

DV11

DV11 ROM TST PRT 2
CZDVDDO

AH-8741D-MC
FICHE 1 OF 1

NOV 1980
COPYRIGHT © 75-80
MADE IN USA



IDENTIFICATION

PRODUCT CODE: AC-8740D-MC
PRODUCT NAME: C?DVDDO DV11 ROM TST PRT2
PRODUCT DATE: JUNE 1980
MAINTAINER: 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) 1975,1980 DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1. ABSTRACT

The function of the DV11 diagnostics are to verify that the option operates according to specifications. The diagnostics verify that there are no malfunctions and the all operations of the DV11 are correct in its environment.

Parameters may be set to alert diagnostics as to the DV11 configuration BY USING THE "TRIAL" PROGRAM (CZDVE SA:210). ALL QUESTIONS SHOULD BE answered and then each diagnostic will "OVERLAY" these parameters which are stored in the "STATUS TABLE" (see section 8.4a). The alternative to "TRIAL" program is "AUTO SIZING" (see section 8.5).

CZDVD LIKE CZDVC ALLOWS THE MICRO PROCESSOR TO "FREE RUN". BECAUSE OF the length of the "free running" test; this test is to catch the OVERFLOW FROM CZDVC.

Currently there are six off line diagnostics that are to be run in sequence to insure that if an error should occur it will be detected at an early stage and insuring that diagnosis of error will be immediate to problem

NOTE: Additional diagnostics may be added in the future.

The six diagnostics are:

1. CZDVA [REV] BASIC R/W TEST AND ROM INSTRUCTION EXERCISER.
2. CZDVB [REV] STATIC LINE CARD TESTS.
3. CZDVC [REV] 'FREE RUNNING' ROM TESTS PART 1.
4. CZDVD [REV] 'FREE RUNNING' ROM TESTS PART 2.
5. CZDVE [REV] MODEM CONTROL AND CABLE TESTS PLUS MANUAL PARAMETER input. [TRIAL PROGRAM]
6. CZDVF [REV] ASYNCHRONOUS LINE CARD TESTS.

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (WITH MINIMUM 8K MEMORY)
 ASR 33 (or equivalent)
 DV11-AA MUX CNTRL UNIT
 AT LEAST ONE OF THE FOLLOWING
 DV11-BA 8 LINE SYNC MODULES
 DV11-BB 8 LINE ASYNC MODULES
 DV11-BC 4 SYNC LINES, 4 ASYNC LINES

2.2 STORAGE

PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABL AND BOOTSTRAP LOADER RESIDE. LOCATION 1500 THRU 1736 ARE ESPECIALLY TO BE NOTED AND TO BE untouched by operator after DV11 trial program has been executed; or after the 'AUTO SIZING' has been done.

3. LOADING PROCEEDURE

3.1 METHOD

All programs are in absolute format and are loaded using the ABSOLUTE LOADER. NOTE: if the diagnostics are on a media such as DISK, MAGTAPE, DECTAPE, or CASSETTE; follow instructions for the monitor which has been provided on that specific media.

ABSOLUTE LOADER starting address *500

MEMORY * SIZE

4k	17
8k	37
12k	57
16k	77
20k	117
24k	137
28k	157

3.1.1 Place address of ABS loader into switch register.
(also place 'HALT' SW up)

3.1.2 Depress 'LOAD ADDRESS' key on console and release.

3.1.3 Depress 'START KEY' on console and release (program should now be loading into CPU)

4. STARTING PROCEEDURE

- A. Set switch register to 000200
- B. Depress 'LOAD ADDRESS' key and release
- C. Set SWR to zero for 'AUTO SIZING' or leave
leave SWR bit 7=1 to use existing parameters set up by DV11 trial
program or a previously run DV11 diagnostic that used the 'AUTO
SIZING'. (section 7.2 and 8.4,8.5 may be helpful)
- D. Depress 'START KEY' and release the program will type Maindec Name
and program name (if this was the first start up of the program) and
also the following:

```

'MAP OF DV11 STATUS'
1500 175000
1502 000300
1504 000226
1506 000062
1510 000226
1512 000062
1514 000226
1516 000062
1520 000226
1522 000062

```

The above is only an example! This would indicate the status table starting at add. 1500 in the program. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. For information of status table see section 8.4 for help.

The program will type 'R' and proceed to run the diagnostic

4.1 CONTROL SWITCH SETTINGS

NOTE: If there is no real SWR (177570); SWR may be modified at Loc:176 or by hitting Control 'G' <^G> on console terminal.

```

SW 15 Set: Halt on error
SW 14 Set: Loop on current test
SW 13 Set: Inhibit error print out
SW 12 Set: Inhibit **ALL** type out/bell on error.
SW 11 Set: Inhibit iterations. (quick pass)
SW 10 Set: Escape to next test
SW 09 Set: Loop with current data
SW 08 Set: Catch error and loop on it
SW 07 Set: Use previous status table. CLR-do AUTO SIZE.
SW 06 Set: Reserved
SW 05 Set: Reserved
SW 04 Set: Reserved
SW 03 Set: Reserved
SW 02 Set: Lock on selected test
SW 01 Set: Restart program at selected test
SW 00 Set: Reselect DV11's desired active.

```

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 RESELECT DV11'S DESIRED ACTIVE. please note that a message is typed out for setting the switch register equal to DV11's active. this means if the system has four DV11s; bits 00,01,02,03 will be set in loc 'DVACTV' from the switch register. Using this switch(SW00) alters that location;therefore if four DV11s are in the system ***DO NOT*** set switches greater than SW 03 in the up position. this would be a fatal error. do not select more active DV11s than has been given information about in trial program.

METHOD: A: Load address 200
 B: Start with SW 00=1
 C: Program will type message
 D: Set the binary number of DV11s desired active EXAMPLE: 1=1
 DV11; 3=2 DV11; 7=3 DV11; 17=4 DV11 37=5 DV11 etc. PRESS
 CONTINUE.
 E: Number (IF VALID) will be in data lights (excluding 11/05)
 F: Set with any other switch settings desired. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST it is strongly suggested that at least one pass has been made before trying to select a test that is not in the order of sequence the reason being is that the program has to clear areas and set up parameters. Also when a test is selected ALWAYS START AT THE VERY BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA: this switch will only work if call 'SCOPI' is in that test. The reason being that most tests deal with blocks of different data to be sent or received all at once thus in block data; one pattern can't be singled out.

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

1. SW 12 Delete print out/bell on error.
2. SW 13 Delete error printout.
3. SW 15 Halt on the error.
4. SW 08 Goto beginning of the test(on error).
5. SW 10 Goto next test(on error).

SCOPE SWITCHES

1. SW 09 (if enabled by 'SCOP1') on an error; If an '*' is printed in front of the test no. (ex. *TEST NO. 10) SW09 is incorporated in that test and therefore SW09 is *usually* the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0). If SW09 is not enabeled; and there is a *HARD* error (constant); SW08 is best.
(SW14=1,0, SW10=0, SW09=0, SW08=1). for intermitent errors; SW14=1 will loop on test regardless of error or not error.
(SW14=1, SW10=0, SW09=0, SW08=1,0)
2. SW 14
3. SW 11

4.2 STARTING ADDRESS

starting address is at 000200 there are no other starting addresses for the DV11 diagnostics previously mentioned except for DZDVE which is: 000200 for the modem control and cable tests and 000210 for the manual parameter input program.

NOTE: If address 000042 is non-zero the program assumes it is under ACT11 or XXDP control and will act accordingly after *ALL* available DV11's are tested the program will return to 'XXDP' or 'ACT-11'.

5. OPERATING PROCEDURE

When program is initially started messages as described in section four will be printed.

and program will begin running the diagnostic

5.2 PROGRAM AND/OR OPERATOR ACTION

The typical approach should be

1. Halt on error (via SW 15=1) when ever an error occurs.
2. Clear SW 15.
3. Set SW 14: (loop on this test)
4. Set SW 13: (inhibit error print out)

The TEST NUMBER and PC will be typed out and possibly an error message (this depends on the test) to give the operator an idea as to the source of the problem. if it is necessary to know more information concerning the error report; LOOK IN THE LISTING for that TEST NUMBER which was typed out and then NOTE THE PC of the ERROR REPORT this way the EXACT FUNCTIONING OF THE TEST CAN BE INTERPRETED.

5.2.1

If the data 'Set Buzy' jumper(s) are removed from the M7833 async line card, the following patch should be installed.

CHANGE LOC. 23106 FROM 403 TO 240
23114 FROM 34000 TO 7000

This patch puts the DV-11BB in internal maintenance for the particular test. This will work with all async line cards; will not work with sync line cards.

6. ERRORS

As described previously there will always be a TEST NUMBER and PC typed out at the time of an error (providing SW 13=0 and SW 12=0). in most cases additional information will be supplied to the the error message which is to give the operator an indication of the error.

6.2 ERROR RECOVERY

If for some reason the DV11 should 'HANG THE BUS' (gain control of bus so that console manual functions are inhibited) an init or power down/up is necessary for operator to regain control of cpu. If this should HAPPEN; LOOK IN LOCATION 'TSTNO' 9ADDRESS 1226) FOR THE NUMBER OF THE test that was running at the time of the catastrophic error. In this way the operator will have an idea as to what the DV11 was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4. (PLEASE)
Status table should be verified regardless of how program was started. Also it is important to use this listing along with the information printed on the TTY to completly isolate problems.

7.2 OPERATING RESTRICTIONS

DV11 trial program must be run prior to the first and only the first running of any DV11 diagnostic if 'AUTO SIZING' is not used.

NOTE: If no program other than a DV11 diagnostic was loaded after DV11 trial or if core memory has not been changed; or if there is no DV11 configuration changes; the DV11 trial program need never be run again. However if any of the above have been violated the DV11 trial program must be run again before running the diagnostics NOTE: An alternative to the above is attempting the 'AUTO SIZING' when program is initially started with SW07=0.

7.3 HARDWARE CONFIGURATION RESTRICTIONS (SYNC LINE CARDS ONLY)

1. Hardware must be set to FULL DUPLEX
2. ALL LINES OF A PARTICULAR LINE CARD MUST BE CONFIGURED THE SAME.

8. MISCELLANEOUS

8.1 EXECUTION TIME

All DV11 device diagnostics will give an 'END PASS' message (providing no errors and sw12=0) within 4 mins. This is assuming SW11=1 (DELETE ITERATIONS) is set to give the fastest possible execution. The actual execution time depends greatly on the PDP11 CPU configuration.

8.2 PASS COMPLETE

NOTE: *EVERY* time the program is started; the tests will run as if SW11 (delete iterations) was up (=1). This is to 'VERIFY NO *HARD* ERRORS' as soon as possible. Therefore the first pass -EACH TIME PROGRAM IS STARTED- will be a 'QUICK PASS' untill all DV11's in system are tested. When the diagnostic has completed a pass the following is an example of the print out to be expected.

END PASS CZDVD CSR: 175000 VEC: 300 PASSES: 000001 ERRORS:

NOTE: The numbers for CSR and VEC are not necessarily the values for the device. They are only for this example.

NOTE: CZDVE (MODEM AND CABLE TEST) END PASS MESSAGE IS A LARGE 'END' typed out on tty. Please note that each character printed is actually and 'END PASS' indication. This was used in place of 'BELL' because if sw12=1 and an error occurred the BELL may be mistaken for END PASS. The pass execution is so fast that the standard END PASS was too lengthy. THEREFORE each char is an 'END PASS and the entire 'END' is not required for acceptance.

8.4 KEY LOCATIONS

RETURN (1214) CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN iteration count is reached or if loop on test is asserted.

NEXT (1216) CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.

TSTNO (1226) CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.

RUN (1304) THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DV11 currently being tested. EXAMPLE: (RUN) 1302/0000000001000000 Means that DV11 no.05 is the DV11 now running.

DVCR00-DVCR17
DVST00-DVST17
(1500)-(1736)

These locations contain the information needed to test up to 8 (decimal) DV11s sequentially. they contain the CSR, VECTOR and STATUS concerning the configuration of each DV11.

DVACTV (1300) EACH BIT SET IN THIS LOCATION INDICATES THAT THE associated DV11 will be tested in turn. EXAMPLE: (DVACTV) 1276/0000000000011111 means that DV11 no. 00,01,02,03,04 will be tested. EXAMPLE: (DVACTV) 1276/0000000000010001 Means that DV11 no. 00,04 will be tested.

DVSCR (1362) CONTAINS THE RECEIVER CSR OF THE CURRENT DV11 UNDER test.

L00.03 (1422)
L04.07 (1424)
L08.11 (1426)
L12.15 (1430)

CONTAINS THE STATUS OF THE CURRENT DV11 UNDER TEST.

BIT 15	Set:	Line card *NOT installed (AND WONT BE TESTED)
BIT 14	SET:	SET FOR PARITY ENABLED.
BIT 13	SET:	SET FOR EVEN PARITY ENABLED.
BIT 12	Set:	One sync, =0: two syncs.
BIT 11	Set:	Async line card, =0 Sync line card.
BIT 10	Set:	Reserved
BIT 09	Set:	Bits per char. (used with bit8)
BIT 08	Set:	Bits per char. (used with bit9)

BIT09 BIT08 BITS PER CHAR.

0	0	8
0	1	7
1	0	6
1	1	5

BIT 07-00 SYNC 'A' for specified line card. Bits 07-00 must be all zeros for testing Async line cards.

'MAP OF DV11 STATUS'

1500	175000
1502	000300
1504	000226
1506	000062
1510	000226
1512	000062
1514	004000
1516	000000
1520	004000
1522	000000

The above information will be repeated for each of up to 8 DV11's in the system (these will follow under this table). EXPLANATION:

1500 175000 This is the system control register for the 1st DV11 in the system.
 1502 000300 This is vector 'A' for the first DV11 in the system.
 1504 000226 This represents 'SYNC A' and the software status for the 1st line card in the 1st DV11. The bits are as follows:

BIT 15 Set: Line card *NOT installed (AND WONT BE TESTED)
 BIT 14 SET: PARITY ENABLED (USED WITH BIT 13)
 BIT 13 SET: PARITY ENABLED (USED WITH BIT 14)

BIT 14 BIT 13 PARITY CONDITION

0	0	NO PARITY
1	0	ODD PARITY
1	1	EVEN PARITY

BIT 12 Set: One sync, =0: two syncs.
 BIT 11 Set: Async line card, =0 Sync line card.
 BIT 10 Set: Reserved
 BIT 09 Set: Bits per char. (used with bit8)
 BIT 08 Set: Bits per char. (used with bit9)

BIT 09 BIT 08 BITS PER CHAR.

0	0	8
0	1	7
1	0	6
1	1	5

BIT 07-00 SYNC 'A' for specified line card.

1506 000062 This represents 'SYNC B' for the 1st line card.
 1510 000226 This is 'SYNC A' and line status for the 2nd line card.
 (for bits defination see explanation for line card 1).
 1512 000062 This is 'SYNC B' for the second line card.
 1514 000226 This is 'SYNC A' and line status for the 3rd line card.
 (for bits defination see explanation for line card 1).
 1516 000062 This is 'SYNC B' for line card no. 3.
 1520 000226 This is 'SYNC A' and line status for the 4th line card.
 (for bits defination see explanation for line card 1).
 1522 000062 This is SYNC B for the 4th line card.

The above is repeated for each DV11 in the system. The table is filled by AUTO SIZING or by the manual parameter input program as described previously. Also if desired by user; the locations may be altered by hand (toggled in) to suit the specific configuration.

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The program will start at address 175000 and start 'REFERENCEING' address. If a NON-EX MEMORY TRAP occurs; the pointer (holding 175000) is updated by 10 and the above is repeated until address 175400 is reached. If a 'SLAVE SYNC RESPONSE' was issued by the DV11 (or any other device) (no nxm trap) (and it (SEL0) was=0) ; pointer plus 12 (SEL12) is tested to contain 177777 (MUST BE EXACTLY 177777); if a trap is encountered or if SEL12 does not contain 177777 the above updating is performed. If SEL12 was equal to 177777 the pointer is stored away and the routine continues as above:
 NOTE: If the program does not find your DV11; something is wrong and AUTO SIZING should not be done.

8.5.2 FINDING THE VECTOR

The vector area (address 300-776) is filled with the instruction IOT and '+2' (next address). Bit7 and Bit6 (RX INTERRUPT AND RX INTERRUPT IE) are set into DVscr register; a delay is made and if no interrupt occurs (because of a bad DV11) the program assumes vector address 300 and the problem should be fixed in the diagnostic. Once the problem is fixed; the program should be re-setup again to get correct vector. If an interrupt occurred; the address to which the DV11 interrupted to is picked up and reported as the vector. NOTE: if the vector reported is not the vector set up by you; there is a problem and AUTO SIZING should not be done.

8.5.3 PARAMETER ASSUMPTIONS.

Since too much hardware would need to be turned on to SIZE the rest of the parameters; the program must assume the remaining variations. The result if not to your specific configuration may be altered by hang (toggle in) is desired. In this way 95% of the parameter setup was done by the program and 5% by you.

THEREFORE:

- 1) ALL LINE CARDS(4) ARE ASSUMED TO BE INSTALLED.
Set Bit15 of status map of any (appropriate) line cards missing
- 2) TWO SYNCs.
Set Bit12 if you have a 4 line group set for 1 sync.
- 3) 5,6,7, OR 8 BITS PER CHARACTER
Adjust bits 9 and bit 8 in status map for your correct config.
- 4) ODD, EVEN, OR NO PARITY ENABLED.
ADJUST BIT14 AND BIT13 IN STATUS MAP FOR YOUR CORRECT CONFIGURATION.
- 5) SYNCHRONOUS LINE CARDS INSTALLED
Set Bit11 of status map for async line card and zero sync chars.
- 6) SYNC 'A'=226 AND SYNC 'B'=062 NOTE: FOR <8 BITS/CHARACTER
THESE NUMBERS MAY HAVE TO BE ADJUSTED TO REFLECT CHARACTER LENGTH.

In all adjustments please refer to section 8.4a for greater detail.

M.

9.0 ***CHANGE HISTORY***

NOTE: HISTORY BEGINS WITH REV. D0

SEQ 0012

D0: MODIFIED PROGRAM TO OPERATE WITH 5,6,7, OR 8 BITS PER
CHARACTER; ODD, EVEN, OR NO PARITY ENABLED.

1
2
3
4
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

```
;*CZDVDD-0/<377>/CZDVDD0 DV11 ROM TST PRT2
;*COPYRIGHT 1972,1980, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
;-----

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE 'CZDVDD-0/<377>/CZDVDD0 DV11 ROM TST PRT2'
;PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
;AND THEN RESUME TESTING

;SWITCH REGISTER OPTIONS
;-----

SW15=100000      ;=1,HALT ON ERROR
SW14=40000       ;=1,LOOP ON CURRENT TEST
SW13=20000       ;=1,INHIBIT ERROR TYPEOUT
SW12=10000       ;=1,DELETE TYPEOUT/BELL ON ERROR.
SW11=4000        ;=1,INHIBIT ITERATIONS
SW10=2000        ;=1,ESCAPE TO NEXT TEST ON ERROR
SW09=1000        ;=1,LOOP WITH CURRENT DATA
SW08=400         ;=1,LOOP ON ERROR
SW07=200         ;=1, DO 'AUTO SIZING' ON INITIAL START UP.
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4           ;LOCK ON TEST SELECT
SW01=2           ;RESTART PROGRAM AT SELECTED TEST
SW00=1           ;RESELECT DV11 DESIRED ACTIVE
                ;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```

```

36
37
38      ;REGISTER DEFINITIONS
39      ;-----
40
41      000000      R0=X0      ;GENERAL REGISTER
42      000001      R1=X1      ;GENERAL REGISTER
43      000002      R2=X2      ;GENERAL REGISTER
44      000003      R3=X3      ;GENERAL REGISTER
45      000004      R4=X4      ;GENERAL REGISTER
46      000005      R5=X5      ;GENERAL REGISTER
47      000006      SP=X6      ;PROCESSOR STACK POINTER
48      000007      PC=X7      ;PROGRAM COUNTER
49
50      ;LOCATION EQUIVALENCIES
51      ;-----
52
53      177776      PS=177776    ;PROCESSOR STATUS WORD
54      001200      STACK=1200   ;START OF PROCESSOR STACK
55
56      100000      BIT15=100000
57      040000      BIT14=40000
58      020000      BIT13=20000
59      010000      BIT12=10000
60      004000      BIT11=4000
61      002000      BIT10=2000
62      001000      BIT9=1000
63      000400      BIT8=400
64      000200      BIT7=200
65      000100      BIT6=100
66      000040      BIT5=40
67      000020      BIT4=20
68      000010      BIT3=10
69      000004      BIT2=4
70      000002      BIT1=2
71      000001      BIT0=1
72      ;-----
73      010000      ALU=BIT12
74      020000      RAM=BIT13
75      030000      XFR=BIT13*BIT12
76      040000      NPR=BIT14
77      050000      S.C=BIT14*BIT12
78      060000      BCC=BIT14*BIT13
79      070000      BRB=BIT14*BIT13*BIT12
80      ;-----
81
82

```

