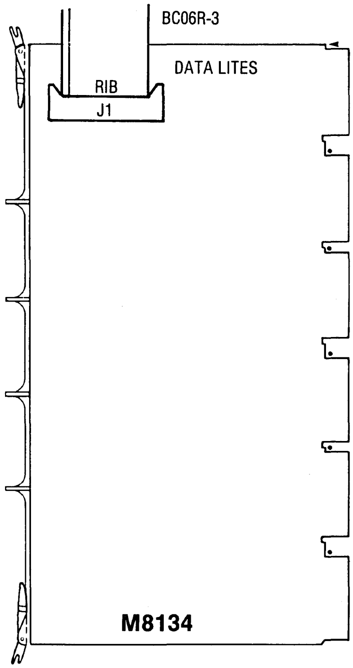


M8255-J1	TO 54-12781-J4
M8255-J2	FOR FACTORY USE ONLY



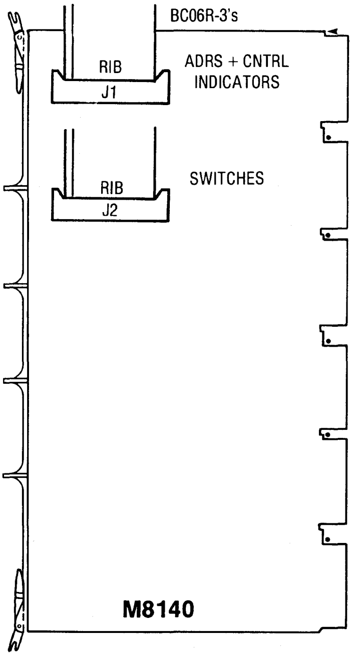
54-12498-J1	TO	54-12781-J6
54-12498-J2	TO	DAA

CONSOLE CABLING



M8134-J1 TO 54-11294-J1 (or 54-12781-J1)*

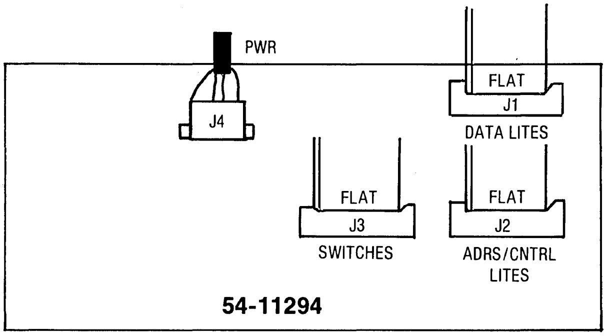
*If RDC front panel.



M8140-J1 TO 54-11294-J2 (or 54-12781-J2)*
M8140-J2 TO 54-11294-J3 (or 54-12781-J3)*

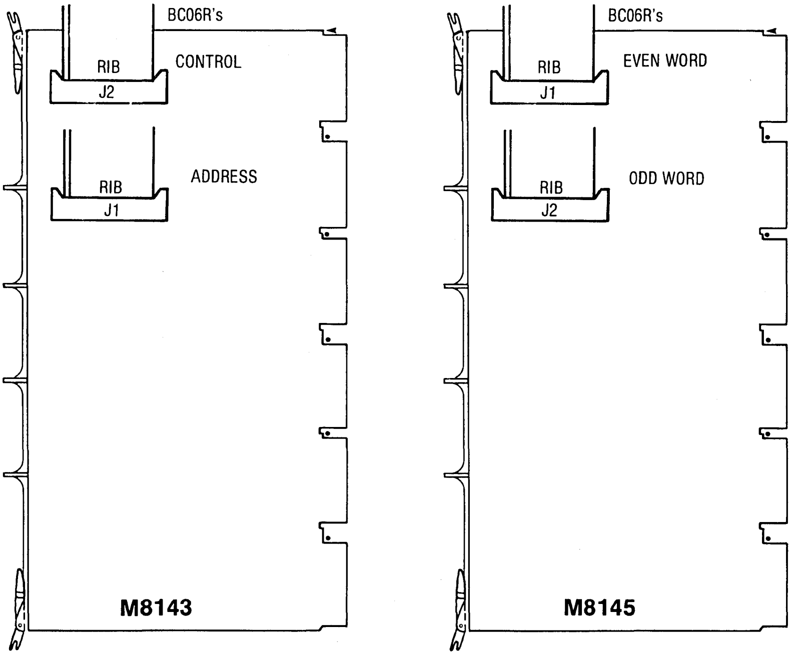
*If RDC front panel.

STANDARD CONSOLE CABLING



54-11298-J1	TO	M8134-J1
54-11294-J2	TO	M8140-J1
54-11294-J3	TO	M8140-J2
54-11294-J4	TO	P1 POWER

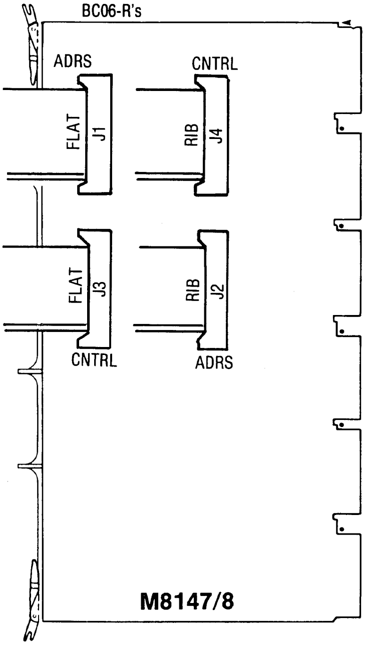
CACHE CABLING



M8143-J1	TO	M8147/8-J1 or M8148-J1
M8143-J2	TO	M8147/8-J3 or M8158-J3

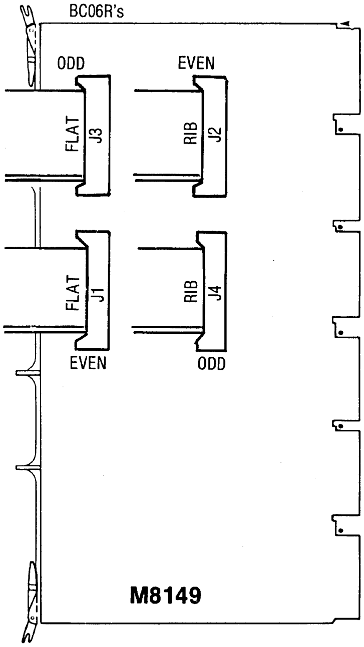
M8145-J1	TO	M8149-J1 or M8159/M8164-J1
M8145-J2	TO	M8149-J3 or M8159/M8164-J3

MJ11 CABLING



M8147/8-J1 FROM M8143-J1 (or M8147/8-J2 or M8158-J2)*
M8147/8-J3 FROM M8143-J2 (or M8147/8-J4 or M8158-J4)*
M8147/8-J2 TO TERMINATOR (or M8147/8-J1 or M8158-J1)*
M8147/8-J4 TO TERMINATOR (or M8147/8-J3 or M8158-J3)*

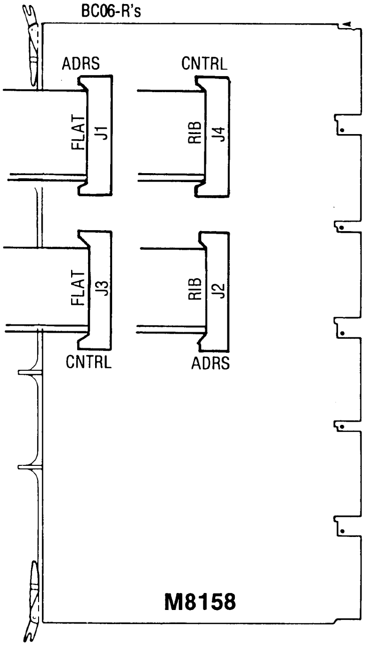
*For more than one memory box.



M8149-J1 FROM M8145-J1 (or M8149-J2 or M8159/M8164-J2)*
M8149-J3 FROM M8145-J2 (or M8149-J4 or M8159/M8164-J4)*
M8149-J2 TO TERMINATOR (or M8149-J1 or M8159/M8164-J1)*
M8149-J4 TO TERMINATOR (or M8149-J3 or M8159/M8164-J3)*

*For more than one memory box.

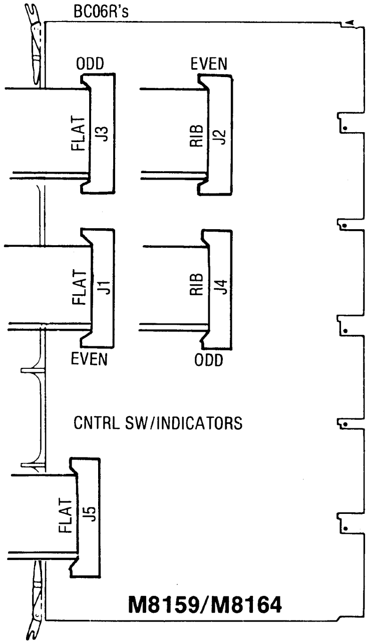
MK11 CABLING



M8158-J1 FROM M8143-J1 (or M8158-J2 or M8147/8-J2)*
M8158-J3 FROM M8143-J2 (or M8158-J4 or M8147/8-J4)*
M8158-J2 TO TERMINATOR (or M8158-J1)*
M8158-J4 TO TERMINATOR (or M8158-J3)*

*For more than one memory box.

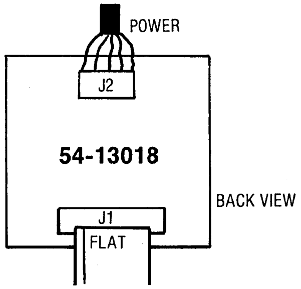
NOTE: M8158-J2 and J4 are not shown as going to M8147/8-J1 and J3 for multi-box configurations because MJ11 must be physically closer to CACHE than MK11.



M8159/M8164-J1 FROM M8145-J1 (or M8159/M8164-J2 or M8149-J2)*
M8159/M8164-J3 FROM M8145-J2 (or M8159/M8164-J4 or M8149-J4)*
M8159/M8164-J2 TO TERMINATOR (or M8159/M8164-J1)*
M8159/M8164-J4 TO TERMINATOR (or M8159/M8164-J3)*
M8159/M8164-J5 TO MK11 CONTROL PANEL — J1

*For more than one memory box.

NOTE: M8159/M8164-J2 and J4 are not shown as going to M8149-J1 and J3 for multi-box configurations because MJ11 must be physically closer to CACHE than MK11.

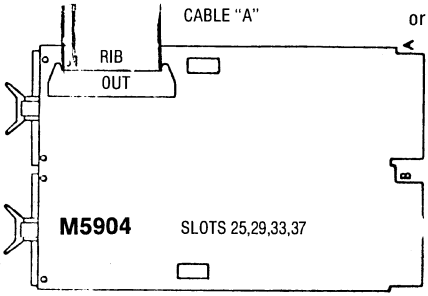


54-13018-J1 TO M8159/M8164-J5
54-13018-J2 TO POWER CONNECTOR J21

RH70 CABLING

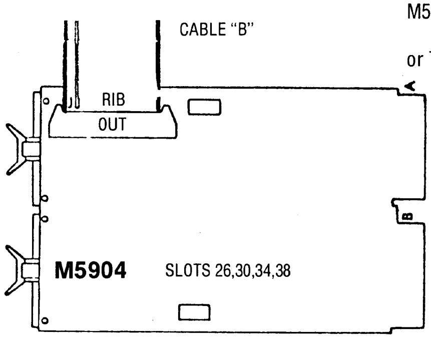
M5904-J1 TO "A" CONNECTOR OF TRANSITION BRACKET

or TO M5903 SLOT 4 OF TM02/3



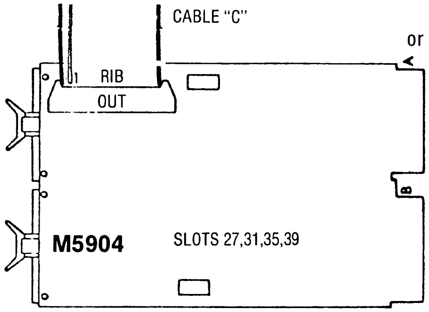
M5904-J1 TO "B" CONNECTOR OF TRANSITION BRACKET

