

DECSA

DECSA LOADABLE IMAGE  
CZLDIA0

AH-FF45A-MC  
1 OF 1 JUL 1985  
COPYRIGHT© 1985

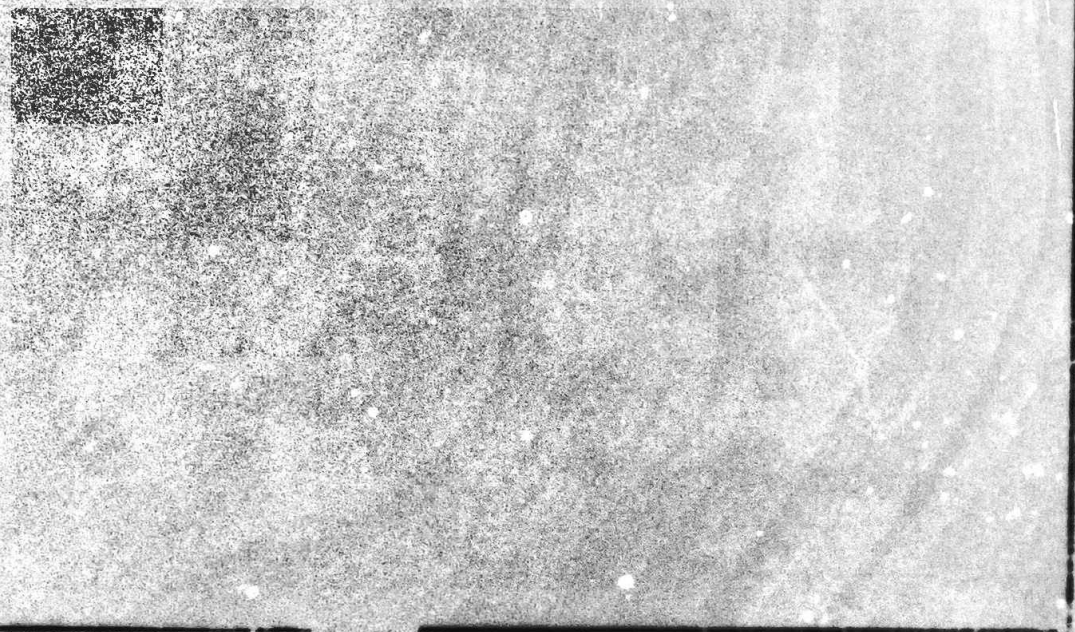
**digital**  
MADE IN USA

The image displays a grid of 40 small, illegible images or data points arranged in 8 rows and 5 columns on the left side of the page. The content is too faint to be transcribed, but it appears to be a structured set of data or a series of small diagrams.



B1

N i w  
A ::



5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37

IDENTIFICATION  
-----

.REM \*

PRODUCT CODE: AC-FF44A-MC  
PRODUCT NAME: CZLDIAO DECSA LOADABLE IMAGE  
PRODUCT DATE: 1-APRIL-85  
MAINTAINER: DISTRIBUTED SYSTEMS DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	



D1

39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58

DECSA  
LOADABLE DIAGNOSTIC IMAGE  
USERS GUIDE  
APRIL 1985  
AC-FF44A-MC



60	SECTION ONE	Loadable Diagnostic Image	
61			
62	1.0	General Information for the Loadable Diagnostic Image	4
63			
64	SECTION TWO	PAM Repair Diagnostics	
65			
66	2.0	General Information for CIDSAA and CIDSBA PAM Tests	7
67	2.1	Program Abstract	7
68	2.2	System Requirements	7
69	2.3	Related Documents and Standards	7
70	2.4	Diagnostic Hierarchy Prerequisites	7
71	2.5	Assumptions - Restrictions	7
72	2.6	Operating Instructions	7
73	2.7	Commands	8
74	2.8	Switches	8
75	2.9	Flags	9
76	2.10	Hardware Questions	10
77	2.11	Types of Error Messages	11
78	2.12	Device Error Messages	11
79	2.13	Test Summaries for CIDSAA PAM Test #1	13
80	2.14	Test Summaries for CIDSBA PAM Test #2	27
81			
82	SECTION THREE	Line Card Repair Diagnostics	
83			
84	3.0	General Info for CIDSCA and CIDSDA Line Card Tests	38
85	3.0	Program Abstract	38
86	3.1	System Requirements	38
87	3.3	Diagnostic Hierarchy Prerequisites	38
88	3.4	Assumptions - Restrictions	38
89	3.5	Operating Instructions	38
90	3.6	Hardware Questions	39
91	3.7	Error Information	40
92	3.8	Configuration Information	41
93	3.9	Test Summaries for CIDSCA Line Card Test #1	42
94	3.10	Test Summaries for CIDSDA Line Card Test #2	47
95			
96	SECTION FOUR	CBT Repair Diagnostic	
97			
98	4.0	General Information for CIDSEA CBT Test	51
99	4.1	Program Abstract	51
100	4.2	System Requirements	51
101	4.5	Assumptions	51
102	4.6	Operating Instructions	52
103	4.7	Hardware Questions	52
104	4.8	Software Questions	53
105	4.9	Error Message Formats	53
106	4.10	Test Summaries for CIDSEA	54
107			
108	SECTION FIVE	System Exerciser	
109			
110	5.0	General Information for SYSEXE - System Exerciser	57
111	5.1	Operating Instructions	57
112	5.2	Line and Slot Identification Under SYSEXE	57
113			
114	SECTION SIX	Updating the LDI to BL06 - CSVLDI.SYS	58
115			
116	SECTION SEVEN	Known Problems with LDI BL06	59

118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169

## 1.0 GENERAL INFORMATION for the Loadable Diagnostic Image

The LDI consists of many software components residing in one large image. The purpose of one image is to allow the testing of the DECSA Subsystem as configured without user interaction.

Execution of the LDI (once the image has been loaded) requires PLUMON to be loaded in a run state. The VMR utility allows you to issue a RUN command to an installed task before the image is saved. Both the RSX-11S and PLUMON (PLU>) will be in this state. PLUMON is the initial controlling task for the LDI. Upon initial execution PLUMON will determine the mode of operation, AUTO or MANUAL. The mode selection is made from a value in a CBT read/write register. The CBT ROM code will deposit a -1 value in this register for AUTO mode and clear it for MANUAL mode.

DECSA short self-test and LDI load is selected by first pressing the "start" button and then when the LEDs are flashing at the quick rate pressing the "test" button.

Manual mode is selected by putting the test button in the out position while the LDI is being loaded, as indicated by the L 5n in the LEDs.

Automatic mode is selected by the "test" button being in the "in" position when the LDI has completed loading and has started.

### Uses of the DECSA TEST BUTTON.

test button	mode	comments
-----	----	-----
in	automatic mode	The 5 diags + sysexe should execute followed by operating system boot. Verify all diags are complete.
out	manual mode	The PLU> should be displayed. Run SYSEXE selecting # of passes and loopback.
out	manual mode	The PLU> should be displayed. Run each of the 5 diags. Run the diags selecting external loopback.
out	manual mode	The PLU> should be displayed. Type in "AUTO". The 5 diags and SYSEXE should run followed by a boot request for the operating system.



171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186

Currently there are five diagnostics and a system exerciser that can be executed either in AUTO mode or executed separately in MANUAL mode.

DIAGNOSTICS:

CIDSAA	REV C	PAM REPAIR TEST #1
CIDSBA	REV C	PAM REPAIR TEST #2
CIDSCA	REV C	LINE CARD REPAIR TEST #1
CIDSDA	REV C	LINE CARD REPAIR TEST #2
CIDSEA	REV C	CBT TEST

SYSTEM EXERCISER:

SYSEXE

188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234

LOADABLE DIAGNOSTIC IMAGE  
MEMORY ALLOCATION

```
*****  
*                               *  
*          RSX11-S             *  
*                               *  
*****  
*                               *  
*          DECNET              *  
*          UNA MICROCODE      *  
*          PAM MICROCODE      *  
*                               *  
*****  
*                               *  
*          PLUMON              *  
*                               *  
*****  
*                               *  
*          FRUMON              *  
*                               *  
*****  
*                               *  
*          DRS/RSX MODULE      *  
*                               *  
*****  
*                               *  
*          PAM DIAGNOSTICS     *  
*                               *  
*****  
*                               *  
*          LINE CARD DIAGNOSTICS *  
*                               *  
*****  
*                               *  
*          CBT DIAGNOSTIC      *  
*                               *  
*****  
*                               *  
*          SYSEXE              *  
*                               *  
*****  
*                               *  
*          COMMON MSG BUFFER   *  
*          COMMON DATA BUFFER *  
*          DEVICE I/O PAGE DEF *  
*                               *  
*****
```



236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284

## 2.0 GENERAL INFORMATION for CIDSAA and CIDSBA PAM Tests

### 2.1 PROGRAM ABSTRACT

The "PAM" repair level diagnostic (1) programs is meant to provide field service and manufacturing with a tool to maintain the "digital ethernet communication server," "protocol assist modules (PAM). "The program will provide the coverage necessary to detect failures in the "PAM" module set only. Fault detection is to the functional level, while fault isolation is to board (M3110 or M3111).

### 2.2 SYSTEM REQUIREMENTS

In order to run this diagnostic program, the following minimum hardware is required:

- A PDP-11 CPU "PROTOCOL PROCESSOR (PP)" (PDP 11/24)
- MINIMUM OF 256K WORDS OF SYSTEM MEMORY
- CONSOLE BOOT TERMINATOR (CBT)
- RSX11-S "LDI" SOFTWARE OR XXDP+ SUPPORTED LOAD MEDIA
- AT LEAST ONE "PAM" MODULE SET CONSISTING OF AN M3110 AND M3111

### 2.3 RELATED DOCUMENTS AND STANDARDS

- XXDP+ USER'S MANUAL (CHQUS?.SEQ WHERE ? IS THE REV. LEVEL OF THE MANUAL - "C" IS THE CURRENT REV.).

### 2.4 DIAGNOSTIC HIERARCHY PREREQUISITES

The goal of the "PAM" diagnostic program is to test the M3110 and M3111 therefore, it is assumed that the "self test diagnostic" has run, and the "CBT" and "system memory" are fully functional. A failure in the aforementioned devices could fail this diagnostic and the user should be aware of this possibility.

### 2.5 ASSUMPTIONS - RESTRICTIONS

It is assumed that the prerequisite diagnostics have been executed (refer to section 2.4). The operator should also be familiar with the operating instructions in section 2.6.

### 2.6 OPERATING INSTRUCTIONS

Section 2.7 - 2.10 contains a brief description of the Pluto runtime services (PLU>). For detailed information, refer to the XXDP+ user's manual (CHQUS).

286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338

## 2.7 COMMANDS

There are eleven legal commands for the diagnostic runtime services (SUPERVISOR). This section lists the commands and gives a very brief description of them. The XXDP+ user's manual has more details.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED
PROCEED	CONTINUE FROM AN ERROR HALT
*EXIT	RETURN TO PLUMON (SEE NOTE)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (NOT IMPLEMENTED BY THE LDI)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS
ZFLAGS	CLEAR ALL FLAGS

A command can be recognized by the first three characters. So you may, for example, type "STA" instead of "START".

\*NOTE: After completion of a diagnostic run, type "EXIT" at the DR> prompt to get back to the PLUMON prompt "PLU>" to run the next diagnostic or SYSEXE. Also refer to the NOTE in section 2.8 on switches.

## 2.8 SWITCHES

There are several switches which are used to modify supervisor operation. These switches are appended to the legal commands. All of the legal switches are tabulated below with a brief description of each. In the descriptions below, a decimal number is designated by "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. (SEE SECTION 2.9)
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)



340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
3  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

The effect of this command will be:  
1) TESTS 1 THROUGH 5 WILL BE EXECUTED.  
2) ALL UNITS WILL TESTED 1000 TIMES.  
3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY.

A switch can be recognized by the first three characters. you may, for example, type "/TES:1-5" instead of "/TESTS:1-5".

NOTE: When running under the LDI it is good practice to set the PASS and HALT ON ERROR Flags, so you can get back to the PLU> prompt by typing "EXIT".