```

83 .....
84 .....
85 :TRAPCATCAER FOR ILLEGAL INTERRUPTS
86 :THE STANDARD "TRAP CATCHER" IS PLACED
87 :BETWEEN ADDRESS 0 TO ADDRESS 776.
88 :IT LOOKS LIKE "PC+2 HALT".
89 .....
90 .....
91
92      000000      .-0
93      :STANDARD INTERRUPT VECTORS
94      :-----
95
96      000024      .-24
97      000024      004402      .PFAIL      :POWER FAIL HANDLER
98      000026      000340      340      :SERVICE AT LEVEL 7
99      000030      004002      .HLT      :ERROR HANDLER
100     000032      000340      340      :SERVICE AT LEVEL 7
101     000034      003750      .TRPSRV      :GENERAL HANDLER DISPATCH SERVICE
102     000036      000340      340      :SERVICE AT LEVEL 7
103
104     000040      000001      .=40      .BLKW 1      :SAVE FOR ACT-11 OR DDP2
105     000042      000001      .BLKW 1      :RETURN ADDRESS IF UNDER ACT-11 OR DDP2
106     000044      000001      .BLKW 1      :SAVE FOR ACT-11 OR DDP2
107     000046      002560      LOGICAL      :FOR USE WITH ACT-11 OR DDP2
108
109     000174      000174      .=174
110     000174      000000      LIGHT: 0
111     000176      000176      .=176
112     000176      000000      SSWR: 0
113
114     000200      000200      .=200
115     000200      000137      001742      JMP      .START      :GO TO START OF PROGRAM
116
117
118     001000      001000      .=1000
119     001000      005377      055103      053104      MTITLE: .ASCIIZ <377><12>/CZDVDD-0/<377>/CZDVDD0 DV11 ROM TST PRT2/<377>
120     001200      001200      .=1200
121     001200      001200      LIGHTS:
122     001200      177570      177570
123     001202      177570      SWR: 177570
124      :INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
125      :-----
126
127     001204      177560      TKCSR: 177560      :TELETYPE KEYBOARD CONTROL REGISTER
128     001206      177562      TKDBR: 177562      :TELETYPE KEYBOARD DATA BUFFER
129     001210      177564      TPCSR: 177564      :TELEPRINTER CONTROL REGISTER
130     001212      177566      TPDBR: 177566      :TELEPRINTER DATA BUFFER
131
132      :PROGRAM CONTROL PARAMETERS
133      :-----
134
135     001214      000000      RETURN: 0      :SCOPE ADDRESS FOR LOOP ON TEST
136     001216      000000      NEXT: 0      :ADDRESS OF NEXT TEST TO BE EXECUTED
137     001220      000000      LOCK: 0      :ADDRESS FOR LOCK ON CURRENT DATA

```


CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 5

CZDVDD.P11 02-JUN-80 09:32

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0016

138	001222	000003	ICOUNT: 3	:NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
139	001224	000000	LPCNT: 0	:NUMBER OF ITERATIONS COMPLETED
140	001226	000000	TSTNO: 0	:NUMBER OF TEST IN PROGRESS
141	001230	000000	PASCNT: 0	:NUMBER OF PASSES COMPLETED
142	001232	000000	ERRCNT: 0	:TOTAL NUMBER OF ERRORS
143	001234	000000	LSTERR: 0	:PC OF LAST ERROR CALL
144				
145			:PROGRAM VARIABLES	
146			:-----	
147				
148	001236	000000	STAT: 0	:DV STATUS WORD STORAGE
149	001240	000000	SYNCX: 0	
150	001242	000000	CLKX: 0	
151	001244	000000	MASKX: 0	
152	001246	000000	TEMP1: 0	:TEMPORARY STORAGE
153	001250	000000	TEMP2: 0	:TEMPORARY STORAGE
154	001252	000000	TEMP3: 0	:TEMPORARY STORAGE
155	001254	000000	TEMP4: 0	:TEMPORARY STORAGE
156	001256	000000	TEMP5: 0	:TEMPORARY STORAGE
157	001260	000000	SAVR0: 0	:R0 STORAGE
158	001262	000000	SAVR1: 0	:R1 STORAGE
159	001264	000000	SAVR2: 0	:R2 STORAGE
160	001266	000000	SAVR3: 0	:R3 STORAGE
161	001270	000000	SAVR4: 0	:R4 STORAGE
162	001272	000000	SAVR5: 0	:R5 STORAGE
163	001274	000000	SAVSP: 0	:STACK POINTER STORAGE
164	001276	000000	SAVPC: 0	:PROGRAM COUNTER STORAGE
165	001300	000001	DVACTV: .BLKB 1	:DV11'S SELECTED ACTIVE.
166	001301	000001	DVNUM: .BLKB 1	:OCTAL NUMBER OF DV11'S.
167	001302	000001	SAVACT: .BLKB 1	:ORIGINAL ACTV. DEVICES.
168	001303	000001	SAVNUM: .BLKB 1	:WORKABLE NUMBER.
169	001304	000001	RUN: .BLKB 1	:POINTER ONE PAST RUNNING DEVICE.
170		001306	.EVEN	
171	001306	001500	CREAM: DV.MAP	:TABLE POINTER.

```
172
173           ;PROGRAM CONTROL FLAGS
174           ;-----
175
176 001310      000      INIFLG: .BYTE 0      ;PROGRAM INITIALIZATION FLAG
177 001311      000      ERRFLG: .BYTE 0      ;ERROR OCCURED FLAG
178 001312      000      LOKFLG: .BYTE 0      ;LOCK ON CURRENT TEST FLAG
179 001313      000      QV.FLG: .BYTE 0      ;QUICK VERIFY FLAG.
180                                           ;ON FIRST PASS OF EACH DV11 ITERATIONS WILL BE SUPPRESSE
181
182           .EVEN
183           $Y=0
184
185           ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
186           ;POINTERS TO SUBROUTINES CAN BE FOUND
187           ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
188
189           ;*****
190           ;-----
191 001314      104400      .TRPTAB:
192 001314      002634      SCOPE=TRAP+0      ;CALL TO SCOPE LOOP AND ITERATION HANDLER
193           104401      .SCOPE
194 001316      003020      SCOP1=TRAP+1      ;CALL TO LOOP ON CURRENT DATA HANDLER
195           104402      .SCOP1
196 001320      003044      TYPE=TRAP+2      ;CALL TO TELETYPE OUTPUT ROUTINE
197           104403      .TYPE
198 001322      003120      INSTR=TRAP+3      ;CALL TO ASCII STRING INPUT ROUTINE
199           104404      .INSTR
200 001324      003224      INSTER=TRAP+4      ;CALL TO INPUT ERROR HANDLER
201           104405      .INSTER
202 001326      003244      PARAM=TRAP+5      ;CALL TO NUMERICAL DATA INPUT ROUTINE
203           104406      .PARAM
204 001330      003444      SAV05=TRAP+6      ;CALL TO REGISTER SAVE ROUTINE
205           104407      .SAV05
206 001332      003504      RES05=TRAP+7      ;CALL TO REGISTER RESTORE ROUTINE
207           104410      .RES05
208 001334      003536      CONVRT=TRAP+10     ;CALL TO DATA OUTPUT ROUTINE
209           104411      .CONVRT
210 001336      003542      CNVRT=TRAP+11     ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
211           104412      .CNVRT
212 001340      004556      MSTCLR=TRAP+12     ;CALL TO ISSUE A MASTER CLEAR
213           104413      .MSTCLR
214 001342      004516      RAMCLR=TRAP+13     ;CALL TO CLEAR THE RAMS
215           104414      .RAMCLR
216 001344      004476      DELAY=TRAP+14     ;CALL TO VARIABLE DELAY COUNTER
217           104415      .DELAY
218 001346      004566      ROMCLK=TRAP+15     ;CALL TO CLOCK ROM ONCE
219           104416      .ROMCLK
220 001350      004576      DATACLK=TRAP+16   ;CALL TO CLK DATA
221           .DATACLK
222
223           ;-----
224           ;*****
```

```

224                                     ;DV11 VECTOR AND REGISTER INDIRECT POINTERS
225
226 001352 000000   DVRVEC: 0           ;POINTER TO DV11 RECEIVER INTERRUPT VECTOR
227 001354 000000   DVRLVL: 0          ;POINTER TO DV11 RECEIVER INTERRUPT SERVICE PS
228 001356 000000   DVTVEC: 0          ;POINTER TO DV11 TRANSMITTER INTERRUPT VECTOR
229 001360 000000   DVTLVL: 0          ;POINTER TO DV11 TRANSMITTER INTERRUPT SERVICE PS
230 001362 000000   DVSCR: 0           ;POINTER TO DV11 SYSTEM CONTROL REGISTER
231 001364 000000   DVSCRH: 0          ;POINTER TO DV11 SYSTEM CONTROL REGISTER HIGH BYTE.
232 001366 000000   DVRIC: 0           ;POINTER TO DV11 NEXT RECEIVED CHARACTER REGISTER
233 001370 000000   DVLCR: 0           ;POINTER TO DV11 LINE PRAMETER REGISTER
234 001372 000000   DVSRs: 0           ;POINTER TO DV11 SECONDARY REGISTER SELECT REGISTER
235 001374 000000   DVSRSRSH: 0        ;POINTER TO DV11 SECONDARY REGISTER SELECT HIGH BYTE.
236 001376 000000   DVSRA: 0           ;POINTER TO DV11 SECONDARY REGISTER ACCESS REGISTER
237 001400 000000   DVSFR: 0           ;POINTER TO DV11 SPECIAL FUNCTIONS REGISTER
238 001402 000000   DVNSR: 0           ;POINTER TO DV11 NPR STATUS REGISTER
239 001404 000000   RESV16: 0          ;POINTER TO RESERVED REGISTER.
240
241
242                                     ;DV11 CONTROL INDICATORS FOR CURRENT DV11 UNDER TEST
243 -----
244
245 001406 000000   MASK.A: .WORD 000   ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 00-03
246 001410 000000   MASK.B: .WORD 000   ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 04-07
247 001412 000000   MASK.C: .WORD 000   ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 08-11
248 001414 000000   MASK.D: .WORD 000   ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 12-15
249
250 001416 010      CLK.A: .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 00-03
251 001417 010      CLK.B: .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 04-07
252 001420 010      CLK.C: .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 08-11
253 001421 010      CLK.D: .BYTE 8.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 12-15
254
255 001422 000000   L00.03: 000000       ;PARAMETERS FOR LINES 00-03
256 001424 000000   L04.07: 000000       ;PARAMETERS FOR LINES 04-07
257 001426 000000   L08.11: 000000       ;PARAMETERS FOR LINES 08-11
258 001430 000000   L12.15: 000000       ;PARAMETERS FOR LINES 12-15
259
260 001432 000000   SYNC2A: 000000       ;SYNC 2
261 001434 000000   SYNC2B: 000000       :
262 001436 000000   SYNC2C: 000000       :
263 001440 000000   SYNC2D: 000000       :
264
265                                     ;SUMMARY
266 -----
267 :      MASK.X      040      5 BITS PER CHAR.
268 :                  100      6 BITS PER CHAR.
269 :                  200      7 BITS PER CHAR.
270 :                  400      8 BITS PER CHAR.
271
272 :      CLK.X       005      5 BITS PER CHAR.
273 :                  006      6 BITS PER CHAR.
274 :                  007      7 BITS PER CHAR.
275 :                  010      8 BITS PER CHAR.
276 :      IF PARITY IS ENABLED; ADD PLUS ONE TO THE ABOVE "CLK.X"
277 :      FOR EACH GROUP THAT PARITY IS ENABLED.

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 8
 CZDVDD.P11 02-JUN-80 09:32

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0019

```

278                                     ;DV11 STATUS TABLE AND ADDRESS ASSIGNMENTS
279                                     ;-----
280
281                                     . =1500
282 001500 DV.MAP:
283 001500 DVCRO0: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 00
284 001502 DVTR00: .BLKW 1 ;VECTOR 'A' FOR DV11 NUMBER 00
285 001504 DV00.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 00
286 001506 SYNA00: .BLKW 1 ;SYNC TWO
287 001510 DV00.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 00
288 001512 SYNBO0: .BLKW 1 ;SYNC TWO
289 001514 DV00.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 00
290 001516 SYNC00: .BLKW 1 ;SYNC TWO
291 001520 DV00.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 00
292 001522 SYND00: .BLKW 1 ;SYNC TWO
293
294 001524 DVCRO1: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 01
295 001526 DVTR01: .BLKW 1 ;VECTOR 'A' FOR DV11 NUMBER 01
296 001530 DV01.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 01
297 001532 SYNA01: .BLKW 1 ;SYNC TWO
298 001534 DV01.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 01
299 001536 SYNBO1: .BLKW 1 ;SYNC TWO
300 001540 DV01.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 01
301 001542 SYNC01: .BLKW 1 ;SYNC TWO
302 001544 DV01.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 01
303 001546 SYND01: .BLKW 1 ;SYNC TWO
304
305 001550 DVCRO2: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 02
306 001552 DVTR02: .BLKW 1 ;VECTOR 'A' FOR DV11 NUMBER 02
307 001554 DV02.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 02
308 001556 SYNA02: .BLKW 1 ;SYNC TWO
309 001560 DV02.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 02
310 001562 SYNBO2: .BLKW 1 ;SYNC TWO
311 001564 DV02.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 02
312 001566 SYNC02: .BLKW 1 ;SYNC TWO
313 001570 DV02.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 02
314 001572 SYND02: .BLKW 1 ;SYNC TWO
315
316 001574 DVCRO3: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 03
317 001576 DVTR03: .BLKW 1 ;VECTOR 'A' FOR DV11 NUMBER 03
318 001600 DV03.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 03
319 001602 SYNA03: .BLKW 1 ;SYNC TWO
320 001604 DV03.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 03
321 001606 SYNBO3: .BLKW 1 ;SYNC TWO
322 001610 DV03.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 03
323 001612 SYNC03: .BLKW 1 ;SYNC TWO
324 001614 DV03.D: .BLKW 1 ;PARAMETER FOR LINES 12 15 FOR DV11 NUMBER 03
325 001616 SYND03: .BLKW 1 ;SYNC TWO
326
327 001620 DVCRO4: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 04
328 001622 DVTR04: .BLKW 1 ;VECTOR 'A' FOR DV11 NUMBER 04
329 001624 DV04.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 04
330 001626 SYNA04: .BLKW 1 ;SYNC TWO
331 001630 DV04.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 04
332 001632 SYNBO4: .BLKW 1 ;SYNC TWO
333 001634 DV04.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 04

```


CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 9
 CZDVDD.P11 02-JUN-80 09:32

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0020

```

334 001636 000001      SYNC04: .BLKW 1      ; SYNC TWO
335 001640 000001      DV04.D: .BLKW 1      ; PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 04
336 001642 000001      SYND04: .BLKW 1      ; SYNC TWO
337
338 001644 000001      DVCRO5: .BLKW 1      ; CONTROL STATUS REGISTER FOR DV11 NUMBER 05
339 001646 000001      DVTR05: .BLKW 1      ; VECTOR 'A' FOR DV11 NUMBER 05
340 001650 000001      DV05.A: .BLKW 1      ; PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 05
341 001652 000001      SYNA05: .BLKW 1      ; SYNC TWO
342 001654 000001      DV05.B: .BLKW 1      ; PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 05
343 001656 000001      SYNBO5: .BLKW 1      ; SYNC TWO
344 001660 000001      DV05.C: .BLKW 1      ; PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 05
345 001662 000001      SYNC05: .BLKW 1      ; SYNC TWO
346 001664 000001      DV05.D: .BLKW 1      ; PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 05
347 001666 000001      SYND05: .BLKW 1      ; SYNC TWO
348
349 001670 000001      DVCRO6: .BLKW 1      ; CONTROL STATUS REGISTER FOR DV11 NUMBER 06
350 001672 000001      DVTR06: .BLKW 1      ; VECTOR 'A' FOR DV11 NUMBER 06
351 001674 000001      DV06.A: .BLKW 1      ; PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 06
352 001676 000001      SYNA06: .BLKW 1      ; SYNC TWO
353 001700 000001      DV06.B: .BLKW 1      ; PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 06
354 001702 000001      SYNBO6: .BLKW 1      ; SYNC TWO
355 001704 000001      DV06.C: .BLKW 1      ; PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 06
356 001706 000001      SYNC06: .BLKW 1      ; SYNC TWO
357 001710 000001      DV06.D: .BLKW 1      ; PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 06
358 001712 000001      SYND06: .BLKW 1      ; SYNC TWO
359
360 001714 000001      DVCRO7: .BLKW 1      ; CONTROL STATUS REGISTER FOR DV11 NUMBER 07
361 001716 000001      DVTR07: .BLKW 1      ; VECTOR 'A' FOR DV11 NUMBER 07
362 001720 000001      DV07.A: .BLKW 1      ; PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 07
363 001722 000001      SYNA07: .BLKW 1      ; SYNC TWO
364 001724 000001      DV07.B: .BLKW 1      ; PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 07
365 001726 000001      SYNBO7: .BLKW 1      ; SYNC TWO
366 001730 000001      DV07.C: .BLKW 1      ; PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 07
367 001732 000001      SYNC07: .BLKW 1      ; SYNC TWO
368 001734 000001      DV07.D: .BLKW 1      ; PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 07
369 001736 000001      SYND07: .BLKW 1      ; SYNC TWO
370
371 001740 000000      DV.END: 000000
372
373      ;PROGRAM INITIALIZATION
374      ;LOCK OUT INTERRUPTS
375      ;SET UP PROCESSOR STACK
376      ;SET UP POWER FAIL VECTOR
377      ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
378      ;TYPE TITLE MESSAGE
379
380 001742 012737 000340 .77776 .START: MOV #340,PS      ;LOCK OUT INTERRUPTS
381 001750 012706 001200      MOV #STACK,SP      ;SET UP STACK
382 001754 012737 004402 000024      MOV #.PFail,#24      ;SET UP POWER FAIL VECTOR
383 001762 113737 001301 001303      MOV# DVNUM,SAVNUM      ;SAVE NUMBER OF DEVICES IN SYSTEM.
384 001770 005037 001230      CLR PASCNT      ;CLEAR PASS COUNT
385 001774 105037 001311      CLRB ERRFLG      ;CLEAR ERROR FLAG
386 002000 105037 001313      CLRB QV.FLG      ;ZERO QUICK VERIFY FLAG
387 002004 012737 001500 001306      MOV #DV.MAP,CREAM      ;GET MAP POINTER.
388 002012 112737 000001 001304      MOV# #1,RUN      ;POINT POINTER TO FIRST DEVICE.
389 002020 005037 001232      CLR ERRCNT      ;CLEAR ERROR COUNT

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 10
 CZDVDD.P11 02-JUN-80 09:32 PROGRAM INITIALIZATION AND START UP.

SEQ 0021

390	002024	005037	001234		CLR	LSTERR	:CLEAR LAST ERROR POINTER
391	002030	012737	000001	001226	MOV	#1,TSTNO	:SET UP FOR TEST 1
392	002036	012737	001742	001214	MOV	#.START,RETURN	:SET UP FOR POWER FAIL BEFORE
393							:TESTING STARTS
394	002044	105737	001310		TSTB	INIFLG	:HAS INITIALIZATION BEEN PERFORMED
395	002050	001063			BNE	1\$:BR IF YES
396	002052	013746	000004		MOV	4,-(SP)	
397	002056	013746	000006		MOV	6,-(SP)	
398	002062	005037	000006		CLR	6	
399	002066	012737	002104	000004	MOV	#80\$,4	
400	002074	005777	177102		TST	@SWR	
401	002100	000240			NOP		
402	002102	000407			BR	81\$	
403	002104	022626			CMP	(SP)+,(SP)+	
404	002106	012737	000174	001200	MOV	#LIGHT,LIGHTS	
405	002114	012737	000176	001202	MOV	#SSWR,SWR	
406	002122	012637	000006		MOV	(SP)+,6	
407	002126	012637	000004		MOV	(SP)+,4	
408	002132	104402	001000		TYPE	,MTITLE	:TYPE TITLE MESSAGE
409	002136	105137	001310		COMB	INIFLG	:IF NOT SET FLAG AND DO
410	002142	105777	177034		TSTB	@SWR	:BIT7=1??
411	002146	100402			BMI	16\$:BR IF NO AUTO SIZE
412	002150	004737	006624		JSR	PC,CSRMAP	:GO DO THE AUTO SIZE
413	002154	104402	005457		TYPE	,XHEAD	:TYPE HEADER
414	002160	012737	001500	001246	MOV	#DV.MAP,TEMP1	:SET POINTER
415	002166	017737	177054	001250	MOV	@TEMP1,TEMP2	:SET DATA
416	002174	022737	177777	001250	CMP	#177777,TEMP2	:ALL DONE?
417	002202	001406			BEQ	1\$:BR IF YES
418	002204	104410			CONVRT		
419	002206	005504			XSTATQ		
420	002210	062737	000002	001246	ADD	#2,TEMP1	:UPDATE POINTER
421	002216	000763			BR	5\$	
422	002220	005737	000042		TST	@#42	:IS PROGRAM RUNNING UNDER MONITOR
423	002224	001030			BNE	3\$:BR IF YES
424	002226	032777	000001	176746	BIT	#SW00,@SWR	:SELECT SPECIFIC DEVICES??
425	002234	001424			BEQ	3\$:BR IF NO.
426	002236	104402	005400		TYPE	,MNEW	:TYPE THE MESSAGE.
427	002242	005000			CLR	R0	:ZERO DATA LIGHTS
428	002244	000000			HALT		:WAIT FOR USER TO TELL WHAT DEVICES TO RUN
429	002246	127737	176730	001302	CMPB	@SWR,SAVACT	:IS THE NUMBER VALID?
430	002254	101404			BLOS	2\$:BR IF NUMBER IS OK.
431	002256	104402	005241		TYPE	,MERR3	:TELL USER OF INVALID NUMBER.
432	002262	000000			HALT		:STOP EVERY THING.
433	002264	000776			BR	.-2	:RESTART THE PROGRAM AGAIN.
434	002266	117737	176710	001300	MOVB	@SWR,DVACTV	:GET NEW DEVICE PATTERN
435	002274	113700	001300		MOVB	DVACTV,R0	:SHOW THE USER WHAT HE SELECTED.
436	002300	042700	177400		BIC	#^C<377>,R0	:USE ONLY LOW BYTE.
437	002304	000000			HALT		:CONTINUE DYNAMIC SWITCHES.
438	002306	012700	000300		MOV	#300,R0	:PREPARE TO CLEAR THE FLOATING
439	002312	012701	000302		MOV	#302,R1	:VECTOR AREA. 300-776
440	002316	010120			MOV	R1,(R0)+	:START PUTTING 'PC+2 - HALT'
441	002320	005021			CLR	(R1)+	:IN VECTOR AREA.
442	002322	022021			CMP	(R0)+,(R1)+	:POP POINTERS
443	002324	022700	001000		CMP	#1000,R0	:ALL DONE??
444	002330	001372			BNE	4\$:BR IF NO.
445							

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 11
 CZDVDD.P11 02-JUN-80 09:32 PROGRAM INITIALIZATION AND START UP.

SEQ 0022