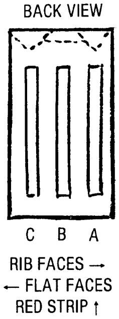
or TO M5903 SLOT 5 OF TM02/3



M5904-J1 TO "C" CONNECTOR OF TRANSITION BRACKET

or TO M5903 SLOT 6 OF TM02/3





TRANSITION BRACKET "A" TO M5904-J1 SLOTS 25,29,33,37
TRANSITION BRACKET "B" TO M5904-J1 SLOTS 26,30,34,38
TRANSITION BRACKET "C" TO M5904-J1 SLOTS 27,31,35,39

SECTION 18

TROUBLESHOOTING

CONTENTS **PAGE**

TROUBLESHOOTING

1) DECX MEMORY MARGIN PROCEDURE

DECX doesn't normally allow you to modify the Cache maintenance register because it clears it when you type RUN. In order to get around this we must NOP this clear out. The following is the absolute locations for the 4 current monitors:

XMONA0 ("E" Monitor)

Location	Is	Change to
6514	5077	240 (NOP)
6516	172502	240 (NOP)
6520	207	207 (Don't alter)

XMONB0 ("E" Monitor)

Location	Is	Change to
6356	5077	240 (NOP)
6360	XXXX	240 (NOP)
6362	207	207 (Don't alter)

XMONC0 ("E" Monitor)

Location	Is	Change to
7032	5077	240 (NOP)
7034	XXXX	240 (NOP)
7036	12600	12600 (Don't alter)

XMOND0 ("E" Monitor)

Location	Is	Change to
7110	5077	240 (NOP)
7112	XXXX	240 (NOP)
7114	12600	12600 (Don't alter)

To utilize this use the following procedure:

1. Load and start your DECX11.
2. After the CMD> prompt utilize the MOD command to implement the changes.
3. Then use the MOD command to set the maintenance register.

```
CMD> MOD 177750
177750/000000 XXXXXX
```

Where XXXXXX is:

	MJ11	MK11
0 =	Normal	Normal
4 =	Early strobe	Early MDR load
6 =	Late strobe	Late refresh
10 =	Low current	Normal
12 =	High current	Normal

4. At this point you can now run DECX11 with the memory margins you desire.
5. If you suspect a particular bank of memory, utilize the RUNL command to lock DECX11 into that area.

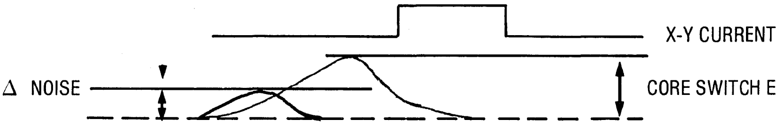
MJ11 CORE MEMORY TUNING PROCEDURE

A) The option consists of the following modules:

MJ11-B		MJ11-A
G236	X-Y Driver	G235
G116	Sense/Inhibit	G114
H224-C	Core Stack	H217-C

B) There are two adjustable components present in the memory.

1) Bias current supply controls the amplitude of the read and write currents. Variation of the currents affects the 0 switching noise level, the 1 switch voltage amplitude, and the time relationship between read current start time and the core output voltage peak.



Bias current is controlled by jumper changes on the G236 module. There are three such jumpers and each has a varying effect on the stack currents.

JUMPER EFFECT:
Maximum effect to least effect: W5,W7,W6.

a) All jumpers in = minimum stack current.
All jumpers out = maximum stack current.

b) Least significant change (W6) = 14 mA = 7 mV.

c) Bias reference voltage is measured on the G236 between AK2 and AA2. It is important that AA2 reference be on the same module.

d) The bias reference voltage is nominally set for 340 mV at 25 degrees Centigrade. (77° F). (365 mV for the G235)

e) If the memory box is hotter, make the bias voltage lower by 1 mV per degree C. (1° C. equals approximately 1.8° F.)

f) The stack should be idle when checking bias voltage levels.

g) Higher bias currents and higher temperatures cause the cores to switch faster. (Affects Strobe Timing.)

2) Strobe timing affects the lead time occurance of the strobe window. (End time is fixed.) Output of the memory core during the window will be interpreted as a "1" if it meets threshold requirements.

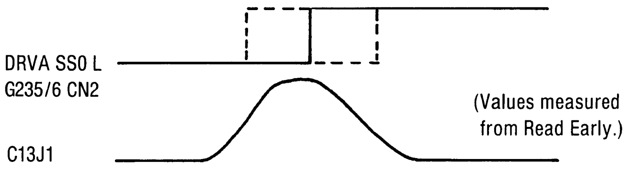
Strobe timing is controlled by jumper changes on the G236 module. There are four such jumpers that affect the strobe timing.

JUMPER EFFECT:
Maximum effect to least effect: W1, W4, W3, W2.
(G235 = W2, W1, W4, W3.)

a) All jumpers in = earliest strobe.
All jumpers out = latest strobe.

b) The least significant change = 3 nSec.

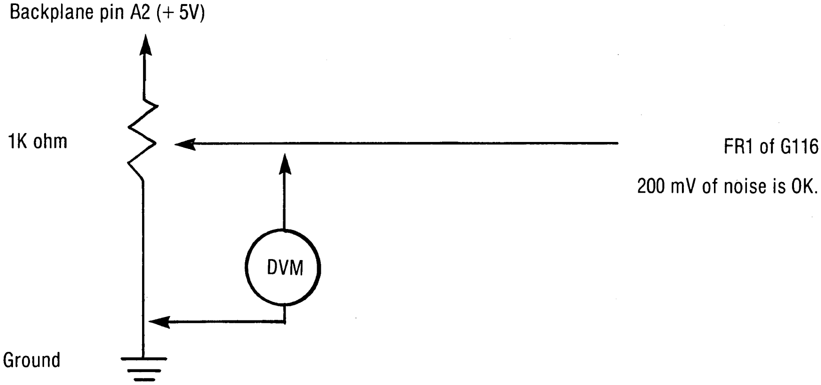
c) Range = 300 to 350 nSec. MJ11-B. (225 to 275 nSec. MJ11-A)



C) The core output voltage is an analog voltage; to digitize the output it is compared to a treshold standard voltage. This voltage is normally constant.

Threshold margining theory maintains that the quality of the parameters affected by the adjustable components can be monitored and optimized by varying the threshold and modifying the bias and strobe jumpers to expand failure limits.

Normally, the threshold input to the sense amps is a constant 1.7 mV level produced on the G116 module. For margining purposes, we can manipulate this voltage by connecting a test potentiometer as follows:



D) The diagnostic used to check the failure limits is EMKA in the Field Service Mode:

1) Control F
2) Command 5
3) Bank = Address/16K (100000)
4) Margin variations:
SWR = 4000 = No margins
SWR = 0 = All Margins

Note: When a test failure occurs, the test must be restarted since the memory control register 17777750 gets zeroed. Margin control occurs in bits 1 thru 3 of this register.

E) Procedure for accomplishing memory tuning.

1) Adjust the +5V, -5V, and +20V to 5% of spec.
2) Adjust the bias voltage for 340 mV nominal. (365 mV for G235.)
3) Connect the Vth potentiometer and adjust to 1.7V.
4) Start the diagnostic with margin 0.
5) Margin Vth toward 1.0v, until it fails and record the voltage.
6) Restart the diagnostic (margin 0), and margin the Vth toward 2.5v, until it fails and record the voltage.
7) Analyze the failure limits and modify the diagnostic selection to improve the failure limits.

   FAILED LOW VTH
   Strobe too early
   G116 Bad
   Bias too high

   FAILED HIGH VTH
   Strobe too late
   Bad G116
   Bias too low

   FAILED BOTH VTH DIRECTIONS
   Bias too high
   Bad G116

8) Modify jumpers dependent on the results of running the affect strobe requirements.
9) Return to steps 4, 5, and 6; retest to final criteria. If unsuccessful, reperform steps 7, 8, and 9.

   FINAL CRITERIA:
   Margin = 0, Vth from 1.0v to 2.5v
   = all, Vth from 1.3v to 2.3v

   Note: If the margins are tightened further, ie. .8v to 2.6v, a large trade off of G116 modules will occur. (50% D.O.A.)

BIAS JUMPERS: (In = 1; Out = 0.)

W7	**W6**	**W5**	...... G236 Module and G235 Module
0	0	0	Highest current
0	1	0	
1	0	0	
1	1	0	
0	0	1	
0	1	1	
1	0	1	
1	1	1	Lowest current

STROBE JUMPERS: (In = 1; Out = 0.)

W4	W3	W2	W1	.... G236 Module
W1	W4	W3	W2	.... G235 Module
0	0	0	0	Latest Strobe
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	
0	0	0	1	
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	1	
1	0	1	1	
1	1	0	1	
1	1	1	1	Earliest Strobe

Note: When ever you remove a module to change a jumper, be sure to allow 5 minutes warm-up before proceeding with tests.

11/70 CPU TIMING MARGINS PROCEDURE

Perform the timing margin procedure as follows:

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

b) Set up oscilloscope as follows:

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

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

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

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

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

f) Repeat step d) from above.

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

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

Proceed with normal acceptance testing.

CPU VOLTAGE MARGINS

All CPU and RH70 logic should run error free when the 5V power regulator is in the 4.75V to 5.25V range. Sometimes intermittent problems can be made solid when setting the +5 regulator to it's hi and low extremes.

Procedure

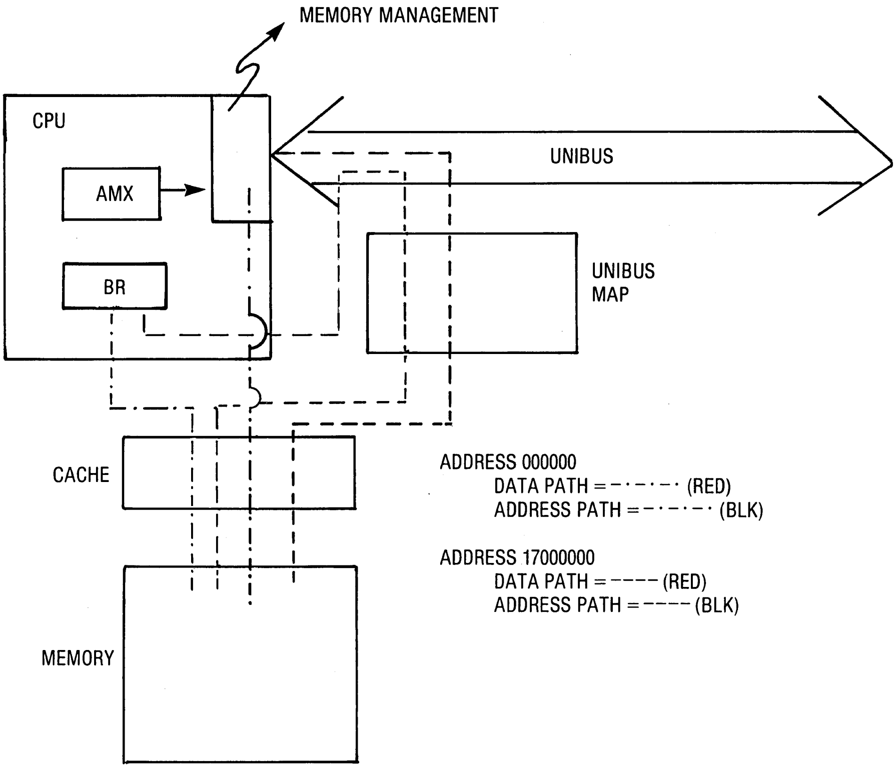
1) Set all +5V regulators (CPU and used RH70) to 4.75V. Use chart on page 26.
2) Run EQKC for half an hour. Then DECX for half an hour (exercise all RH70's).
3) If errors occur, readjust regulators one at a time until problem disappears. By using chart on page 26, find bad module.
4) If no errors, repeat above procedure, but adjust to 5.25 in step 1.

   Good Luck!!

DATA PATH CHECK

1) Assure memory location 0 is good.
   Deposit all ones
   Examine all ones
   Deposit all zeros
   Examine all zeros
2) Deposit known pattern in location 0.
   Load address 000000
   Deposit 070707
3) Read location 0 via Unibus Map.
   Load address 17000000
   Examine
   You should read 070707

   If data differs, one of the data or address paths is bad. See diagram below.
4) Reverse the procedure, i.e. deposit known pattern in location 17000000 and examine location 000000. Again the data should be the same.