STA/PASS:1/FLA:HOE

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.9 FLAGS

Flags are used to set up certain operational parameters such as looping on error. All flags are cleared at startup and remain cleared until explicitly set using the flags switch. Flags are also cleared after a start command unless set using the flag switch. The ZFLAGS command may also be used to clear all flags. With the exception of the START and ZFLAGS commands, no commands affect the state of the flags they remain set or cleared as specified by the last flag switch.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBE*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)

398		
399		
400	PRI	DIRECT MESSAGES TO LINE PRINTER
401	PNT	PRINT TEST NUMBER AS TEST EXECUTES
402	BOE	"BELL" ON ERROR
403	UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
404	ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
405	IDR	INHIBIT PROGRAM DROPPING OF UNITS
406	ADR	EXECUTE AUTODROP CODE
407	LOT	LOOP ON TEST
408	EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)
409		
410		

\*ERROR MESSAGES ARE DESCRIBED IN SECTIONS 2.11, 3.10 AND 4.7

See the XXDP+ user's manual for more details on flags. You may specify more than one FLAG with the flag switch. For example, to cause the program to loop on error, inhibit error reports and type a "BELL" on error, you may use the following string:

/FLAGS:LOE:IER:BOE

## 2.10 HARDWARE QUESTIONS

When a diagnostic is started, the runtime services will prompt the user for hardware information by typing "CHANGE HW (L) ?" you must answer "Y" after a start command unless the hardware information has been "preloaded" using the setup utility (see chapter 6 of the XXDP+ user's manual). When you answer this question with a "Y", the runtime services will ask for the number of units (in decimal).

The "PAM" repair diagnostic will test up to two units. However, the diagnostic automatically checks to see if the requested units for test are there and drops any not responding. Also, the "CBT" is checked for a one or two "PAM" system indicator (CBT DCR BIT0) and drops those units that do not, according to the sizing program, belong. The user may wish to inhibit the dropping of units by setting the flag "inhibit program drop macro (IDU)".

411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437



439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491

# UNITS (D) ? 2<CR>

UNIT 0

Unibus Address of "PAM" ? 171200<CR>

Hard Error Interrupt Vector ? 130 <CR>

Soft Error Interrupt Vector ? 134 <CR>

UNIT 1

Unibus Address of "PAM" ? 171000<CR>

Hard Error Interrupt Vector ? 140 <CR>

Soft Error Interrupt Vector ? 144 <CR>

## 2.11 TYPES OF ERROR MESSAGES

There are three levels of error messages that may be issued by a diagnostic: general, basic and extended. General error messages are always printed unless the "IER" flag is set (section 2.9). The general error message is of the form:

```
NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:XXXXXX
ERROR MESSAGE
```

WHERE NAME = DIAGNOSTIC NAME  
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)  
NUMBER = ERROR NUMBER  
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)  
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED  
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

Basic error messages are messages that contain some additional information about the error. These are always printed unless the "IER" or "IBE" flags are set (section 2.9). These messages are printed after the associated general message.

Extended error messages contain supplementary error information such as register contents or good/bad data. These are always printed unless the "IER", "IBE" or "IXE" flags are set (section 2.9). These messages are printed after the associated general error message and any associated basic error messages.

## 2.12 DEVICE ERROR MESSAGES

Error messages that occur in the initialize code, due to the SIZING program finding fault with the expected and received PAM configuration, are as follows:

a. The SIZE program couldn't find PAM1 in the system.

a.1. PAM1 is not in the system and should be: Unit 0 dropped

b. The SIZE program couldn't find PAM2 in the system but the CBT indicates it should be there ( BIT0=0 in DCR ).

493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546

b.1. PAM2 is not in the system and should be: Unit 1 dropped

c. The SIZE program found PAM2 in the system but the CBT indicates that it shouldn't be there ( BIT0=1 in DCR ).

c.1. PAM2 is present and should not be.

The following is a list of the basic format followed in printing Device Error messages in this diagnostic.

-----

This message says that the Micro-Instruction LSI[] failed to move data to Local Storage correctly.

Local Storage Address Mux Test Failed

Local Storage Addressing Scheme LSI[] Failed

Address in Error == 171234

Expected Data == 125

Received Data == 333

Contents of (SEQA) == 00043

-----

This message says that the Soft error Interrupt occurred before the hard error interrupt.

Force Hard/Soft error Interrupt test failed

Interrupts occurred out of sequence

Last Interrupt Expected == 130

last Interrupt Received == 134

-----

This message says that an ADD instruction failed in the high nibble 2901 slice.

ALU (2901) Function test failed

Expected results == 340

Oprnd 1 ==000

Oprnd 2 ==340

Function == ADD



548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595

Results == 240

## 2.13 TEST SUMMARIES For CIDSAA PAM Test #1

### TEST 1

This test will check the ability to Read/Write all locations in the PAM address space. The interrupt service routine (VECTOR 4) will set an error flag to indicate that an interrupt occurred. The diagnostic does a Read, checks the error flag, Write and checks the error flag again. If an error flag was set after the read or write, the diagnostic will report the address and the function in error.

### TEST 2

This test will check that CSR1 R/W bits can be set and cleared from the Unibus and can be cleared by "INIT" (Bit 03). All bits are first written with ones (except "FORCE PE", "LCPRS", "INTENB", "INIT", "RUN" "SINGLE STEP") and then checked to see that the correct bits were set. CSR1 is then written with zeros and reread to check that the bits cleared. CSR1 is again written with ones ( except "FORCE PE", "LCPRS", "INTENB", "RUN" and "SINGLE STEP") but this time "INIT" is set (Bit 03) also. All bits, except Line card Present which is not checked, should be cleared when reread.

### TEST 3

This test will check that CSR2 R/W bits can be set and cleared from the Unibus and can be cleared by "INIT" (CSR1 Bit 03). The register is written with different data patterns and checked to see that the correct bits were set or cleared. CSR2 is again written with ones but this time "INIT" is set in CSR1. All bits should be cleared when read and rechecked by the diagnostic.

### TEST 4

This test checks for SA1 and SA0 bits in the WCSA register by writing several data pattern to the register and reading/verifying the results of the write. The following patterns are used:

125252  
052525  
031463  
007417

597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650

## TEST 5

This test will check for SA0 and SA1 bits in WCS by first writing all location with a given pattern and then reading WCS to verify the data. The diagnostic will dump the address in error, expected data, received data and XOR data. The following patterns are used:

252  
125  
063  
017

## TEST 6

This test will perform a dynamic check of WCS by using a modified Moving Inversions algorithm. Starting with WCS cleared to all zeros, 24 passes are made (one for each data bit) from the lowest to the highest address. Each location is first read to verify that the background pattern was stored correctly, then a single bit is rewritten to the location and reread to verify that the new-pattern was stored correctly. This process is repeated until WCS is filled with all ones. The next step ( step 2 ) is to repeat the above process on WCS, now with an all ones background pattern, but this time each individual bit is cleared. This will leave WCS filled with zeros and ready for the next step. Step 3 and 4 are the same as 1 and 2 but the sequence starts at the Highest WCS address and works to the lowest. The key to the moving inversions test is doing the Read-Write-Read sequence as fast as possible. Therefore, the check of data is done after the Read-Write-Read sequence has completed.

## TEST 7

This test will check for SA1 and SA0 bits in Local Storage. All Local Storage location are first written with a given data pattern. The diagnostic then reads all locations and verifies the data. If an error occurs, the LS Address, Data Written, Data Read and the XOR Data are output to the terminal.

## TEST 8

This test will perform a dynamic check of Local Storage by using a modified Moving Inversions algorithm. Starting with local storage cleared to all zeros, 8 passes are made (one for each data bit) from the lowest to the highest address. Each location is first read to verify that the background pattern was stored correctly, then a single bit is rewritten to the location and reread to verify that the new-pattern was stored correctly. This process is repeated until local storage is filled with all ones. The next step ( step 2 ) is to repeat the above process on local storage, now with an all ones background pattern, but this time each individual bit is



652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698

cleared. This will leave local storage filled with zeros and ready for the next step. Step 3 and 4 are the same as 1 and 2 but the sequence starts at the Highest local storage address and works to the lowest. The key to the moving inversions test is doing the Read-Write-Read sequence as fast as possible. Therefore, the check of data is done after the Read-Write-Read sequence has completed.

## TEST 9

This test will check that a Local Storage parity error can be forced by using "Force Parity Error" in CSR1. Force Parity error (FPE) is set in CSR1 and then data is written in Local Storage. The data written should have bad parity and should cause a parity error (LSPE) when read. The diagnostic will read the Local Storage location and then check that LSPE and PE both set in CSR1. Several data patterns are used when loading local Storage to assure the integrity of the Parity checkers and generators.

\*\*\*\*\* Interrupts are disabled in this test \*\*\*\*\*

## TEST 10

This test will check that the 2911 Microsequencers are able to Sequence through all WCS Addresses. This is accomplished by loading all locations in WCS with A HALT instruction then overwriting locations as follows:

Location	Instruction	
0000	R[0] <-- [1]	; Load A number in Reg.
0001	LS[7760] <-- R[0]	; Write LS
0001	BR[1463]	; Branch
1463	LS[7761] <-- R[0]	; Write LS+1
1464	BR[2525]	; Branch
2525	LS[7762] <-- R[0]	; Write LS+2
2526	BR[7417]	; Branch
7417	LS[7763] <-- R[0]	; Write LS+3
7420	LS[END] <-- [-1]	; Set done
7421	HALT	; HALT

When this Microroutine Runs, Local Storage Locations 7760 to 7763 will be Incremented and the Microsequencers should halt at Location 7421. The Macrocode will report, if an error occurs, the Expected and Received HALTED SEQUENCER ADDRESS, and the Expected and Received contents of LOCAL STORAGE.

700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753

## TEST 11

This is a test of the ability to read and write the MSR. Command-Segment-Descriptor-Block-Entry is first set in CSR1 and then microcode is started. The micro-code will write zeros to all bits except 6 ( Clock ) in the MSR and then store the contents of MSR in Local Storage. All bits in the MSR are then written with ones except 7,6 and 1 ( hrd-err-int, clock and sft-err-int ). After several micro-cycles the MSR is again read and the contents stored. The micro-cycles between write and read of the MSR is to allow bit 6 (clock) to reset. This test is not a check of Clock timing, only a check of read/write bits in the MSR.

## TEST 12

This test will check to see that the Local Storage Address Mux can properly select the correct input for the different Local Storage Addressing modes. Local Storage 2525 and 5252 are the locations used as the sources and destinations for for the MOV instructions.

## LIMITATIONS:

The Programmable Line Number register must be operational for this test to work. The Local Storage addressed by Special Character MOV instruction is not used in this test. This instruction will be tested in a later test.

## TEST 13

This test will check that there are no Ram A/B address lines SA1/SA0 or shorted together. All Ram locations, except locations "5", "12", and "14", are first written with Zero. The locations 5,12, and 14 are then written with Ones, followed by a read of all other locations to Local Storage. The action of writing the ones will overwrite any zero'd locations address with ones if the address lines are tied together. Fore example: if address lines 0 and 1 are shorted, then when address 5 is written, location 7 would be overwritten with Ones. The next step is to rewrite all locations with zero except locations 5,12 and 14 and then read and save in Local Storage the unwritten locations ( 5,12,14 ). The action of writing the locations will again force an overwrite into one of the unwritten locations ( 5,12,14 ) if the address lines are shorted. The diagnostic will read Local Storage an verify the integrity of the data written to each Ram location.

## TEST 14

This test will check the 2901 Ram locations for SA1 and SA0 bits. Data patterns are written to Ram and the Ram is written to local storage for verification by the diagnostic. The following patterns are used: 125, 252, 314 and 360.