```

446                                     ;TEST START AND RESTART
447                                     ;-----
448
449 002332 012737 000340 177776 .BEGIN: MOV    #340,PS          ;LOCK OUT INTERRUPTS
450 002340 012706 001200          MOV    #STACK,SP          ;SET UP STACK
451 002344 005737 000042          TST    @#42              ;IS PROGRAM UNDER MONITOR CONTROL
452 002350 001023          BNE     3$                      ;BR IF YES
453 002352 032777 000004 176622  BIT    #BIT2,@SWR          ;CHECK FOR LOCK ON TEST
454 002360 001411          BEQ     1$                      ;BR IF NO LOCK DESIRED.
455 002362 104402 005277          TYPE   ,MLOCK             ;TYPE LOCK SELECTED.
456 002366 012737 000240 002702  MOV    #NOP,TTST          ;ADJUST SCOPE ROUTINE.
457 002374 012737 000240 002704  MOV    #NOP,TTST+2        ;SET UP TO LOCK
458 002402 000406          BR      2$                      ;CONTINUE ALONG.
459 002404 013737 003014 002702 1$:  MOV    BRW,TTST         ;PREPARE NORMAL SCOPE ROUTINE
460 002412 013737 003016 002704  MOV    BRX,TTST+2        ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
461 002420          2$:
462 002420 012737 005664 001214 3$:  MOV    #CYCLE,RETURN    ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
463 002426 104402 005167          TYPE   ,MR              ;TYPE R
464 002432 000177 176556          JMP    @RETURN           ;START TESTING

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 12
 CZDVDD.P11 02-JUN-80 09:32 END OF PASS ROUTINE

SEQ 0023

```

465                                     :END OF PASS
466                                     :TYPE NAME OF TEST
467                                     :UPDATE PASS COUNT
468                                     :CHECK FOR EXIT TO ACT-11
469                                     :RESTART TEST
470
471 002436 000005 .EOP: RESET                                     :MAKE THE WORLD CLEAN AGAIN.
472 002440 005037 001234 CLR LSTERR                               :CLEAR LAST ERROR PC
473 002444 105037 001311 CLRB ERRFLG                             :CLEAR ERROR FLAG
474 002450 005237 001230 INC PASCNT                               :UPDATE PASS COUNT
475 002454 013777 001230 176516 MOV PASCNT,@LIGHTS             :DISPLAY PASS COUNT
476 002462 104402 005145 TYPE ,MEPASS                           :TYPE END PASS
477 002466 104402 005326 TYPE ,MCSR                               :TYPE CSR
478 002472 104411 002604 CNVRT ,XCSR                             :SHOW IT
479 002476 104402 005334 TYPE ,MVECX                             :TYPE VECTOR
480 002502 104411 002612 CNVRT ,XVEC                             :SHOW IT
481 002506 104402 005342 TYPE ,MPASSX                           :TYPE PASSES
482 002512 104411 002620 CNVRT ,XPASS                           :SHOW IT
483 002516 104402 005353 TYPE ,MERRX                             :TYPE ERRORS
484 002522 104411 002626 CNVRT ,XERR                             :SHOW IT
485 002526 105337 001303 DECB SAVNUM                             :ARE ALL DEVICES TESTED?
486 002532 001017 BNE RESTRT                                     :BR IF NO.
487 002534 112737 000377 001313 MOVB #377,QV.FLG               :SET THE QUICK VERIFY FLAG.
488 002542 113737 001301 001303 MOVB DVNUM,SAVNUM              :RESTORE THE COUNT
489 002550 013701 000042 MOV @#42,R1                           :CHECK FOR ACT-11 OR DDP
490 002554 001406 BEQ RESTRT                                     :IF NOT, CONTINUE TESTING
491 002556 000005 RESET                                         :STOP THE SHOW--CLEAR THE WORLD
492 002560
493 002560 004711 LOGICAL: JSR PC,(R1)
494 002562 000240 NOP
495 002564 000240 NOP
496 002566 000240 NOP
497 002570 000240 NOP
498 002572 012737 005664 001214 RESTRT: MOV #CYCLE,RETURN
499 002600 000137 005664 JMP CYCLE
500 002604 000001 XCSR: 1
501 002606 006 002 .BYTE 6,2
502 002610 001362 DVSCR
503 002612 000001 XVEC: 1
504 002614 003 002 .BYTE 3,2
505 002616 001352 DVRVEC
506 002620 000001 XPASS: 1
507 002622 006 002 .BYTE 6,2
508 002624 001230 PASCNT
509 002626 000001 XERR: 1
510 002630 006 002 .BYTE 6,2
511 002632 001232 ERRCNT
512
513                                     :SCOPE LOOP AND INTERATION HANDLER
514                                     :-----
515
516 002634 .SCOPE:
517 002634 022737 177570 001202 CMP #177570,SWR             :IS THERE A REAL SWR?
518 002642 001411 BEQ 64$                                         :BR IF YES
519 002644 017746 176336 MOV @TKDBR,-(SP)                       :SAVE KEYBOARD CHAR
520 002650 042716 000200 BIC #BIT7,(SP)           :CLEAR PARITY BIT

```


CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 13

CZDVDD.P11 02-JUN-80 09:32

GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0024

```

521 002654 122726 000007      CMPB    #7,(SP)+      ;WAS IT CNTRL 'G' ?
522 002660 001002      BNE      .+6      ;BR IF NO.
523 002662 004737 004640      JSR      PC,SERV.G      ;SERVICE 'CNTRL 'G''.
524 002666 005037 001234      CLR      LSTERR      ;CLEAR LAST ERROR PC.
525 002672 010016      MOV      R0,(SP)      ;SAVE R0 ON THE STACK
526 002674 032777 040000 176300  BIT      #BIT14,@SWR      ;'LOOP ON THIS TEST'?
527 002702 001407      TTST:  BEQ      1$      ;BR IF NO. (IF LOCK SW01=1; THIS LOC =240)
528 002704 000437      BR      3$      ;GOTO 3$ (IF LOCK SW01=1; THIS LOC =240)
529 002706 105777 176272      TSTB    @TKCSR      ;KEYBOARD DONE?
530 002712 100034      BPL      3$      ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
531 002714 017700 176266      MOV      @TKDBR,R0      ;CLEAR DONE BIT
532 002720 000415      BR      2$      ;CONTINUE
533 002722 032777 004000 176252 1$:  BIT      #SW11,@SWR      ;DELETE ITERATION? (QUICK PASS)
534 002730 001011      BNE      2$      ;BR IF YES
535 002732 105737 001313      TSTB    QV.FLG      ;HAVE PASSES BEECOMPLETED?
536 002736 001406      BEQ      2$      ;BR IF QUICK PASS.
537 002740 005237 001224      INC      LPCNT      ;UPDATE ITERATION COUNTER
538 002744 023737 001224 001222      CMP      LPCNT,ICOUNT      ;ARE ALL ITERATIONS DONE??
539 002752 001014      BNE      3$      ;BR IF NOT YET
540 002754 105037 001311 2$:  CLRB    ERRFLG      ;PREPARE FOR NEW TEST
541 002760 005037 001224      CLR      LPCNT      ;START ICOUNTER AT 0
542 002764 005037 001220      CLR      LOCK      ;
543 002770 012737 000024 001222      MOV      #20.,ICOUNT      ;RESET ITERATIONS
544 002776 013737 001216 001214      MOV      NEXT,RETURN      ;GET NEXT TEST
545 003004 011600 3$:  MOV      (SP),R0      ;POP R0 OFF OF THE STACK
546 003006 022626      POP2SP      ;FAKE AN 'RTI'
547 003010 000177 176200      JMP      @RETURN      ;GO DO THE TEST
548 003014 001407      BRW:  1407
549 003016 000437      BRX:  437

550
551      ;CHECK FOR FREEZE ON CURRENT DATA
552      -----
553
554 003020 032777 001000 176154 .SCOP1: BIT      #SW09,@SWR      ;IS SW09=1(SET)?
555 003026 001405      BEQ      1$      ;BR IF NOT SET.
556 003030 005737 001220      TST      LOCK
557 003034 001402      BEQ      1$
558 003036 013716 001220      MOV      LOCK,(SP)      ;GOTO THE ADDRESS IN LOCK.
559 003042 000002 1$:  RTI      ;GO BACK.

560
561      ;TELETYPE OUTPUT ROUTINE
562      -----
563
564 003044 010546 .TYPE:  MOV      R5,-(SP)      ;SAVE R5 ON THE STACK.
565 003046 017605      MOV      @2(SP),R5      ;GET ADDRESS OF MESSAGE.
566 003052 062766 000002 000002      ADD      #2,2(SP)      ;POP OVER ADDRESS.
567 003060 032777 010000 176114 1$:  BIT      #SW12,@SWR      ;INHIBIT ALL PRINT OUT??
568 003066 001012      BNE      3$      ;BR IF NO PRINT OUT WANTED (SW12=1)
569 003070 105715      TSTB    (R5)      ;IS NUMBER MINUS? (MSB=1(BIT7))
570 003072 100002      BPL      2$      ;BR IF NUMBER IS PLUS
571 003074 104402 005104      TYPE    ,MCRLF      ;TYPE A CR/LF'
572 003100 105777 176104 2$:  TSTB    @TPCSR      ;TTY READY?
573 003104 100375      BPL      2$      ;BR IF NO.
574 003106 112577 176100      MOVB    (R5)+,@TPDBR      ;PRINT CURRENT CHAR.
575 003112 001362      BNE      1$      ;IF NOT ZERO KEEP PRINTING!
576 003114 012605 3$:  MOV      (SP)+,R5      ;END OF OUTPUT. RESTORE R5

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 14
 CZDVDD.P11 02-JUN-80 09:32 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0025

```

577 003116 000002          RTI          ;GO HOME
578          ;-----
579
580 003120 010346          .INSTR: MOV    R3,-(SP)          ;SAVE R3 ON STACK
581 003122 010446          MOV    R4,-(SP)          ;SAVE R4 ON STACK
582 003124 017637 0C0004 003142  MOV    @4(SP),.MSG
583 003132 062766 000002 000004  ADD    #2,4(SP)
584 003140 104402          .INST1: TYPE
585 003142 000000          .MSG: 0
586 003144 012704 005516          MOV    #INBUF,R4
587 003150 012703 000007          MOV    #7,R3
588 003154 105777 176024          1$:  TSTB   @TKCSR
589 003160 100375          BPL     1$
590 003162 117714 176020          MOVB   @TKDBR,(R4)
591 003166 142714 000200          BICB   #200,(R4)
592 003172 122427 000015          CMPB   (R4)+,#15
593 003176 001417          BEQ     INSTR2
594 003200 105777 176004          2$:  TSTB   @TPCSR
595 003204 100375          BPL     2$
596 003206 017777 175774 175776  MOVB   @TKDBR,@TPDBR
597 003214 005303          DEC     R3
598 003216 001356          BNE     1$
599 003220 012604          MOV     (SP)+,R4
600 003222 012603          MOV     (SP)+,R3
601 003224 104402 005100          .INSTE: TYPE ,MQM
602 003230 010346          MOV     R3,-(SP)
603 003232 010446          MOV     R4,-(SP)
604 003234 000741          BR     .INST1
605 003236 012604          INSTR2: MOV    (SP)+,R4          ;RESTORE R4
606 003240 012603          MOV     (SP)+,R3          ;RESTORE R3
607 003242 000002          RTI
608
609          ;CONVERT ASCII STRING TO OCTAL
610          ;-----
611
612 003244 010546          .PARAM: MOV    R5,-(SP)
613 003246 010446          MOV    R4,-(SP)
614 003250 016605 000004          MOV    4(SP),R5
615 003254 012537 003434          MOV    (R5)+,LOLIM
616 003260 012537 003436          MOV    (R5)+,HILIM
617 003264 012537 003440          MOV    (R5)+,DEVADR
618 003270 112537 003442          MOVB   (R5)+,LOBITS
619 003274 112537 003443          MOVB   (R5)+,ADRCNT
620 003300 010566 000004          MOV    R5,4(SP)
621 003304 005005          PARAM1: CLR    R5
622 003306 012704 005516          MOV    #INBUF,R4
623 003312 122714 000015          CMPB   #15,(R4)
624 003316 001420          BEQ     PARERR
625 003320 121427 000060          1$:  CMPB   (R4),#60
626 003324 002415          BLT     PARERR
627 003326 121427 000067          CMPB   (R4),#67
628 003332 003012          BGT     PARERR
629 003334 142714 000060          BICB   #60,(R4)
630 003340 152405          BISB   (R4)+,R5
631 003342 122714 000015          CMPB   #15,(R4)
632 003346 001406          BEQ     LIMITS

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 15
 CZDVDD.P11 02-JUN-80 09:32 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0026

```

633 003350 006305      ASL      R5
634 003352 006305      ASL      R5
635 003354 006305      ASL      R5
636 003356 000760      BR        1$
637 003360 104404      PARERR: INSTER
638 003362 000750      BR        PARAM1
639
640                      ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
641                      ;-----
642
643 003364 020537 003436  LIMITS: CMP      R5,HILIM
644 003370 101373      BHI      PARERR
645 003372 020537 003434  CMP      R5,LOLIM
646 003376 103770      BLO      PARERR
647 003400 133705 003442  BITB     LOBITS,R5
648 003404 001365      BNE      PARERR
649
650                      ;STORE NUMBER AT SPECIFIED ADDRESS
651
652 003406 013704 003440  1$:      MOV      DEVADR,R4
653 003412 010524      MOV      R5,(R4)+
654 003414 062705 000002  ADD      #2,R5
655 003420 105337 003443  DECB     ADCNT
656 003424 001372      BNE      1$
657 003426 012604      MOV      (SP)+,R4
658 003430 012605      MOV      (SP)+,R5
659 003432 000002      RTI
660 003434 000000      LOLIM: 0
661 003436 000000      HILIM: 0
662 003440 000000      DEVADR: 0
663 003442 000000      LOBITS: 0
664                      ADCNT=LOBITS+1
665
666                      ;SAVE PC OF TEST THAT FAILED AND R0-R5
667                      ;-----
668
669 003444 016637 000004 001276 .SAV05: MOV      4(SP),SAVPC      ;SAVE R7 (PC)
670
671                      ;SAVE R0-R5
672
673 003452 010537 001272  SAV05:  MOV      R5,SAVR5      ;SAVE R5
674 003456 010437 001270      MOV      R4,SAVR4      ;SAVE R4
675 003462 010337 001266      MOV      R3,SAVR3      ;SAVE R3
676 003466 010237 001264      MOV      R2,SAVR2      ;SAVE R2
677 003472 010137 001262      MOV      R1,SAVR1      ;SAVE R1
678 003476 010037 001260      MOV      R0,SAVR0      ;SAVE R0
679 003502 000002      RTI                      ;LEAVE.
680
681                      ;RESTORE R0-R5
682
683 003504 013700 001260  .RES05:  MOV      SAVR0,R0      ;RESTORE R0
684 003510 013701 001262      MOV      SAVR1,R1      ;RESTORE R1
685 003514 013702 001264      MOV      SAVR2,R2      ;RESTORE R2
686 003520 013703 001266      MOV      SAVR3,R3      ;RESTORE R3
687 003524 013704 001270      MOV      SAVR4,R4      ;RESTORE R4
688 003530 013705 001272      MOV      SAVR5,R5      ;RESTORE R5

```

```

689 003534 000002 RTI ;LEAVE
690
691 ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
692 -----
693
694 003536 104402 005104 .CONVR: TYPE ,MCRLF
695 003542 010046 .CNVRT: MOV R0,-(SP)
696 003544 010146 MOV R1,-(SP)
697 003546 010346 MOV R3,-(SP)
698 003550 010446 MOV R4,-(SP)
699 003552 010546 MOV R5,-(SP)
700 003554 017601 000012 MOV @12(SP),R1
701 003560 062766 000002 C00012 ADD #2,12(SP)
702 003566 012137 003742 MOV (R1)+,WRDCNT
703 003572 112137 003744 18: MOVB (R1)+,CHRCNT
704 003576 112137 003745 MOVB (R1)+,SPACNT
705 003602 013137 003746 MOV @ (R1)+,BINWRD
706 003606 013704 003746 28: MOV BINWRD,R4
707 003612 113705 003744 MOVB CHRCNT,R5
708 003616 012700 005560 MOV #TEMP,R0
709 003622 010403 38: MOV R4,R3
710 003624 042703 177770 BIC #177770,R3
711 003630 062703 000060 ADD #060,R3
712 003634 110320 MOVB R3,(R0)+
713 003636 000241 CLC
714 003640 006004 ROR R4
715 003642 000241 CLC
716 003644 006004 ROR R4
717 003646 000241 CLC
718 003650 006004 ROR R4
719 003652 005305 DEC R5
720 003654 001362 BNE 38
721 003656 012703 005622 MOV #MDATA,R3
722 003662 114023 48: MOVB -(R0),(R3)+
723 003664 105337 003744 DECB CHRCNT
724 003670 001374 BNE 48
725 003672 105737 003745 TSTB SPACNT
726 003676 001405 BEQ 68
727 003700 112723 000040 58: MOVB #040,(R3)+
728 003704 105337 003745 DECB SPACNT
729 003710 001373 BNE 58
730 003712 105013 68: CLRB (R3)
731 003714 104402 005622 TYPE ,MDATA
732 003720 005337 003742 DEC WRDCNT
733 003724 001322 BNE 18
734 003726 012605 MOV (SP)+,R5
735 003730 012604 MOV (SP)+,R4
736 003732 012603 MOV (SP)+,R3
737 003734 012601 MOV (SP)+,R1
738 003736 012600 MOV (SP)+,R0
739 003740 000002 RTI
740 003742 000000 WRDCNT: 0
741 003744 000000 CHRCNT: 0
742 003745 003745 SPACNT=CHRCNT+1
743 003746 000000 BINWRD: 0
744
    
```

```

745
746           ;TRAP DISPATCH SERVICE
747           ;ARGUMENT OF TRAP IS EXTRACTED
748           ;AND USED AS OFFSET TO OBTAIN POINTER
749           ;TO SELECTED SUBROUTINE
750
751 003750 011646 .TRPSR: MOV      (SP),-(SP)           ;GET PC OF RETURN
752 003752 162716 000002 SUB      #2,(SP)           ;=PC OF TRAP
753 003756 017616 000000 MOV      @ (SP),(SP)         ;GET TRP
754 003762 006316 TPPOK: ASL      (SP)           ;MULTIPLY TRAP ARG BY 2
755 003764 042716 177001 BIC      #177001,(SP)        ;CLEAR UNWANTED BITS
756 003770 062716 001314 ADD      #.TRPTAB,(SP)       ;POINTER TO SUBROUTINE ADDRESS
757 003774 017616 000000 MOV      @ (SP),(SP)       ;SUBROUTINE ADDRESS
758 004000 000136 JMP      @ (SP)+          ;GO TO SUBROUTINE
759
760           ;ERROR HANDLER
761           ;-----
762
763 004002 .HLT:
764 004002 022737 177570 001202 CMP      #177570,SWR           ;IS THERE A REAL SWR?
765 004010 001411 BEQ      64$              ;BR IF YES
766 004012 017746 175170 MOV      @TKDBR,-(SP)        ;SAVE KEYBOARD CHAR
767 004016 042716 000200 BIC      #BIT7,(SP)         ;CLEAR PARITY BIT
768 004022 122726 000007 CMPB     #7,(SP)+          ;WAS IT CNTRL 'G' ?
769 004026 001002 BNE      .+6              ;BR IF NO.
770 004030 004737 004640 JSR      PC,SERV.G          ;SERVICE 'CNTRL 'G''.
771 004034 032777 010000 175140 64$: BIT      #SW12,@SWR          ;BELL ON ERROR?
772 004042 001406 BEQ      XB$              ;BR IF NO BELL
773 004044 105777 175140 TSTB     @TPCSR           ;TTY READY.
774 004050 100003 BPL      XB$              ;DON'T WAIT IF TTY NOT READY.
775 004052 112777 000207 175132 MOVB     #207,@TPDBR        ;PUSH A BELL AT THE TTY.
776 004060 032777 020000 175114 XB$: BIT      #SW13,@SWR          ;DELETE ERROR PRINT OUT?
777 004066 001105 BNE      HALTS              ;BR IF NO PRINT OUT WANTED.
778 004070 021637 001234 CMP      (SP),LSTERR        ;WAS THIS ERROR FOUND LAST TIME?
779 004074 001404 BEQ      1$              ;BR IF YES
780 004076 011637 001234 MOV      (SP),LSTERR        ;RECORD BEING HERE
781 004102 105037 001311 CLRB      ERRFLG          ;PREPARE HEADER
782 004106 104406 1$: SAVO5          ;SAVE ALL PROC REGISTERS
783 004110 011605 MOV      (SP),R5              ;GET THE PC OF ERROR
784 004112 162705 000002 SUB      #2,R5              ;GET ADDRESS OF TRAP CALL
785 004116 011504 MOV      (R5),R4              ;GET HLT INSTRUCTION
786 004120 006304 ASL      R4              ;MULT BY TWO
787 004122 061504 ADD      (R5),R4              ;DOUBLE IT
788 004124 006304 ASL      R4              ;MULT AGAIN
789 004126 042704 177001 BIC      #177001,R4          ;CLEAR JUNK
790 004132 062704 037360 ADD      #.ERRTAB,R4          ;GET POINTER
791 004136 012437 004252 MOV      (R4)+,ERRMSG        ;GET ERROR MESSAGE
792 004142 012437 004264 MOV      (R4)+,DATAHD        ;GET DATA HEADER
793 004146 011437 004276 MOV      (R4),DATABP        ;GET DATA TABLE
794 004152 105737 001311 TSTB     ERRFLG          ;TYPE HEADREER
795 004156 001403 BEQ      TYPMSG          ;BR IF YES
796 004160 005737 004276 TST      DATABP          ;DOES DATA TABLE EXIST?
797 004164 001040 BNE      TYPDAT          ;BR IF YES.
798 004166 104402 005104 TYPMSG: TYPE      ,MCRLF
799 004172 104402 005104 TYPE      ,MCRLF
800 004176 005737 001220 TST      LOCK