CUSTOMER CRASH/HALT SHEET

STANDARD FRONT PANEL

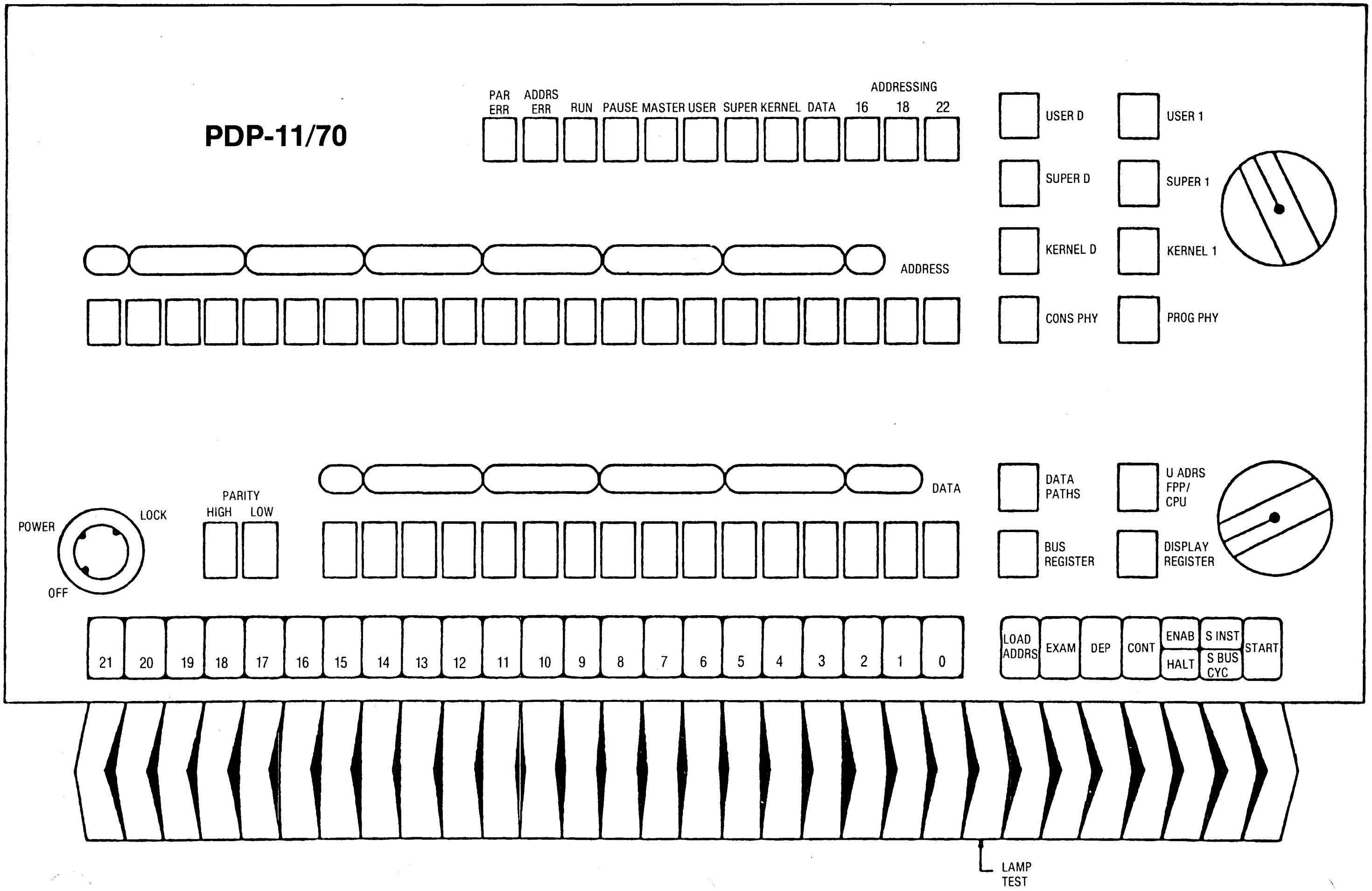
If your system halts please collect the following information prior to restarting your system.

1. Without touching anything please record which lights are on using the attached sheet.
2. Press the halt key down.
3. Set the switches for the ADDRESS (see table) then press LOAD ADDRS. The contents of the switches will now be in the address lights. Now press EXAM and record the condition of the data lights in the table. If you encounter any problems with this procedure please contact your local Field Service for assistance.

SWITCHES SET	ADDRESS/PROCEDURE	DATA LIGHTS
All up except 0,3	17777766/LOAD ADDRS	
	Press EXAM	____________
	Press EXAM again	____________
	Press EXAM again	____________
	Press EXAM again	____________
	Press EXAM again	____________
All up except 0,2,7	17777572/LOAD ADDRS	
	Press EXAM	____________
	Press EXAM again	____________
	Press EXAM again	____________
All up except 0,4,5,7,9,11	17772516/LOAD ADDRS	
	Press EXAM	____________
All up except 0,1,2,3,4	17777740/LOAD ADDRS	
	Press EXAM	____________
	Press EXAM again	____________
	Press EXAM again	____________
	Press EXAM again	____________
	Press EXAM again	____________
	Press EXAM again	____________
All up except 0,1,2,3,4,5,9	17776700/LOAD ADDRS	
	Press EXAM	____________

If your system has MK11 memory (MOS) please do the following as before.

SWITCHES SET	ADDRESS/PROCEDURE	DATA LIGHTS
All up except 0,3,4	1777746/LOAD ADDRS	
All down except 2,3	14/DEP	
All up except 0,1,2,3,4,6, 8,9,10,11	17770200/LOAD ADDRS	
All down except 12,13,14,15	170000/DEP	
All down except 0,1,2,3,4,5	77/DEP	
All up except 0,4,5,7,9,11	17772516/LOAD ADDRS	
All down except 5	40/DEP	
All down except 6,10,18,19, 20,21	17002100/LOAD ADDRS	
	Press EXAM	____________
	Press EXAM again	____________


PDP-11/70
PAR ERR
ADDRS ERR
RUN
PAUSE
MASTER
USER
SUPER
KERNEL
DATA
ADDRESSING
16
18
22
USER D
SUPER D
KERNEL D
CONS PHY
USER 1
SUPER 1
KERNEL 1
PROG PHY
ADDRESS
DATA
DATA PATHS
BUS REGISTER
U ADRS FPP/ CPU
DISPLAY REGISTER
POWER
LOCK
OFF
PARITY
HIGH
LOW
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
LOAD ADDRS
EXAM
DEP
CONT
ENAB
HALT
S INST
S BUS CYC
START
LAMP TEST

REMOTE CONSOLE FRONT PANEL

If your system halts please collect the following information prior to restarting it. Please type in the following in response to the "CON=" prompt. The system will type out the information we need to help isolate the cause. The following only shows what you should type in and not what the system responds with.

```
       ΛP
CONS= T U 0M 1M 2M 3M <CR>
CONS= 0A 1A 2A 3A 4A <CR>
CONS= 5A 6A 7AΛP
CONS= $60/ <LF> <LF> <LF> <LF> <LF> <LF> <LF> <CR>
CONS= $572/ <LF> <LF> <CR>
CONS= $2516/ <CR>
CONS= $40/ <LF> <LF> <LF> <LF> <LF> <CR>
CONS= 17776700ΛD
CONS= 17772440ΛD
CONS= $46/ 14                                ---------> Only if
CONS= $0200/ 170000 <LF> 77 <CR>                      > You have
CONS= $2516/ 40 <CR>                                  > MK11 mos
CONS= 17002100/ <LF> <CR>                    ---------> memory.
CONS= IΛE
CONS= V
CONS=
```

You may now attempt a system restart. If you encounter any problems with this procedure, please contact the local Field Service office for assistance.
$

NOTE: V01 consoles allow the use of the $ sign with a maximum of 2 trailing digits and the V02 consoles allow any number of trailing digits. If you have a V01 console (ΛE at the CONS= prompt will print the version) you must make the following changes to the above procedure:

V02	V01
$572/	17777572/
$2516	17772516/
$0200/	17770200/

TROUBLESHOOTING DBE'S

INTRODUCTION

Intermittent DBE's in MK11 memory can sometimes be very difficult to isolate. There are two causes for DBE's of which the most common is a failing array module. The other cause is bad data being written into an array. When this problem occurs we get no indication until a subsequent read of that location. We usually waste many hours and parts before resolving the problem. But there is hope!!!

MK11 doesn't store parity bits but stores ECC check bits. Because of this it must check the incoming data and parity for correct parity. If it detects a parity error it inverts four of the check bits and then writes the data and bad check bits into the array. This will cause a DBE when the location is read.

With a Logic Analyzer we can trigger when it is writing the inverted check bits and use this to determine where the bad data was coming from. The following setup is for the K100D logic analyzer.

SIGNAL NAME	PIN	PRINTS	ANALYZER CHANNEL
IN MK11:			
CAS TIM L	A11M1	CBA5	F
WR TIM L	A11N2	CBA5	E
WRITE PAR ERR H	*	CBB5	D
IN 1170:			
AMX S0 H	D17L1	CCBB	6
AMX S1 H	D17K1	CCBB	5
DONE(1) H	D17N1	CCBC	4
CTRLA REQ L	F21K1	CDPH	3
CTRLB REQ L	F21K2	CDPH	2
CTRLC REQ L	F21E2	CDPH	1
CTRLD REQ L	F21C1	CDPH	0

* CBB5 WRITE PAR ERR H doesn't come to a backplane pin so you can either hook up to the chip (E92-12 on M8161 right hand) or run a wire to spare pin D10V1 and use it. This wire must be removed when you resolve the problem because if someone replaces that board and returns it for repair the branch will not get credit for the module!

TRIGGER

Set up the trigger to combinational on channels F,E,D. ie:

```
Trigger (C) (T) F  E  D  C    B  A  9  8    7  6  5  4    3  2  1  0
                0  0  1  x    x  x  x  x    x  x  x  x    x  x  x  x
```

WHAT TO LOOK FOR IF IT TRIGGERS

1) The logic analyzer is set up to trigger when CAS TIM L and WR TIM L both go low at the same time WRITE PAR ERR H is high. This is the point we are writing the inverted check bits into the memory chips. In other words a WRITE PARITY ERROR has occurred which will result in a DBE when we read that location.

2) Channels C and B are set up to identify who was using cache at the time. Use the following table to determine the source of the bad data.

AMX S0 H (C)	AMX S1 H (B)	CACHE PORT
0	0	CPU
0	1	RH70
1	0	PWR UP
1	1	MAP (UNIBUS)

3) Channel A indicates completion of cache cycles whether it was a Hit in Cache or a main memory cycle.

4) Channels 9,8,7 and 6 will indicate which RH70 (A thru D) was accessing Cache if AMX 0+1 show the RH70 port into cache was selected.

5) Now you need a couple of Write Parity Errors triggers on the analyzer to determine where to start looking for the cause of the bad data. The logic analyzer is set up to trigger only if a Write Parity Error occurs on Control 0 side (addresses ending in 0 or 2). If you get a DBE in Control 1 (addresses ending in 4 or 6) the analyzer will not trigger. If the problem is a Write Parity Error, the chances of it happening in both sides are high. If your DBE's only occur on one side and it's not an Array, you should suspect the M8161 on the failing side or the MK11 backplane.

6) Use the following tables to determine where to start with in isolating the source of the bad data.