755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808

TEST 15

This test will check the ability of the 2901 to rotate the RAM left. A ram location is loaded with data to be shifted. The data is then shifted and written to Local Storage for examination by the diagnostic. Local Storage should look as follows when the test completes:

LS Address	Data
7760	002
7761	004
7762	010
7763	020
7764	040
7765	100
7766	200
7767	001

TEST 16

This test will check that the 2901 RAM can be shifted Right. Data is loaded into a ram location to be shifted. The RAM location is then shifted and the results written to Local Storage for examination by the diagnostic. Local Storage should look as follows when the test completes:

Local Storage Address	DATA
7760	100
7761	040
7762	020
7763	010
7764	004
7765	002
7766	001
7767	200

TEST 17

This test will check that the Q register and the RAM can be shifted left. Both the Q and a RAM location are loaded with data to be shifted. The registers are shifted eight times and read after each shift to Local Storage. Local Storage should look as follows when the test completes:

Local Storage address	Data
7740	002 ( Q register data )
7741	004
7742	010
7743	020
7744	040
7745	100

810			
811		7746	200
812		7747	001
813			
814		7750	002 ( RAM data )
815		7751	004
816		7752	010
817		7753	020
818		7754	040
819		7755	100
820		7756	200
821		7757	001

TEST 18

This test will check that the Q register and the RAM can be shifted Right. Both the Q and a RAM location are loaded with data to be shifted. The registers are shifted eight times, each time writing the shifted data to Local Storage. Local Storage should look as follows when the test completes:

Local Storage address	Data	
	7740	100 ( Q register data )
	7741	040
	7742	020
	7743	010
	7744	004
	7745	002
	7746	001
	7747	200
	7750	100 ( RAM data )
	7751	040
	7752	020
	7753	010
	7754	004
	7755	002
	7756	001
	7757	200

TEST 19

This test will check the Q register for SA1/SA0 bits and Check that writing the Q/RAM locations do not affect each other. The Q is first written with data patterns and each time the contents is saved in Local Storage. Next, Ram location 0 is cleared and the Q written with 377. The RAM location is then saved in local storage and again written with ZERO. The contents of the Q is then saved in Local Storage.

810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858



860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912

TEST 20

This test will check the ALU ( 2901 ) functions using the microcode CALC instructions ( the opcode roms are not tested). The microcode will fetch two Operands from Local Storage. Each function is executed on the Operands and the results written to an assigned Local Storage location. The diagnostic will read and verify the results of each operation. Several passes through the diagnostic are made with different operand pairs to fully check 2901 operation. Local Storage locations are assigned as follows:

Local Storage Address	Function assigned
7760	"OR" results
7761	"AND" results
7762	"XOR" results
7763	"XNOR" results
7764	"NOTRS" results
7765	"ADD" results
7766	"SUBR" results
7767	"SUBS" results
7770	Operand 1
7771	Operand 2

TEST 21

This test will check the ALU ( 2901 ) functions using the microcode Opcode "G" instructions in an attempt to check the I/O of the opcode roms. Each function is executed on an Operand and the results written to an assigned Local Storage location. This test is not an attempt to check the 2901 ALU, only the opcode roms inputs and outputs. Local Storage locations are assigned as follows:

Local Storage Address	Function assigned
7760	"ADD" results (Opcode 40)
7761	"ADD" results (Opcode 50)
7762	"SUBS" results (Opcode 41)
7763	"SUBS" results (Opcode 51)
7764	"SUBD" results (Opcode 42)
7765	"SUBD" results (Opcode 52)
7766	"OR" results (Opcode 43)
7767	"OR" results (Opcode 53)
7770	"AND" results (Opcode 44)
7771	"AND" results (Opcode 54)
7772	"XOR" results (Opcode 46)
7773	"XOR" results (Opcode 56)
7774	"XNOR" results (Opcode 47)
7775	"XNOR" results (Opcode 57)
7776	
7777	DONE FLAG

914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965

TEST 22

This test will check that all read modify write instructions used on Local Storage work correctly. The test mainly checks the two Opcode Decode Roms on the M3110 board. Microcode operates on instruction dependent operands stored in Local Storage. The operands are chosen to assure that Both 2901 slices must work on the data. The Diagnostic will then check Local Storage locations for correctness of data and report any errors.

TEST 23

This test will check that the CALL and RTS functions in Microcode work and the Micro-Stack is the correct depth. Four consecutive CALLS are made to different routines in WCS. Each routine does a CALL to the next routine until the last routine is reached. The last ( routine 4 ) Microroutine writes a location in Local Storage and then does an RTS to the previously called routine which also increments a Local Storage location and an RTS. The process is continued (Increment Local Storage then do RTS) until the Micro-Stack is empty and the Micro-PC has returned to the starting Micro-Address+1. Local Storage will then be read by the Diagnostic to verify that all Micro-Routines were hit. Local Storage should contain the following:

Address	Data
7760	001
7761	001
7762	001
7763	001
7764	001

It should be noted that the Micro-Routines are NOT loaded in contiguous WCS locations as it may appear in the listing.

TEST 24

This test will check the POP function in Microcode. Four consecutive CALLS are made to different routines in WCS. Each routine does a CALL to the next routine until the last routine is reached. The last Microroutine ( routine 4 ) writes a location in Local Storage and then does three consecutive POPs of the Micro-Stack followed by an RTS. The RTS should bring the Micro-PC back to the starting Micro-Address+1. Local Storage will then be read by the Diagnostic to verify that the first and last Microroutines increment Local Storage (NO OTHER MICROROUTINE WAS RETURNED TO). Local Storage should contain the following:



967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015

Address	Data
:-----:	
7760	001
7761	000
7762	000
7763	000
7764	001

It should be noted that the Micro-Routines are NOT loaded in contiguous WCS locations as it may appear in the listing.

## TEST 25

This test checks that the Micro-sequencer is capable of Single Stepping through a Microroutine and the Micro-sequencer address register is operating correctly. The micro-routine is single-stepped through each micro-instruction while examining the contents of the Sequencer address register. If the address is correct in the register, then the Sequencer is single-stepped again. This process continues until the Micro-routine has halted. Local Storage is then examined to verify that all instructions functioned correctly.

Local Storage should contain the following:

Address	Data
:-----:	
7760	001
7761	001
7762	001
7763	001
7764	001

It should be noted that the Micro-Routines are NOT loaded in contiguous WCS locations as it may appear in the listing.

## TEST 26

This test will check that the SYNC bit in the Microword will halt the Microprocessor and will set PE and SYNC ACK in CSR1. A Microroutine is loaded that has SYNC set in two of the microwords. The diagnostic will start the microcode and wait for SYNC ACK and PE to set and the RUN bit to clear in CSR1. SYNC ACK and PE are then cleared and RUN reset to allow the Microroutine to continue. Again the aforementioned sequence is repeated and the expected results of the Microroutine is examined. Any errors in status or the Microroutine results is reported.

1017  
 1018  
 1019  
 1020  
 1021  
 1022  
 1023  
 1024  
 1025  
 1026  
 1027  
 1028  
 1029  
 1030  
 1031  
 1032  
 1033  
 1034  
 1035  
 1036  
 1037  
 1038  
 1039  
 1040  
 1041  
 1042  
 1043  
 1044  
 1045  
 1046  
 1047  
 1048  
 1049  
 1050  
 1051  
 1052  
 1053  
 1054  
 1055  
 1056  
 1057  
 1058  
 1059  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065  
 1066  
 1067

## TEST 27

This test will check that a WCS parity error can be forced and the correct bits set in CSR1. A Microroutine is loaded in WCS with a bad parity microwords. The diagnostic will start the routine and check that WCSPE and PE both set in CSR1. Process on error is also set so the Microroutine should complete the Microroutine correctly.

\*\*\*\*\* Interrupts are disabled in this test \*\*\*\*\*

Local Storage locations are assigned as follows:

Local Storage Address	Function assigned
7760	"OR" results
7761	"AND" results
7762	"XOR" results
7763	"XNOR" results
7764	"NOTRS" results
7765	"ADD" results
7766	"SUBR" results
7767	"SUBS" results
7770	Operand 1
7771	Operand 2

## TEST 28

This is a check of the Special Character Recognition Register bits 0 to 2 and Mux. The microcode loads the Special Character register (SCR) by doing consecutive loads of the Destination Register (DR). Bits 0 to 2 of the SCR select the bit in the DR to be tested. If the selected bit is set then the Branch on Special Condition will be taken. Both branch and no branch conditions are tested ( bit under test set and cleared ) for all bits in the destination register.

## TEST 29

This is a check that Local Storage can be addressed by Special Char. register. Local Storage can be addressed by a combination of Special Character register, Line Number register and Microword. The Line Number register contents is used as LS address bits 3 to 7, while the Special Char. reg. contents is used as LS address bits 0 to 2 and 8 to 9. LS address bits 10 and 11 are derived from the Microword. Locations 2525 and 5252 in Local Storage are used in the transfer of data. These locations correspond to setting and clearing each bit in the Local Storage address. The test is successful if data is correctly moved to and from these locations.



1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121

TEST 30

This test will check that the Interlock function is working correctly. The microcode loads a location in Local Storage with Bit 0 Set. The location is then complemented-leaving the location with bit 0 clear ( 376 )- and the interlock branch is tested. Since Bit 0 was set before the compliment function, the interlock flop will set and the branch will be taken. The local storage location is again complimented (001) leaving bit 0 set and the branch is again tested. Since bit 0 was clear before the compliment, the branch will not be taken.

TEST 31

This test will check that the microcode can do a CALL, RTS and POP on a condition code. The interlock condition is used only because forcing interlock requires minimal hardware and because it has already been tested. No attempt is made to test all the possible condition codes only the Micro-sequencer control bits are tested. Local Storage locations are used to save function indicator codes as follows:

Local Storage Address	function	code
7760	Work Location	1 or 376
7761	CALL_IL [ ]	1 or 2 (condition set - clear)
7761	RTS_IL	3 or 4 (condition set - clear)
7761	POP_IL	5 or 6 (condition set - clear)
7775	ERROR FLAG	1 ( Micro-Stack error )
7776	ERROR FLAG	1 (Function error)
7777	DONE FLAG	377

When the function is started (CALL\_IL, POP\_IL or RTS\_IL), the routine will write its code to Local Storage. If the routine worked then the test continues to the next function; but, if the routine failed then the error flag is set and the Microroutine halts leaving the failing function code in Local Storage.

TEST 32

This test will check the Micro-Sequencer control logic and the Oring Mux during a CALL on Low Nibble. The destination register is loaded with a data pattern and a CALL\_LN [ ] is done to a 16 location target area. The CALL\_LN will "OR" the low nibble of the destination register with bits 0 to 3 of the called address. Each location in the target area will increment the same location in Local Storage until the table is exhausted and the RTS is performed. If the first location in the table is hit (lowest address in the table), then the number in Local Storage when when the RTS is performed will be "20" (octal) if the location "12" (oct.) is hit then the Local Storage location will contain a "6". Several data patterns are

1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177

loaded to the destination register to check the integrity of the Oring Mux during the call and are as follows:

005  
012  
014  
165  
172  
174

#### TEST 33

This test will check the Micro-Sequencer control logic and the Oring Mux during a CALL on High Nibble. The destination register is loaded with a data pattern and a CALL\_HN [ ] is done to a 16 location target area. The CALL\_HN will "OR" the high nibble of the destination register with bits 0 to 3 of the called address. Each location in the target area will increment the same location in Local Storage until the table is exhausted and the RTS is performed. If the first location in the table is hit (lowest address in the table), then the number in Local Storage when the RTS is performed will be "20" (decimal) if the location "12" (oct) is hit then the Local Storage location will contain a "6". Several data patterns are loaded to the destination register to check the integrity of the Oring Mux during the call and are as follows:

120  
240  
300  
137  
257  
317

#### TEST 34

This test will check that the Micro-sequencer is able to branch correctly on High Nibble and Low Nibble. The test only checks that the branches can be taken properly without affecting the micro-stack. It should be assumed that the Oring Mux is working properly by virtue of previous testing.

#### TEST 35

This test will check Bit-Test Mux and the Micro-sequencer functionality by using micro-code Bit-Test instructions. The micro-code floats a "1" on a background pattern of zeros and a "0" on a background pattern of ones through the destination register. The Micro-code sequence is as follows:

- A. Do the following for bits 0 to 7
1. Set the bit to test.
  2. Test the bit.
  3. Compliment the bit pattern.
  4. Test the bit.