```

```

801 004202 001402 BEQ 1$
802 004204 104402 005376 TYPE ,MASTEK
803 004210 104402 005364 1$: TYPE ,MTSTN
804 004214 104411 004374 CNVRT ,XTSTN ;SHOW IT
805 004220 104402 005452 TYPE ,MERRPC ;TYPE PC.
806 004224 104411 004366 CNVRT ,ERTABO ;SHOW IT
807 004230 104402 005104 TYPE ,MCRLF ;GIVE A CR/LF
808 004234 112737 177777 001311 MOVB #-1,ERRFLG ;NO MORE HEADER UNLESS NO DATA TABLE.
809 004242 005737 004252 TST ERRMSG ;IS THERE AN ERROR MESSAGE?
810 004246 001402 BEQ WRKO.FM ;BR IF NO.
811 004250 104402 TYPE ;TYPE
812 004252 000000 ERRMSG: 0 ; ERROR MESSAGE
813 004254 WRKO.FM: ;
814 004254 005737 004264 TST DATAHD ;DATA HEADER?
815 004260 001402 BEQ TYPDAT ;BR IF NO
816 004262 104402 TYPE ;TYPE
817 004264 000000 DATAHD: 0 ; DATA HEADER
818 004266 005737 004276 TYPDAT: TST DATABP ;DATA TABLE?
819 004272 001402 BEQ RESREG ;BR IF NO.
820 004274 104410 CNVRT ;SHOW
821 004276 000000 DATABP: 0 ; DATA TABLE
822 004300 104407 RESREG: RES05 ;RESTORE PROC REGISTERS
823 004302 005777 174674 HALTS: TST @SWR ;HALT ON ERROR?
824 004306 100005 BPL EXITER ;BR IF NO HALT ON ERROR
825 004310 010046 PUSHRO ;SAVE RO
826 004312 016600 000002 MOV 2(SP),RO ;SHOW ERROR PC IN DATA LIGHTS
827 004316 000000 HALT ;HALT
828 004320 012600 POPRO ;GET RO
829 004322 005237 001232 EXITER: INC ERRCNT ;UPDATE ERROR COUNT
830 004326 032777 000400 174646 BIT #SW08,@SWR ;GOTO TOP OF TEST?
831 004334 001007 BNE 1$ ;BR IF YES
832 004336 032777 002000 174636 BIT #SW10,@SWR ;GOTO NEXT TEST?
833 004344 001407 BEQ 2$ ;BR IF NO
834 004346 013737 001216 001214 MOV NEXT,RETURN ;SET FOR NEXT TEST
835 004354 012706 001200 1$: MOV #STACK,SP ;RESET SP
836 004360 000177 174630 JMP @RETURN ;GOTO SPECIFIED TEST
837 004364 000002 2$: RTI ;RETURN
838 004366 000001 ERTABO: 1
839 004370 006 002 .BYTE 6,2
840 004372 001276 SAVPC
841 004374 000001 XTSTN: 1
842 004376 003 002 .BYTE 3,2
843 004400 001226 TSTNO
844 ;ENTER HERE ON POWER FAILURE
845 ;-----
846
847
848 004402 .PFAIL:
849 004402 012737 004414 000024 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
850 004410 000000 HALT ;HALT ON POWER DOWN NORMAL
851 004412 000777 BR .
852
853 ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
854
855 004414 RESTAR:
856 004414 012737 004402 000024 MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE

```

CZDVN-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 19
 CZDVDD.P11 02-JUN-80 09:32

GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0030

```

857 004422 012706 001200      MOV      #STACK,SP      ;RESET THE STACK POINTER
858 004426 005037 005560      CLR      TEMP          ;READY FOR TIMER
859 004432 005237 005560      INC      TEMP          ;PLUS ONE TO THE TIMER!
860 004436 001375              BNE      .-4              ;BR IF MORE TO GO
861 004440 104402 005107      TYPE      ,MPFAIL        ;TYPE THE MESSAGE
862 004444 104411 004470      CNVRT     ,P,TAB          ;TELL WHAT TEST TO RETURN TO.
863 004450 105037 001311      CLRB     ERRFLG        ;START CLEAN
864 004454 005037 001234      CLR      LSTERR        ;*****
865 004460 104412              MSTCLR          ;START CLEAN UP OF DEVICE
866 004462 104413              RAMCLR          ;CLEAR IT ALL!
867 004464 000177 174524      JMP      @RETURN        ;START DOING THAT TEST AGAIN.
868 004470 000001              FCTAB: 1
869 004472      003          .BYTE 3,2
870 004474 001226              .DELAY: TSTNO
871 004476 010046              MOV      R0,-(SP)
872 004500 013700 004514      MOV      1$,R0
873 004504 005300              DEC      R0
874 004506 001376              BNE      .-2
875 004510 012600              MOV      (SP)+,R0
876 004512 000002              RTI
877 004514 000036              1$: 30.
878
879 004516              .RAMCLR:
880 004516 012777 004000 174636      MOV      #MRESET,@DVSCR ;ISSUE A MASTER CLEAR
881 004524 010146              MOV      R1,-(SP) ;SAVE R1 ON THE STACK
882 004526 010446              MOV      R4,-(SP) ;SAVE R4 ON THE STACK
883 004530 013701 001372              MOV      DVSRS,R1 ;GET SECONDARY SEL. REG.
884 004534 013704 001376              MOV      DVSRA,R4 ;GET SECONDARY REGISTER ACCESS REG.
885 004540 005014              1$: CLR      (R4) ;ZERO THE SECONDARY REGISTER.
886 004542 062711 170361              ADD      #^C<BIT11+BIT10+BIT9+BIT8+BIT3+BIT2+BIT1+BIT0>+BIT0,(R1)
887 004546 001374              BNE      1$
888 004550 012604              MOV      (SP)+,R4 ;RESTORE R4
889 004552 012601              MOV      (SP)+,R1 ;RESTORE R1
890 004554 000002              RTI
891
892 004556              .MSTCLR:
893 004556 012777 004000 174576      MOV      #MRESET,@DVSCR ;ISSUE MASTER CLEAR.
894 004564 000002              RTI
895
896 004566              .ROMCLK:
897 004566 052777 000002 174566      BIS      #BIT1,@DVSCR
898 004574 000002              RTI
899
900 004576              .DATACLK:
901 004576 010046              MOV      R0,-(SP)
902 004600 005000              CLR      R0
903 004602 052777 000400 174560      BIS      #BIT8,@DVLCR
904 004610 017737 174554 004636      1$: MOV      @DVLCR,3$
905 004616 106037 004637              RORB     3$,1
906 004622 103003              BCC      2$
907 004624 005200              INC      R0
908 004626 001370              BNE      1$
909 004630 104000              HLT      0
910 004632 012600              2$: MOV      (SP)+,R0
911 004634 000002              RTI
912 004636 000001              3$: .BLKW 1

```


CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 20
 CZDVDD.P11 02-JUN-80 09:32 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0031

```

913
914 004640 032777 004000 174336 SERV.G: BIT #4000,@TKCSR ;RX BUSY?
915 004646 001374 BNE SERV.G ;BR IF YES
916 004650 017737 174326 005072 MOV @SWR,90$ ;SAVE (SWR).
917 004656 013777 005072 174316 1$: MOV 90$,@SWR ;
918 004664 104402 005052 TYPE ,89$ ;
919 004670 104411 005064 CNVRT ,88$ ;
920 004674 104402 005074 TYPE ,91$ ;
921 004700 105777 174300 TSTB @TKCSR ;WAIT FOR DONE.
922 004704 100375 BPL , -4 ;
923 004706 017746 174274 MOV @TKDBR,-(SP) ;
924 004712 042716 000200 BIC #BIT7,(SP) ;
925 004716 122726 000015 CMPB #15,(SP)+ ;
926 004722 001450 BEQ 5$ ;
927 004724 005077 174252 CLR @SWR ;
928 004730 105777 174254 2$: TSTB @TPCSR ;
929 004734 100375 BPL , -4 ;
930 004736 016677 177776 174246 MOV -2(SP),@TPDBR ;
931 004744 000241 CLC ;
932 004746 006177 174230 ROL @SWR ;
933 004752 006177 174224 ROL @SWR ;
934 004756 006177 174220 ROL @SWR ;
935 004762 103735 BCS 1$ ;ERROR
936 004764 026627 177776 000060 CMP -2(SP),#60 ;
937 004772 002731 BLT 1$ ;
938 004774 026627 177776 000067 CMP -2(SP),#67 ;
939 005002 003325 BGT 1$ ;
940 005004 042766 177770 177776 BIC #^C<7>,-2(SP) ;
941 005012 056677 177776 174162 BIS -2(SP),@SWR ;
942 005020 105777 174160 TSTB @TKCSR ;
943 005024 100375 BPL , -4 ;
944 005026 017746 174154 MOV @TKDBR,-(SP) ;
945 005032 042716 000200 BIC #BIT7,(SP) ;
946 005036 122726 000015 CMPB #15,(SP)+ ;
947 005042 001332 BNE 2$ ;
948 005044 104402 005104 5$: TYPE ,MCRLF ;
949 005050 000207 RTS PC ;
950
951 005052 020377 051450 051127 89$: .ASCIIZ <377>? (SWR)=/?
952 005060 036451 000057
953 .EVEN
954 005064 000001 88$: 1
955 005066 006 000 .BYTE 6,0
956 005070 005072 90$: .WORD 0
957 005072 000000 91$: .ASCIIZ ?/=/?
958 005074 036457 000057 .EVEN
959
960 005100 020040 000077 MQM: .ASCIIZ / ?/
(2) 005104 005015 000 MCRLF: .ASCIIZ <15><12>
(2) 005107 377 053520 020122 MPFAIL: .ASCIIZ <377>/PWR FAILED. RESTART AT TEST /
(2) 005145 377 047105 020104 MEPASS: .ASCIIZ <377>/END PASS CZDVDD /
(2) 005167 377 000122 MR: .ASCIIZ <377>/R/
(2) 005172 050377 047522 051107 MERR2: .ASCIIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 005241 377 047111 052523 MERR3: .ASCIIZ <377>/INSUFFICIENT DATA!/
(2) 005265 377 042524 052123 MTSTPC: .ASCIIZ <377>/TEST PC-/
(2) 005277 377 047514 045503 MLOCK: .ASCIIZ <377>/LOCK ON SELECTED TEST/

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 21

CZDVDD.P11 02-JUN-80 09:32

GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0032

```

(2) 005326 051503 035122 000040 MCSR: .ASCIIZ /CSR: /
(2) 005334 042526 035103 000040 MVEC: .ASCIIZ /VEC: /
(2) 005342 040520 051523 051505 MPASS: .ASCIIZ /PASSES: /
(2) 005353 105 051122 051117 MERR: .ASCIIZ /ERRORS: /
(2) 005364 042524 052123 047040 MTSTN: .ASCIIZ /TEST NO: /
(2) 005376 000052 MASTE: .ASCIIZ /*/
(2) 005400 051777 052105 051440 MNEW: .ASCIIZ <377>/SET SWITCH REG TO DV11'S DESIRED ACTIVE./
(2) 005452 041520 020072 000 MERRPC: .ASCIIZ /PC: /
(2) 005457 377 040515 020120 XHEAD: .ASCIIZ <377>/MAP OF DV11 STATUS/<377>
(2) .EVEN
(2) 005504 000002 XSTATQ: 2
961 005506 006 003 .BYTE 6,3
962 005510 001246 TEMP1
963 005512 006 002 .BYTE 6,2
964 005514 001250 TEMP2
965 .EVEN
966 ;BUFFERS FOR INPUT-OUTPUT
967
968
969 005516 000000 INBUF: 0
970 005560 .=. +40
971 005560 000000 TEMP: 0
972 005622 005622 .=. +40
973 005622 000000 MDATA: 0
974 005664 .=. +40

```

```

975
976
977      ;ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DV11'S
978      ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
979      ;AND RUNS THE SPECIFIED DV11'S.  THIS ROUTINE *MUST*
980      ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
981      ;SETUP NECESSARY.
982
983
984 005664 105737 001300      CYCLE:  TSTB    DVACTV    ;ARE ANY DV11'S TO BE TESTED?
985 005670 001004          BNE      1$          ;BR IF OK.
986 005672 104402 005172    TYPE    ,MERR2      ;NO DV11'S SELECTED!!
987 005676 000000          HALT                    ;STOP THE SHOW.
988 005700 000776          BR      -2            ;DISQUALIFY CONT. SW.
989 005702 133737 001304 001300 1$:  BITB    RUN,DVACTV ;IS THIS ONE 'ACTIVE'
990 005710 001020          BNE      2$          ;BR IF GOOD ONE FOUND.
991 005712 000241          CLC                    ;CLEAR PROC. CARRY BIT.
992 005714 106137 001304    ROLB    RUN          ;UPDATE POINTER
993 005720 105537 001304    ADCB    RUN          ;CATCH CARRY FROM RUN
994 005724 062737 000024 001306    ADD    #24,CREAM ;UPDATE ADDRESS POINTER.
995 005732 022737 001740 001306    CMP    #DV.END,CREAM
996 005740 001360          BNE      1$          ;KEEP GOING; NOT ALL TESTED FOR.
997 005742 012737 001500 001306    MOV    #DV.MAP,CREAM ;RESET ADDRESS POINTER.
998 005750 000754          BR      1$          ;KEEP LOOKING FOR ACTIVE DV11
999 005752 000241          2$:  CLC                    ;CLEAR PROC. CARRY.
1000 005754 106137 001304    ROLB    RUN          ;UPDATE POINTER.
1001 005760 105537 001304    ADCB    RUN          ;CATCH CARRY.
1002 005764 013700 001306    MOV    CREAM,RO      ;GET ADDRESS POINTER.
1003 005770 062737 000024 001306    ADD    #24,CREAM ;UPDATE.
1004 005776 022737 001740 001306    CMP    #DV.END,CREAM
1005
1006 006004 001003          BNE      3$          ;ALL DONE?
1007 006006 012737 001500 001306    MOV    #DV.MAP,CREAM ;BR IF NO.
1008 006014 012037 001362          3$:  MOV    (RO)+,DVSCR ;RESTORE POINTER.
1009 006020 012037 001352          MOV    (RO)+,DVRVEC ;LOAD SYSTEM CTRL. REG
1010 006024 012037 001422          MOV    (RO)+,LO0.03 ;LOAD VECTOR
1011 006030 012037 001432          MOV    (RO)+,SYNC2A ;GET LINE PARAMETERS. 00-03
1012 006034 012037 001424          MOV    (RO)+,LO4.07 ;
1013 006040 012037 001434          MOV    (RO)+,SYNC2B ;
1014 006044 012037 001426          MOV    (RO)+,LO8.11 ;
1015 006050 012037 001436          MOV    (RO)+,SYNC2C ;
1016 006054 012037 001430          MOV    (RO)+,L12.15 ;
1017 006060 012037 001440          MOV    (RO)+,SYNC2D ;
1018 006064 012700 000002          MOV    #2,RO      ;SAVE CORE THIS WAY!
1019 006070 013737 001362 001364    MOV    DVSCR,DVSCRH ;GET SYS CTRL. REG HIGH BYTE.
1020 006076 005237 001364          INC    DVSCRH ;GOT IT.
1021 006102 013737 001364 001366    MOV    DVSCRH,DVRIC ;GET NXT REC. CHAR REG.
1022 006110 005237 001366          INC    DVRIC ;GOT IT
1023 006114 013737 001366 001370    MOV    DVRIC,DVLCR ;GET LN. PAR.REG.
1024 006122 060037 001370          ADD    RO,DVLCR ;GOT IT
1025 006126 013737 001370 001372    MOV    DVLCR,DVSRS ;GET SEC. REG. SEL. REG.
1026 006134 060037 001372          ADD    RO,DVSRS ;GOT IT
1027 006140 013737 001372 001374    MOV    DVSRS,DVSRSH ;GET HIGH BYTE.
1028 006146 005237 001374          INC    DVSRSH ;GOT IT
1029 006152 013737 001374 001376    MOV    DVSRSH,DVSRA ;SEC. REG. ACCESS.
1030 006160 005237 001376          INC    DVSRA ;GOT IT

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 23
 CZDVDD.P11 02-JUN-80 09:32 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0034

1031	006164	013737	001376	001400	MOV	DVSRA,DVSFR	;SPEC. FUN. REG.
1032	006172	060037	001400		ADD	RO,DVSFR	;
1033	006176	013737	001400	001402	MOV	DVSFR,DVNSR	;NPR STAT. REG.
1034	006204	060037	001402		ADD	RO,DVNSR	;
1035	006210	013737	001402	001404	MOV	DVNSR,RESV16	;RESERVED REG
1036	006216	060037	001404		ADD	RO,RESV16	;
1037							
1038	006222	013737	001352	001354	MOV	DVRVEC,DVRLVL	;PTY LVL
1039	006230	060037	001354		ADD	RO,DVRLVL	;
1040	006234	013737	001354	001356	MOV	DVFLVL,DVTEC	;TX VEC
1041	006242	060037	001356		ADD	RO,DVTEC	;
1042	006246	013737	001356	001360	MOV	DVTEC,DVTLVL	;TX LVL
1043	006254	060037	001360		ADD	RO,DVTLVL	;
1044							
1045	006260	012700	001422		MOV	#L00.03,RO	;LOAD STAU 00-03
1046	006264	012701	001406		MOV	#MASK.A,R1	;PREPARE MASK.
1047	006270	012702	001416		MOV	#CLK.A,R2	;PREPARE CLOCKS
1048	006274	004737	006514		JSR	PC,FIX.00	;GO AND CALCULATE CONFIGURATION.
1049							
1050	006300	012700	001424		MOV	#L04.07,RO	;LOAD STAU 00-03
1051	006304	012701	001410		MOV	#MASK.B,R1	;PREPARE MASK.
1052	006310	012702	001417		MOV	#CLK.B,R2	;PREPARE CLOCKS
1053	006314	004737	006514		JSR	PC,FIX.00	;GO AND CALCULATE CONFIGURATION.
1054							
1055	006320	012700	001426		MOV	#L08.11,RO	;LOAD STAU 00-03
1056	006324	012701	001412		MOV	#MASK.C,R1	;PREPARE MASK.
1057	006330	012702	001420		MOV	#CLK.C,R2	;PREPARE CLOCKS
1058	006334	004737	006514		JSR	PC,FIX.00	;GO AND CALCULATE CONFIGURATION.
1059							
1060	006340	012700	001430		MOV	#L12.15,RO	;LOAD STAU 00-03
1061	006344	012701	001414		MOV	#MASK.D,R1	;PREPARE MASK.
1062	006350	012702	001421		MOV	#CLK.D,R2	;PREPARE CLOCKS
1063	006354	004737	006514		JSR	PC,FIX.00	;GO AND CALCULATE CONFIGURATION.
1064	006360	032777	000002	172614	BIT	#SW01,2SWR	
1065	006366	001445			BEQ	7\$	
1066	006370						
1067	006370	005737	000042		TST	@#42	
1068	006374	001042			BNE	7\$	
1069	006376	104402	005104		TYPE	,MCRLF	
1070	006402	104403			INSTR		
1071	006404	005364			MTSTN		
1072	006406	104405			PARAM		
1073	006410	000001			1		
1074	006412	001000			1000		
1075	006414	001226			TSTNO		
1076	006416	000			0		
1077	006417	001			1		
1078	006420	012700	007256		MOV	#TST1,RO	
1079	006424	022710			CMP	(PC)+,(RO)	
1080	006426	012737			MOV	(PC)+,@(PC)+	
1081	006430	001015			BNE	6\$	
1082	006432	023760	001226	000002	CMP	TSTNO,2(RO)	
1083	006440	001011			BNE	6\$	
1084	006442	022760	001226	000004	CMP	#TSTNO,4(RO)	
1085	006450	001005			BNE	6\$	
1086	006452	010037	001214		MOV	RO,RETURN	

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 24
 CZDVDD.P11 02-JUN-80 09:32 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0035

```

1087 006456 104402 005104          TYPE      MCRLF
1088 006462 000412          BR          8$
1089 006464 005720          6$: TST      (R0)+
1090 006466 020027 022516      CMP      R0,#TLAST+10
1091 006472 001354          BNE      5$
1092 006474 104402 005100          TYPE      ,MQM
1093 006500 000733          BR          4$
1094 006502 012737 007256 001214 7$: MOV      #TST1,RETURN ;PREPARE RETURN ADDRESS
1095 006510 000177 172500      8$: JMP      @RETURN ;GO START TESTING.
1096
1097 006514 011003          FIX.00: MOV      (R0),R3 ;GET PARAMETERS.
1098 006516 042703 176377      BIC      #*C<1400>,R3 ;CLEAR JUNK.
1099 006522 005703          TST      R3 ;TEST FOR EIGHT BITS.
1100 006524 001005          BNE      1$ ;BR IF NOT 8 BITS.
1101 006526 012711 000400      MOV      #400,(R1) ;SET FOR 8 BITS PER CHAR
1102 006532 112712 000010      MOVB     #8.,(R2) ;
1103 006536 000424          BR          4$
1104 006540 022703 000400      1$: CMP      #400,R3 ;CHECK FOR SEVEN BITS.
1105 006544 001005          BNE      2$ ;BR IF NOT 7 BITS.
1106 006546 112711 000200      MOVB     #200,(R1) ;
1107 006552 112712 000007      MOVB     #7,(R2) ;
1108 006556 000414          BR          4$
1109 006560 022703 001000      2$: CMP      #1000,R3 ;CHECK FOR SIX BITS.
1110 006564 001005          BNE      3$ ;BR IF NOT SIX BITS.
1111 006566 112711 000300      MOVB     #300,(R1) ;
1112 006572 112712 000006      MOVB     #6,(R2) ;
1113 006576 000404          BR          4$
1114 006600 112711 000340      3$: MOVB     #340,(R1) ;IF NONE OF THE ABOVE; MUST BE 5 BITS.
1115 006604 112712 000005      MOVB     #5,(R2) ;
1116 006610 032710 040000      4$: BIT      #PARBIT,(R0) ;PARITY ENABLED?
1117 006614 001401          BEQ      5$ ;IF =0; THEN NO PARITY.
1118 006616 105212          INCB     (R2) ;PLUS ONE TO THE CLOCK!
1119 006620 000207          5$: RTS      PC ;
1120
1121          ;*ROUTINE USED TO "AUTO SIZE" THE DV11
1122          ;*CSR AND VECTOR.
1123          ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
1124          ;* ADDRESS RANGE (175000:175400)
1125          ;* AND THE VECTOR MAY BE ANY WHERE IN THE
1126          ;* FLOATING VECTOR RANGE (300:770)
1127          ;*
1128
1129          AUTO.SIZE:
1130          006622 000005          RESET
1131 006624 012702 001500      CSRMAP: MOV      #DV.MAP,R2 ;INSURE A BUS INIT.
1132 006630 005022          1$: CLR      (R2)+ ;LOAD MAP POINTER.
1133 006632 022702 001740      CMP      #DV.END,R2 ;ZERO ENTIRE MAP
1134 006636 001374          BNE      1$ ;ALL DONE?
1135 006640 105037 001301      CLRB     DVNUM ;BR IF NO
1136 006644 012702 001500      MOV      #DV.MAP,R2 ;SET OCTAL NUMBER OF DV11'S TO 0
1137 006650 012701 175000      MOV      #175000,R1 ;SET FOR FIRST ADDRESS TO BE TESTED
1138 006654 012737 007074 000004      MOV      #6$,@#4 ;SET FOR NON-EXISTANT DEVICE TIME OUT
1139 006662 005711          2$: TST      (R1) ;IF DV11 DVSCR S/B 0
1140 006664 001037          BNE      3$ ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO DV11
1141 006666 022761 177777 000012      CMP      #177777,12(R1) ;IF DV11 THEN DVSCR S/B ALL 1'S ON INIT!
1142 006674 001033          BNE      3$ ;BR IF NOT DV11

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 25
 CZDVDD.P11 02-JUN-80 09:32 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0036