POSSIBLE CAUSES IF MORE THAN ONE PORT:

1. M8145 Cache data path
2. Main Memory Bus data cables
3. M8159 Data Buffer
4. M8161 Control B0/1
5. MK11 Backplane

POSSIBLE CAUSES IF ONLY RH70 PORT:

1. M8150 in selected RH70
2. M8145 Cache Data Path
3. CPU Backplane

POSSIBLE CAUSES IF ONLY MAP PORT:

1. M8141 Unibus Map
2. M8145 Cache Data Path
3. Any DMA Unibus Device
4. Unibus
5. CPU Backplane

POSSIBLE CAUSES IF ONLY CPU PORT:

1. Follow steps in ANY PORT first

**

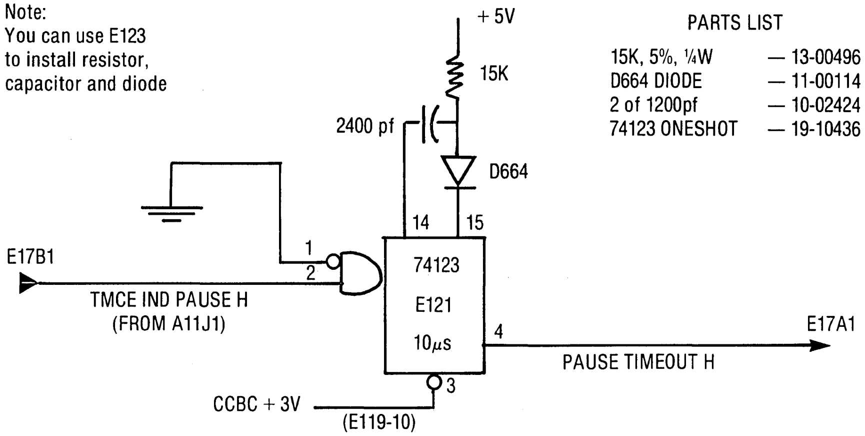
2. M8134 Processor Data Path
3. M8145 Cache Data Path
4. CPU Backplane

** Since the CPU port uses Cache much much more than the MAP or the RH70, two Analyzer triggers are not conclusive enough to rule out it happening on ANY PORT. The problem is more likely to be between Cache and Main Memory, so start looking there first.

TROUBLESHOOTING PAUSE HANGS

INTRODUCTION

1170 systems that hang in the pause state can become real headaches for everyone, especially when the obvious logic has been swapped out, because the problem could be almost anywhere in the system. If a logic analyzer could be triggered at the time of the pause, the signal that put the system in the pause state could be seen, and as a result, we could target in on a specific area within the system. Since all system activities prior to the pause hang is quite normal, it is almost impossible to trigger most logic analyzers at the time of the hang. For this reason, I have put a 10 micro-second one-shot circuit on the M8142 to provide a trigger for the logic analyzer if any pause exceeds the 10 micro-second time frame (no normal pause will). The circuit is as follows:



Note:
You can use E123
to install resistor,
capacitor and diode

PARTS LIST

15K, 5%, ¼W	— 13-00496
D664 DIODE	— 11-00114
2 of 1200pf	— 10-02424
74123 ONESHOT	— 19-10436

The input to the one-shot is brought from the TMC module pin 11AJ1 (IND PAUSE H) via a backplane jumper to the M8142 unused pin 17EB1. The output of the one-shot, which I have called "Pause Timeout L", is brought to another M8142 unused pin 17EA1 for analyzer hook-up. The signal at pin 17EA1 will go high if any pause exceeds 10 micro-seconds. The following pages state the logic analyzer setup for the various pause cycles (the analyzer in this case is a KD100). The analyzer is set up as events happen, starting with Channel F. A detailed description of the events that happen on the analyzer then follows. When the analyzer triggers, by following the description, along with what you see on the analyzer, you should be able to find the signal that put the system in the pause hang.

Trigger: Trigger on "Pause Timeout H" on pin 17EA1.

* **Warning:** The modification of the M8142 should be done only on a module owned by field service and used only when troubleshooting pause hangs!!

Note: The above circuit could be built in a test box external to the CPU.

A) CACHE PAUSE T5

1) Set Logic Analyzer as follows:

SIG NAME	PIN	PRINTS	ANAL CHANNEL
BUST H	17ET2	CCBC	F
CACHE ADRS H	11CM1	TMCF	E
PAUSE B H	14DK1	SAPC	D
CONTROL OK H	17DV2	CCBC	C
HIT EITHER L	17FL2	CCBC	B
START H	18BE1	ADML	A
ADRS ACKN H	17DM2	CCBE	9
RDY CLK L	17DF1	CCBE	8
MEM SYNC H	17FE2	CCBC	7
MEM SYNC H	13CJ2	TIGA	*
DONE H	17DN1	CCBC	6
PAUSE TIMEOUT H	17EA1	****	5

2) Sequence of events:

1) A data transfer is completed in two Rom cycles; a Bus Start Cycle, which is used to initialize memory management and cache (in case of a cache address), and a Pause Cycle, which will do the actual data transfer.
During the Bus Start Rom Cycle, "Bust H" will be asserted and will be sent to memory management to start relocation and address decoding, and also to cache to start cache timing (in case of cache address). In this example, memory management will decode a cache address and "Cache Adrs H" will be asserted.

2) When the CPU leaves the bus start cycle and enters the pause cycle, "Bust H" will be negated and "Pause B H" will be asserted.

3) With both "Pause B H" and "Cache Address H" asserted "Control Ok H" will be asserted (TMCE) and will be sent to cache to start a data transfer.

4) If doing a read and a hit is detected, then "Hit Either L" is asserted and no main memory cycle is required. "Mem Sync H" will be generated with both "Control Ok H" and "Hit Either L" asserted. "Mem Sync H" will be sent to the CPU to restart the timing and clock the data into the CPU BR register.

4A) If a write operation is specified when "Control Ok H" is received in cache, "Start H" is asserted and sent down the main memory bus to start an MJ11 or an MK11 data cycle. Once the MJ or the MK has completed the cycle, it sends "Adrs Ackn H" to cache. When cache receives "Adrs Ackn H", it generates "Mem Sync H" and sends it to the CPU to restart the timing.

4B) If a main memory read cycle (Read Miss) is specified when cache receives "Control Ok H", "Start H" will be asserted and an MJ or an MK read cycle will be started. Once the MJ or MK has started it's timing, it will assert "Adrs Ackn H" which is sent to cache. When the read cycle is completed, the MJ or the MK will generate "Data Rdy H". With both "Adrs Ackn H" and "Data Rdy H" asserted in cache, "Mem Sync H" is generated and sent to the CPU to restart the timing and clock the data in the CPU BR register.

5) "Done H" will then be asserted in cache to place cache in it's idle mode.
The data transfer is now over and the CPU moves to it's next Rom cycle.

3) Analyzer Trigger:

Trigger on "Pause Timeout H" on pin 17EA1. The analyzer will trigger if a cache pause is longer than 10 micro-seconds.
Set time to 10-50 nSec.

B) UNIBUS ADDRESS PAUSE T2

1) Set Logic Analyzer as follows:

SIG NAME	PIN	PRINTS	ANAL CHANNEL
BUST H	17ET2	CCBC	F
PAUSE B H	14DK1	SAPC	E
UNIBUS ADRS H	12AM2	UBCA	D
PSEUDO T3 H	13FC1	TIGA	C
BUS NPR L	12DR1	UBCD	B
PROC NPG H	12FD2	UBCD	A
BUS SACK L	12DJ1	UBCD	9
BUS SSYN L	12ER2	UBCB	8
BUS BBSY L	12FK2	UBCA	7
CPBSY B H	12FL1	UBCA	6
BUS MSYN L	12FL2	UBCA	5
**Waiting for SSYN, see Channel 8.			
TIG RESTART H	12AM1	UBCB	4
TIG RESTART H	13CD1	TIGA	*
PAUSE TIMEOUT H	17EA1	****	3

2) Sequence of events:

1) As mentioned previously, during the Bus Start Rom Cycle, "Bust H" is sent to memory management to start relocation and address decode. In this example, a unibus address will be decoded from memory management and "Unibus Adrs H" will be asserted and sent to the Unibus Control Module "UBC".

2) When the processor leaves the bus start rom cycle and enter the pause cycle, "Bust H" will be negated and "Pause B H" will be asserted.

3) When "Pause B H" is asserted "Pseudo T3" will be generated. Pseudo T3 is only generated in a Pause Rom Cycle.

4) "Pseudo T3" will be sent to the Unibus Control Module "UBC" and if "Unibus Adrs H" is asserted, and the following conditions are *not true*; NPR-NPG-SACK-SSYN-BBSY; "CPBSY B H" will be asserted.

5) "CPBSY B H" will generate "BUS MSYN L".

6) When the unibus control module receives "SSYN L" from the unibus, "TIG Restart H" will be generated and sent to the TIG module to restart the CPU timing and clock the data into the CPU BR register (if dati).
The data transfer is over and the CPU moves on to it's next rom cycle.

3) Analyzer Trigger:

Once again trigger on "Pause Timeout H" on pin 17EA1. The analyzer will trigger if a unibus address pause at T2 is longer than 10 micro-seconds. Set time to 10-50 nanoSec.

C) INTR PAUSE T2

1) Set Logic Analyzer as follows:

SIG NAME	PIN	PRINTS	ANAL CHANNEL
EXT BRQ H	12DD1	UBCD	F
INTR PAUSE H	12DD2	UBCD	E
TIGE TS2 L	12ES1	UBCD	D
PROC NPG H	12FD2	UBCD	C
BUS NPR L	12DR1	UBCD	B
BUS SACK L	12DJ1	UBCD	A
PROC BG4 H	12ET2	UBCD	9
PROC BG5 H	12EM2	UBCD	8
PROC BG6 H	12EP2	UBCD	7
PROC BG7 H	12ES2	UBCD	6
**Waiting for SACK, see Channel A.			
BUS BBSY L	12FK2	UBCA	5
BUS INTR L	12DT2	UBCC	4
TIG RESTART H	13CD1	TIGA	3
TIGE TS3 L	12DP2	UBCB	2
BUS SSYN L	12ER2	UBCB	1
PAUSE TIMEOUT H	17EA1	****	0

2) Sequence of events:

1) In between instruction execution, the CPU will do a BR Strobe to see if any bus request are present. If a request is present, the CPU will not fetch the next instruction but will go to an interrupt micro-code routine to get a vector address on the unibus. At the time of the BR Strobe, if any BR are present, then "EXT BRQ H" will be asserted.

2) The CPU will go in an INTR Pause. "INTR Pause H" will be asserted and the timing will stop at T2.

3) When "TIGE TS2 L" is generated during the pause cycle, and with both "EXT BRQ H" and "Intr Pause H" asserted, and the following conditions *not true*; NPG-NPR-SACK; the UBC module will assert "PROC BG* H" on the unibus (*=4,5,6, or 7).

4) The interrupting device will assert "Bus Sack L" on the unibus. When the UBC module receives "Bus Sack L", it will negate "PROC BG* H" and disable the sack timeout circuit.

5) When the unibus becomes available (BBSY negated) the interrupting device will take the bus by asserting "BUS BBSY L" and negating "BUS SACK L".

6) The interrupting device will then put it's vector on the unibus data lines and assert "BUS INTR L".

7) When the Unibus Control Module (UBC) receives "BUS INTR L", it will generate "TIG Restart H", which will be sent to the TIG module to restart the timing and clock the vector into the CPU BR register.

8) When the CPU timing restarts, "TIGE TS3 L" will be generated.

9) When the UBC module receives "TIGE TS3 L", it will generate "BUS SSYN L" out on the unibus.

10) The device will then negate "BUS INTR L", "BUS BBSY L", and also remove it's vector from the unibus data lines. The transaction is over and the CPU now fetches the next instruction.

3) Analyzer Trigger:

Again trigger on "Pause Timeout H" on pin 17EA1. The analyzer will trigger if an INTR Pause is longer than 10 micro-seconds.
Set time to 10-50 nano-Sec.

TROUBLESHOOTING RED ZONE ABORTS

A) In between program instructions (i.e. before fetching the next instruction) the micro-code will do a "BRQ Strobe" to see if there are any pending internal or external break requests. If a request exists, the micro-code will branch away from the instruction fetch rom cycle and go into a break request rom cycle, in order to obtain a vector that will be used as a new PC (software subroutine to handle the request).

Internal Requests Are:

1) Internal Traps.
Memory Management traps or parity traps for example.
2) Program Interrupt Requests (PIRQ) which are software controlled.

External Requests Are:

1) Unibus Requests.
BR4 for example.

Vectors for the internal requests are obtained internally. That is, the CPU itself provides the vector (see DAPE). The CPU must go out onto the unibus to obtain a vector when an external request exists (device vectors), and therefore must go into an INTR Pause Cycle.

As stated previously, before fetching a new instruction with the updated PC, a BRQ Strobe is done, and if a request exists, the instruction is not fetched and the updated PC is pushed into the stack so that we can return to it after the request has been serviced. If a request exists while in the request subroutine (there is a BRQ Strobe in between all instructions in the subroutine) then that PC is pushed on the stack. If we cannot get out of the subroutine because of continuous requests, then we will continuously push onto the stack, and decrement it to the red zone.