1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232

## TEST 36

This test will check that the "N" bit will set/clear and not effect, or be affected by the Z,C or V condition codes. The Micro-code will write a register with a negative number and then store the register contents in Local Storage. The condition codes are then checked by taking the correct branch. If the branch fails, an error flag, CC set/clear flag, data used and a function code are written to Local Storage. The function codes are as follows:

BMI == 0  
BPL == 1  
BEQ == 2  
BNE == 3  
BCS == 4  
BCC == 5  
BVS == 6  
BVC == 7

## TEST 37

This test will check that the "C" bit will set/clear and not effect, or be affected by the Z,N or V condition codes. The Micro-code will write two registers with different operands for the ALU to to ADD. The condition codes are then checked by taking the correct branch. If the branch fails, an error flag, CC set/clear flag, data used, results of the add and a function code are written to Local Storage. The function codes are as follows:

BMI == 0  
BPL == 1  
BEQ == 2  
BNE == 3  
BCS == 4  
BCC == 5  
BVS == 6  
BVC == 7

## TEST 38

This test will check that the "V" bit will set/clear and not effect, or be affected by the Z,N or C condition codes. The Micro-code will write two registers with different operands for the ALU to to ADD. The condition codes are then checked by taking the correct branch. If the branch fails, an error flag, CC set/clear flag, data used, results of the add and a function code are written to Local Storage. The function codes are as follows:

BMI == 0  
BPL == 1  
BEQ == 2

1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287

BNE == 3  
BCS == 4  
BCC == 5  
BVS == 6  
BVC == 7

## TEST 39

This test will check that the "Z" bit will set/clear and not effect, or be affected by the N,C or V condition codes. The Micro-code will write a register with data patterns to set or clear the Z Bit. The condition codes are then checked by taking the correct branch. If the branch fails, an error flag, CC set/clear flag, data used and a function code are written to Local Storage. The function codes are as follows:

BMI == 0  
BPL == 1  
BEQ == 2  
BNE == 3  
BCS == 4  
BCC == 5  
BVS == 6  
BVC == 7

## TEST 40

This test will check the hard error interrupt control function through interrupt vector 130 or 140 (PAM1 or PAM2). An interrupt is forced by setting "Hard Error Interrupt" in the Micro-code Status register. The interrupt service routine (HARDSERV) will save the status in CSR1, clear the error condition and restart the micro-code.

The interrupt service routines for hard and soft error interrupts use a common flag word in memory and its format is as follows:

Bits 0 to 7 = Interrupt Vector (Written by Service routine)  
Bit 8 = Hard Error Interrupt (Vector 130/140 Ser. rout. sets)  
Bit 9 = Soft Error Interrupt (Vector 134/140 Ser. rout. sets)  
Bit 10 = Double interrupt through vector 130/140  
Bit 11 = Double interrupt through vector 134/144

If the reported CSR1 status is 0 then the interrupt service routine did not read status at time of interrupt.

## TEST 41

This test will check the Soft error interrupt control function through interrupt vector 134 or 144 (PAM1 OR PAM2). An interrupt is forced by setting "Status Segment Descriptor Block Interrupt" in the Micro-code Status register. The interrupt service routine (SOFTSERV) will save the status



1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341

in CSR1 at interrupt time and load the interrupt flag word.  
The status in CSR1 should reveal that the Run bit remained set.

The interrupt service routines for hard and soft error interrupts use a common flag word in memory and its format is as follows:

Bits 0 to 7 = Interrupt Vector (Written by Service routine)  
Bit 8 = Hard Error Interrupt (Vector 130/140 Serv. rou. sets)  
Bit 9 = Soft Error Interrupt (Vector 134/144 Serv. rou. sets)  
Bit 10 = Double interrupt through vector 130/140  
Bit 11 = Double interrupt through vector 134/144

If the reported CSR1 status is 0 then the interrupt service routine did not read status at time of interrupt.

#### TEST 42

This test will attempt to force both Hard error and Soft error interrupts through interrupt vectors 130/140 and 134/144, respectively. The interrupts are forced by setting "Hard Error Interrupt" and "Status Segment Descriptor Block Interrupt" in the Microcode Status register. Both bits, in the microcode Status register, are set at the same time this should cause the Hard Error Interrupt to occur first ( Vector 130/140 ) and then the Soft Error Interrupt ( Vector 134/144 ). The hard error interrupt will halt the PAM ; therefore, the RUN bit is reset by the interrupt service routine. The soft error interrupt has no effect on the PAM microprocessor and will not halt the PAM.

The interrupt service routines use a common flag word in memory to indicate which interrupt occurred first.  
The Flag Word is written as follows:

Bits 0 to 7 = Interrupt Vector (Written by Service routine)  
Bit 8 = Hard Error Interrupt (Vector 130/140 Ser. rout. sets)  
Bit 9 = Soft Error Interrupt (Vector 134/144 Ser. rout. sets)  
Bit 10 = Double interrupt through vector 130 or 140  
Bit 11 = Double interrupt through vector 134 or 140

#### 2.14 TEST SUMMARIES for CIDSBA PAM Test #2

##### TEST 1

This test will check the path to and from the Dash Bus using Scanner "Maintenance Mode" in "Address Wrap" and the 11/24 Dash Bus Window.

The test sequence is as follows:

1. Set CSR2 Bit 13 ( Data Address Wrap )
2. Set CSR1 Bit 5 ( Maintenance Mode )
3. Read Dash Bus address window

1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395

The data read from the Dash Bus window should consist of the contents of CSR2 Bits 0-3 in the high nibble (Bits 4-7) the current Dash Bus window number should be in the low nibble (Bits 0-3).

#### TEST 2

This test will check the path to and from the Dash Bus using Scanner "Maintenance Mode" in "Data Wrap" and the 11/24 Dash Bus Window.

The test sequence is as follows:

1. Clear CSR2 Bit 13 ( Data Address Wrap )
2. Set CSR1 Bit 5 ( Maintenance Mode )
3. Write the Dash Bus address window
4. Read the Dash Bus address window

The data read from the Dash Bus window should be the same as the data written. Any window location read should fetch the same data, indifferent to the window location written.

#### TEST 3

This test will attempt to force a DASH BUS parity error through the 11/24 DASH BUS Window. The diagnostic will set "Maintenance Mode" and "Force Parity Error" in CSR1 ( Bits 5 and 4 ) and then read an address in the Dash Bus Window. Status is then checked to see that "Dash PE" sets in CSR1 and "11/24 Dash PE" sets in CSR2. The process is again repeated and the error bits are written with a 1 to check that both clear. The final check is to force the error and then set INIT to again check that the error bits clear.

#### TEST 4

This test will check the ability of the PAM to read data from the Dash Bus. This is accomplished by setting the "Address Wrap" bit in CSR2, "Maintenance Mode" in CSR1 and having the PAM microcode do reads to the Dash Bus. The microcode loads the desired line number, using the programmable line register, and reads the desired Dash Bus Register (DBR). The data read should be a combination of the Line Number and the Register number:

BITS 7 TO 4 == Line Number BITS 4 TO 0 == Register Number

Local Storage will look as follows:

7760 == 17 ( last line number used )

If the branch condition fails for a specific line number, the line number in error will be saved in 7760 and one of the error flags will set as described below.



1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449

7761 == 125 ( Register and line number wrapped )  
7762 == 252 ( Register and line number wrapped )  
7763 == 314 ( Register and line number wrapped )  
7764 == 360 ( Register and line number wrapped )

7775 == Dash Bus Parity Error (BPE) Branch was taken  
if bit 0 == 1.  
7776 == Read Not Done OR Dash Bus Parity Error (BDE)  
Branch did not clear if bit 0 == 1.

NOTE!! This is the first test that will check the branch  
conditions "BPE" ( Branch on Dash Parity error ) and  
"BDE" ( Branch on Read Not Done or Dash Parity Error).

#### TEST 5

This test will check the ability of the PAM to do WRITES to  
the Dash Bus. This is accomplished by clearing Data/Adrs wrap  
in CSR2, Setting "Maintenance Mode" in CSR1 and having the PAM  
microcode write data to the DBR's. The microcode loads the the  
line number and writes a Dash Bus Register with a data  
pattern. A different DBR is read to verify that the data  
pattern is the same as was written. The above process ( write  
DBR - read different DBR ) is done with several data patterns  
to verify the integrity of the data path. An attempt is is  
also made to test the STALL feature by doing successive writes,  
with different data patterns, to the dash bus each time a  
write/read cycle is done.

Local Storage will look as follows:

7760 == 360 ( Last Data Pattern Written )

If the branch condition fails for a specific data pattern  
used, the pattern in error will be saved in 7760 and one of  
the error flags will set as described below.

7761 == 125(Data Pattern Read)\*\*If NO Parity Error on read \*\*  
7762 == 252(Data Pattern Read)\*\*If NO Parity Error on read \*\*  
7763 == 314(Data Pattern Read)\*\*If NO Parity Error on read \*\*  
7764 == 360(Data Pattern Read)\*\*If NO Parity Error on read \*\*

7775 == Dash Bus Par Error (BPE) Branch was taken if bit 0=1.  
7776 == Read Not Done OR Dash Bus Parity Error (BDE) Branch  
did not clear if bit 0 == 1.

#### TEST 6

This test will attempt to force an Underrun condition and  
Transmit Error in Scanner "Maintenance Mode". The PAM  
microcode writes data to the Dash Bus that has "Bit 2" set  
(Bit 2 corresponds to XMIT ERR in the line status registers).

1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503

Microcode informs the diagnostic that the data was written and then waits for a response.

The Macrocode will set "Sync", "Transmit Flag" and "Maintenance Mode", then tell the Microcode to proceed. The PAM microcode will, when "Scan Entry" sets in the MSR, read and ,in Local Storage, store the contents of the "Data FIFO ", "Status FIFO" and the "MSR". The Microcode then informs the Macrocode that the function is done.

If the microcode is unable to flush the FIFO's correctly, which indicates that Scanner Entry remains set, the microcode will set a Timeout flag in LS location 7776.

#### TEST 7

This test will attempt to force a "Receive Error" condition in "Synchronous mode", using Maintenance mode.

The PAM microcode writes a data pattern, that the Macrocode has passed to Local Storage, to the Dash Bus and waits for a response from the diagnostic.

The Macrocode will then set "Sync" and "Receive Flag" in CSR2, set "Maintenance Mode" in CSR1 and tell the microcode to continue.

The PAM microcode will, when "Scan Entry" sets in the MSR, read and store (in Local Storage) the contents of the "Data FIFO ", "Status FIFO" and the "MSR". The Microcode then informs the Macrocode that the function was done.

The contents of the "Status FIFO" and "Data FIFO" is dependent on the data pattern written to the "Dash Bus" and whether the "Sync" bit is set in "CSR2". In "Synchronous" mode bits 0 to 3 will cause a "Receive Error" to set in the "Status FIFO". Two entries will be entered in each FIFO for the error condition as follows:

STATUS FIFO	DATA FIFO
1. Error set	Line Reg. 1 ( DATA WRITTEN )
2. No error set	Char. in Err.( DATA WRITTEN )

Bits 4 to 7 should not cause an error condition and the FIFOs' will look as follows:

STATUS FIFO	DATA FIFO
1. No Error set	Received Char.
2. NO Error set	Received Char.

If the microcode is unable to flush the FIFO's correctly, which indicates that Scanner Entry remains set, the microcode will set a Timeout flag in LS location 7776.



1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558

TEST 8

This test will attempt to force a "Receive Error" condition in "Asynchronous mode", using Maintenance mode.

The PAM microcode writes a data pattern, that the Macrocode has passed to Local Storage, to the Dash Bus and waits for a response from the diagnostic.

The Macrocode will then set "Enable Scan Cntr" and "Receive Flag" in CSR2, then set "Maintenance Mode" in CSR1 and again wait for response from the PAM.

The PAM microcode will, when "Scan Entry" sets in the MSR, read and store (in Local Storage) the contents of the "Data FIFO", "Status FIFO" and the "MSR". The Microcode then informs the Macrocode that the function was done.

The contents of the "Status FIFO" and "Data FIFO" is dependent on the data pattern written to the "Dash Bus" and whether the "Sync" bit is set in "CSR2".

In "Asynchronous" mode bits 3 to 5 will cause a "Receive Error" to set in the "Status FIFO". Two entries will be entered in each FIFO for the error condition as follows:

STATUS FIFO	DATA FIFO
1. Error set	Line Reg. 1 ( DATA WRITTEN )
2. No error set	Char. in Err.( DATA WRITTEN )

All bits , other than bits 3 to 5, should not cause an error condition and the FIFOs' will look as follows:

STATUS FIFO	DATA FIFO
1. No Error set	Received Char.
2. No Error set	Received Char.