```

1143 006676 005761 000016      TST      16(R1)      ;IF DV11 THEN RESV16 S/B ALL 0'S
1144 006702 001030      BNE      3$          ;BR IF NOT DV11
1145      ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DV11 CSR ADDRESS.
1146 006704 010122      MOV      R1,(R2)+      ;STORE CSR IN CORE TABLE.
1147 006706 005722      TST      (R2)+      ;POP OVER VECTOR STORE AREA
1148 006710 052722 000226      BIS      #226,(R2)+      ;SET LINE CARD 1 STAT AND SYNC
1149 006714 052722 000062      BIS      #62,(R2)+      ;
1150 006720 052722 000226      BIS      #226,(R2)+      ;SET LINE CARD 2 STAT AND SYNC
1151 006724 052722 000062      BIS      #62,(R2)+      ;
1152 006730 052722 000226      BIS      #226,(R2)+      ;SET LINE CARD 3 STAT AND SYNC
1153 006734 052722 000062      BIS      #62,(R2)+      ;
1154 006740 052722 000226      BIS      #226,(R2)+      ;SET LINE CARD 4 STAT AND SYNC
1155 006744 052722 000062      BIS      #62,(R2)+      ;
1156 006750 105237 001301      INCB     DVNUM      ;UPDATE DEVICE COUNTER
1157 006754 122737 000010 001301  CMPB     #10,DVNUM      ;ARE MAX. NO. OF DEV FOUND?
1158 006762 001405      BEQ      100$      ;YES DON'T LOOK FOR ANY MORE.
1159 006764 062701 000010      3$:      ADD      #10,R1      ;UPDATE CSR POINTER ADDRESS
1160 006770 022701 175400      CMP      #175400,R1
1161 006774 001332      BNE      2$          ;BR IF MORE ADDRESS TO CHECK.
1162 006776 012722 177777      100$:    MOV      #177777,(R2)+      ;TERMINATER.
1163 007002 105637 001300      CLRB     DVACTV
1164 007006 105737 001301      TSTB     DVNUM      ;WERE ANY DV11'S FOUND AT ALL?
1165 007012 001423      BEQ      5$          ;ERROR AUTO SIZER FOUND NO DV11'S IN THIS SYS.
1166 007014 113701 001301      MOVB     DVNUM,R1
1167 007020 110137 001303      MOVB     R1,SAVNUM      ;SAVE NUMBER OF DEVICES
1168 007024 000241      4$:      CLC
1169 007026 106137 001300      ROLB     DVACTV      ;GENERATE ACTIVE REGISTER OF DEVICES.
1170 007032 105237 001300      INCB     DVACTV      ;SET THE BIT
1171 007036 005301      DEC      R1
1172 007040 001371      BNE      4$          ;BR IF MORE TO GENERATE
1173 007042 012737 000006 000004  MOV      #6,#4      ;RESTORE TRAP VECTOR
1174 007050 113737 001300 001302  MOVB     DVACTV,SAVACT      ;SAVE ACTIVE REGISTER
1175 007056 000137 007102      JMP      VECMAP      ;GO FIND THE VECTOR NOW.
1176 007062 104402 005172      5$:      TYPE     ,MERR2      ;NOTIFY OPR THAT NO DV11'S FOUND.
1177 007066 005000      CLR      R0      ;MAKE DATA LIGHTS ZERO
1178 007070 000000      HALT
1179 007072 000776      BR      -2      ;STOP THE SHOW
1180 007074 012716 006764      6$:      MOV      #3$,(SP)      ;DISABLE CONT. SW.
1181 007100 000002      RTI      ;ENTERED BY NON-EXISTANT TIME-OUT.
1182      ;RETURN TO MAINSTREAM
1183 007102 012737 000340 000022  VECMAP: MOV      #340,#22      ;SET IOT TRAP PRIO TO 7
1184 007110 012737 007232 000020  MOV      #4$,#20      ;SET IOT TRAP VECTOR
1185 007116 012702 001500      MOV      #DV.MAP,R2      ;SET SOFTWARE POINTER
1186 007122 012700 000300      MOV      #300,R0      ;FLOATING VECTORS START HERE.
1187 007126 012701 000302      MOV      #302,R1      ;PC OF IOT INSTR.
1188 007132 010120      1$:      MOV      R1,(R0)+      ;START FILLING VECTOR AREA
1189 007134 012721 000004      MOV      #4,(R1)+      ;WITH .+2; IOT
1190 007140 022021      CMP      (R0)+,(R1)+      ;ADD 2 TO R0 +R1
1191 007142 020127 001000      CMP      R1,#1000
1192 007146 101771      BLOS     1$          ;BR IF MORE TO FILL
1193 007150 113737 001300 001246  MOVB     DVACTV,TEMP1      ;STORE TEMPORALLY
1194 007156 006037 001246      2$:      ROR      TEMP1      ;BRING OUT A BIT
1195 007162 103034      BCC      5$          ;BR IF ALL DONE
1196 007164 005037 177776      CLR      PS      ;ZERO CPU PRIO
1197 007170 012772 001300 000000  MOV      #BIT9+BIT7+BIT6,#(R2)
1198 007176 005000      CLR      R0      ;ATTEMPT TO FORCE AN INTERRUPT

```


CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 27
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0038

1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 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

007256 012737 000001 001226
 007264 012737 007664 J01216
 007272 012700 000000
 007276 013737 001422 001236
 007304 100402
 007306 004737 007374
 007312 012700 000004
 007316 013737 001424 001236
 007324 100402
 007326 004737 007374
 007332 012700 000010
 007336 013737 001426 001236
 007344 100402
 007346 004737 007374
 007352 012700 000014
 007356 013737 001430 001236
 007364 100402
 007366 004737 007374
 007372 104400
 007374
 007374 012737 007422 001220
 007402 104413
 007404 005003
 007406 005001
 007410 112737 000025 025472
 007416 012702 000004
 007422 110137 026517
 007426 010077 171740
 007432 004537 025032
 007436 000 001
 007440 025472
 007442 177777
 007444 004537 025032
 007450 013 010
 007452 000004
 007454 026472
 007456 004537 025032
 007462 014 014
 007464 000000
 007466 000000
 007470 032737 004000 001236
 007476 001407
 007500 004537 025076
 007504 015000
 007506 004537 025076
 007512 072000

; TEST 1

```

TST1:  MOV    #1,TSTNO
        MOV    #TST2,NEXT
        MOV    #0,R0
        MOV    L00.03,STAT
        BMI    100$
        JSR    PC,105$
100$:   MOV    #4,R0
        MOV    L04.07,STAT
        BMI    101$
        JSR    PC,105$
101$:   MOV    #8,R0
        MOV    L08.11,STAT
        BMI    102$
        JSR    PC,105$
102$:   MOV    #12,R0
        MOV    L12.15,STAT
        BMI    103$
        JSR    PC,105$
103$:   SCOPE
105$:   MOV    #1$,LOCK
        RAMCLR
        CLR    R3
        CLR    R1
        MOVB   #25,TXBAP
        MOV    #4,R2
1$:     MOVB   R1,XTAB+25
        MOV    R0,ADVSR5
        PERFORM SETREG
        .BYTE  000,001
        TXBAP
        -1
        PERFORM SETREG
        .BYTE  013,010
        BIT2
        XTAB
        PERFORM SETREG
        .BYTE  014,014
        0
        0
        BIT    #ASYNC,STAT
        BEQ    60$
        PERFORM LOAD.MODE
        <BIT12+BIT11>+BIT9
        PERFORM LOAD.MODE
        <BIT14+BIT13+BIT12>+BIT10
        #9600 BAUD.

```

***** TEST 1 *****
 *TEST OF TRANSMITTER CONTROL BYTES.
 *TEST OF 'NEXT MODE' FOR TRANSMITTER.
 *THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.

;PLACE LINE NUMBER INTO R0
 ;LOAD LINE CARD STATUS INTO STAT
 ;BR IF LINE CARD NOT TO BE TESTED
 ;GO DO THE TEST FOR LINE CARD 1
 ;PLACE LINE NUMBER INTO R0
 ;LOAD LINE CARD STATUS INTO STAT
 ;BR IF LINE CARD NOT TO BE TESTED
 ;GO DO THE TEST FOR LINE CARD 2
 ;LOAD LINE NUMBER
 ;LOAD LINE CARD STATUS INTO STAT
 ;BR IF LINE CARD NOT TO BE TESTED
 ;DO THE TEST FOR LINE CARD 3
 ;LOAD LINE NO.
 ;LOAD LINE CARD STATUS
 ;BR IF LINE CARD NOT TO BE TESTED
 ;DO THE TESTS FOR LINE CARD 4
 ;SCOPE THIS TEST.
 ;TEST ENTRANCE.
 ;SET IF SW09=1 (LOCK)
 ;CLEAR ALL SEC REGISTERS
 ;SET IMAGE EXPECTED MODE=0
 ;SET IMAGE 'NEXT MODE'=0
 ;SET TX DATA CHAR
 ;SET FOR 4 LINE GROUP
 ;LOAD CONTROL BYTE(MODE)
 ;LOAD LINE NUMBER
 ;TX PRINCIPLE BA, PRINCIPLE BC
 ;
 ;LINE STATE, CNTRI TABLE
 ;TXGO
 ;
 ;TX MODE REG
 ;MAKE
 ;IT=0
 ;IS THIS ASYNC LINE CARD?
 ;BR IF NO.
 ;
 ;8 BITS/PER/CHAR
 ;
 ;9600 BAUD.

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 28
 CZDVD.D.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0039

```

1269 007514 000403
1270 007516 004537 025076 60$: BR 61$
1271 007522 014000 :PERFORM ,LOAD.MODE :LOAD
1272 007524 012737 000340 177776 61$: BIT12+BIT11 :MODE
1273 007532 012777 007574 171616 :MOV #340,PS :LOCK OUT INTERRUPTS
1274 007540 012777 000340 171612 :MOV #3$,@DVTVEC :SET TRANS VECTOR
1275 007546 052777 020001 171606 :MOV #340,@DVTLVL :LOAD PRIO.
1276 007554 005005 :BIS #BIT13+BIT0,@DVSCR :SET STATUS IE AND UCPU GO.
1277 007556 104414 2$: CLR R5 :WAIT
1278 007560 005037 177776 :DELAY :STALL FOR TIME
1279 007564 005205 :CLR PS :ALLOW INTERRUPTS (NSR ENTRY)
1280 007566 001373 :INC R5 :ENTRY
1281 007570 104000 :BNE 2$
1282 007572 024646 :HLT :NO SILO ENTRY (DVSCR 15 NOT=1)
1283 007574 042777 020000 171560 3$: CMP -(SP),-(SP) :FAKE INTERRUPT BECAUSE NO REAL ONE HAPPENED.
1284 007602 005037 177776 :BIC #BIT13,@DVSCR :CLR IE
1285 007606 022626 :CLR PS :ZERO PSW
1286 007610 112777 000014 171556 :CMP (SP)+,(SP)+ :FAKE AN RTI
1287 007616 017704 :MOVB #14,@DVSRSH :SEL TX MODE REGISTER
1288 007622 010305 :MOV @DVSR,R4 :READ MODE REG.
1289 007624 020504 :MOV R3,R5 :SET EXPECTED
1290 007626 001401 :CMP R5,R4 :WAS 'NEXT MODE' LOADED CORRECTLY?
1291 007630 104003 :BEQ 4$ :BR IF YES
1292 007632 104412 4$: HLT 3 :TX MODE REGISTER WRONG
1293 007634 104401 :MSTCLR :INIT DV11
1294 007636 005203 :SCOP1 :LOCK ON MODE, LOCK ON LINE?
1295 007640 062701 000040 :INC R3 :UPDATE EXPECTED MODE
1296 007644 105701 :ADD #BIT5,R1 :UPDATE CNTRL BYTE IMAGE
1297 007646 001665 :TSTB R1 :ALL DONE??
1298 007650 005001 :BEQ 1$ :BR IF NO
1299 007652 005003 :CLR R1 :ZERO EXPECTE MODE
1300 007654 005200 :CLR R3 :ZERO CNTRL BYTE MODE
1301 007656 005302 :INC R0 :UPDATE LINE NO POINTER
1302 007660 001260 :DEC R2 :4 LINES DONE
1303 007662 000207 :BNE 1$ :BR IF YES
1304 :RTS PC :EXIT FOR NEXT GROUP OF LINES
1305
1306 :***** TEST 2 *****
1307 :*TEST OF TRANSMITTER IDLE FUNCTIONS.
1308 :*TEST THAT THE TRANSMITTER WILL IDLE
1309 :*SYNC (IDLE) CHARS WHEN BIT 0 OF
1310 :*DLE/PROTOCOL REGISTER IS CLEARED.
1311 :*THIS TEST IS DONE FOR SYNC LINE CARDS ONLY.
1312 :*****
1313
1314 : TEST 2
1315 :-----
1316 007664 012737 000002 001226 1ST2: MOV #2,TSTNO
1317 007672 012737 010422 001216 :MOV #TST3,NEXT
1318 007700 012700 000000 :MOV #0.,R0 :PLACE LINE NUMBER INTO R0
1319 007704 013737 001406 001244 :MOV MASK.A,MASKX :PLACE 'MASK' FOR CHARS INTO MASKX
1320 007712 013737 001422 001236 :MOV L00.03,STAT :LOAD LINE CARD STATUS INTO STAT
1321 007720 100402 :BMI 100$ :BR IF LINE CARD NOT TO BE TESTED
1322 007722 004737 010032 :JSR PC,105$ :GO DO THE TEST FOR LINE CARD 1
1323 007726 012700 000004 100$: MOV #4.,R0 :PLACE LINE NUMBER INTO R0
1324 007732 013737 001410 001244 :MOV MASK.B,MASKX :GET MASK

```

(ZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 29
 (ZDVDG.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0040

1325	007740	013737	001424	001236	MOV	L04.07,STAT	:LOAD LINE CARD STATUS INTO STAT
1326	007746	100402			BMI	101\$:BR IF LINE CARD NOT TO BE TESTED
1327	007750	004737	010032		JSR	PC,105\$:GO DO THE TEST FOR LINE CARD 2
1328	007754	012700	000010		101\$: MOV	#8.,R0	:LOAD LINE NUMBER
1329	007760	013737	001412	001244	MOV	MASK.C,MASKX	:GET MASK
1330	007766	013737	001426	001236	MOV	L08.11,STAT	:LOAD LINE CARD STATUS INTO STAT
1331	007774	100402			BMI	102\$:BR IF LINE CARD NOT TO BE TESTED
1332	007776	004737	010032		JSR	PC,105\$:DO THE TEST FOR LINE CARD 3
1333	010002	012700	000014		102\$: MOV	#12.,R0	:LOAD LINE NO.
1334	010006	013737	001414	001244	MOV	MASK.D,MASKX	:GET MASK
1335	010014	013737	001430	001236	MOV	L12.15,STAT	:LOAD LINE CARD STATUS
1336	010022	100402			BMI	103\$:BR IF LINE CARD NOT TO BE TESTED
1337	010024	004737	010032		JSR	PC,105\$:DO THE TESTS FOR LINE CARD 4
1338	010030	104400			103\$: SCOPE		:SCOPE THIS TEST.
1339	010032				105\$:		:TEST ENTRANCE.
1340	010032	032737	004000	001236	BIT	#ASYNC,STAT	:#IS THIS AN ASYNC LINE CARD?
1341	010040	001401			BEQ	.+4	:#BR IF NOT ASYNC.
1342	010042	000207			RTS	PC	:#EXIT TEST. (ASYNC LINE CARD NOT TESTED)
1343	010044	012737	010116	001220	MOV	#3\$,LOCK	:SET FOR RETURN IF SW09=1
1344	010052	104413			RAMCLR		:CLEAR ALL SEC REGISTERS
1345	010054	012705	026472		MOV	#TXTAB,R5	:CLEAR
1346	010060	012704	033072		MOV	#RXTAB,R4	:RECEIVER
1347	010064	005001			CLR	R1	:AND
1348	010066	005025			1\$: CLR	(R5)+	:TRANSMITTER
1349	010070	005024			CLR	(R4)+	:CONTROL
1350	010072	105201			INCB	R1	:TABLES
1351	010074	100374			BPL	1\$:
1352	010076	012737	000001	025472	MOV	#1,TXBAP	:LOAD TX
1353	010104	112737	000015	025473	MOVB	#1\$,TXBAP+1	:DTA
1354	010112	012702	000004		MOV	#4,R2	:SET FOR 4 LINE GROUP
1355	010116	010077	171250		3\$: MOV	R0,ADVSR5	:LOAD LINE NUMBER
1356	010122	005037	032472		LLR	RXBA	:CLEAR
1357	010126	005037	032474		CLR	RXBA+2	:RECEIVER
1358	010132	005037	032476		CLR	RXBA+4	:BUFFER
1359	010136	032737	004000	001236	BIT	#ASYNC,STAT	:#IS THIS AN ASYNC LINE CARD?
1360	010144	001406			BEQ	80\$:#BR IF NOT ASYNC.
1361	010146	004537	025032		PERFORM	SETREG	:#ADJUST FOR ASYNC LINE CARD
1362	010152	000	001		.BYTE	000,001	:#REGISTERS
1363	010154	025472			TXBAP		:#LOAD FOR ASYNC
1364	010156	177776			-2		:#LOAD FOR ASYNC
1365	010160	000405			BR	81\$:#CONTINUE TEST
1366	010162	004537	025032		80\$: PERFORM	SETREG	:
1367	010166	000	001		.BYTE	000,001	:TX PRINCIPLE BA, PRINCIPLE BC
1368	010170	025470			SYNC		:
1369	010172	177774			-4		:
1370	010174	004537	025032		81\$: PERFORM	SETREG	:
1371	010200	004	005		.BYTE	004,005	:RX BA, RX BC
1372	010202	032472			RXBA		:
1373	010204	177772			-6		:
1374	010206	004537	025032		PERFORM	SETREG	:
1375	010212	010	011		.BYTE	010,011	:TX TABLE, RXTABLE
1376	010214	026472			TXTAB		:
1377	010216	033072			RXTAB		:
1378	010220	004537	025032		PERFORM	SETREG	:
1379	010224	013	012		.BYTE	013,012	:LINE STATE, LINE PROTOCOL
1380	010226	000004			BIT2		:TX GOOD