B) Here is a list of the internal and external requests, their priority and their vector assignment:

1 CONSOLE FLAG 2 PARITY TRAP 3 SEG MEM MANAGEMENT TRAP 4 STACK LIMIT YELLOW 5 POWER FAIL 6 FP EXEPTION TRAP	— 114 250 4 24 244
CP LEVEL 7	
7 PIRQ 7 (PIR 15) 8 BUS REQUEST 7	240 INTR
CP LEVEL 6	
9 PIRQ 6 (PIR 14) 10 BUS REQUEST 6	240 INTR
CP LEVEL 5	
11 PIRQ 5 (PIR 13) 12 BUS REQUEST 5	240 INTR

CP LEVEL 4 13 PIRQ 4 (PIR 12) 14 BUS REQUEST 4	 240 INTR
CP LEVEL 3	
15 PIRQ 3 (PIR 11)	240
CP LEVEL 2	
16 PIRQ 2 (PIR 10)	240
CP LEVEL 1	
17 PIRQ 1 (PIR 09)	240
CP LEVEL 0	
18 T BIT (PS 04) AND NOT RRT & RTT NOT LOCKED OUT BY PROC STATUS LEVELS BUT SERVICED LAST	14

C) By triggering the logic analyzer with "TMCD Yellow Trap H" we should see which one of the above listed request was continuously asserted and caused the red zone.

Analyzer Setup:

Analyzer Setup Yellow/Red Zone Violations:

F	TRIGGER	11AD2	YELLOW TRAP H	
E	LATCH	11ER1	HONOR BR 7 L	
D	LATCH	11ED1	HONOR BR 6 L	
C	LATCH	11EE2	HONOR BR 5 L	
B	LATCH	11EL1	HONOR BR 4 L	
A	LATCH	11EU1	PARITY TRAP H	
9	LATCH	11DJ1	MEM MNGT TRAP L	
8	LATCH	11EN1	PWR FAIL (1) H	
7	LATCH	11FA1	FP TRAP L	
6	LATCH	11DC1	UNIBUS TIMEOUT B L	
5	LATCH	11AN1	ODD ADDR ERR L	
4	LATCH	11DF2	KT ABORT FLG L	
3	LATCH	11CR2	PE ABORT L	
2	LATCH	11CL2	NOT CACHE ADDR H	NEXM L
1	LATCH	11EF1	UNIBUS ADDR L	

Set timing to 1 mSec.

Trigger Setup:

Trigger (C) (T)	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1
	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Analyzer — GNDS = Any pin with a black wire on it
— CLK = 10ns for all except UB TOUT + BR = 10ms
— Input Mode = Latch
— Threshold = TTL
— Delay = 500 Clks

RED ZONE TEST PROGRAMS

The following programs can be used to verify the logic analyzer setup for a particular RED ZONE. Use a analyzer clock of 10ns for all except the "BR" and "UB TIMEOUT" RED ZONES which use a clock of 10ms.

1. Memory Management Abort Red Zone Program

   The program executes from PAR0 while trying to access location thru PAR1 (physical memory address 20000) which has been setup to abort all access.

```
1000/12737/0/772340            MOV #0, PAR0
1006/12737/77406/772300        MOV #77406, PDR0
1014/12737/200/772342          MOV #200, PAR1
1022/12737/77400/772302        MOV #77400, PDR1
1030/12737/1/777572            MOV #1, MMR0
1036/137/20000                 JMP @20000

250/1036                       MMU vector service routine
17777706/600                   Stack pointer
4/1036                         Restart after yellow zone

20000/777                      BR (Should not set here)
```

   Result is MMR0 = 100003, CPER = 14

2. Odd Address Error Red Zone Program

   The following program will try and access an odd address then try again and again until Red Zone.

```
1000/012737/100/2001           MOV #100, 2001
1006/137/1000                  JMP 1000

17777706/600                   Set stack pointer = 600

4/1006                         Odd adrs error service
6/0                            PSW
```

   Result is CPER = 114

RED ZONE TEST PROGRAMS *(continued)*

3. BR6 Red Zone Program

The following program will turn on interrupts from the KW11 and never service them which will result in a Red Zone.

```
1000/777                        BR
1002/012737/100/777546          MOV #100, 777546        (KW11 IE)
1010 137/1000                   JMP 1000

17777706/600                    Set stack pointer = 600

100/1000                        KWW11 Vector service
102/0                           PSW

4/1000                          Restart after yellow zone
6/0                             PSW
```

Result is CPER = 14

4. Unibus Timeout Red Zone Program

The following program will try and access a non-existant device on the unibus again and again until Red Zone.

```
1000/013737/777400/2000         MOV 777400, 2000  (RKDS->2000)
1006/137/1000                   JMP 1000

17777706/600                    Set stack pointer = 600

4/1006                          UB Timeout Service
6/0                             PSW
```

Result is CPER = 34

RED ZONE TEST PROGRAMS *(concluded)*

5. NXM Red Zone Program

The following program causes NXM aborts which never get serviced resulting in a Red Zone.

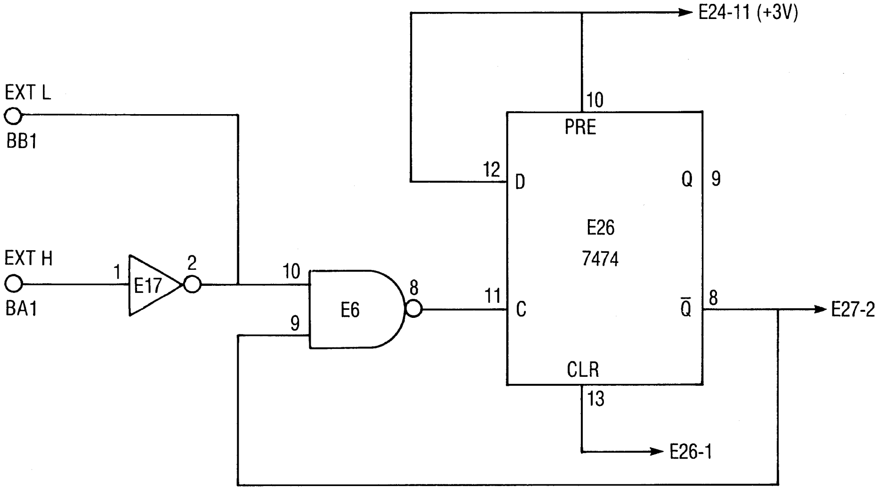
```
1000/12737/0/772340          MOV #0, PAR0
1006/12737/77406/772300      MOV #77406, PDR0
1014/12737/60000/772342      MOV #60000, PAR1
1022/12737/77406/772302      MOV #77406, PDR1
1030/12737/20/772516         MOV #20, MMR3
1036/12737/1/777572          MOV #1, MMR0
1044/12737/4/20000           MOV #4, 20000
1052/774                     BR Back

17777706/600                 Set SP = 600

4/1044                       NXM Service
6/0                          PSW
```

Result is CPER = 54

UNIBUS CAUSED RED ZONES AND UNIBUS TIMEOUT TROUBLESHOOTING

When a RED ZONE occurs as a result of a multiple BRs or UNIBUS TIMEOUTs it can be difficult to isolate to a device without knowing what was happening on the unibus. If we could see the vector in the case of a BR or the address in the case of a UNIBUS TIMEOUT the solution would be simple. The Unibus Trace Analyzer (UTA) can be used to help get this information. Also if you are troubleshooting an intermittant UNIBUS TIMEOUT problem this technique will be very useful.

In order to catch the state of the unibus prior to the RED ZONE or UNIBUS TIMEOUT we need to add a couple of wires to the UTA to provide it with a latching external input. The circuit below will stop the UTA from recording any further Unibus activity once the external input goes high (or low). It will keep the UTA locked with the first occurrence of the trigger even if the external input toggles again.



CHANGE LIST

1. Add wire E26-8 to E27-2
2. Add wire E26-8 to E6 -9
3. Add wire E26-10 to E24-11
4. Add wire E26-11 to E6 -8
5. Add wire E26-12 to E26-10
6. Add wire E26-13 to E26-1
7. Add wire E6 -10 to E17-2
8. Add wire E17-1 to module pin BA1
9. Add wire E17-2 to module pin BB1

*** With this addition if you are using the module in any of it's other recording modes the EXT H should be connected to a ground on the backplane to prevent false triggering.

1. Catching Unibus caused RED ZONES on the PDP1170

   Install the modified UTA in the first Unibus SPC slot so all devices on the Unibus are covered. Run the diagnostic (G5445C.BIN) with the Enable SW in the down position. If all checks out ok hookup the EXT H input to pin A11D2, YELLOW TRAP H. Then set the Enable SW down and load the CSR address (17776570) and deposit 100000 (inits the module). At this point it is a good idea to verify every thing is working with the following program. If everything works, init the module again and boot the customer's software.

   TEST PROGRAM:

   The following program will turn on interrupts from the KW11L and never service them which will result in a RED ZONE.

```
1000/777                      BR
1002/012737/100/777546        MOV #100, 777546          (KW11 IE)
1010/137/1000                 JMP 1000

17777706/600                  SP = 600

100/1000                      KW11 Vector Service
102/0                         PSW

4/1000                        Restart after a RED ZONE
6/0                           PSW
```

   The result when you halt the system is CPU ERROR REGISTER = 14. If you then run BUSDMP.BIN to examine the UTA memory you will see it recorded a BR6 and a VECTOR = 100 prior to stopping the recording. Bingo, we know 'who done it'.

Here is an example of what the BUSDMP looks like for the test program. You will notice that 33 decimal interrupts occured at vector 100. This would result in 66 decimal (102 octal) pushes onto the Kernal stack (PC and PSW are pushed). The 102 octal pushes converts to 204 octal word addresses. Since our Kernal stack started at 600, we went into yellow zone (600-204 = 374) and stopped the UTA from recording.

```
.R BUSDMP
BUSDMP.BIN

UNIBUS CRASH DUMP ANALYZER PROGRAM

IS THE MODULE SET FOR THE STANDARD ADDRESS? Y/N Y
RESTART ADDRESS IS 2000
THE MODULE STORES 2048 (10) UNIBUS SAMPLES.
THE LAST SAMPLE WAS STORED AT LOCATION 00033(10)
TYPE HLP FOR COMMAND SUMMARY
>DMP
FM>0
TO>33

SAMPLE    ADDRESS   DATA     MISC

00000     777546    000100   C1 MSYN INIT BBSY
00001     000000    000100   INTR BR6 SACK BG6 BBSY
00002     000000    000100   INTR BR6 SACK BG6 BBSY
00003     000000    000100   INTR BR6 SACK BG6 BBSY
00004     000000    000100   INTR BR6 SACK BG6 BBSY
00005     000000    000100   INTR BR6 SACK BG6 BBSY
00006     000000    000100   INTR BR6 SACK BG6 BBSY
00007     000000    000100   INTR BR6 SACK BG6 BBSY
00008     000000    000100   INTR BR6 SACK BG6 BBSY
00009     000000    000100   INTR BR6 SACK BG6 BBSY
00010     000000    000100   INTR BR6 SACK BG6 BBSY
  '         '         '                 '
  '         '         '                 '
  '         '         '                 '
00030     000000    000100   INTR BR6 SACK BG6 BBSY
00031     000000    000100   INTR BR6 SACK BG6 BBSY
00032     000000    000100   INTR BR6 SACK BG6 BBSY
00033     000000    000100   INTR BR6 SACK BG6 BBSY
```

2. Catching UNIBUS TIMEOUTS on the PDP1170

Install the modified UTA in the first Unibus SPC slot so all devices on the Unibus are covered. Run the diagnostic (G5445C.BIN) with the Enable SW in the down position. If all checks out ok hookup the EXT H input to pin B11A1, UNIBUS TIMEOUT H. Then set the Enable SW down and load the CSR address (17776570) and deposit 100000 (inits the module). At this point it is a good idea to verify everything is working with the following program. If everything works, bring up the customer's software and then use online ODT or halt the system and init the module again (continue the system if you halted it). You have to do it in that order because most operating systems size the Unibus for devices on startup and will set a UNIBUS TIMEOUT.