If the microcode is unable to flush the FIFO's correctly, which indicates that Scanner Entry remains set, the microcode will set a Timeout flag in LS location 7776.

TEST 9

This test will attempt to force a "Receive Error" condition in "Synchronous mode", using Address Wrap.

The Microcode writes a location in Local Storage informing the Macrocode that it is ready to proceed.

The Macrocode will then set "Sync", "Receive Flag" and "Address Wrap" in CSR2 and then set "Maintenance mode" in CSR1. The Diagnostic then informs the PAM that the function was done.

The PAM Microcode will, when "Scan Entry" sets in the MSR,

1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613

read and store (in Local Storage) the contents of the "Data FIFO", "Status FIFO" and the "MSR". The Microcode then informs the Diagnostic that the function is done. This test loops through all the line numbers.

The contents of the "Status FIFO" and "Data FIFO" is dependent on the "Dash Bus" address the "Scanner" is referencing and whether the "Sync" bit is set in "CSR2".

The Scanner reads Line Register "1", when it sees a Receive Flag in Line Register "9" the address currently on the Dash Bus ( Line Register 1 ) will appear as data of Line Register "1". Therefor, an error will be recorded in the status FIFO because "Bit 00" will be set.

In "Synchronous" mode bits 0 to 3 will cause "Error" to set in the "Status FIFO". Two entries will be entered in each FIFO for the error condition as follows:

STATUS FIFO	DATA FIFO
1. Error set	Line Reg. 1 (ADDRESS WRITTEN)
2. No error set	Line Reg. 0 (ADDRESS WRITTEN)

If the microcode is unable to flush the FIFO's correctly, which indicates that Scanner Entry remains set, the microcode will set a Timeout flag in LS location 7776.

#### TEST 10

This test will attempt to force a "Receive Error" condition in "Asynchronous mode", using Address Wrap.

The Microcode writes a location in Local Storage informing the Macrocode that it is ready to proceed.

The Macrocode will then set "Receive Flag" and "Address Wrap" in CSR2 and then sets "Maintenance mode" in CSR1. The Diagnostic then informs the PAM that the function was done.

The PAM Microcode will, when "Scan Entry" sets in the MSR, read and store (in Local Storage) the contents of the "Data FIFO", "Status FIFO" and the "MSR". The Microcode then informs the Macrocode that the function is done. This test will loop through all the line numbers.

The contents of the "Status FIFO" and "Data FIFO" is dependent on the "Dash Bus" address the "Scanner" is referencing and whether the "Sync" bit is set in "CSR2".

The Scanner reads Line Register "1", when it sees a Receive Flag in Line Register "9", and the address currently on the Dash Bus ( Line Register 1 ) will appear as the data of Line Register "1". An error will "NOT" be recorded in the status FIFO for the following Line Numbers: 0,4,8 and 12 all Line



1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667

Numbers, except those listed, will cause an Error bit to set in the Status FIFO.

In "Asynchronous" mode bits 3 to 5 will cause "Error" to set in the "Status FIFO". Two entries will be entered in each FIFO for the error condition as follows:

STATUS FIFO	DATA FIFO
1. Error set	Line Reg. 1 ( ADDRESS WRITTEN )
2. No error set	Char. in Err.( ADDRESS WRITTEN )

If the microcode is unable to flush the FIFO's correctly, which indicates that Scanner Entry remains set, the microcode will set a Timeout flag in LS location 7776.

#### TEST 11

This test will force the scanner to record a Modem Change for all line numbers. The test will start by forcing a Modem Change for all lines, with a known data pattern ( zeros' ). The Status and FIFO entries are ignored for the first data pattern used, since the initial values in the modem change ram is unknown. Subsequent patterns should yield the following: MEC should set in the STATUS FIFO the DATA FIFO should have the EXCLUSIVE "OR" of the previous pattern and the pattern written. The pattern used ( after the pattern of ZEROS ) is incrementing from 1 to 20 (OCT.). This pattern sequence will verify the DEPTH of the Modem Change Ram (16 Decimal locations).

If the microcode is unable to flush the FIFO's correctly, which indicates that Scanner Entry remains set, the microcode will set a Timeout flag in LS location 7776.

#### TEST 12

This test will force the scanner to record a Modem Change for all line numbers. The test will start by forcing a Modem Change for all lines with a known data pattern ( zeros' ). The Status and FIFO entries are ignored for the first data pattern used, since the initial values in the modem change ram is unknown. Subsequent patterns should yield the following: MEC should set in the STATUS FIFO the DATA FIFO should have the EXCLUSIVE "OR" of the previous pattern and the pattern written. Four data patterns are used to verify the data integrity of the ram, as follows:

1. 252 Alternate zeros and ones
2. 125 Above shifted right
3. 063 Adjacent bits set and cleared
4. 017 Adjacent nibbles set and cleared

If the microcode is unable to flush the FIFO's correctly,

1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721

which indicates that Scanner Entry remains set, the microcode will set a Timeout flag in LS location 7776.

#### TEST 13

This test will force a Dash Bus parity error for PAM Writes to the Dash Bus. The Microroutine first flushes the FIFO's and waits for the diagnostic to setup the function. The diagnostic will set Force PE, in CSR1, and informs the microcode that the function was done. When the microcode writes the Dash Bus window, Line in error and Dash Bus Parity error bit, along with the bad data, will load in the FIFOs. The contents of the Data and Status FIFOs is stored in local Storage.

The Microcode will then test two Condition codes while the Force PE is set. A read will indicate the parity error by setting two microbranch condition codes: "Read Not Done or Parity Error" and "Read Dash Bus Parity Error". The state of the condition codes is saved in Local Storage for the Read and Write operations.

The transfer of data to and from Local Storage will cause a "Local Storage Parity error" when Force Parity error is set. Therefore, the existence of this error bit is expected. It should also be noted that the contents of the Status and Data FIFO's is invalid unless Scan Entry is set in the MSR for each entry read.

#### TEST 14

This test will check that the Scanner can be disabled by setting Disable Scan in CSR2 (Bit 6) and starting a XMIT/REC function in "Maintenance Mode". The diagnostic first does a valid Scanner function to assure that known data will appear in the FIFO's. Scanner Disable is then set, the FIFO's flushed and a different type of Scanner function started. There should be no entries into any of the FIFO's from subsequent transfers and this will be verified by the diagnostic.

If the microcode is unable to flush the FIFO's correctly, which indicates that Scanner Entry remains set, the microcode will set a Timeout flag in LS location 7776.

#### TEST 15

This test will check the Block Mover memory address register bits 0 to 21. A pattern is passed to Local Storage for the microcode to read and pass to the Block Mover address register. The microcode then starts a one word DATA-IN to Local Storage with the Block Mover. When the block move stops, the microcode will pass the contents of Last Memory Address register to Local Storage for verification by the program. The following patterns are used as addresses:



1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775

1. 05252525
2. 12525252
3. 14631463

## TEST 16

The block mover is given an address of ALL ONES and a block move is started. The "PAM" should detect an "NXM" in the Micro-Status register and the block mover should stop. The microcode should be "Forced to Zero" when the "NXM" condition occurs and a check is made to see that the force condition occurred once only.

## TEST 17

This is a test of the Block Movers ability to do a DATA-IN from system memory. A data pattern is first written into system memory for the Block Mover to transfer. The Microcode fetches the memory address to write the data, number of words to transfer and Local Storage location to write, from the Pseudo CSR locations. The Block mover should be able to read the data from system memory ( BUFFER ) and write it to contiguous Local Storage locations. The pattern used is incrementing from 1 to 40 (octal).

The microcode is "Forced to Zero" for the following conditions: "NXM" ( Non Existent Memory ), "MPE" ( Memory Parity Errors). A check is made to see that only one traverse through micro-location zero is made (START) by the Microroutine.

## TEST 18

This is a test of the Block Movers ability to do a DATA-OUT to system memory. A data pattern is first written into Local Storage for the Block Mover to transfer. The Microcode fetches the memory address to write the data, number of words to transfer and Local Storage location to read from the Pseudo CSR locations. The Block mover should be able to read the data from Local Storage and write it to a system memory location called BUFFER. The pattern used is incrementing from 1 to 40 (octal). The total transfer should be 16 Words. The microcode is "Forced to Zero" for the following conditions: "NXM" ( Non Existent Memory ), "MPE" ( Memory Parity Errors). A check is made to see that only one traverse through micro-location zero is made (START) by the Microroutine.

## TEST 19

This is a test of the Block Mover Local Storage address register. A data pattern is first written into Local Storage for the Block Mover to READ. The Microcode fetches the memory address to write the data and number of words to transfer from the Pseudo CSR locations. The Block mover should be able to

1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829

read the data from Local Storage and write it to a system memory location called BUFFER.

The following Local Storage locations are used:

LS Address	LS Data
5252	1
5253	2
2525	3
2526	4
1463	5
1464	6
3777	7
4000	10

The microcode is "Forced to Zero" for the following conditions: "NXM" ( Non Existent Memory ), "MPE" (Memory Parity Errors). A check is made to see that only one traverse through micro-location zero is made (START) by the Microroutine.

#### TEST 20

This test will check the Block Movers ability to do a DATA\_IN followed by a DATA\_OUT. The PAM microcode first reads Local Storage to fetch Memory address, Word transfer count and starting function ( DATA\_IN ). The microcode will then start, wait for BM to finish and then write the data read back to system memory. A check is made to see that STATUS, MSR and LAST MEMORY ADDRESS registers are correct. The data in system memory is then checked for correctness.

The microcode is "Forced to Zero" for the following conditions: "NXM" ( Non Existent Memory ), "MPE" (Memory Parity Errors). A check is made to see that only one traverse through micro-location zero is made (START) by the Microroutine.

#### TEST 21

This test will force a Local Storage Parity error and see that the Block Mover will stop when the parity error is detected. Force Parity error is set in CSR1 and a location in Local Storage is written causing that location to have bad parity. When the Block Mover is started and reads the Bad Parity Location, it should stop the transfer. LSPE and PE should set in CSR1 but the microcode should not be Forced to Zero for this error condition. The first pass through the diagnostic POERR is set in CSR1. This will allow the block mover to complete the transfer of data. The second pass will CLEAR POERR and should halt the BLOCK MOVER and MICROCODE when bad parity is read.

The microcode is "Forced to Zero" for the following conditions: "NXM" ( Non Existent Memory ), "MPE" (Memory Parity Errors). A



1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871

check is made to see that only one traverse through micro-location zero is made (START) by the Microroutine.

#### TEST 22

This test will check the Block Movers ability to do a DATA\_IN followed by a DATA\_OUT while the previous Block Move is still in progress. The PAM microcode first reads Local Storage to fetch Memory address, Word transfer count and function for both transfers. The microcode will start the DATA\_IN then immediately start a Data-out with both transfers using the same Local Storage locations. If the Block Mover hasn't finished when another block move is started, a STALL of the microcode takes place until the first block move has finished. The Block Mover should be able to complete both block move operations. A check is made to see that Status, MSR and Last Memory address registers are correct; the data in system memory is then checked for correctness.

The microcode is "Forced to Zero" for the following conditions: "NXM" ( Non Existent Memory ), "MPE" (Memory Parity Errors). A check is made to see that only one traverse through micro-location zero is made (START) by the Microroutine.

#### TEST 23

This test will check that the STEAL IBUS cycle operates correctly from the Fast-bus (Unibus) and PAM sides of the IBUS. A DATA\_OUT block move is started by the Pam Microcode. While the block mover is operating, a series of reads and writes are done to Local Storage from the Pam and Unibus at the same time. Each time the IBUS is requested while the block mover is operating, the requesting operation will Steal an IBUS cycle from the block mover. Both the block move and the function that did the steal should continue to completion without error.

The microcode is "Forced to Zero" for the following conditions: "NXM" ( Non Existent Memory ), "MPE" (Memory Parity Errors). A check is made to see that only one traverse through micro-location zero is made (START) by the Microroutine.

1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924

### 3.0 GENERAL INFORMATION For CIDSCA and CIDSDA Line Card Tests

#### 3.1 PROGRAM ABSTRACT

The line card repair level diagnostic (1) programs is meant to provide field service and manufacturing with a tool to maintain "digital ethernet communication server" digital manufactured line cards. The program will provide the coverage necessary to detect a failure in a line card function. The diagnostic is usually capable of isolating a fault to a particular line card.