```

1381 010230 000000 0 ;DEFAULT-IDLE SYNC
1382 010232 032737 004000 001236 BIT #ASYNC,STAT ;#IS THIS ASYNC LINE CARD?
1383 010240 001412 BEQ 60$ ;#BR IF NO.
1384 010242 004537 025076 PERFORM ,LOAD.MODE ;#LOAD PARAMETERS.
1385 010246 020000 BIT13 ;#RECEIVER ENABLE
1386 010250 004537 025076 PERFORM ,LOAD.MODE ;#
1387 010254 015000 <BIT12+BIT11>+BIT9 ;#8 BITS/PER/CHAR
1388 010256 004537 025076 PERFORM ,LOAD.MODE ;#
1389 010262 072000 <BIT14+BIT13+BIT12>+BIT10 ;#9600 BAUD.
1390
1391 010264 000405 BR 4$
1392 010266 004537 025076 6$: PERFORM ,LOAD.MODE ;LOAD
1393 010272 034000 BIT13+BIT12+BIT11 ;MODE AND RX ENABLE
1394 010274 004537 024620 PERFORM ,SETSYNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
1395 010300 005277 171056 4$: INC @DVSCR ;SET MICRO CPU GO
1396 010304 105777 171052 TSTB @DVSCR ;WAIT FOR
1397 010310 100375 BPL -4 ;DVSCR07=1
1398 010312 005004 CLR R4 ;
1399 010314 012705 000001 MOV #1,R5 ;SET EXPECTED
1400 010320 113704 032472 MOVB RXBA,R4 ;READ 1ST CHAR
1401
1402 010324 004737 023304 ;*****
JSR PC,PAREN ;CHECK FOR PARITY ENABLED (REV. DO)
1403 ;1ST CHAR S/B = 1!
1404 ;*****
1405 010330 112705 000015 5$: MOVB #15,R5 ;SET EXPECTED
1406 010334 113704 032473 MOVB RXBA+1,R4 ;GET 2ND CHAR
1407 ;*****
1408 010340 004737 023304 JSR PC,PAREN ;CHECK FOR PARITY ENABLED (REV. DO)
1409 ;2ND CHAR S/B = 15
1410 ;*****
1411 010344 6$: MOVB STAT,R5 ;SET EXPECTED=SYNC CHAR
1412 010344 113705 001236 BIC #^C<377>,R5 ;CLEAR HIGH BYTE
1413 010350 042705 177400 ;*****
1414 BIC MASKX,R5 ;CLEAR BITS/PER/CHAR MASK (REV. DO)
1415 010354 043705 001244 ;*****
1416 MOV #4,R3 ;SET TO LOOK AT 4 CHARS
1417 010360 012703 000004 MOV #RXBA+2,R1 ;GET RX DATA POINTER
1418 010364 012701 032474 7$: MOVB (R1)+,R4 ;GET FOUND DATA
1419 010370 112104 BIC #^C<377>,R4 ;CLEAN HIGH BYTE
1420 010372 042704 177400 ;*****
1421 JSR PC,MRKCK ;GO CHECK DATA (REV. DO)
1422 010376 004737 024414 ;IF ERROR, XMITR IDLED WRONG
1423 ;*****
1424 8$: DEC R3 ;4 CHARS CHECKED?
1425 010402 005303 BNE 7$ ;BR IF NO
1426 010404 001371 MSTCLR ;INIT DV11
1427 010406 104412 SCOP1 ;LOCK ON LINE?
1428 010410 104401 INC R0 ;UPDATE LINE POINTER
1429 010412 005200 DEC R2 ;4 LINE GROUP DONE?
1430 010414 005302 BNE 3$ ;BR IF NO
1431 010416 001237 RTS PC ;EXIT FOR NEXT GROUP
1432 010420 000207
1433
1434
1435 ;***** TEST 3 *****
1436 ;*TEST OF TRANSMITTER IDLE FUNCTIONS.

```

```
1437 ;*TEST THAT THE TRANSMITTER WILL IDLE
1438 ;*MARK STATE (377) WHEN BIT0 IS
1439 ;*SET IN THE DLE/PROTOCOL REGISTER.
1440 ;*THIS TEST IS DONE FOR SYNC LINE CARDS ONLY.
1441 ;*****
1442
1443 ; TEST 3
1444 -----
1445 010422 012737 000003 001226 TST3: MOV #3,TSTNO
1446 010430 012737 011160 001216 MOV #TST4,NEXT
1447 010436 012700 000000 MOV #0.,R0 ;PLACE LINE NUMBER INTO R0
1448 010442 013737 001406 001244 MOV MASK.A,MASKX ;PLACE 'MASK'FOR CHARS INTO MASKX
1449 010450 013737 001422 001236 MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
1450 010456 100402 BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
1451 010460 004737 010570 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 1
1452 010464 012700 000004 100$: MOV #4.,R0 ;PLACE LINE NUMBER INTO R0
1453 010470 013737 001410 001244 MOV MASK.B,MASKX ;GET MASK
1454 010476 013737 001424 001236 MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
1455 010504 100402 BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
1456 010506 004737 010570 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 2
1457 010512 012700 000010 101$: MOV #8.,R0 ;LOAD LINE NUMBER
1458 010516 013737 001412 001244 MOV MASK.C,MASKX ;GET MASK
1459 010524 013737 001426 001236 MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
1460 010532 100402 BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED
1461 010534 004737 010570 JSR PC,105$ ;DO THE TEST FOR LINE CARD 3
1462 010540 012700 000014 102$: MOV #12.,R0 ;LOAD LINE NO.
1463 010544 013737 001414 001244 MOV MASK.D,MASKX ;GET MASK
1464 010552 013737 001430 001236 MOV L12.15,STAT ;LOAD LINE CARD STATUS
1465 010560 100402 BMI 103$ ;BR IF LINE CARD NOT TO BE TESTED
1466 010562 004737 010570 JSR PC,105$ ;DO THE TESTS FOR LINE CARD 4
1467 010566 104400 103$: SCOPE ;SCOPE THIS TEST.
1468 010570 105$: ;TEST ENTRANCE.
1469 010570 032737 004000 001236 BIT #ASYNC,STAT ;#IS THIS AN ASYNC LINE CARD?
1470 010576 001401 BEQ .+4 ;#BR IF NOT ASYNC.
1471 010600 000207 RTS PC ;#EXIT TEST. (ASYNC LINE CARD NOT TESTED)
1472 010602 012737 010654 001220 MOV #3$,LOCK ;SET FOR RETURN IF SW09=1
1473 010610 104413 RAMCLR ;CLEAR ALL SEC REGISTERS
1474 010612 012705 026472 MOV #TXTAB,R5 ;CLEAR
1475 010616 012704 033072 MOV #RXTAB,R4 ;RECEIVER
1476 010622 005001 CLR R1 ;AND
1477 010624 005025 1$: CLR (R5)+ ;TRANSMITTER
1478 010626 005024 CLR (R4)+ ;CONTROL
1479 010630 105201 INCB R1 ;TABLES
1480 010632 100374 BPL 1$ ;
1481 010634 012737 000001 025472 MOV #1,TXBAP ;LOAD TX
1482 010642 112737 000015 025473 MOVB #15,TXBAP+1 ;DTA
1483 010650 012702 000004 MOV #4,R2 ;SET FOR 4 LINE GROUP
1484 010654 010077 170512 3$: MOV R0,@DVSRS ;LOAD LINE NUMBER
1485 010660 005037 032472 CLR RXBA ;CLEAR
1486 010664 005037 032474 CLR RXBA+2 ;RECEIVER
1487 010670 005037 032476 CLR RXBA+4 ;BUFFER
1488 010674 032737 004000 001236 BIT #ASYNC,STAT ;#IS THIS AN ASYNC LINE CARD?
1489 010702 001406 BEQ 80$ ;#BR IF NOT ASYNC.
1490 010704 004537 025032 PERFORM SETREG ;#ADJUST FOR ASYNC LINE CARD
1491 010710 000 001 .BYTE 000,001 ;#REGISTERS
1492 010712 025472 TXBAP ;#LOAD FOR ASYNC
```

Line	Address	Hex	Label	Instruction	Comment
1493	010714	177776		BR 81\$;#LOAD FOR ASYNC
1494	010716	000405			;#CONTINUE TEST
1495	010720	004537	025032	PERFORM ,SETREG	
1496	010724	000	001	.BYTE 000,001	;TX PRINCIPLE BA, PRINCIPLE BC
1497	010726	025470		SYNC	
1498	010730	177774		-4	
1499	010732	004537	025032	PERFORM ,SETREG	
1500	010736	004	005	.BYTE 004,005	;RX BA, RX BC
1501	010740	032472		RXBA	
1502	010742	177772		-6	
1503	010744	004537	025032	PERFORM ,SETREG	
1504	010750	010	011	.BYTE 010,011	;TX TABLE, RXTABLE
1505	010752	026472		TXTAB	
1506	010754	033072		RXTAB	
1507	010756	004537	025032	PERFORM ,SETREG	
1508	010762	013	012	.BYTE 013,012	;LINE STATE, LINE PROTOCOL
1509	010764	000004		BIT2	;TX GOOD
1510	010766	000001		BIT0	;IDLE MARK ON BYTE CNT=0
1511	010770	032737	004000	BIT #ASYNC,STAT	;#IS THIS ASYNC LINE CARD?
1512	010776	001412	001236	BEQ 60\$;#BR IF NO.
1513	011000	004537	025076	PERFORM ,LOAD.MODE	;#LOAD PARAMETERS.
1514	011004	020000		BIT13	;#RECEIVER ENABLE
1515	011006	004537	025076	PERFORM ,LOAD.MODE	;#
1516	011012	015000		<BIT12+BIT11>+BIT9	;#8 BITS/PER/CHAR
1517	011014	004537	025076	PERFORM ,LOAD.MODE	;#
1518	011020	072000		<BIT14+BIT13+BIT12>+BIT10	;#9600 BAUD.
1519					
1520	011022	000405		BR 4\$	
1521	011024	004537	025076	PERFORM ,LOAD.MODE	;LOAD
1522	011030	034000		BIT13+BIT12+BIT11	;MODE AND RX ENABLE
1523	011032	004537	024620	PERFORM ,SETSYNC	;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
1524	011036	005277	170320	INC @DVSCR	;SET MICRO CPU GO
1525	011042	105777	170314	TSTB @DVSCR	;WAIT FOR
1526	011046	100375		BPL -4	;DVSCR07=1
1527	011050	005004		CLR R4	
1528	011052	012705	000001	MOV #1,R5	;SET EXPECTED
1529	011056	113704	032472	MOVB RXBA,R4	;READ 1ST CHAR
1530					
1531	011062	004737	023304	JSR PC,PAREN	;CHECK FOR PARITY ENABLED (REV. DO)
1532					;1ST CHAR S/B = 1!
1533					
1534	011066	112705	000015	MOV #15,R5	;SET EXPECTED
1535	011072	113704	032473	MOVB RXBA+1,R4	;GET 2ND CHAR
1536					
1537	011076	004737	023304	JSR PC,PAREN	;CHECK FOR PARITY ENABLED (REV. DO)
1538					;2ND CHAR S/B = 15
1539					
1540	011102				
1541	011102	012705	000377	MOV #377,R5	;SET EXPECTED=MARK CHAR
1542	011106	042705	177400	BIC #^C<377>,R5	;CLEAR HIGH BYTE
1543					
1544	011112	043705	001244	BIC MASKX,R5	;CLEAR BITS/PER/CHAR MASK (REV. DO)
1545					
1546	011116	012703	000004	MOV #4,R3	;SET TO LOOK AT 4 CHARS
1547	011122	012701	032474	MOV #RXBA+2,R1	;GET RX DATA POINTER
1548	011126	112104		7\$: MOVB (R1)+,R4	;GET FOUND DATA

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 33
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEG 0044

```

1549 011130 042704 177400      BIC      #^C<377>,R4      ;CLEAN HIGH BYTE
1550                               ;*****
1551 011134 004737 024414      JSR      PC,MRKCK      ;GO CHECK DATA      (REV. DO)
1552                               ;IF ERROR, XMITR IDLED WRONG
1553                               ;*****
1554 011140 005303      8$:      DEC      R3      ;4 CHARS CHECKED?
1555 011142 001371      BNE      7$      ;BR IF NO
1556 011144 104412      MSTCLR      ;INIT DV11
1557 011146 104401      SCOP1      ;LOCK ON LINE?
1558 011150 005200      INC      R0      ;UPDATE LINE POINTER
1559 011152 005302      DEC      R2      ;4 LINE GROUP DONE?
1560 011154 001237      BNE      3$      ;BR IF NO
1561 011156 000207      RTS      PC      ;EXIT FOR NEXT GROUP
1562
1563
1564

```

```

;***** TEST 4 *****
;*TEST OF RECEIVER CONTROL BYTE OPERATIONS.
;*TEST OF THE 'STORE/DISCARD' FUNCTIONS.
;*TEST THAT CHARS:
;*      2$      STORED
;*      23      DISCARDED
;*      31      STORED
;*      32      DISCARDED
;*SINCE TWO CHRS SHOULD BE THROWN AWAY;
;*THE TX LINE IS SET TO GO BACK TO A MARK STATE;
;*THEREFORE THE RX BUFFER S/B:
;*RXBA 31,25
;*      377,377
;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
;*****

```

```

1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580      : TEST 4
1581      :-----
1582 011160 012737 000004 001226 1ST4: MOV      #4,TSTNO
1583 011166 012737 012116 001216      MOV      #TST5,NEXT
1584 011174 012700 000000      MOV      #0.,R0      ;PLACE LINE NUMBER INTO R0
1585 011200 013737 001406 001244      MOV      MASK.A,MASKX      ;PLACE 'MASK' FOR CHARS INTO MASKX
1586 011206 013737 001422 001236      MOV      L00.03,STAT      ;LOAD LINE CARD STATUS INTO STAT
1587 011214 100402      BMI      100$      ;BR IF LINE CARD NOT TO BE TESTED
1588 011216 004737 011326      JSR      PC,105$      ;GO DO THE TEST FOR LINE CARD 1
1589 011222 012700 000004      100$: MOV      #4.,R0      ;PLACE LINE NUMBER INTO R0
1590 011226 013737 001410 001244      MOV      MASK.B,MASKX      ;GET MASK
1591 011234 013737 001424 001236      MOV      L04.07,STAT      ;LOAD LINE CARD STATUS INTO STAT
1592 011242 100402      BMI      101$      ;BR IF LINE CARD NOT TO BE TESTED
1593 011244 004737 011326      JSR      PC,105$      ;GO DO THE TEST FOR LINE CARD 2
1594 011250 012700 000010      101$: MOV      #8.,R0      ;LOAD LINE NUMBER
1595 011254 013737 001412 001244      MOV      MASK.C,MASKX      ;GET MASK
1596 011262 013737 001426 001236      MOV      L08.11,STAT      ;LOAD LINE CARD STATUS INTO STAT
1597 011270 100402      BMI      102$      ;BR IF LINE CARD NOT TO BE TESTED
1598 011272 004737 011326      JSR      PC,105$      ;DO THE TEST FOR LINE CARD 3
1599 011276 012700 000014      102$: MOV      #12.,R0      ;LOAD LINE NO.
1600 011302 013737 001414 001244      MOV      MASK.D,MASKX      ;GET MASK
1601 011310 013737 001430 001236      MOV      L12.15,STAT      ;LOAD LINE CARD STATUS
1602 011316 100402      BMI      103$      ;BR IF LINE CARD NOT TO BE TESTED
1603 011320 004737 011326      JSR      PC,105$      ;DO THE TESTS FOR LINE CARD 4
1604 011324 104400      103$: SCOPE      ;SCOPE THIS TEST.

```


CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 34
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0045

```

1605 011326 105$: :TEST ENTRANCE.
1606 :*****
1607 011326 012737 011524 001220 :MOV #LINES,LOCK :SET RETURN IF SW09=1 (REV. D0)
1608 :*****
1609 011334 104413 :RAMCLR :CLEAR ALL DV11 SEC REGISTERS
1610 011336 105037 026517 :CLRB TXTAB+25 :ZERO
1611 011342 105037 026515 :CLRB TXTAB+23 :
1612 011346 105037 026523 :CLRB TXTAB+31 :USED
1613 011352 105037 026524 :CLRB TXTAB+32 :CONTROL
1614 011356 105037 027071 :CLRB TXTAB+37 :BYTES
1615 011362 012705 025472 :MOV #TXBAP,R5 :FOR TRANSMITTER
1616 011366 012725 :MOV (PC)+,(R5)+ :LOAD
1617 011370 025 023 :.BYTE 25,23 :TRANSMITTER
1618 011372 012715 :MOV (PC)+,(R5) :DATA
1619 011374 031 032 :.BYTE 31,32 :CHARS
1620 :*****
1621 011376 013703 001244 :MOV MASKX,R3 :GET BITS/CHAR (REV. D0)
1622 011402 032703 000400 :BIT #400,R3 :8 BIT?
1623 011406 001004 :BNE 84$ :BR IF YES
1624 011410 032737 020000 001236 :BIT #20000,STAT :EVEN PARITY EN?
1625 011416 001013 :BNE 85$ :BR IF YES
1626 011420 112737 000020 033115 84$: :MOVB #BIT4,RXTAB+23 :DISCARD
1627 :*****
1628 011426 112737 000020 033124 :MOVB #BIT4,RXTAB+32 :DISCARD
1629 011434 105037 033117 :CLRB RXTAB+25 :DEFAULT-STORE
1630 011440 105037 033123 :CLRB RXTAB+31 :DEFAULT-STORE
1631 :*****
1632 011444 000425 :BR 87$ :SKIP EVEN DATA LOAD (REV.D0)
1633 :
1634 :LOAD EVEN DATA
1635 :
1636 011446 132703 000040 85$: :BITB #40,R3 :5 BITS/CHAR?
1637 011452 001403 :BEQ 90$ :BR IF NOT
1638 011454 012701 024114 :MOV #FIVTAB,R1 :BASE ADDR. OF 5 BIT DATA TABLE
1639 011460 000410 :BR 86$ :GET OUT
1640 011462 132703 000100 90$: :BITB #100,R3 :6 BITS/CHAR?
1641 011466 001403 :BEQ 91$ :BR IF NOT
1642 011470 012701 024104 :MOV #SIXTAB,R1 :LOAD ADDR. OF 6 BIT DATA.
1643 011474 000402 :BR 86$ :GET OUT
1644 011476 012701 024074 91$: :MOV #SEVTAB,R1 :MUST BE 7 BITS/CHAR DATA
1645 011502 012731 010000 86$: :MOV #BIT12,@(R1)+ :DISCARD
1646 011506 012731 000020 :MOV #BIT4,@(R1)+ :DISCARD
1647 011512 005031 :CLR @(R1)+ :DEFAULT STORE
1648 011514 005071 000000 :CLR @(R1) :DEFAULT STORE
1649 011520 012702 000004 87$: :MOV #4,R2 :SET FOR 4 LINE GROUP
1650 011524 010077 167642 LINES: :MOV R0,@DVSR5 :LOAD LINE NO.
1651 :*****
1652 011530 005037 032472 :CLRB RXBA :MAKE SURE
1653 011534 005037 032474 :CLRB RXBA+2 :RX BUFFER=0
1654 011540 032737 004000 001236 :BIT #ASYNC,STAT :IS THIS AN ASYNC LINE CARD?
1655 011546 001406 :BEQ 80$ :BR IF NOT ASYNC.
1656 011550 004537 025032 :PERFORM SETREG :ADJUST FOR ASYNC LINE CARD
1657 011554 000 001 :.BYTE 000,001 :REGISTER
1658 011556 025472 :TXBAP :LOAD FOR ASYNC
1659 011560 177774 : -4 :LOAD FOR ASYNC
1660 011562 000405 :BR 81$ :CONTINUE TEST

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 35
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0046

```

1661 011564 004537 025032      80$:  PERFORM ,SETREG      ;
1662 011570      000      001      .BYTE 000,001      ;TX PRINCIPLE BA, PRINCIPLE BC
1663 011572 025470      SYNC      ;SYNC CHAR
1664 011574 177772      -6      ;2 SYNC, 4 DATA=6
1665 011576
1666 011576 032737 004000 001236 81$:  BIT      #ASYNC,STAT      ;#IS THIS AN ASYNC LINE CARD?
1667 011604 001406      BEQ      82$      ;#BR IF NOT ASYNC.
1668 011606 004537 025032      PERFORM ,SETREG      ;#ADJUST FOR ASYNC LINE CARD
1669 011612      004      005      .BYTE 004,005      ;#REGISTERS
1670 011614 032472      RXBA      ;#LOAD FOR ASYNC
1671 011616 177776      -2      ;#LOAD FOR ASYNC
1672 011620 000405      BR      83$      ;#CONTINUE TEST
1673 011622 004537 025032      82$:  PERFORM ,SETREG      ;
1674 011626      004      005      .BYTE 004,005      ;RXBA, RXBC
1675 011630 032472      RXBA      ;
1676 011632 177774      -4      ;
1677 011634 004537 025032      83$:  PERFORM ,SETREG      ;
1678 011640      010      011      .BYTE 010,011      ;TX TABLE, RX TABLE
1679 011642 026472      TXTAB      ;
1680 011644 033072      RXTAB      ;
1681 011646 004537 025032      PERFORM ,SETREG      ;
1682 011652      013      012      .BYTE 013,012      ;LINE STATE, LINE PROTOCOL
1683 011654 000004      BIT2      ;TX GO
1684 011656 000001      BIT0      ;IDLE MARK ON BYTE COUNTS=0
1685 011660 032737 004000 001236 84$:  BIT      #ASYNC,STAT      ;ASYNC LINE CARD?
1686 011666 001412      BEQ      60$      ;BR IF NOT
1687 011670 004537 025076      PERFORM ,LOAD.MODE      ;LOAD PARAMETERS
1688 011674 020000      BIT13      ;RECEIVER ENABLED
1689 011676 004537 025076      PERFORM ,LOAD.MODE      ;
1690 011702 015000      <BIT12+BIT11>+BIT9      ;8 BITS/CHAR
1691 011704 004537 025076      PERFORM ,LOAD.MODE      ;
1692 011710 072000      <BIT14+BIT13+BIT12>+BIT10      ;
1693
1694 011712 000405      BR      2$      ;9600 BAUD
1695 011714 004537 025076      60$:  PERFORM ,LOAD.MODE      ;SKIP SYNC SETUP
1696 011720 034000      BIT13+BIT12+BIT11      ;LOAD
1697 011722 004537 024620      PERFORM ,SETSYNC      ;MODE+RX ENABLE
1698 011726 005277 167430      2$:  INC      @DVSCR      ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
1699 011732 005005      CLR      R5      ;SET MICRO CPU GO
1700 011734 105777 167422      3$:  TSTB      @DVSCR      ;DELAY
1701 011740 100404      BMI      4$      ;FOR
1702 011742 104414      DELAY      ;RX INTERRUPT (BIT 7)
1703 011744 005205      INC      R5      ;WASTE TIME
1704 011746 001372      BNE      3$      ;KEEP COUNTING.
1705 011750 104000      HLT      ;BR
1706 011752      4$:  ;BIT 7 OF DVSCR NOT SET!
1707 011752 112705 000025      MOV B      #25,R5      ;SET EXPECTED
1708 011756 113704 032472      MOV B      RXBA,R4      ;GET FOUND
1709
1710 011762 032737 040000 001236 ;*****
1711 011770 001402      BIT      #PARBIT,STAT      ;PARITY ENABLED?
1712 011772 043704 001244      BEQ      52$      ;IF NO, BRANCH
1713 011776 120504      BIC      MASKX,R4      ;ELSE CLEAR BIT
1714
1715 012000 001401      52$:  CMPB      R5,R4      ;OK?
1716 012002 104002      BEQ      5$      ;*****
1716 012002 104002      HLT      2      ;'25' NOT FIRST IN RX BUFFER