TEST PROGRAM:

The following program will try and access an RK05's CSR which will result in a UNIBUS TIMEOUT. If your system has an RK05, pick an address of some device you do not have and substitute it in location 1004.

```
1000/012737/5/777404                 MOV #5, RKCS
1006/0                               HLT

4/6                                  UB TIMEOUT Trap Catcher
6/0                                  HLT
```

The result when the system halts is the CPU ERROR REGISTER = 20. If you then run BUSDMP.BIN to examine the UTA memory you will see it recorded a ADDRESS = 777404 (or the one you selected) prior to stopping the recording. Bingo, we know 'who done it'.

*** Remember to INIT the module after the customer's system is up or the UTA will trigger on system startup.

Here is what the BUSDMP printout will look like for the test program. You will notice there is only 1 entry since the address sent ot the Unibus caused a timeout and the UTA stopped recording.

NOTE: Since the UTA clocks the memory with MSYN on a DATO and SSYN on a DATI if the Unibus Timeout occurs on a DATI the address causing the Timeout will not be stored (no SSYN). This means you have a 50/50 chance of getting the bad address.

```
.R BUSDMP
BUSDMP.BIN

UNIBUS CRASH DUMP ANALYZER PROGRAM

IS THE MODULE SET FOR THE STANDARD ADDRESS? Y/N Y
RESTART ADDRESS IS 2000
THE MODULE STORES 2048 (10) UNIBUS SAMPLES.
THE LAST SAMPLE WAS STORED AT LOCATION 00000(10)
TYPE HLP FOR COMMAND SUMMARY
>DMP
FM>0
TO>0

SAMPLE     ADDRESS     DATA      MISC

00000      777404      000005    MSYN INIT BBSY
```

TROUBLESHOOTING MAIN MEMORY TIMEOUTS

Main Memory Timeouts are a result of a loss of communications between the Main Memory and Cache. At the start of a memory cycle Cache sends the ADDRESS and MAIN START to Memory and starts a 4us timer. Memory decodes the address and responds with a MAIN ACK. If a signal is lost along the way the Cache will timeout with the 4us one shot setting the Main Memory Timeout if Cache was doing a CPU cycle (MSER bit 0 set), set the RH70 Timeout if Cache was doing an MBA cycle (RH70 CSR2 bit 11 set) or cause the requesting Unibus Device to set it's NXM bit if cache was doing a MAP cycle.

The following analyzer setup can be used to isolate which signal was lost and where for a Main Memory Timeout. To test this setup run a BR SELF instruction with Cache off and ground CPU backplane pin A18A1. This should trigger the analyzer showing MAIN ACK was never received from memory but memory sent it out.

Logic Analyzer Setup:

Module	Signal Name	Prints	Pin	Channel
M8142	CCBD CP TIMEOUT L	CCBD	D17H2	F
	CCBD MBC TIMEOUT H	CCBD	C17V2	E
	CCBD UB TIMEOUT L	CCBD	D17A1	D
M8143	CCBD START H	ADML	B18E1	C
	ADML ADRS ACK H	ADML	A18A1	B
M8158	ABB9 GO H	ABB9	D13U1	1
	DBB2 IACK L	ABB3	D13T2	0
M8147/8	MCTA START H	MCTA	E87-3	1
	MCTF IACK L	MCTF	E39-8	0

Trigger setup:

* Trigger (C) (F)	F	E	D	1	0
	1	0	1	x	x

Analyzer	— GNDS	= MK11 = any T1 or C2 pin = CPU = any pin with black wire on it
	— CLK	= 10ns
	— Input Mode	= Latch
	— Threshold	= TTL
	— Delay	= 100 clks

* Note: Trigger is setup for FALSE (F) so that any one of the timeout signals setting (Going LOW) will trigger the analyzer.

RSTS ERROR LOGGER

To run the RSTS Error Logger proceed as follows after logging in and answer the questions presented. All of these questions may not appear depending on what options are selected. The defaults are in <> and are accepted by typing a <CR>.

```
$ RUN $ERRDIS
ERRDIS V9.1-05 RSTS V9.2-10 SYSTEM 1170
Input File <ERROR$:ERRLOG.FIL>?                              ; note 1
Output to <KB:ERRDIS.OUT>?                                   ;note 2
He[lp], Ba[d Blocks[, Su[mmary] of Fu[ll] Report <Summary>?
Specific Error Type <All>?                                   ;note 3
Starting Date <First Error>?                                 ;note 4
Starting Time <First Error>?
Ending Date <Last Error>?
Ending Time <Last Error>?
List Bad Blocks (Yes/No) <No>?
Zero Error File upon completion (Yes/No) <No>?
```

Notes:

1. *This is the default account (may vary in different RSTS versions) and error log file. You can also select ERRCRS.FIL which if it exists will contain the error log buffer at the time of the system crash.*

2. *This can be any file specification, eg: LP0: for output to printer, ERROR.RPT for a file, etc.*

3. *Specify ALL (default) or device mnemonic. Request a He[lp] report for a list of mnemonics. You can also specify ALL/NOTAPE for everything but magtape errors.*

4. *Date format = dd-mm-yy and Time format = hh:mm (24 hour format).*

RSTS CRASH DUMP ANALSYS

When RSTS is having a serious problem and it doesn't expect to recover it will attempt to create [0,1]CRASH.SYS on the system disk. If it is able to create this file it will reboot to "OPTION". What it does from here depends on the state of SWR bit 0. If bit 0 is clear it will stay at "OPTION". If bit 0 is set or the SWR is disabled it will attempt to come all the way up using the crash procedure in a command file. This should run ANALYS on the crash and generate a report. If the command procedure is not setup this way or the system didn't restart, you can run ANALYS at any time on the last [0,1]CRASH.SYS file. This file is overwritten on each crash.

To run ANALYS proceed as follows after logging in and answer the questions presented. The defaults are in <> and are accepted by typing a <CR>.

```
$ RUN $ANALYS
ANALYS V9.0 RSTS V9.2-10 SYSTEM 1170
INPUT <[0,1]CRASH.SYS>?                                    ;note 1
OUTPUT <ANALYS.DMP>?                                       ;note 2
Crash error log filename<[0,3]ERRCRS.FIL>?                 ;note 3
```

Notes:

1. *ANALYS uses the currently installed monitor SIL for symbol references. If the CRASH.SYS being analyzed is for a different monitor use the /SIL:sil_name switch on the input file.*
2. *The following switches apply to the output file spec:*

/NARROW	*Create an 80 column report*
/NOSTB	*Omit the Symbol table printout*
/NODUMP	*Omit the memory dump*

3. *If you enter /DET here it will cause ANALYS to detach.*

RSTS CRASH DUMP ANALSYS

RSTS CRASH ERROR CODES

Code	Description
177777	Power Fail
177776	Trap 0 (Jump to 0)
177775	Continue from 52
177774	Software forced Crash
000041	Trap 4 (CPU error)
000042	Trap 10 (Illegal Instruction)
000043	Trap 250 (MMU abort)
000044	Red Zone, Kernal stack overflow
000046	Trap 114 (Memory system error)
0 or other	Forced Dump

CRASH DUMP ENABLED?

To find out if crash dump is enabled RUN $UTILITY and type SNAP (for V9.x type $DUMP/SYSTEM at the DCL prompt). This will attempt to create a [0,1]CRASH.SYS file of the current system state. If the function works, then crash dump is enabled. If the function returns a "Can't find file or account" error, then crash dump was NOT enabled at system startup or through DEFAULT. In V9.x crash dump is always enabled unless there isn't enough room on the disk for CRASH.SYS in which case RSTS prints an error message.

To re-enable crash dump, do the following:

1. Get to the OPTION: prompt of INIT.SYS by shutting down timesharing.
2. Execute the REFRESH option to create the CRASH.SYS file (if it does NOT already exist).
3. Finally, execute the DEFAULT option and answer YES to the crash dump enabled question.

FORCING A CRASH DUMP

At times it is useful to force a crash dump to gather information on a RSTS/E system. This can be used to get a dump when a system hangs. These all require the system to be put into console mode. In order to do this, halt the processor, either by the switch on the front panel or typing ΛP H on the console. After the commands are typed, you will need to proceed, which refers to either switching the front panel switch back to the RUN position and pressing the CONTINUE switch or typing the P console command.

1. Poke the clock. Load address 100 (which is the clock vector) and deposit a 1 into this location, then proceed.
2. Force a trap to 4. Examine the contents at location 4 and 6. Load the contents from location 4 into the PC (address 17777707), and load the contents of location 6 into the PSW (address 17777776), then proceed.
3. Force an odd address trap. Examine the contents of the PC (address 17777707) and deposit back into the PC it contents plus 1, then proceed.

RSTS will now do a crash dump and initiate a system reload/auto-restart if crash dump is enabled and switch register bit 0 = 1 (Auto Restart bit). After time-sharing is resumed, the standard diagnostic tools such as ANALYS and ERRDIS can be used to analyse the crash dump. When analyzing these dumps, however, one should remember that the reported error code (41) is no longer valid since the dump was taken by forcing a trap through 4.

ANALYZING RSTS HALTS

When RSTS has a serious problem it attempts to create a crash file on the disk. If it is unable to create or fully complete [0,1]CRASH.SYS or restart RSTS it will halt at 1 of 6 locations. To identify which halt you are at you will need to look at the memory locations around the halt. BECAREFUL, the halt address the KY11R gives you is VIRTUAL. You must insure you are examining the correct memory locations. This can be done by relocating the virtual address using KIPARx manually or on the KY11R by typing the following on the console IMMEDIATELY following the halt, eg:

```
*H00033054/T44410 <CR>     ;type a <CR>
CONS= 0A00377054 ^P        ;type a 0A then ^P, 377054 is the
CONS=                      ; physical relocation of virtual 33054
```

If after examining the memory locations, the halt is not one listed here, you may have just been transferred into hyperspace (incorrect relocation, memory cleared or corrupted, etc.). Your best approach in this case is to find out where you are and if the code you examined is what is supposed to be there. Another CAUTION is always verify R7 matches the halt address. The KY11R doesn't always display the virtual PC on a halt but R7 is always correct.

Here are the 6 halts. Following this summary is a listing of the memory locations around each halt. Once the halt is identified there is a detailed description of each halt and how to troubleshoot it.

1. *H00000056/T44410 HLT54, Double PWR FAILS or T114's
2. *Hxxxxxxxx/T44410 C.HLT1, Unable to map Crash phase
3. *Hxxxxxxxx/T44410 C.HLT2, Crashed while Crashing
4. *Hxxxxxxxx/T44410 REBOOT+72 Halt, Disk read error
5. *Hxxxxxxxx/T44410 WRILOW+106 Halt, Disk write error
6. *H00000xxx/T44410 TRAP CATCHER Halt, Restart error

ANALYZING RSTS HALTS

IDENTIFYING THE HALT

1. HLT54

   This address is fixed and is the same on all RSTS systems. The pattern in in memory is as follows:

```
00000052/000434          BR DODUMP     ;do a crash dump
00000054/000000          HALT          ;halt
00000056/000770          BR 10$        ;do a system reload
```

2/3. C.HLT1, C.HLT2

   The C.HLT1 and C.HLT2 addresses may vary from system to system and to be sure thats where you actually halted examine your halt address +-6 locations and look for the following pattern:

```
xxxxxxx/000000   C.HLT2: HALT          ;double error halt
xxxxxxx/000776           BR C. HLT2    ;no continue allowed
xxxxxxx/000000   C.HLT1: HALT          ;crash not mapped halt
xxxxxxx/000776           BR C.HLT1     ;no continue allowed
```

4. REBOOT+72

   The REBOOT+72 halt address may vary from system to system and to be sure thats where you actually halted examine your halt address +-6 locations and look for the following pattern:

```
xxxxxxxx/022737          CMP #NOP,@#0  ;boot read in ok?
xxxxxxxx/000240
xxxxxxxx/000000
xxxxxxxx/001402          BEQ 10$       ;yes, jump to boot
xxxxxxxx/000000          HALT          ;no, halt REBOOT+72
xxxxxxxx/000741          BR REBOOT     ;retry after halt
```

5. WRILOW+106

   The WRILOW+106 halt addresses may vary from system to system and to be sure thats where you actually halted examine your halt address +-6 locations and look for the following pattern:

```
xxxxxxxx/004767          JSR PC,B.READ ;go to dump writer
xxxxxxxx/?
xxxxxxxx/103001          BCC 50$       ;write error?
xxxxxxxx/000000          HALT          ;yes, halt WRILOW+106
xxxxxxxx/000207     50$  RTS PC        ;no; return or continue
```

IDENTIFYING THE HALT (cont.)