Line card types covered are M3100 sync, M3101 high speed sync, and the M3102 dual async line card.

#### 3.2 SYSTEM REQUIREMENTS

In order to run this diagnostic program, the following minimum hardware is required:

- A PDP-11 CPU "PROTOCOL PROCESSOR (PP)" (PDP 11/24)
- MINIMUM OF 256K WORDS OF SYSTEM MEMORY
- CONSOLE BOOT TERMINATOR (CBT)
- RSX11-S "LDI" SOFTWARE OR XXDP. SUPPORTED LOAD MEDIA
- AT LEAST ONE "PAM" MODULE SET CONSISTING OF AN M3110 & M3111
- THE LINE CARD UNDER TEST

#### 3.3 DIAGNOSTIC HIERARCHY PREREQUISITES

The goal of the "PAM" diagnostic program is to test digital manufactured line cards. It is assumed that the "self test diagnostic" has run, and the "CBT", "SYSTEM MEMORY" and "PAM(S)" are fully functional. A failure in the aforementioned devices could fail this diagnostic and the user should be aware of this possibility.

#### 3.4 ASSUMPTIONS - RESTRICTIONS

It is assumed that the prerequisite diagnostics have been executed (refer to section 3.3). The operator should also be familiar with the operating instructions in section 3.5.

#### 3.5 OPERATING INSTRUCTIONS

Refer to section 2.6 for a complete description of the operating instructions.

NOTE: After making one pass of the diagnostic the UNIT flag can be used to test a single unit or more.

STA/PASS:1/FLA:HOE/UNIT:1 !test unit 1 only.

STA/PASS:1/FLA:HOE/UNIT:0-4 !tests units 0-4 only.



1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974

STA/PASS:1/FLA:HOE/UNIT:1:3:6 !tests units 1,3 and 6

### 3.6 HARDWARE QUESTIONS

When a diagnostic is started, the runtime services will prompt the user for hardware information by typing "CHANGE HW (L) ?".

If you answer "NO" the program will run with parameters in the hard coded hardware P-tables.

If you answer "Y" after a start command, the runtime services will ask for the number of units (in decimal).

\*\*\*\*\* WARNING \*\*\*\*\*

[ THE NUMBER OF UNITS MUST ALWAYS BE 16. ]

The line card repair diagnostic will test up to 16 units. However, the diagnostic automatically checks to see if the requested units for test are there and drops any not responding. Also, the "CBT" is checked for a one or two "PAM" system indicator and drops line card unit associated with any PAM not present or not responding. If the PAM configuration does not agree with valid PLUTO configurations or with information in the CBT configuration register an initialization error message is output. An initialization error message is also output if the program has difficulty sizing line cards. Initialization error messages are indicated by error numbers of the form INI XXXXXX.

The hardware P-tables exist to communicate operational parameters for each unit to the diagnostic. These parameters consist of an "LOOPBACK" flag. Loopback indicates that loopback connector(s) are permanently installed on all the line cards that are selected and that external loopback tests may be run without operator intervention. The DRS prompting for P-table parameters includes a indication of the default value which may be used by responding with a <CR>. All remaining P-table questions for any unit may be defaulted by typing a single <CTRL Z>.

The operational parameters are:

LOOP-BACK MODE - Indicates if external loopback connectors are permanently installed on all the selected line cards.

The following P-table dialog alters the default by setting loopback mode for units 0 and 1.

1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028

# UNITS (D) ? 16<CR>

UNIT 0  
INPUT MARGIN 0 CONDITION: (0) 0 ?<CR>  
INPUT MARGIN 1 CONDITION: (0) 252 ?<CR>  
INPUT MARGIN 2 CONDITION: (0) 252 ?<CR>  
INPUT MARGIN 3 CONDITION: (0) 152 ?<CR>  
INPUT MARGIN 4 CONDITION: (0) 125 ?<CR>  
TEST IN AUTO-LOOPBACK MODE ? (L) N ? <CR>

UNIT 1  
INPUT MARGIN 0 CONDITION: (0) 0 ?<CTRL Z>  
UNIT 2

.

UNIT 15  
INPUT MARGIN 0 CONDITION: (0) 0 ?<CTRL Z>

3.7 ERROR INFORMATION

There are three levels of error messages that may be issued by a diagnostic: general, basic and extended. General error messages are always printed unless the "IER" flag is set (section 2.7). The general error message is of the form:

```

NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:XXXX
ERROR MESSAGE
WHERE
NAME = DIAGNOSTIC NAME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL
    
```

Basic error messages are messages that contain some additional information about the error. These are always printed unless the "IER" or "IBE" flags are set (section 2.7). These messages are printed after the associated general message. Extended error messages contain supplementary error information such as register contents or good/bad data. These are always printed unless the "IER", "IBE" or "IXE" flags are set (section 2.7). These messages are printed after the associated general error message and any associated basic error messages. This diagnostic does not use any extended error messages.

Initialization error messages are of the format :

```

NAME INI NUMBER MESSAGE
    
```

These are always printed and occur because of configuration errors found in the diagnostic initialization, problems sizing



2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076

line cards or operational parameters which should not be used with this specific diagnostic. After the error is output, the diagnostic is aborted.

A warning is output when the diagnostic is run and no standard line card is found. The diagnostic is then aborted.

### 3.8 CONFIGURATION INFORMATION

The Pluto system configuration presumes that 1 or 2 PAMS are attached to the PDP-11/24 protocol processor and that each PAM has 8 dash bus slots. The PAM UNIBUS addresses for PAM0 and PAM1 are known. PAM sizing is done via accessing a PAM0 and PAM1 register. If a Timeout interrupt results then it is assumed that either the PAM is not present or it is incapable of responding. The number of PAMS which should reside in a system is determined by reading the display/configuration register in the console-boot-terminator module. If one PAM exists in a system it should be PAM0.

Line cards of any type(s) may be arbitrarily inserted into the dash bus slots subject to system constraints. The dash bus is sized to determine what type of line cards, if any, are attached to each dash bus slot.

Default hardware P-tables are set up to run diagnostics on all line cards in both PAMS. Automatic sizing determines the appropriate line card tests to be run. Empty dash bus slots are skipped, i.e. no tests are run. User and undefined line cards are not tested. If line card types are such that no tests can be run from this or companion line card diagnostics, a warning message will be displayed.

Errors or system configuration violations, if found by the diagnostic initialization, will result in initialization error messages and will cause an abort.

The auto-loopback question is asked for each unit. This informs DRS that a loopback connector (2 connectors for dual line cards) is permanently installed on that unit. All applicable tests, including external loopback tests, are executed.

If the auto-loopback question was answered NO and DRS is run in UNATTENDED MODE, no external loopback tests are run. If NOT in UNATTENDED MODE, the operator is prompted to install a loopback connector on the appropriate line card port/unit.

2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132

3.9 TEST SUMMARIES For CIDSCA Line Card Test #1

T	M	M	M	2	2	I	E
E	3	3	3	6	6	N	X
S	1	1	1	6	5	T	T
T	0	0	0	1	2	.	.
N	0	1	2	.	.	L	L
O	.	.	.	A	.	O	O
.	.	.	.	S	.	O	O
.	.	.	.	Y	.	P	P
.	.	.	.	N	.	.	.
.	.	.	.	C	.	.	.
.	.	.	.	.	.	.	.

TEST 1  
Line card INIT, led and dash bus dual addressing test. X X X - - - -

This test verifies that the led bit can be set via a line card INIT or by writing a 1 to the led bit, and that the bit can be cleared. this test also checks for dash bus dual addressing.

TEST 2  
Line card generate bad parity check. X X X - - - -

This test bit bangs the line parameter reg for the purpose of determining if the generate bad parity bit has any stuck at or short type faults.

TEST 3  
Line card reg clear on INIT test. X - X - - - -

This test inits the line card and then checks if the 2661 registers which should be cleared on INIT are in fact cleared.

TEST 4  
Line card register initialization test. X X - - - - -

This test inits the line card and then checks if the 2652 registers which should be cleared on INIT are in fact cleared.

TEST 5  
Line card 2661 register dual addressing test. - - X - - - -

This test inits the line card and checks the 2661 registers (reg space 0 - 7) and generic registers (reg space 8 - 15) for dual addressing.

TEST 6  
Line card 2652 register dual addressing test. X X - - - - -

This test inits the line card and checks the 2652 registers (reg space 0 - 7) and generic registers (reg space 8 - 15) for dual addressing.



2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183

T		M	M	M	2	2	I	E
E		3	3	3	6	6	N	X
S		1	1	1	6	5	T	T
T		0	0	0	1	2	.	.
	N	0	1	2	.	.	L	L
	O	.	.	.	A	.	O	O
	.	.	.	.	S	.	O	O
	.	.	.	.	Y	.	P	P
	.	.	.	.	N	.	.	.
	.	.	.	.	C	.	.	.
	.	.	.	.	.	.	.	.

TEST 7  
Line card 2661/2652 register interference test.  
(THIS TEST IS SKIPPED BECAUSE 2661 MODE WAS REMOVED FROM THE M3100.)  
This test checks for interference between registers of the 2661 and 2652 protocol chips.

TEST 8  
Line card 2661/2661 register interference test.       - - X - - - -  
  
This test checks for interference between registers of the the 2 2661 protocol chips on the line card.

TEST 9  
Bit bang 2661 and generic registers.                       - - X - - - -  
  
This test bit bangs the line card generic registers (reg addr 8 - 15) and the 2661 registers (reg addresses 0 - 7). Also checks scanner retry on mode registers with forced par err.

TEST 10  
Bit bang line card 2652 and generic registers.               X X - - - - -  
  
This test bit bangs the line card generic registers (reg addr 8 - 15) and the 2652 registers (reg addresses 0 - 7).

TEST 11  
Modem in register external loopback test.                   X X X - - - X  
  
This test bit bangs the modem in register via the modem output register and an external loopback connector.

TEST 12  
2652 select and 2661 xmitter ready test.                   - - X - X - -  
  
This test verifies that 2661 and 2652 mode can be selected if applicable. Functioning of 2661 xmit buff avail <TXBAV>, xmitter empty <TXEMT> and xmitter ready <TXRDY> bits is verified.

2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234

T  
E  
S  
T  
  
N  
O  
  
.  
.  
.

M	M	M	2	2	I	E
3	3	3	6	6	N	X
1	1	1	6	5	T	T
0	0	0	1	2	.	.
0	1	2	.	.	L	L
.	.	.	A	.	O	O
.	.	.	S	.	O	O
.	.	.	Y	.	P	P
.	.	.	N	.	.	.
.	.	.	C	.	.	.
.	.	.	.	.	.	.

TEST 13  
2652 xmitter ready test.

This test checks the functioning of the 2652 transmitter related bits <TXBAV>, <TXEN2> and <TSOM>. No data is actually looped. Checks are made in both BOP and BCP modes.

TEST 14  
2661 receiver check.

This test checks the functioning of the 2661 RCV data avail <RCVDAV> and RCVR enable <RXEN1> bits. The test is performed in async mode. Verifies operation of line par <SYNC XMIT ERR>.

TEST 15  
2652 receiver check.

This test checks the functioning of the 2652 RCV data avail <RXDAV> and RCVR enable <RXEN2> bits.

TEST 16  
2661 all character length data xfer test.

Loop data pattern through 2661 (async mode) for 5, 6, 7 and 8 bit characters at 19.2 kbaud.

TEST 17  
M3101 transmit buffer ram address sequence test.

Insure that transmitter buffer ram address pointer is being autoincremented, following each write to the ram control byte, when 'load ram' is set in transmit buffer control register.

TEST 18  
M3101 transmit ram data test.

Verify that all transmit data, and command byte ram bytes are free of stuck bits.

X	X	-	-	X	-	-
-	-	X	-	X	-	X
X	X	-	-	X	X	-
-	-	X	X	-	X	-
-	X	-	-	-	-	-
-	X	-	-	-	-	-



2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279  
2280  
2281  
2282  
2283  
2284  
2285

T  
E  
S  
T  
  
N  
O  
  
.  
.  
.