```

(REV. D0)

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 36
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0047

```

1717 012004 112705 000031      5$:  MOVB  #31,R5      ;NEXT CHAR S/B '31'
1718 012010 113704 032473      MOVB  RXBA+1,R4      ;GET NEXT CHAR
1719                                     ;*****
1720 012014 032737 040000 001236  BIT  #PARBIT,STAT  ;PARITY ENABLED? (REV. DO)
1721 012022 001402                BEQ  51$          ;IF NO, BRANCH
1722 012024 043704 001244        BIC  MASKX,R4        ;ELSE CLEAR BIT
1723 012030 120504                51$: CMPB  R5,R4      ;OK?
1724                                     ;*****
1725 012032 001401                BEQ  6$          ;
1726 012034 104002                HLT  2              ;'31' NOT SECOND IN RX BUFFER
1727 012036 032737 004000 001236  6$:  BIT  #ASYNC,STAT  ;#IS THIS AN ASYNC LINE CARD?
1728 012044 001014                BNE  8$          ;#BR IF YES.
1729 012046 112705 000377        MOVB  #377,R5        ;MARK=377 (NEXT CHAR)
1730
1731
1732                                     ;*****
1733 012052 043705 001244        BIC  MASKX,R5          ;CLEAR BITS/PER/CHAR MASK. (REV. DO)
1734                                     ;*****
1735 012056 113704 032474        MOVB  RXBA+2,R4        ;GET FOUND
1736                                     ;*****
1737 012062 004737 024414        JSR  PC,MRKCK          ;GO COMPARE DATA (REV. DO)
1738                                     ;*****
1739 012066 113704 032475        7$:  MOVB  RXBA+3,R4      ;NEXT CHAR
1740                                     ;*****
1741 012072 004737 024414        JSR  PC,MRKCK          ;GO COMPARE DATA (REV. DO)
1742                                     ;*****
1743 012076 104412                8$:  MSTCLR          ;INIT DV11
1744 012100 104401                SCOP1          ;LOCK ON CURRENT LINE?
1745 012102 005200                INC  R0          ;UPDATE LINE POINTER
1746 012104 005302                DEC  R2          ;4 LINES DONE?
1747                                     ;*****
1748 012106 001402                BEQ  55$          ;BR IF TRUE (REV. DO)
1749 012110 000137 011524        JMP  LINES        ;NO, GO DO NEXT
1750 012114 000207                55$: RTS  PC          ;EXIT FOR NEXT GROUP
1751                                     ;*****
1752
1753
1754                                     ;***** TEST 5 *****
1755                                     ;*TEST OF RECEIVER CONTROL BYTE OPERATIONS.
1756                                     ;*TEST OF THE 'INCLUDE IN BCC YES/NO FUNCTION'
1757                                     ;*TEST THAT THE CHAR '031' IS INCLUDED
1758                                     ;*IN THE BCC WHEN AT:
1759                                     ;*LRC8      NOTE: ONLY LRC8 WILL TEST FOR < 8 BITS/CHAR (REV. DO)
1760                                     ;*CRC16
1761                                     ;*CRC.CCITT
1762                                     ;*THE RECEIVER BCC STARTS AT 0 AND CALCULATES
1763                                     ;*ONLY ONE CHAR (31).
1764                                     ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
1765                                     ;*****
1766
1767                                     ; TEST 5
1768                                     ;-----
1769 012116 012737 000005 001226  TST5: MOV  #5,TSTNO
1770 012124 012737 013032 001216  MOV  #TST6,NEXT
1771 012132 012700 000000        MOV  #0.,R0          ;PLACE LINE NUMBER INTO R0
1772 012136 013737 001406 001244  MOV  MASK.A,MASKX      ;PLACE 'MASK' FOR CHARS INTO MASKX

```

CZDVJ-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 37
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0048

1773	012144	013737	001422	001236	MOV	L00.03,STAT	;LOAD LINE CARD STATUS INTO STAT	
1774	012152	100402			BMI	100\$;BR IF LINE CARD NOT TO BE TESTED	
1775	012154	004737	012264		JSR	PC,105\$;GO DO THE TEST FOR LINE CARD 1	
1776	012160	012700	000004	100\$:	MOV	#4.,R0	;PLACE LINE NUMBER INTO R0	
1777	012164	013737	001410	001244	MOV	MASK.B,MASKX	;GET MASK	
1778	012172	013737	001424	001236	MOV	L04.07,STAT	;LOAD LINE CARD STATUS INTO STAT	
1779	012200	100402			BMI	101\$;BR IF LINE CARD NOT TO BE TESTED	
1780	012202	004737	012264		JSR	PC,105\$;GO DO THE TEST FOR LINE CARD 2	
1781	012206	012700	000010	101\$:	MOV	#8.,R0	;LOAD LINE NUMBER	
1782	012212	013737	001412	001244	MOV	MASK.C,MASKX	;GET MASK	
1783	012220	013737	001426	001236	MOV	L08.11,STAT	;LOAD LINE CARD STATUS INTO STAT	
1784	012226	100402			BMI	102\$;BR IF LINE CARD NOT TO BE TESTED	
1785	012230	004737	012264		JSR	PC,105\$;DO THE TEST FOR LINE CARD 3	
1786	012234	012700	000014	102\$:	MOV	#12.,R0	;LOAD LINE NO.	
1787	012240	013737	001414	001244	MOV	MASK.D,MASKX	;GET MASK	
1788	012246	013737	001430	001236	MOV	L12.15,STAT	;LOAD LINE CARD STATUS	
1789	012254	100402			BMI	103\$;BR IF LINE CARD NOT TO BE TESTED	
1790	012256	004737	012264		JSR	PC,105\$;DO THE TESTS FOR LINE CARD 4	
1791	012262	104400		103\$:	SCOPE		;SCOPE THIS TEST.	
1792	012264			105\$:			;TEST ENTRANCE.	
1793	012264	012737	012300	001220	MOV	#1\$,LOCK	;SET FOR 'LOCK ON LINE'.	
1794	012272	104413			RAMCLR		;CLEAR ALL SEC REGISTERS	
1795	012274	012702	000004		MOV	#4,R2	;4 LINE GROUP	
1796					;*****			
1797	012300			1\$:	LOADIT:	JSR PC,MODDAT	;GO CALCULATE CNTRL BYTE OFFSETS	(REV. DO)
1798	012300	004737	023734		CLRB	TXTAB+31	;CLEAR CNTRL BYTE (TX)	
1799	012304	105037	026523		MOVB	#BIT3,RXTAB(R4)	;SET 'INC/BCC' IN RX CNTRL BYTE	
1800	012310	112764	000010	033072	MOVB	#31,TXBAP	;SET TX DATA CHAR	
1801	012316	112737	000031	025472	MSTCLR		;INIT DV11	
1802	012324	104412			;*****			
1803					MOV	R0,@DVSR	;LOAD LINE #	
1804	012326	010077	167040		JSR	PC,DV11ON	;GOSUB DV11ON	
1805	012332	004737	025136		PERFORM	SETREG		
1806	012336	004537	025032		.BYTE	007,012		
1807	012342	007	012				;RXBCC,LINE PROTOCOL	
1808	012344	000000			0		;START BCC AT 0	
1809	012346	000000			0		;POLYNOMIAL SELECT	
1810	012350	005277	167006		INC	@DVSCR	;SET MICRO CPU GO	
1811	012354	105777	167002	44\$:	TSTB	@DVSCR	;WAIT FO	(REV. DO)
1812	012360	100375			BPL	44\$;BIT 7 OF DVSCR=1	(REV. DO)
1813	012362	112777	000007	167004	MOVB	#7,@DVSRSH	;SEL RX BCC REG	
1814	012370	017704	167002		MOV	@DVSR,R4	;READ BCC	
1815	012374	005037	025030		CLR	CALBCC	;SET SOFTWARE BCC=0	
1816	012400	012737	000200	025024	MOV	#LRCB,XPOLY	;SET SOFTWARE POLYNOMIAL	
1817					;*****			
1818	012406	012737	000010	001246	MOV	#10,TEMP1	;LOAD # OF SHIFTS	(REV. DO)
1819	012414	012737	000000	001252	MOV	#0,TEMP3	;GET PREVIOUS BCC	
1820	012422	010046			MOV	R0,-(SP)	;SAVE R0	
1821	012424	010146			MOV	R1,-(SP)	;SAVE R1	
1822	012426	010246			MOV	R2,-(SP)	;SAVE R2	
1823	012430	005037	025026	31\$:	CLR	BCCFBK		
1824	012434	013700	001252		MOV	TEMP3,R0		
1825	012440	006037	001250		ROR	TEMP2		
1826	012444	005500			ADC	R0		
1827	012446	032700	000001		BIT	#BIT0,R0		
1828	012452	001402			BEQ	32\$		

1829	012454	005137	025026		COM	BCCFBK	
1830	012460	013700	025024	32\$:	MOV	XPOLY,R0	
1831	012464	005100			COM	R0	
1832	012466	040037	025026		BIC	R0,BCCFBK	
1833	012472	000241			CLC		
1834	012474	006037	001252		ROR	TEMP3	
1835	012500	013700	025026		MOV	BCCFBK,R0	
1836	012504	013701	001252		MOV	TEMP3,R1	
1837	012510	010102			MOV	R1,R2	
1838	012512	040100			BIC	R1,R0	
1839	012514	043702	025026		BIC	BCCFBK,R2	
1840	012520	050200			BIS	R2,R0	
1841	012522	047737	025024	001252	BIC	XPOLY,TEMP3	
1842	012530	050037	001252		BIS	R0,TEMP3	
1843	012534	005337	01246		DEC	TEMP1	
1844	012540	001333			BNE	31\$	
1845	012542	013737	001252	025030	MOV	TEMP3,CALBCC	
1846	012550	012602			MOV	(SP)+,R2	:RESTORE R2
1847	012552	012601			MOV	(SP)+,R1	:RESTORE R1
1848	012554	012600			MOV	(SP)+,R0	:RESTORE R0
1849					;*****		
1850	012556	013705	025030		MOV	CALBCC,R5	:GET SOFTWARE BCC
1851	012562	020504			CMP	R5,R4	:SOFT=HARD?
1852	012564	001401			BEQ	+.4	
1853	012566	104004			HLT	4	:RECEIVER BCC INCORRECT!
1854	012570	032737	000002	001252	BIT	#BIT1,TEMP3	:LESS THAN 8 BITS/CHAR? (REV. DO)
1855	012576	001506			BEQ	62\$:YES,MUST SKIP CRC16 & CCITT (REV. DO)
1856	012600	104412			MSTCLR		:INIT DV11
1857	012602	110077	166564		MOV	R0,ADVSR5	:LOAD LINE NO.
1858	012606	004737	025136		JSR	PC,DV110N	:GOSUB DV110N
1859	012612	004537	025032		PERFORM	SETREG	
1860	012616	007	012		.BYTE	007,012	:RXBCC, LINE PROTOCOL
1861	012620	000000			0		:START BCC AT 0.
1862	012622	000010			BIT3		:POLYNOMIAL SELECT
1863	012624	005277	166532		INC	ADVSCR	:SET MICRO CPU GO
1864	012630	105777	166526		TSTB	ADVSCR	:WAIT FOR
1865	012634	100375			BPL	65\$:BIT 7 OF DVSCR=1
1866	012636	112777	000007	166530	MOVB	#7,ADVSR5H	:SEL RX BCC REG
1867	012644	017704	166526		MOV	ADVSR4,R4	:READ BCC
1868	012650	005037	025030		CLR	CALBCC	:SET SOFTWARE BCC=0
1869	012654	012737	120001	025024	MOV	#CRC16,XPOLY	:SET SOFTWARE POLYNOMIAL
1870	012662	004537	024652		JSR	R5,SIMBCC	:GO GET SOFTWARE BCC
1871	012666	000010			8.		:SHIFTS
1872	012670	000031			31		:DATA
1873	012672	000000			0		:PREVIOUS BCC
1874	012674	013705	025030		MOV	CALBCC,R5	:GET SOFTWARE BCC
1875	012700	020504			CMP	R5,R4	:SOFT=HARD?
1876	012702	001401			BEQ	+.4	
1877	012704	104004			HLT	4	:RECEIVER BCC INCORRECT!
1878	012706	104412			MSTCLR		:INIT DV11
1879	012710	010077	166456		MOV	R0,ADVSR5	:LOAD LINE NO.
1880	012714	004737	025136		JSR	PC,DV110N	:GOSUB DV110N
1881	012720	004537	025032		PERFORM	SETREG	
1882	012724	007	012		.BYTE	007,012	:RXBCC, LINE PROTOCOL
1883	012726	000000			0		:START BCC AT 0.
1884	012730	000030			BIT4+BIT3		:POLYNOMIAL SELECT

1885	012732	005277	166424		INC	@DVSCR	;SET MICRO CPU GO
1886	012736	105777	166420	67\$:	TSTB	@DVSCR	;WAIT FOR
1887	012742	100375			BPL	67\$;BIT 7 OF DVSCR=1
1888	012744	112777	000007	166422	MOVB	#7,@DVSRSH	;SEL RX BCC REG
1889	012752	017704	166420		MOV	@DVSR,R4	;READ BCC
1890	012756	005037	025030		CLR	CALBCC	;SET SOFTWARE BCC=0
1891	012762	012737	102010	025024	MOV	#CRC.CCITT,XPOLY	;SET SOFTWARE POLONOMIAL
1892	012770	004537	024652		JSR	R5,SIMBCC	;GO GET SOFTWARE BCC
1893	012774	000010			8.		;SHIFTS
1894	012776	000031			31		;DATA
1895	013000	000000			0		;PREVIOUS BCC
1896	013002	013705	025030		MOV	CALBCC,R5	;GET SOFTWARE BCC
1897	013006	020504			CMP	R5,R4	;SOFT=HARD?
1898	013010	001401			BEQ	.+4	
1899	013012	104004			HLT	4	;RECEIVER BCC INCORRECT!
1900	013014	104401		62\$:	SCOP1		;LOCK ON SELECTED LINE?
1901	013016	005200			INC	R0	;UPDATE LINE NO. POINTER
1902	013020	005302			DEC	R2	;ALL LINES DONE?
1903	013022	001402			BEQ	36\$;BR IF YES (REV. DO)
1904	013024	000137	012300		JMP	LOADIT	;DO NEXT LINE (REV. DO)
1905	013030	000207		36\$:	RTS	PC	;EXIT FOR NEXT GROUP

***** TEST 6 *****
 ;*TEST OF RECEIVER CONTROL BYTE OPERATIONS.
 ;*TEST OF THE 'NEXT MODE' FUNCTION.
 ;*TEST THAT THE NEXT MODE REGISTER (015)
 ;*CAN BE LOADED FROM THE CONTROL BYTES.
 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
 ;*****

; TEST 6

1917					TST6:	MOV	#6,TSTNO	
1918	013032	012737	000006	001226		MOV	#TST7,NEXT	
1919	013040	012737	013346	001216		MOV	#0.,R0	;PLACE LINE NUMBER INTO R0
1920	013046	012700	000000			MOV	MASK.A,MASKX	;PLACE 'MASK'FOR CHARS INTO MASKX
1921	013052	013737	001406	001244		MOV	L00.03,STAT	;LOAD LINE CARD STATUS INTO STAT
1922	013060	013737	001422	001236		MOV	L00.03,STAT	;LOAD LINE CARD STATUS INTO STAT
1923	013066	100402				BMI	100\$;BR IF LINE CARD NOT TO BE TESTED
1924	013070	004737	013200			JSR	PC,105\$;GO DO THE TEST FOR LINE CARD 1
1925	013074	012700	000004		100\$:	MOV	#4.,R0	;PLACE LINE NUMBER INTO R0
1926	013100	013737	001410	001244		MOV	MASK.B,MASKX	;GET MASK
1927	013106	013737	001424	001236		MOV	L04.07,STAT	;LOAD LINE CARD STATUS INTO STAT
1928	013114	100402				BMI	101\$;BR IF LINE CARD NOT TO BE TESTED
1929	013116	004737	013200			JSR	PC,105\$;GO DO THE TEST FOR LINE CARD 2
1930	013122	012700	000010		101\$:	MOV	#8.,R0	;LOAD LINE NUMBER
1931	013126	013737	001412	001244		MOV	MASK.C,MASKX	;GET MASK
1932	013134	013737	001426	001236		MOV	L08.11,STAT	;LOAD LINE CARD STATUS INTO STAT
1933	013142	100402				BMI	102\$;BR IF LINE CARD NOT TO BE TESTED
1934	013144	004737	013200			JSR	PC,105\$;DO THE TEST FOR LINE CARD 3
1935	013150	012700	000014		102\$:	MOV	#12.,R0	;LOAD LINE NO.
1936	013154	013737	001414	001244		MOV	MASK.D,MASKX	;GET MASK
1937	013162	013737	001430	001236		MOV	L12.15,STAT	;LOAD LINE CARD STATUS
1938	013170	100402				BMI	103\$;BR IF LINE CARD NOT TO BE TESTED
1939	013172	004737	013200			JSR	PC,105\$;DO THE TESTS FOR LINE CARD 4
1940	013176	104400			103\$:	SCOPE		;SCOPE THIS TEST.

1941	013200			105\$:	MOV	#1\$,LOCK	;TEST ENTRANCE.
1942	013200	012737	013236	001220	RAMCLR		;SET IF SW09=1
1943	013206	104413			CLR	R3	;CLEAR ALL SEC REGISTERS
1944	013210	005003			CLR	R1	;SET EXPECT RESULTS OF MODE REGISTER
1945	013212	005001			MOV	#4,R2	;SET CNTRL BYTE MODE
1946	013214	012702	000004				;SET FOR4 LINE GROUP
1947					;*****		
1948	013220	004737	023734		JSR	PC,MODDAT	;GO CALCULATE CNTRL BYTE OFFSETS (REV. DO)
1949	013224	012737	000031	025472	MOV	#31,TXBAP	;LOAD TX DATA CHAR
1950	013232	105037	026523		CLRB	TXTAB+31	;ZERO TX CNTRL BYTE
1951	013236	110164	033072		1\$:	MOVB	R1,RXTAB(R4)
1952							;LOAD RX CNTRL BYTE (WITH MODE)
1953	013242	004737	025136		;*****		
1954	013246	004537	025032		JSR	PC,DV11ON	;GO SETUP ROUTINE THINGS (BA,BC,LS,LP)
1955	013252	015	015		PERFORM	SETREG	;ZERO
1956	013254	000000			.BYTE	015,015	;RECEIVER
1957	013256	000000			0		;MODE
1958	013260	005277	166076		0		;REGISTER
1959	013264	105777	166072		INC	@DVSCR	;SET MICRO CPU GO
1960	013270	100375			TSTB	@DVSCR	;WAIT FOR
1961	013272	112777	000015	166074	BPL	.-4	;DVSCRO7=1
1962	013300	017704	166072		MOVB	#15,@DVSRSH	;SEL PX MODE REGISTER
1963	013304	010305			MOV	@DVSRSA,R4	;READ MODE REGISTER
1964	013306	020504			MOV	R3,R5	;SET EXPECTED MODE
1965	013310	001401			CMP	R5,R4	
1966	013312	104002			BEQ	3\$	
1967	013314	104412			HLT	2	;RX MODE REGISTER WRONG
1968	013316	005203			3\$:	MSTCLR	;INIT DV11
1969	013320	062701	000040		INC	R3	;UPDATE EXPECTED MODE
1970	013324	105701			ADD	#BIT5,R1	;UPDATE LOADED (NEXT) MODE
1971	013326	001743			TSTB	R1	;ALL DONE?
1972	013330	005001			BEQ	1\$;BR IF NO
1973	013332	005003			CLR	R1	;ZERO LOAD MODE
1974	013334	104401			CLR	R3	;ZERO EXPECTED MODE
1975	013336	005200			SCOP1		;LOCK ON SELECTED LINE?
1976	013340	005302			INC	R0	;UPDATE LINE POINTER
1977	013342	001335			DEC	R2	;4 LINE GROUP DONE?
1978	013344	000207			BNE	1\$;BR IF NO
1979					RTS	PC	;EXIT FOR NEXT GROUP OF LINES
1980					;***** TEST 7 *****		
1981					;*TEST OF TRANSMITTER CONTROL BYTE OPERATIONS.		
1982					;*TEST OF THE "SEND DLE NEXT" FUNCTION		
1983					;*THE "TRANSMITTER DLE REGISTER" IS LOADED		
1984					;*WITH CHAR '025'. THE RECEIVER IS SET TO RECEIVE		
1985					;*ONE CHAR (THE DLE) SO RX BA S/B=25		
1986					;*THE TRANSMITTER DATA CHAR IS '031'.		
1987					;*****		
1988							
1989							
1990					; TEST 7		
1991					;-----		
1992	013346	012737	000007	001226	TST7:	MOV	#7,TSTNO
1993	013354	012737	013670	001216		MOV	#TST10,NEXT
1994	013362	012700	000000			MOV	#0,R0
1995	013366	013737	001406	001244		MOV	MASK.A,MASKX
1996	013374	013737	001422	001236		MOV	LOO.03,STAT
							;PLACE LINE NUMBER INTO R0
							;PLACE 'MASK' FOR CHARS INTO MASKX
							;LOAD LINE CARD STATUS INTO STAT