6. TRAP CATCHER

The halt addresses for TRAP CATCHER halts are always the address of the vector that caused the trap, interrupt or abort + 4. For example if the system halted at 10 memory would look as follows:

```
00000004/000006              ;TRAP 4 catcher, new PC
00000006/000000    HALT      ;TRAP 4 halt
00000010/000012              ;TRAP 10 catcher, new
                             PC
```

ANALYZING THE HALT

1. HLT54

If you halted at 56 you have either had a powerfail while servicing a powerfail or a parity error while servicing a parity error. To determine which it was look at the following physical memory locations:

```
00034/000054        You had a double powerfail!
00114/000054        You had a double parity error!
```

In the case of double parity errors the $40,$42,$44 and the MK11 CSR's will still contain the failure information. A double powerful problem should be troubleshot as any normal powerfail.

2. C.HLT1

If you halted at C.HLT1 this means RSTS was having problems mapping the Crash phase and gave up. Before giving up it did however write the ERROR CODE into a location called CRASAV+776. For RSTS V8.x and 9.x, CRASAV is physical 24000. In this location will be the reason RSTS attempted to crash (see table of ERROR CODES in RSTS CRASH DUMP ANALSYS section).

Also be sure to look at the CPU error register ($66), MMRO ($572), Memory error registers ($40, $42, $44) and the MK11 CSR's which may also contain information about the crash.

ANALYZING THE HALT (cont.)

3. C.HLT2
   A halt at C.HLT2 indicates we were in the process of crashing when another fatal error occured. Unfortunately when this happens it usually always means the error registers have been cleared. This happens because RSTS issues a RESET instruction before it attempts to reboot. In any case it has assembled all or part of the CRASH file in memory starting at location CRASAV for 776 locations. Use an old Crash dump to get this address and the addresses for LOWAD, HGHAD and MEMERR (for RSTS V8.x and 9.x, CRASAV is physical 24000, but the locations were the memory registers are saved vary from system to system). These locations will reflect the 1st crash and the current registers ($40, $42, $44, $66, $572) should contain the information on the second crash. Refer to the table on the layout of the CRASAV, LOWAD, HGHAD and MEMERR area in memory.

4. REBOOT+72
   A halt here indicates the system was in the process of doing a software BOOT and an disk error occured. It tries to load the primary bootstrap off of block 0 of the system disk and then checks physical memory location 0 to insure it contains a NOP (240) instruction. If it doesn't, the system halts. If you halted here you should check the SYSTEM DISK's REGISTERS for the source of the problem. It is also useful to check the CRASAV, LOWAD, HGHAD and MEMERR area of memory incase this boot was a reboot of RSTS following a crash.

5. WRILOW+6
   RSTS will halt here if a disk error occurs while trying to write the file [0,1]CRASH.SYS. If you halted here you should check the SYSTEM DISK's REGISTERS for the source of the problem. It it also important to check the CRASAV, LOWAD, HGHAD, MEMERR area of memory to find out why we were crashing in the first place.

6. TRAP CATCHER
   During the initial loading or reloading of RSTS the vector area of memory is loaded up with trapcatchers. This prevents unexpected ABORTS, TRAPS or INTERRUPTS from affecting the startup/restart. If you halt at a TRAP CATCHER location the vector that caused the trap will tell you what to look at. For example if I halted at 120, this would mean I had a TRAP 114. It would then make sense to look at the memory system error registers for the cause of the problem. It is also useful to check the CRASAV, LOWAD, HGHAD, and MEMERR area of memory incase this boot was a reboot of RSTS following a crash.

RSTS V8.x CRASH DUMP AREA IN MEMORY

Location	Contents
CRASAV + 0	KISAR5
2	KDSAR5
4 .. 12	Instruction Space -6(PC) to (PC)
14 .. 22	Instruction Space 2(PC) to 8(PC)
24	CPU ID register
26	CPU error resister
30	PDP 11/60 MED data length
32 .. 54	PDP 11/60 MED data (reserved for 10. words)

The following area is unassigned and must be at least 10. words long for stack space.

56 .. 260	UNASSIGNED

These next two words may be at higher addresses depending on exclusion of other conditional data below.

262	Complement of size of dumped monitor image
264	Size of dumped monitor image

The following data is only present if the system has Unibus Mapping Registers. If I and D space is not present this data will occupy slots 332 thru 524.

226 .. 304	UMR's 0-3
306 .. 324	UMR's 4-7
326 .. 344	UMR's 8-11
346 .. 364	UMR's 12-15
366 .. 404	UMR's 16-19
406 .. 424	UMR's 20-23
426 .. 444	UMR's 24-27
446 .. 460	UMR's 28-30

The following data is only present if the CPU's has I and D space.

462 .. 464	KDSAR0 KDSDR0
466 .. 470	KDSAR1 KDSDR1
472 .. 474	KDSAR2 KDSDR2
476 .. 500	KDSAR3 KDSDR3
502 .. 504	KDSAR4 KDSDR4
506 .. 510	KDSAR5 KDSDR5
512 .. 514	KDSAR6 KDSDR6
516 .. 520	KDSAR7 KDSDR7

Location	Contents
CRASAV + 522	MMR3
524	MMR1
526	MMR2
530	MMR0
532 .. 540	UISAR0 UISDR0 KISAR0 KISDR0
542 .. 550	UISAR1 UISDR1 KISAR1 KISDR1
552 .. 560	UISAR2 UISDR2 KISAR2 KISDR2
562 .. 570	UISAR3 UISDR3 KISAR3 KISDR3
572 .. 600	UISAR4 UISDR4 KISAR4 KISDR4
602 .. 610	UISAR5 UISDR5 KISAR5 KISDR5
612 .. 620	UISAR6 UISDR6 KISAR6 KISDR6
622 .. 630	UISAR7 UISDR7 KISAR7 KISDR7
646 .. 632	User XRB (in reverse order)
706 .. 650	User FIRQB (in reverse order)
710	User keyword
712 .. 720	16(USP) 14(USP) 12(USP) 10(USP)
722 .. 730	6(USP) 4(USP) 2(USP) (USP)
732	User stack pointer, USP
734 .. 742	16(KSP) 14(KSP) 12(KSP) 10(KSP)
744 .. 752	6(KSP) 4(KSP) 2(KSP) (KSP)
754	Processor status word, PSW
756	Virtual PC
760 .. 772	R0 R1 R2 R3 R4 R5
774	Kernal stack pointer, KSP
CRASAV + 776	ERROR CODE
LOWAD	$40, Low order address of error
HGHAD	$42, High order address of error and cycle type
MEMERR	$44, Memory System Error register

*NOTE 1 The CRASAV area is normally filled with all 1's (177777). This is helpful in determining what is valid data and how far the crash got.

*NOTE 2 CRASAV = 24000 physical for RSTS V8.x and V9.x. The LOWAD, HGHAD and MEMERR vary per system. Look at an old CRASH DUMP for the following:

```
                 Memory parity/ECC log
               LOWADD      HI ADD     MEMERR . . .
        067554/ 040122      000001     144014 . . .
```

Therefore LOWAD = 067554, HGHAD = 067556 and MEMERR = 067560 virtual.

RSTS V9.x CRASH DUMP AREA IN MEMORY

Below are listed the locations that differ from V8.x

Location	Contents

The following area is unassigned and must be at least 10. words long for stack space.

CRASAV 56 .. 220	UNASSIGNED

These next two words may be at higher addresses depending on exclusion of other conditional data below.

222	Complement of size of dumped monitor image
224	Size of sumped monitor image

The following data is only present if the system has Unibus Mapping Registers. If I and D space is not present this data will occupy slots 332 thru 524

226 .. 244	UMR's 0-3
246 .. 264	UMR's 4-7
266 .. 304	UMR's 8-11
306 .. 324	UMR's 12-15
326 .. 344	UMR's 16-19
346 .. 364	UMR's 20-23
366 .. 404	UMR's 24-27
406 .. 420	UMR's 28-30

The following data is only present if the CPU's has I and D space.

422 .. 430	UDSAR0	UDSDR0	KDSAR0	KDSDR0
432 .. 440	UDSAR1	UDSDR1	KDSAR1	KDSDR1
442 .. 450	UDSAR2	UDSDR2	KDSAR2	KDSDR2
452 .. 460	UDSAR3	UDSDR3	KDSAR3	KDSDR3
462 .. 470	UDSAR4	UDSDR4	KDSAR4	KDSDR4
472 .. 480	UDSAR5	UDSDR5	KDSAR5	KDSDR5
502 .. 510	UDSAR6	UDSDR6	KDSAR6	KDSDR6
512 .. 520	UDSAR7	UDSDR7	KDSAR7	KDSDR7

RSX11M V3.2 ERROR LOGGER

To run the RSX11M V3.2 Error Logger proceed as follows after logging in:

```
>RUN $PSE
PSE>[output_file]=[input_dev:]
  (default=SY0:[1,6]ERROR.SYS=SY0:)
PSE>∧Z
>RUN $SYE
SYE>[report_file][/sw's]=input_file
```

Example of a full error report:

```
RUN $PSE
PSE>SY0:[1,6]ERROR.SYS=SY0:
PSE>∧Z
>RUN $SYE
SYE> SY0:ERRLOG.LST/RP=SY0:[1,6]ERROR.SYS
SYE>∧Z
>PIP TI:=ERRLOG.LST
```

Switches:

```
            /RP[:class]
                HDW[:type]
                     :MEM (all cache/memory errors)
                     :DSK (disk errors)
                     :MAG (tape errors)
                      null (all types of hardware errors)
                TM0[:type]
                     :DSK (interrupt timeout errors)
                     :MAG (interrupt timeout errors)
                SYS[:type]
                     :PSE (all entries from PSE operations)
                     :STA (errlog startups)
                      null (all types of system entries)
     /SU            (create Summary)

     /QU            (create short summary)

     /-RP           (not include individual error reports)

     /DV:DEV[n]     (DB1: ,DR1: etc)
          :CMM      (comm devices)
          :UDI      (unidentified Interrupts)
          :PWR      (power fail)

     /BEG:time      (dd-mmm-yy:hh:mm:ss)
     /END:time      (dd-mmm-yy:hh:mm:ss)
```

RSX11M V4.x/RSX11M+ ERROR LOGGER

To run the RSX11M V4.x/RSX11M+ Error Logger proceed as follows after logging in:

```
>RUN $RPT
RPT> [report_file][/sw's]=[input_file][/sw's]
```

Example of a full error report:

```
RPT> [1,6]ERRLOG.LST=[1,6]LOG.RPT/F:F/T:A/W:N/DE:A
```

Switches:

```
  /T:A    [ll]
    :C    [ontrl]
    :E    [rrors]
    :M    [emory]
    PE    [ripherals]
   :PR    [ocessor]
    :S    [ystem_Info]

 /DA:P    [revious]:ndays
    :R    [ange]:start:end (DD-MMM-YY HH:MM:SS)
    :T    [oday]
    :Y    [esterday]

  /F:B    [rief]
    :F    [ull]
    :N    [one]
    :R    [egister]

  /W:N    [arrow]
    :W    [ide]

 /DE:A    [ll]
    :(device mnemonic)

 /PA:bbbb.xxx(:bbbb.xxx) [b=block #, x=record #]

  /R:D    [ay]
     M    [onth]
     W    [eek]
     S    [ystem]

 /SE:D    [rive]:number and/or
    :P    [ack]:number

  /V:volume_label

 /SU:parameter
     (one of — ALL,ERROR,GEOMETRY,HISTORY) (M+ only)
```

RSX CRASH DUMP ANALYZER (CDA)

When RSX crashes it will either go into XDT, if it was gen'd into the system, or transfer control to the EXECUTIVE CRASH DUMP ROUTINE. From XDT you can transfer control by typing an X to the XDT> prompt. You can also transfer control manually by restarting the processor at location 40. This can be used in the case of system halts or hangs. After control has been transferred, the following message will appear on the console terminal:

```
CRASH-CONT WITH SCRATCH MEDIA ON (device mnemonic):
```

The CPU then halts waiting for you to put the scratch media online and hit the CONTINUE switch on the processor console. When the dump is complete the CPU halts again and you may now reboot the system. Once the system is up you can run the CRASH DUMP ANALYZER program (CDA) to analyze the dump. The procedure is as follows:

```
>MOUNT ddu:/FOR (crash dump media on device ddu:)
>RUN $CDA
CDA>[list_file/sw's],[binary_file/sw]=[symbol_file/STB],crash_input[/sw's]
CDA>∧Z
>
```

Example, crash dump on MM0:

```
>MOUNT MM0:/FOR
>RUN $CDA
CDA>CRASH/-SP,CRASH/MEMSIZ:256.=[1,54]RSX11M.STB/STB,MM0:/
ALL/DUMP:0:1000000
CDA> ∧Z
>PIP TI:=CRASH.LST
```

Crash_Input Analysis Switches

/ACT or /ATL	= Control Blocks for Active Tasks
/ADV	= Control Blocks for all Devices
/ALL	= All Switches Except /DUMP /KDS /KIS /TASK /TDS /TIS
/CLI or /CPB	= Command Line Interpreter Parser Blocks
/CLQ	= Clock Queue
/CTL	= Device Control Tables and Request Blocks
/DEV or /SCB	= Control Blocks for all Active Devices
/DUMP:a:b	= Physical Memory Dump from Address a to b
/HDR	= Headers for Memory Resident Tasks
/KDS:a:b	= Kernel Dspace Virtual Dump from Address a to b (M+)
/KIS:a:b	= Kernel Ispace Virtual Dump from Address a to b (M+)
/PCB or /PAR	= Partition Control Blocks
/POOL[:a:b]	= System Pool Dump

Crash_Input Analysis Switches (cont.)