M	M	M	2	2	I	E
3	3	3	6	6	N	X
1	1	1	6	5	T	T
0	0	0	1	2	.	.
0	1	2	.	.	L	L
.	.	.	A	.	O	O
.	.	.	S	.	O	O
.	.	.	Y	.	P	P
.	.	.	N	.	.	.
.	.	.	C	.	.	.
.	.	.	.	.	.	.

TEST 19  
M3101 autoloading of transmit control ram test.

- X - - - -

Verify that with 'load ram' set in transmit buffer control, that a data byte written to the multi memory register will be placed into the data portion of the transmit ram, and at the same time, causes a default value to be placed into the same ram address, then autoincrements the ram address pointer.

TEST 20  
M3101 low speed transmit ram data transfer test.

- X - - - X -

This test, operating in the maintenance mode at 19.2 kbaud, will verify operation of the transmit ram, in the buffered mode at 19.2 kbaud

TEST 21  
M3101 transmit buffer ram address overflow test.

- X - - - X -

Operating in the maintenance mode at 19.2 kbaud this test will verify that an overflow of the ram address buffer will set 'end of buffer' and 'transmit buffer avail'.

TEST 22  
M3101 buffered mode transmitter underrun test.

- X - - - X -

Operating in the maintenance mode at 19.2 kbaud this test will verify that while a data transmission from the transmit buffer ram is taking place, clearing 'send ram' will cause a transmitter underrun.

TEST 23  
M3101 high speed bop internal loopback test.  
(UTILIZES DIAGNOSTIC MICROCODE)

- X - - - X -

Operating in maintenance mode at 500 kbaud, internal loopback, bop mode, with the line card in the buffered mode, will verify that data can be successfully transferred.

2287	T		M	M	2	2	I	E
2288	E		3	3	6	6	N	X
2289	S		1	1	6	5	T	T
2290	T		0	0	1	2	.	.
2291			0	1	2	.	.	L
2292	N		.	.	.	A	.	O
2293	O		.	.	.	S	.	O
2294			.	.	.	Y	.	P
2295	.		.	.	.	N	.	.
2296	.		.	.	.	C	.	.
2297	.		.	.	.	.	.	.
2298	TEST 24		.	.	.	.	.	.
2299	M3101 high speed bop external loopback test.		-	X	-	-	-	X
2300	(UTILIZES DIAGNOSTIC MICROCODE)							
2301								
2302	SAME AS #24, EXCEPT IN EXTERNAL LOOPBACK							
2303								
2304	TEST 25							
2305	M3101 high speed BCP internal loopback test.		-	X	-	-	-	X
2306	(UTILIZES DIAGNOSTIC MICROCODE)							
2307								
2308	Operating in maintenance mode at 500 kbaud, internal							
2309	loopback, BCP mode, with line card in the buffered							
2310	mode, will verify that data can be successfully							
2311	transferred.							
2312								
2313	TEST 26							
2314	M3101 high speed, BOP mode, force XMT BUFF RAM		-	X	-	-	-	X
2315	parity error. (UTILIZES DIAGNOSTIC MICROCODE)							
2316								
2317	This test will verify that on detection of a transmit							
2318	buffer ram parity error, during a data transfer attempt,							
2319	'transmitter error' bit will set.							



2321  
2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339  
2340  
2341  
2342  
2343  
2344  
2345  
2346  
2347  
2348  
2349  
2350  
2351  
2352  
2353  
2354  
2355  
2356  
2357  
2358  
2359  
2360  
2361  
2362  
2363  
2364  
2365  
2366  
2367  
2368  
2369

3.9 TEST SUMMARIES For CIDS DA Line Card Test #2

T	M	M	M	2	2	I	E
E	3	3	3	6	6	N	X
S	1	1	1	6	5	T	T
T	0	0	0	1	2	.	.
	0	1	2	.	.	L	L
N	.	.	.	A	.	O	O
O	.	.	.	S	.	O	O
.	.	.	.	Y	.	P	P
.	.	.	.	N	.	.	.
.	.	.	.	C	.	.	.

TEST 1 All baud rates data xfer test. - - X X - X -

Loop and check data data pattern through 2661 (async mode) via internal at all baud rates from 50mbaud to 19.2 kbaud. Baud rate accuracy is not checked.

TEST 2 2661 All stop bit length data xfer test. - - X X - X -

Loop data pattern via 2661 async mode with 1, 1.5 and 2 stop bits. Check data and relative timing to verify that the correct number of stop bits are being used.

TEST 3 2652 Sync generation test. X X - - X - X

Check the ability of the 2652 to generate syn characters from the syn register and xmit holding register. Also check the ability to strip the 1st 2 syn characters and the ability to discriminate against non-syn characters.

TEST 4 2652 Transmitter flag generation test. X X - - X X -

Check the ability of the 2661 (bop mode) to generate and strip flags characters. transmit data where data = flag. Data integrity verifies 0 stuffing.

TEST 5 2652 BOP mode 2ndary addr RSOM and REOM test. X X - - X - X

Loop data in 2652 bop internal loopback mode with secondary address recognition enabled. Data integrity, RCVR errors and the ability of REOM to set and clear is checked.

TEST 6 2652 BCP mode internal data wrap test. X X - - X X -

Loop a data pattern in 2652 internal loopback mode and verify the data integrity. Test is performed at 19.2 kbaud and for 5 thru 8 bit character lengths.

2371  
2372  
2373  
2374  
2375  
2376  
2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385  
2386  
2387  
2388  
2389  
2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422

T	M	M	M	2	2	I	E
E	3	3	3	6	6	N	X
S	1	1	1	6	5	T	T
T	0	0	0	1	2	.	.
	0	1	2	.	.	L	L
N	.	.	.	A	.	O	O
O	.	.	.	S	.	O	O
.	.	.	.	Y	.	P	P
.	.	.	.	N	.	.	.
.	.	.	.	C	.	.	.
.	.	.	.	.	.	.	.
TEST 7 2652 BOP mode data wrap/ bit stuff test.	X	X	-	-	X	X	-

Loop a data pattern in 2652 bop mode at 19.2 kbaud for 4 thru 8 bits/char, and at 4.8 kbaud for 2-3 bits/character. The data pattern exercises the 2652 bit stuffing feature.

TEST 8 2652 0/1/2 Starting syn test.	X	X	-	-	X	X	-
--------------------------------------	---	---	---	---	---	---	---

Attempt to loop data in 2652 internal loopback bop mode with 0, 1 and 2 starting syn characters. The receiver should sync up only with 2 leading syns.

TEST 9 2652 Mult start syns w/wo strip sync.	X	X	-	-	X	X	-
--	---	---	---	---	---	---	---

Loop a data pattern with multiple starting and embedded syns. with strip syn disabled verify that 2 starting syns are stripped. With strip syn enabled, verify that all starting syns are stripped.

TEST 10 2652 Multiple syn character test.	X	X	-	-	X	X	-
---	---	---	---	---	---	---	---

Data patterns are looped using different syn characters to find stuck bits or lines in the syn related circuitry.

TEST 11 2652 Syn character discrimination test.	X	X	-	-	X	X	-
---	---	---	---	---	---	---	---

An attempt is made to loop data using xmitted syn characters differing from syn characters in the low byte par reg by 1 bit. If the RCVR syncs up an error is indicated. Correct syn chars are also xmitted to verify that the RCVR can sync up.

TEST 12 2652 Secondary address mode test.	X	X	-	-	X	X	-
---	---	---	---	---	---	---	---

This test checks the ability of the 2652 2ndary address mode bit to put the 2652 into 2ndary address mode.

TEST 13 Right/wrong secondary address test.	X	X	-	-	X	X	-
---	---	---	---	---	---	---	---

Attempt to loop data patterns 2ndary addresses which are incorrect by 1 bit. No data xfer should occur. Correct 2ndary addresses are also used to verify that data xfers can occur.



2424  
2425  
2426  
2427  
2428  
2429  
2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465  
2466  
2467  
2468  
2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478

T		M	M	M	2	2	I	E
E		3	3	3	6	6	N	X
S		1	1	1	6	5	T	T
T		0	0	0	1	2	.	.
		0	1	2	.	.	L	L
N		.	.	.	A	.	O	O
O		.	.	.	S	.	O	O
.		.	.	.	Y	.	P	P
.		.	.	.	N	.	.	.
.		.	.	.	C	.	.	.
.		.	.	.	.	.	.	.
TEST 14	2652 All parties addressed enable test.	X	X	-	-	X	X	-

This test checks the ability of the 2652 to reject an all parties message when not in the all parties addressed mode.

TEST 15	2652 All parties addressed detection discrimination test.	X	X	-	-	X	X	-
---------	---	---	---	---	---	---	---	---

With all parties addressed (apa) set, attempt to loop data with 2ndary addresses differing from all parties address (377) by 1 bit. Data should not be received. The correct all parties address (377) is also used to verify that data can be received in this mode. A correct 2nd addr is sent to verify reception in apa mode.

TEST 16	2652 Abort detection/generation test.	X	X	-	-	X	X	-
---------	---------------------------------------	---	---	---	---	---	---	---

While looping data a check is made that setting TEOM sends abort if idle = 0 and sends a flag if idle = 1. Abort reception should cause RAB and REOM to set. A check is made that these bits properly clear. Flag reception should allow receipt of the character before the flag.

TEST 17	2652 Go-ahead gen/detect, abort with go-ahead test.	X	X	-	-	X	X	-
---------	---	---	---	---	---	---	---	---

Check the functionality of the 2652 go-ahead generation and detect features incl the <RAB/GA>, <TEOM> and <TGA> bits.

TEST 18	2661 Async forced break test.	-	-	X	X	-	X	-
---------	-------------------------------	---	---	---	---	---	---	---

Verify that the 2661 command register break bit is functional.

TEST 19	2661 Async mode parity error test (no error).	-	-	X	X	-	-	X
---------	---	---	---	---	---	---	---	---

Verify 2661 async mode data can be looped with odd and even parity checking enabled without a parity error occurring.

TEST 20	2661 Async mode odd/even parity gen/det test.	-	-	X	X	-	X	-
---------	---	---	---	---	---	---	---	---

Loop data with odd and even parity checking enabled. Parity errors are forced and verified via the parity error bit. With with parity disabled, the parity err bit should not set.

2480  
2481  
2482  
2483  
2484  
2485  
2486  
2487  
2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495  
2496  
2497  
2498  
2499  
2500  
2501  
2502  
2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510  
2511  
2512  
2513  
2514  
2515  
2516  
2517  
2518

T  
E  
S  
T

N  
O

TEST 21 2661 Async overrun test.

Generate an overrun while looping data in 2661 async mode and verify that the overrun bit sets.

TEST 22 2661 Async mode framing error test.

Check the ability to detect a framing error. Framing errors are generated by looping data at different xmit and RCV clock rates.

TEST 23 2652 Error control modes test.

This test loops data in each of the 2652 error control modes. Errors are generated and error detection is verified. A check is also made to verify that error detection can be disabled.

TEST 24 2652 Underrun test.

The 2652 response to an underrun is checked in both BOP and BCP mode with the IDLE bit both set and clear.

TEST 25 2652 Overrun test.

Generate an overrun in both BOP and BCP mode. Verify that the RCVR status reg overrun bit is set, and that it can be cleared via a RCVR status reg read, a reset error command, or by disabling the receiver.

```
MMH 22IEE
333 66NXX
111 65TTT
000 12.LL
012 .LLOO
. . . A.OOP
. . . S.OOP
. . . Y.PP
. . . N.
. . . C.
. . .
. . .
. . .
```

- - X X - X -

X X - - X X -

X X - - X X -

X X - - X X -



2520  
2521  
2522  
2523  
2524  
2525  
2526  
2527  
2528  
2529  
2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539  
2540  
2541  
2542  
2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568

#### 4.0 GENERAL INFORMATION For CIDSEA CBT Test

##### 4.1 PROGRAM ABSTRACT

This program is a repair level diagnostic for the M3112 CBT (Console,Boot,Terminator) module.

The CBT (M3112) is A standard Hex module with Unibus SPC pinout that contains:

1. ROM Bootstrap supporting 8 ROM sockets (64kb total).
2. Off/On/Lock/Standby key switch
3. Start and Test pushbuttons
4. Four seven segment leds to identify operator action (e.g. replace bad unit)
5. Serial Line Unit fixed at 1200 baud, for a virtual console.
6. EIA console serial line connector, for local control of the 11/24.
7. Unibus terminator for the end of the Unibus.

This diagnostic has been written for use with the diagnostic runtime services software (DR>). These services provide the interface to the operator and to the software environment.

##### 4.2 SYSTEM REQUIREMENTS

The Minimum system required is:

1. 11/24 Processor with its SLU1 set to 1200 baud
2. 28Kw of Unibus memory
3. CBT (Console,Boot,Terminator) module

##### 4.4 PREREQUISITES

The 11/24 option diagnostic (CJDFA) or equivalent must be run to insure a working SLU1.

##### 4.5 ASSUMPTIONS

The SLU1 in the 11/24 Processor must be functional.

2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593  
2594  
2595  
2596  
2597  
2598  
2599  
2600  
2601  
2602  
2603  
2604  
2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620

#### 4.6 OPERATING INSTRUCTIONS

Refer to section 2.6 for a complete description of the operating instructions.

The following is a sample CBT diagnostic run:

Start the Diagnostic under DRS

DR> STA/FLA:PNT:HOE/PAS:1

CHANGE HW (L) ? N

TST 001: READ/WRITE REGISTER TEST  
 TST 002: PCR REGISTER TEST  
 TST 003: MAINTENANCE REGISTER TEST  
 TST 004: RECEIVER CSR REGISTER TEST  
 TST 005: DLART INTERNAL LOOPBACK TEST (10 SECS)  
 TST 006: CBT TO 11/24 SLU1 TEST (20 SECS)  
 TST 007: UNIBUS REGISTER ADDRESS DECODE TEST  
 TST 008: ROM CRC-16 CHECKWORD TEST  
 TST 009: SEVEN SEGMENT DISPLAY REGISTER TEST  
 TST 010: SINGLE LEDS DISPLAY TEST  
 TST 011: CONFIGURATION REGISTER PRINTOUT TEST  
 CONFIGURATION REGISTER CONTENTS  
 BIT<8>=1 JUMPER W4 IS NOT INSTALLED  
 BIT<7>=1 JUMPER W1 IS NOT INSTALLED  
 BIT<6>=1 JUMPER W2 IS NOT INSTALLED  
 BIT<5>=1 JUMPER W3 IS NOT INSTALLED  
 BIT<4>=1 BATTERY BACKUP IS NOT PRESENT  
 BIT<3>=0 MARGINING BOX IS NOT PRESENT  
 BIT<2>=0 UNUSED  
 BIT<1>=1 TEST PUSHBUTTON IS OFF  
 BIT<0>=1 ONE PAM SET IS PRESENT  
 TST 012: ROM CONFIGURATION PRINTOUT TEST  
 ROM 0 - PART NUMBER IN ROM = 23-abcde-fg  
 SLOT NUMBER IN ROM = 0  
 SIZE IN ROM = 1KB  
 CRC CALCULATED = 000000

( THIS IS REPEATED FOR ALL EIGHT ROMS )

DR>

Tests 11 and 12 will be skipped when the UAM flag is set.

Example: DR> START/FLA:PNT:UAM

#### 4.7 HARDWARE QUESTIONS

When a diagnostic is started, the runtime services will prompt



2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669

the user for hardware information by printing:

CHANGE HW (L) ?

You must enter Y after a START command, unless the information has been preloaded via the setup utility. See the XXDP+ manual for more information on the setup utility.

The DRS will then ask for the number of units to test. For this diagnostic always answer 1.

For Example:  
CHANGE HW (L) ? Y  
# UNITS (D) ? 1

#### 4.8 SOFTWARE QUESTIONS

This diagnostic does NOT ask any software questions.

#### 4.9 ERROR MESSAGE FORMATS

The error messages are in the following format:

M3112 HRD ERR 00514 ON UNIT 00 TST 005 SUB 007 PC:12762  
CBT Data error in loopback mode

EXPD: 000005 RECV: 000004 XOR: 000001

Where:

1. "M3112" is the CBT module name
2. "HRD ERR" indicates a non-recoverable (hard) error. All CBT errors are considered hard errors, or fatal (FTL ERR) errors.
3. "00514" is the test and error number. This example is test 5 error number 14.
4. "ON UNIT 00" is Fixed. The CBT consists of only one unit per processor.
5. "TST 005 SUB 007" indicates test 5 subtest 7 was executing at error call.
6. "PC:12762" is the virtual pc at the error. The program may actually be executing at a different physical PC if it is running under a monitor other than XXDP+.

2671  
2672  
2673  
2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694  
2695  
2696  
2697  
2698  
2699  
2700  
2701  
2702  
2703  
2704  
2705  
2706  
2707  
2708  
2709  
2710  
2711  
2712  
2713  
2714  
2715  
2716  
2717  
2718  
2719  
2720

7. "EXPD:" is the expected data.
8. "RECV:" is the received data.
9. "XOR: " is the bits that are different between the EXPD and RECV data.

#### 4.10 TEST SUMMARIES for CIDSEA

##### TEST 1: READ/WRITE REGISTER TEST

This test verifies the READ/WRITE register is addressable from the 11/24 and has no bits shorted together or stuck at a high or low level.

##### TEST 2: PAGE CONTROL REGISTER (PCR) TEST

This test verifies the PCR is addressable from the 11/24 and has no bits shorted together or stuck at a high or low level.

##### TEST 3: MAINTENANCE REGISTER TEST

This test verifies the MAINTENANCE register is addressable from the 11/24 and has no bits shorted together or stuck at a high or low level.

##### TEST 4: CBT RECEIVER CSR REGISTER TEST

This test verifies the DLART RECEIVER CSR register is addressable from the 11/24 and has no bits shorted together or stuck at a high or low level.

##### TEST 5: CBT DLART INTERNAL LOOPBACK TEST

This test verifies that data can be transmitted to the CBT serial line unit (DLART) and received in loop back mode. In addition the DLART status bits are checked.

##### TEST 6: CBT TO 11/24 SLU1 TEST

This test verifies data can be transmitted between the CBT serial line unit (DLART) and the 11/24 serial line unit (SLU1).

##### TEST 7: UNIBUS REGISTER ADDRESS DECODE TEST

This test verifies that the CBT register address decode logic is functioning correctly so that each register will only respond to their valid Unibus addresses.



2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769

## TEST 8: ROM CRC-16 CHECKWORD TEST

This test verifies each ROM CRC-16 CHECKWORD (checksum) is correct.

The test first searches all eight ROM slots for roms. A slot is assumed empty if -1 is read back from the first and last locations.

Each ROM is read and a CRC-16 CHECKWORD is calculated. It is verified the result of the CRC calculation including the CHECKWORD blasted into the ROM is zero.

## TEST 9: SEVEN SEGMENT LEDS DISPLAY TEST

This test will verify the display register will cause a Unibus Timeout if written into with a byte instruction. The Seven segment led display is also tested.

Here is how the digits are formed in the seven segment display test:

```

  -- : : -- : : -- : : -- : : -- : :
  : : : : : : : : : : : : : :
  -- : : -- : : -- : : -- : : -- : :

```

Each segment and decimal points are lit individually in sequence.

## TEST 10: SINGLE LEDS DISPLAY TEST

This test will light the single leds in a fixed sequence. The CBT contains Four single leds (with room reserved for two more) arranged in a row. They will be used to indicate status such as cable faults or line faults. There are two other leds located in the TEST and START switches.

The test will light the leds in the following sequences:

1. All Leds
2. No Leds
3. Light each Led in turn from left to right
4. Turn off each led in turn from left to right

This is not a Manual Intervention test but will require an operator to determine if the display is correct.

2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786

TEST 11: CONFIGURATION PRINTOUT TEST

This is a Manual Intervention test to printout the contents of the Configuration Register. The Configuration Register is a Read-only register that indicates which options are present, such as hardware jumpers on the CBT module.

TEST 12: ROM CONFIGURATION PRINTOUT TEST

This is a Manual Intervention test to printout the configuration of the Roms currently installed in the CBT. The test sizes automatically for the roms and calculates the CRC-16 on each ROM.

The calculated CRC-16 is always required to be zero, since it includes the CHECKWORD blasted into the ROM.



2788  
2789  
2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799  
2800  
2801  
2802  
2803  
2804  
2805  
2806  
2807  
2808  
2809  
2810  
2811  
2812  
2813  
2814  
2815  
2816  
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825  
2826  
2827  
2828  
2829  
2830  
2831  
2832  
2833  
2834  
2835  
2836  
2837

### 5.0 GENERAL INFORMATION for SYSEXE

The system exerciser (SYSEXE) is a RSX11-S task which is part of the DECSA Loadable Diagnostic Image (LDI). Its purpose is to create as much activity between the PDP-11/24 and PAM/LINE units as possible. This is accomplished by transmitting and receiving 576 byte data messages on all available lines. These lines may be M3100/M3101 SYNC or M3102 ASYNC and the DEUNA. If mixed line cards are sized by SYSEXE, the majority line type will be exercised.

Activity is built up gradually by phases.

PHASE 0 - START PAM 0 AND LOOP DATA MESSAGES ON EACH LINE PRESENT.

PHASE 1 - START PAM 1 WHILE KEEPING PAM 0 GOING. LOOP DATA MESSAGES ON EACH LINE PRESENT.

PHASE 2 - KEEPING BOTH PAMS GOING START THE UNA LOOPING DATA MESSAGES.

### 5.1 OPERATING INSTRUCTIONS

To execute SYSEXE at the PLU> prompt type "RUN SYSEXE" it will then prompt you for number of passes and if you want to run with loopbacks connected.

### 5.2 LINE AND SLOT IDENTIFICATION UNDER SYSEXE

LINE		LINE	
----		----	
slot		slot	
1	) 0 8 )	2	) 1 9 )
3	) 2 10 )	4	) 3 11 )
5	) 4 12 )	6	) 5 13 )
7	) 6 14 )	8	) 7 15 )
	) )		) )
9	) 0 8 )	10	) 1 9 )
11	) 2 10 )	12	) 3 11 )
13	) 4 12 )	14	) 5 13 )
15	) 6 14 )	16	) 7 15 )

2839  
2840  
2841  
2842  
2843  
2844  
2845  
2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857  
2858  
2859  
2860  
2861  
2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891

#### 6.0 UPDATING CSVLDI.SYS (LDI BL06)

These instructions are intentionally general due to the large number of possible load device names on VMS and RSX11m+ systems. See the system manager for the specific device name of RL02 or magtape on the target system. It is also recommended that you verify the location of the disk:[targetuic](this area was created when the operational DECSA software package was installed) for the LDI with the system manager.

The distribution media is in Files11 format. Label name is CZLDIA. After copying, verify CSVLDI.sys size is 1002 blocks.

VMS installation do:

For RL02 do:

```
$ mount rl02:czldia
$ copy/contiguous
$_From: rl02:[ldi]csvldi.sys
$_To: sys$system:csvldi.sys
```

End RL02.

For Tape do:

```
$ mount tape:czldia
$ copy/contiguous
$_From: tape:csvldi.sys
$_To: sys$system:csvldi.sys
```

End Tape.

End VMS installation.

RSX11m+ installation do:

For RL02 do (DCL assumed):

```
> mount rl02:czldia
> copy/contiguous
From? rl02:[ldi]csvldi.sys
To? disk:[targetuic]csvldi.sys
```

End RL02.

For Tape do (DCL assumed):

```
> mount tape:czldia
> copy/contiguous
From? tape:csvldi.sys
To? disk:[targetuic]csvldi.sys
```

End Tape.

End RSX11.



2893  
2894  
2895  
2896  
2897  
2898  
2899  
2900  
2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910

000001

.END

## 7.0 KNOWN PROBLEMS WITH LDI BL06

This is the current list of problems with LDI BL06. It is assumed that these are software problems and should be fixed in BL07.

- o SYSEXE errors when started the second time from PLU >. The LDI must be reloaded.

- o Control C not handled by PLUMON.

Refer to sections 2.7 and 2.8 "NOTES" for help on this problem.

- o M3101 in slot 1 causing M3100s in other slots to error while running SYSEXE.

\*

. ABS. 000000 000  
000000 001  
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 19 WORDS ( 1 PAGES)  
DYNAMIC MEMORY: 20324 WORDS ( 78 PAGES)  
ELAPSED TIME: 00:01:16  
.CZLDIA.SEQ/-SP=CZLDIA.MEM