/SECPOOL[:a:b]	= Secondary Pool Dump (M+)
/STD or /TCB	= Control Blocks for the System Task Directory
/-SYS	= Suppress the System Information (first 5 pages)
/TASK:name:a:b	= Task "name"'s Virtual Space from address a to b
/TDS:name[:a:b]	= Task "name"'s Virtual Dspace from Address a to b (M+)
/TIS:name[:a:b]	= Task "name"'s Virtual Ispace form Address a to b (M+)

Crash_Input Function Switches

/BL:n	= Starting Octal Block of Crash_Input Device <BL:1>
/DENS:n	= MagTape Density, n = 800, 1600, low, high, <DENS:800>
/KMR	= Forces assignment of KAPR's for crash <-KMR>

List_File Function Switches

*/EXIT:n	= Terminate after n analysis errors <-->
*/LIMIT:n	= Limits List_File to n Pages <LIMIT:300.>
*/LINES:n	= Limits Page length to n Lines <LINES:60.>
/-SP	= Don't print the List_File <SP>

Binary_File Function Switch

*/MEMSIZ:n	= Save nK of Memory from Crash <MEMSIZ:124.>

Note: *n can be specified in decimal by putting a "." after the number, for example 256 decimal is specified in the command line as "256."*

ANALYZING RSX HALTS

RSX has several halts in the code but are almost always preceded by an ERROR message describing the problem. The halts are located in the CRASH code, PARITY code, POWER code and the INITL code (at least the only ones I could find!). The most common halt is in the CRASH code as RSX always halts on a system crash after asking you to specify the crash dump device and put it online.

With all halts there is one very important rule to follow and that is to always dump the registers (see pages 146-149). The halts can occur in 1 of 4 RSX modules (look these up on RSX uFiche for more detail). I have indicated if they apply to M and/or M+.

1. CRASH module

a) DOCRSH (M+)

This is where RSX11M+ halts after printing the "CRASH-CONT WITH SCRATCH.." message on the console. Remember to examine all registers before taking a dump as they may help in determining why we crashed. The halt can be identified by examining the data around the halt address (remember the console halt address is virtual and must be relocated, see RSTS-5 for details) for the following pattern:

```
xxxxxxxx/000000             HALT          ;hit continue for dump
xxxxxxxx/000167             JMP DUMP      ;do a crash dump
xxxxxxxx/000076
```

b) CRSHLT (M,M+)

This halt is where RSX11M/M+ goes after completing a dump or getting an error on the I/O on the crash device (exam registers). RSX11M also uses it as the location to halt at after printing the "CRASH-CONT WITH SCRATCH.." message on the console. When you halt here after taking a dump you can either reboot the system or hit continue to take another dump. The pattern in memory around this halt is?

For RSX11M:

```
xxxxxxxx/000773             BR 10$        ;branch
xxxxxxxx/000207             RTS           ;return
xxxxxxxx/000000   $CRSHLT:  HALT          ;crash halt, hit cont
xxxxxxxx/000413             BR DUMP       ;to do a crash dump
```

b) CRSHLT (M,M+) (cont.)

For RSX11M+:

```
xxxxxxxx/000000     $CRSHLT: HALT           ;dump complete, hit
xxxxxxxx/000167              JMP $CRALT     ;continue for another
xxxxxxxx/176700                             ;crash dump
```

c) CKSUM+16 (M/M+)

This halt will occur after starting a crash dump if the crash has corrupted the crash device's address table. A checksum is done on the address. No error message is printed. Since a dump can't be taken your only hope is the register dump you took earlier. The pattern in memory will be as follows:

```
xxxxxxxx/001402              BEQ 10$        ;checksum ok?
xxxxxxxx/000000              HALT           ;no, halt system
xxxxxxxx/000767              BR CKSUM       ;try checksum again
xxxxxxxx/000207              RTS            ;return
```

d) HALT 40 (M,M+)

If no crash dump support was included in the system it will halt at location 40 in memory. Look at the registers to find the cause of the crash. In this case memory will look like:

```
00000040/000000              HALT           ;no crash support
00000043/000776              BR 40          ;no cont possible
```

2. PARITY module

a) RSX will halt here if it gets a parity error in the EXECUTIVE or if it gets another parity error while servicing the first. The registers will give the error information on the most recent parity error and an area in memory called $MSTAT will contain info on the first parity error in the case of double error halts. The halt pattern in memory looks like:

```
xxxxxxxx/000771              BR 30$                 ;branch
xxxxxxxx/012600      50$:    MOV (SP)+,RO           ;pop stack to R0
xxxxxxxx/000000      60$:    HALT                   ;fatal parity error
xxxxxxxx/000776              BR 60$                 ;no cont possible
```

To find $MSTAT's address for the first error's information you need to EXAM location 114 to get the address of the service routine ($PARER). Remember this is a virtual address! Then EXAM $PARER for about 30 locations and look for this pattern where "aaaaaa" is the virtual address of $MSTAT.

```
xxxxxxxx/012700              MOV #$MPCSR,R0         ;move CSR address
xxxxxxxx/?                                          ;table to R0
xxxxxxxx/012701              MOV #$MSTAT,R1         ;move save status
xxxxxxxx/aaaaaa                                     ;area to R1
```

If you now EXAM aaaaaa ($MSTAT) the following registers were saved:

MSTAT:	+0	$40 Low error address register
	+2	$42 High error address register
	+4	$44 Memory system error register
	+6	$46 Cache control register
	+10	$50 Cache maintenance register
	+12	$52 Cache hit/miss register

3. POWER module

a) PWRFAIL (M/M+)

This halt occurs when RSX is finished it's power down code and is waiting for a powerup. The pattern in memory will look as follows:

```
xxxxxxxx/000000          HALT              ;wait for powerup
xxxxxxxx/016706          MOV $POWSP,SP     ;restore SP
xxxxxxxx/?                                 ;index to $POWSP
xxxxxxxx/052767          BIS #PMODE,PS     ;setup PSW mode
xxxxxxxx/030000                            ;PM=user
xxxxxxxx/177776                            ;PSW
```

b) PWRUP (M+)

This halt occurs is power is restored but a powerfail was not indicated. This can happen if the system is halted, powered off and then back on again or possibly on double powerfails. The area in memory will be as follows:

```
xxxxxxxx/000000     $5:  HALT              ;pwrup + no pwrdwn
xxxxxxxx/000776          BR 5$             ;no cont possible
xxxxxxxx/005067    10$:  CLR $PWKA0(R2)    ;reference only
xxxxxxxx/?                                 ;reference only
xxxxxxxx/005301          DEC R1            ;reference only
```

4. INITL module

There are various halts in the RSX initialization code. They are all preceded by an ASCII message on the console describing the problem. Examine the registers to determine the cause of the halt.

UNIX ERROR MESSAGES

Panic Traps

An unexpected system fault occured. On PDP11 the message goes:

KA6=#
APS=#
PC=#, PS=#
Trap type #

KA6 — is the contents of the segmentation register for the area in which the system stack is kept.
APS — is the location where the hardware stored the PSW during the trap.
PC — is the contents of the program counter.
PS — is the contents of the processor status word.

Trap type is one of the following:

0	Bus error
1	Illegal instruction
2	BPT/trace
3	IOT
4	Power fail
5	EMT
6	Recursive system call (trap instruction)
7	1170 Cache parity or programmed interrupt
8 or 10	Floating point trap
9 or 11	Segmentation violation (almost always software)

Note: *In some versions of UNIX, Panic trap types get echoed in octal. Thus panic 8 will be type 10 and 9 will be type 11.*

Other Panics:

Panic:parity

1# 2# 3# 4#

A parity error has occured in main memory. If this occurs in user mode, then the users process is simply terminated and no panic occurs. The four registers printed out in order are:

LEAR	HEAR	MSER	CCR
17777740	17777742	17777744	17777746

Panic:buffers

Insufficient memory space was found when the system was allocating the non-addressable buffer pool space. SOFTWARE! Either decrease these system parameters or the main memory size should be increased.

Panic:IO err in swap

An unrecoverable error has occurred during a system swap operation. Could be a hardware problem in the disk drive or controller. Could also be a bad spot on the pack.

Panic:no clock

A KW11L or KW11P was not found at the standard address during system startup. UNIX requires a PDP11 to have a clock.

Panic:double

You occasionally may see a "DOUBLE PANIC" message. This simply means that the system was processing one panic trap when a second occured.

Some Other Possible Messages

Bad block on "device name" drive #,[cntrl #], [slice #]

A block number not in the valid range of available free blocks on a file system has been detected. Device type may be any of those listed in the device error message description. SOFTWARE PROBLEM. (File system must be unmounted and checked).

Bad free count on "device name" drive #,[cntrl #],[slice #]

Corrupted free list block on file system was detected. Same device types as above. SOFTWARE PROBLEM. Same fix as above.

Stray interrupt at # (#=interrupt vector)

A device has interrupted through an unexpected vector on the unibus. The vector number printed is usually the correct value for a device. Could possibly be caused by a device specified at an incorrect vector in the system descriptor file. If this is not the case, then hardware is suspect.

**Death

Red zone stack violation always

Device Error Messages

Device error messages indicate that a hardware error has occured on a block type device. The error will appear as follows:

Device error on "device type", drive #,[cntrl #], [slice #]
bn=#,er=#,#

"device type": RM02/3/5,RM80,RP04/5/6,RP07,RP03,RK05,RL01/2, RF11,RS03/4,TU16,TU78,ML11 (TU16 for TU16,TE16, TRU45,TU77)

[contrl #]: 0 will appear if more than one controller is gen'd into your system (RH0,RH1,TM0,TM1).

bn=#: is the logical block number in error, followed by the contents of two of the device registers:

Device		Error Register	Control Register
RL01/2		RLCS	RLDA
RF11		RFCS	RFDAE
RK05		RKDS	RKER
RM05		RMER1	RMDS
RM80		RMER1	RMDS
RP03		RPER	RPDS
RP04/5/6	(HP.C)	RPER1	RPCS2
	(GD.C)	RPER1	MBA Status reg
RP07		RPER1	RPDS
RS03/4		RPCS2	
TU16	(HT.C)	MTER	MTCS2
		MTER	MBA Status reg

All register contents are printed in octal.

UNIX ERROR MESSAGES (cont.)

If after a crash, a tape dump is initiated and a tape error occurs the following message will appear:

ERR #,#

The first number is the address of the block being transmitted at the time of the error. The second number is MTER in the case of a TU16 and TUDTE in the case of a TU78.

In the case of a hard error on a disk the message will appear as follows:

Hard error on "drive type",drive #, [cntrl #],[slice #],
DS=#,ER=#,#,#

This message pertains only to RP04/5/6, RM05,RM80,RP07,ML11. In the case of the RP's the registers are RPDS followed by RPER1,RPER2,RPER3. In the case of the RM's they are RMDS followed by RMER1,RMMR2,RMER2..

To initiate a crash dump

1. Take a Hardware Register Dump first
2. Load address 44 or deposit 44 into R7
3. Press start or press proceed/continue