1997	013402	100402			BMI	100\$:BR IF LINE CARD NOT TO BE TESTED
1998	013404	004737	013514		JSR	PC,105\$:GO DO THE TEST FOR LINE CARD 1
1999	013410	012700	000004		MOV	#4.,R0	:PLACE LINE NUMBER INTO R0
2000	013414	013737	001410	001244	MOV	MASK.B,MASKX	:GET MASK
2001	013422	013737	001424	001236	MOV	L04.07,STAT	:LOAD LINE CARD STATUS INTO STAT
2002	013430	100402			BMI	101\$:BR IF LINE CARD NOT TO BE TESTED
2003	013432	004737	013514		JSR	PC,105\$:GO DO THE TEST FOR LINE CARD 2
2004	013436	012700	000010		MOV	#8.,R0	:LOAD LINE NUMBER
2005	013442	013737	001412	001244	MOV	MASK.C,MASKX	:GET MASK
2006	013450	013737	001426	001236	MOV	L08.11,STAT	:LOAD LINE CARD STATUS INTO STAT
2007	013456	100402			BMI	102\$:BR IF LINE CARD NOT TO BE TESTED
2008	013460	004737	013514		JSR	PC,105\$:DO THE TEST FOR LINE CARD 3
2009	013464	012700	000014		MOV	#12.,R0	:LOAD LINE NO.
2010	013470	013737	001414	001244	MOV	MASK.D,MASKX	:GET MASK
2011	013476	013737	001430	001236	MOV	L12.15,STAT	:LOAD LINE CARD STATUS
2012	013504	100402			BMI	103\$:BR IF LINE CARD NOT TO BE TESTED
2013	013506	004737	013514		JSR	PC,105\$:DO THE TESTS FOR LINE CARD 4
2014	013512	104400			SCOPE		:SCOPE THIS TEST.
2015	013514				105\$:		:TEST ENTRANCE.
2016	013514	012737	013554	001220	MOV	#1\$,LOCK	:SET IF SW09=1
2017	013522	104413			RAMCLR		:CLEAR ALL SEC REGISTERS
2018					*****		
2019	013524	004737	023734		JSR	PC,MODDAT	:GO CALCULATE CNTRL BYTE OFFSETS (REV. D0)
2020	013530	112737	000002	026523	MOVB	#BIT1,TXTAB+31	:SET 'SND/DLE' IN CNTRL BYTE
2021	013536	112737	000031	025472	MOVB	#31,TXBAP	:SET TX DATA CHAR
2022	013544	105064	033072		CLRB	RXTAB(R4)	:ZERO RX CNTRL BYTE
2023					*****		
2024	013550	012702	000004		MOV	#4,R2	:SET FOR 4 LINE GROUP
2025	013554	004737	025136		JSR	PC,DV110N	:SET ROUTINE THING
2026	013560	004537	025032		PERFORM	SETREG	
2027	013564	012	012		.BYTE	012,012	:LINE PROTOCOL REG
2028	013566	012400			#25*400		:DATA
2029	013570	012400			#25*400		:IN HIGH BYTE
2030	013572	005037	032472		CLR	RXBA	:ZERO RX BUFFER
2031	013576	005277	165560		INC	@DVSCR	:SET MICRO CPU GO
2032	013602	105777	165554		TSTB	@DVSCR	:WAIT FOR
2033	013606	100375			BPL	.-4	:DVSC07=1
2034	013610	013704	032472		MOV	RXBA,R4	:GET DATA
2035	013614	012705	000025		MOV	#25,R5	:LOAD DLE INTO EXPECTED
2036					*****		
2037	013620	032737	040000	001236	BIT	#PARBIT,STAT	:PARITY ENABLED? (REV. D0)
2038	013626	001402			BEG	16\$:IF NO, BRANCH
2039	013630	043704	001244		BIC	MASKX,R4	:ELSE CLEAR BIT
2040	013634	120504			16\$:	CMPB R5,R4	
2041					*****		
2042	013636	001401			BEG	2\$	
2043	013640	104003			HLT	3	:25 (DLE) NOT 1ST IN RX BUFFER
2044	013642	104412			MSTCLR		:INIT DV11
2045	013644	112777	000012	165522	MOVB	#12,@DVSRSH	:SEL LINE PROTOCOL
2046	013652	005077	165520		CLR	@DVSR	:ZERO IT.
2047	013656	104401			SCOP1		:LOCK ON SELECTED LINE?
2048	013660	005200			INC	R0	:UPDATE LINE POINTER
2049	013662	005302			DEC	R2	:4 LINE GROUP DONE?
2050	013664	001333			BNE	1\$:BR IF NO
2051	013666	000207			RTS	PC	:EXIT FOR NEXT GROUP OF LINES
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
2077
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

013670 012737 000010 001226
013676 012737 014200 001216
013704 012700 000000
013710 013737 001422 001236
013716 100402
013720 004737 014006
013724 012700 000004
013730 013737 001424 001236
013736 100402
013740 004737 014006
013744 012700 000010
013750 013737 001426 001236
013756 100402
013760 004737 014006
013764 012700 000014
013770 013737 001430 001236
013776 100402
014000 004737 014006
014004 104400
014006
014006 012737 014036 001220
014014 104413
014016 012702 000004
014022 112737 000340 026507
014030 113737 026507 033107
014036 004737 025136
014042 112777 000012 165324
014050 052777 000140 165320
014056 112737 000015 025472
014064 005277 165272
014070 105777 165266
014074 100375
014076 005005
014100 112777 000014 165266
014106 017704 165264
014112 001401
014114 104001
014116 105277 165252
014122 017704 165250
014126 001401
014130 104001
014132 112777 000006 165234
014140 017704 165232
014144 001001
014146 104000

: TEST 10

```

:-----
TST10: MOV #10,TSTNO
        MOV #TST11,NEXT
        MOV #0,R0
        MOV L00.03,STAT
        BMI 100$
        JSR PC,105$
100$: MOV #4,R0
        MOV L04.07,STAT
        BMI 101$
        JSR PC,105$
101$: MOV #8,R0
        MOV L08.11,STAT
        BMI 102$
        JSR PC,105$
102$: MOV #12,R0
        MOV L12.15,STAT
        BMI 103$
        JSR PC,105$
103$: SCOPE
105$: MOV #18,LOCK
        RAMCLR
        MOV #4,R2
        MOV #BIT7+BIT6+BIT5,XTAB+15
        MOV #TXTAB+15,RXTAB+15
1$: JSR PC,DV110N
        MOV #12,@DVSRSH
        BIS #BIT6+BIT5,@DVSRSH
        MOV #15,TXBAP
        INC @DVSCR
        TSTB @DVSCR
        BPL -4
        CLR R5
        MOV #14,@DVSRSH
        MOV @DVSRSH,R4
        BEQ +4
        HLT 1
        INCB @DVSRSH
        MOV @DVSRSH,R4
        BEQ 3$
        HLT 1
3$: MOV #6,@DVSRSH
        MOV @DVSRSH,R4
        BNE 4$
        HLT 0

```

***** TEST 10 *****
 *TEST OF BOTH BITS 6 AND 5 OF THE LINE PROTOCOL REG.
 *TEST THAT NEITHER THE TRANSMITTER OR RECEIVER
 *CONTROL BYTES ARE USED AND THAT
 *THE CHARS ARE AUTOMATICALLY INCLUDED INTO THE BCC.
 *THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.

```

;PLACE LINE NUMBER INTO R0
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;GO DO THE TEST FOR LINE CARD 1
;PLACE LINE NUMBER INTO R0
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;GO DO THE TEST FOR LINE CARD 2
;LOAD LINE NUMBER
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;DO THE TEST FOR LINE CARD 3
;LOAD LINE NO.
;LOAD LINE CARD STATUS
;BR IF LINE CARD NOT TO BE TESTED
;DO THE TESTS FOR LINE CARD 4
;SCOPE THIS TEST.
;TEST ENTRANCE.
;SET RETURN IF SW09=1
;CLEAR ALL SEC REGISTERS
;SET FOR 4 LINE GROUP
;SET RX AND TX NEXT MODE=7
;SET UP MINOR DETAILS
;GET LINE PROTOCOL REGISTER
;SET TX AND RX DDCMP MODE
;LOAD DATA CHAR
;SET MICRO CPU GO
;WAIT FOR
;DVSCR07=1
;EXPECTED=0
;GET TX MODE REG
;READ MODE REG
;S/B=0
;TX MODE REG S/B=0
;GET RX MODE REG
;READ RX MODE
;RX MODE REG S/B=0
;TX BCC REG.
;READ TXBCC REG.
;S/B NOT=0
;NOTHING IN BCC!! (TX)

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 43
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0054

2109	014150	105277	165220	4\$:	INCB	@DVSRSH	:GET RX BCC REG
2110	014154	017704	165216		MOV	@DVSR4,R4	:READ INTO R4
2111	014160	001001			BNE	5\$	
2112	014162	104000			HLT	0	:NOTHING IN RXBCC!!
2113	014164	104412		5\$:	MSTCLR		:INIT DV11
2114	014166	104401			SCOPI		:LOCK ON SELECTED LINE?
2115	014170	005200			INC	R0	:UPDATE LINE POINTER
2116	014172	005302			DEC	R2	:4 LINES DONE?
2117	014174	001320			BNE	1\$:BR IF NO
2118	014176	000207			RTS	PC	:EXIT FOR NEXT 4 LINE GROUP

2119

2120

2121

2122

2123

2124

2125

2126

2127

2128

2129

2130

2131

2132

2133

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

***** TEST 11 *****
 :*TEST OF BIT 1 IN LINE PROTOCOL PARAMETER REGISTER.
 :*TEST OF 'STRIP LEADING SYNC'S'.
 :*TEST TO XMIT 10. SYNC CHARS, 1 NON-SYNC, AND 2 SYNC'S
 :*[13 CHARS TOTAL].
 :*DV11 SHOULD RECEIVE 1 NON-SYNC,AND TWO SYNC'S.
 :*[3 CHARS TOTAL].
 :*THE TEN LEADING SYNC'S S/B STRIPPED
 :*THIS TEST IS DONE FOR SYNC LINE CARDS ONLY.
 :*****

: TEST 11

:-----

2134	014200	012737	000011	001226	TST11:	MOV	#11,TSTNO	
2135	014206	012737	014614	001216		MOV	#TST12,NEXT	
2136	014214	012700	000000			MOV	#0.,R0	:PLACE LINE NUMBER INTO R0
2137	014220	013737	001406	001244		MOV	MASK.A,MASKX	:PLACE 'MASK'FOR CHARS INTO MASKX
2138	014226	013737	001422	001236		MOV	L00.03,STAT	:LOAD LINE CARD STATUS INTO STAT
2139	014234	100402				BMI	100\$:BR IF LINE CARD NOT TO BE TESTED
2140	014236	004737	014346			JSR	PC,105\$:GO DO THE TEST FOR LINE CARD 1
2141	014242	012700	000004		100\$:	MOV	#4.,R0	:PLACE LINE NUMBER INTO R0
2142	014246	013737	001410	001244		MOV	MASK.B,MASKX	:GET MASK
2143	014254	013737	001424	001236		MOV	L04.07,STAT	:LOAD LINE CARD STATUS INTO STAT
2144	014262	100402				BMI	101\$:BR IF LINE CARD NOT TO BE TESTED
2145	014264	004737	014346			JSR	PC,105\$:GO DO THE TEST FOR LINE CARD 2
2146	014270	012700	000010		101\$:	MOV	#8.,R0	:LOAD LINE NUMBER
2147	014274	013737	001412	001244		MOV	MASK.C,MASKX	:GET MASK
2148	014302	013737	001426	001236		MOV	L08.11,STAT	:LOAD LINE CARD STATUS INTO STAT
2149	014310	100402				BMI	102\$:BR IF LINE CARD NOT TO BE TESTED
2150	014312	004737	014346			JSR	PC,105\$:DO THE TEST FOR LINE CARD 3
2151	014316	012700	000014		102\$:	MOV	#12.,R0	:LOAD LINE NO.
2152	014322	013737	001414	001244		MOV	MASK.D,MASKX	:GET MASK
2153	014330	013737	001430	001236		MOV	L12.15,STAT	:LOAD LINE CARD STATUS
2154	014336	100402				BMI	103\$:BR IF LINE CARD NOT TO BE TESTED
2155	014340	004737	014346			JSR	PC,105\$:DO THE TESTS FOR LINE CARD 4
2156	014344	104400			103\$:	SCOPE		:SCOPE THIS TEST.
2157	014346				105\$:			:TEST ENTRANCE.
2158	014346	032737	004000	001236		BIT	#ASYNC,STAT	:#IS THIS AN ASYNC LINE CARD?
2159	014354	001401				BEQ	.+4	:#BR IF NOT ASYNC LINE CARD.
2160	014356	000207				RTS	PC	:#EXIT TEST. (ASYNC LINE CARD NOT TESTED)
2161	014360	012737	014430	001220		MOV	#2\$,LOCK	:SET RETURN IF SW09=1
2162	014366	104413				RAMCLR		:CLEAR ALL SEC REGISTERS
2163	014370	012702	000004			MOV	#4,R2	:4 LINE GROUP
2164	014374	012704	000010			MOV	#8.,R4	:LOAD TX BUFFER

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 44
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0055

```

2165 014400 012705 025472      MOV    #TXBAP,R5      ;WITH
2166 014404 113725 001236      1$:  MOVB   STAT,(R5)+    ;8. SYNC
2167 014410 005304              DEC    R4                ;CHARS
2168 014412 001374              BNE    1$                  ;
2169 014414 112725 000005      MOVB   #5,(R5)+    ;LOAD 'NON-SYNC' CHAR
2170 014420 113725 001236      MOVB   STAT,(R5)+    ;SYNC
2171 014424 113725 001236      MOVB   STAT,(R5)+    ;SYNC
2172 014430 004737 025136      2$:  JSR    PC,DV110N    ;MINOR DETAIL SETUP
2173 014434 005037 032472      CLR    RXBA        ;CLEAR
2174 014440 005037 032474      CLR    RXBA+2      ;RX BUFFER
2175 014444 004537 025032      PERFORM ,SETREG    ;
2176 014450 001005 005        .BYTE  001,005      ;TX PRINCIPLE BC, RX BC
2177 014452 177763            -13.
2178 014454 177775            -3
2179 014456 112777 000012 164710  MOVB   #12,@DVSRSH    ;LINE PROTOCOL REG
2180 014464 012777 000143 164704  MOV    #BIT6+BIT5+BIT1+BIT0,@DVSRSH
2181 014472 005277 164664      INC    @DVSCR        ;LP=TX+RX DDCMP, STRIP SYNC, IDLE MARK
2182 014476 105777 164660      TSTB   @DVSCR        ;WAIT FOR
2183 014502 100375            BPL    .-4            ;DVSCR07=1
2184 014504 012705 000005      MOV    #5,R5          ;1ST DATA S/B=15
2185 014510 113704 032472      MOVB   RXBA,R4        ;GET DATA
2186 014514 042704 177400      BIC    #^C<377>,R4    ;STRIP HIGH BYTE
2187 *****
2188 014520 004737 023304      JSR    PC,PAREN    ;CHECK PARITY ENABLE (REV. D0)
2189 *****
2190 *****
2191 014524 113705 001236      3$:  MOVB   STAT,R5      ;LOAD SYNC INTO EXPECTED
2192 014530 042705 177400      BIC    #^C<377>,R5    ;STRIP HIGH BYTE
2193 *****
2194 014534 032737 020000 001236  BIT    #20000,STAT    ;EVEN PARITY EN? (REV. D0)
2195 014542 001402            BEQ    25$          ;IF NO, BRANCH. DO NOT MASK PARITY BIT
2196 *****
2197 *****
2198 *****
2199 014544 043705 001244      BIC    MASKX,R5      ;CLEAR BITS/PER/CHAR MASK.
2200 014550 113704 032473      25$: MOVB   RXBA+1,R4    ;GET 2ND CHAR
2201 *****
2202 014554 042704 177400      BIC    #^C<377>,R4    ;STRIP HIGH BYTE
2203 *****
2204 014560 004737 023304      JSR    PC,PAREN    ;CHECK PARITY ENABLED (REV. D0)
2205 *****
2206 *****
2207 014564 113704 032474      4$:  MOVB   RXBA+2,R4    ;GET 3RD CHAR
2208 014570 042704 177400      BIC    #^C<377>,R4    ;STRIP HIGH BYTE
2209 *****
2210 014574 004737 023304      JSR    PC,PAREN    ;CHECK PARITY ENABLED (REV. D0)
2211 *****
2212 *****
2213 *****
2214 014600 104412            5$:  MSTCLR    ;INIT DV11
2215 014602 104401            SCOP1    ;LOCK ON SELECTED LINES?
2216 014604 005200            INC     R0          ;UPDATE LINE POINTER
2217 014606 005302            DEC     R2          ;4 LINE GROUP DONE?
2218 014610 001307            BNE     2$          ;BR IF NO
2219 014612 000207            RTS     PC          ;EXIT FOR NEXT GROUP
2220

```

2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
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

014614 012737 000012 001226
014622 012737 015360 001216
014630 012700 000000
014634 013737 001406 001244
014642 013737 001422 001236
014650 100402
014652 004737 014762
014656 012700 000004
014662 013737 001410 001244
014670 013737 001424 001236
014676 100402
014700 004737 014762
014704 012700 000010
014710 013737 001412 001244
014716 013737 001426 001236
014724 100402
014726 004737 014762
014732 012700 000014
014736 013737 001414 001244
014744 013737 001430 001236
014752 100402
014754 004737 014762
014760 104400
014762
014762 012737 015100 001220
014770 104413
014772 005001
014774 012702 000004
015000 005005
015002 012704 025472
015006 110524
015010 005205
015012 022705 000007
015016 001373
015020 005003
015022 005005
015024 012704 033072
015030 013703 001244
015034 032703 000400
015040 001004
015042 032737 040000 001236
015050 001007

; TEST 12

```

TST12:  MOV    #12,TSTNO
        MOV    #TST13,NEXT
        MOV    #0.,R0
        MOV    MASK.A,MASKX
        MOV    L00.03,STAT
        BMI    100$
        JSR    PC,105$
100$:    MOV    #4.,R0
        MOV    MASK.B,MASKX
        MOV    L04.07,STAT
        BMI    101$
        JSR    PC,105$
101$:    MOV    #8.,R0
        MOV    MASK.C,MASKX
        MOV    L08.11,STAT
        BMI    102$
        JSR    PC,105$
102$:    MOV    #12.,R0
        MOV    MASK.D,MASKX
        MOV    L12.15,STAT
        BMI    103$
        JSR    PC,105$
103$:    SCOPE
105$:    MOV    #66$,LOCK
        RAMCLR
        CLR    R1
        MOV    #4,R2
        CLR    R5
        MOV    #TXBAP,R4
1$:      MOV    R5,(R4)+
        INC    R5
        CMP    #7,R5
        BNE    1$
        CLRR   R3
        CLRR   R5
        MOV    #RXTAB,R4
        MOV    MASKX,R3
        BIT    #400,R3
        BNE    11$
        BIT    #PARBIT,STAT
        BNE    12$

```

```

;PLACE LINE NUMBER INTO R0
;PLACE 'MASK' FOR CHARS INTO MASKX
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;GO DO THE TEST FOR LINE CARD 1
;PLACE LINE NUMBER INTO R0
;GET MASK
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;GO DO THE TEST FOR LINE CARD 2
;LOAD LINE NUMBER
;GET MASK
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;DO THE TEST FOR LINE CARD 3
;LOAD LINE NO.
;GET MASK
;LOAD LINE CARD STATUS
;BR IF LINE CARD NOT TO BE TESTED
;DO THE TESTS FOR LINE CARD 4
;SCOPE THIS TEST.
;TEST ENTRANCE.

```

```

;CLEAR ALL SEC REGISTERS
;SET FOR 4 LINE GROUP
;LOAD
;TX DATA

```

```

;POINT TO RX CNTRL TABLE
;GET BITS/CHAR. DATA
;8 BITS/CHAR?
;BN IF YES
;FAR. EN?
;BR IF YES

```

(REV. D0)

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 46
CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0057

2277	015052	012705	000006		11\$: MOV #6,R5 ;SET UP TABLE LINE COUNTER	
2278					;*****	
2279	015056	112724	000001		2\$: MOVB #BIT0,(R4)+ ;RX CNTRL TABLE	
2280	015062	005305			DEC R5 ;WITH SPECIAL	
2281	015064	001374			BNE 2\$;CHAR BITS	
2282					;*****	
2283	015066	000404			BR 66\$;SKIP PARITY EN. SETUP	(REV. DO)
2284	015070	012705	000006		12\$: MOV #6,R5 ;SET UP TABLE LINE COUNTER	
2285	015074	004737	023366		JSR PC,BTCHAR ;GO SETUP PAR. EN. DATA	
2286					;*****	
2287	015100	004737	025136		66\$: JSR PC,DV11ON ;SET UP DV11	
2288	015104	004537	025032		PERFORM ,SETREG ;	
2289	015110	001	005		.BYTE 001,005 ;TX PRINCIPLE BC, RX BC	
2290	015112	177770			-8. ;	
2291	015114	177772			-6 ;	
2292	015116	112777	000012	164250	MOVB #12,@DVSRSH ;LINE PROTOCOL	
2293	015124	052777	000100	164244	BIS #BIT6,@DVSRA ;TX DDCMP	
2294	015132	012737	000340	177776	MOV #340,PS ;LOCK OUT INTERRUPTS	
2295	015140	012777	015210	164204	MOV #67\$,@DVRVEC ;SET RX INTER VECTOR.	
2296	015146	012777	000340	164200	MOV #340,@DVRLVL ;SET PRIO. LEVEL TO 7	
2297	015154	052777	000101	164200	BIS #BIT6+BIT0,@DVSCR ;	
2298					;SET RX IE AND UCPU GO.	
2299	015162	005003			CLR R3 ;DATA IMAGE	
2300	015164	005037	015356		3\$: CLR 69\$;STALL COUNTER	
2301	015170	005037	177776		CLR PS ;ENABLE INTERRUPTS	
2302	015174	104414			DELAY ;WASTE TIME	
2303	015176	005237	015356		INC 69\$;UPDATE STALL	
2304	015202	001372			BNE .-12 ;BR BACK	
2305	015204	104000			HLT ;NO INTERRUPT OCCURED.	
2306	015206	024646			CMP -(SP),-(SP) ;FAKE AN INTERRUPT	
2307	015210	010005			67\$: MOV R0,R5 ;LOAD LINE NO.	
2308	015212	000305			SWAB R5 ;PUT IN HIGH BYTE	
2309	015214	050305			BIS R3,R5 ;SET DATA	
2310	015216	017704	164144		MOV @DVRIC,R4 ;READ FOUND RESULT	
2311					;*****	
2312	015222	004737	023304		JSR PC,PAREN ;CHECK PARITY ENABLED	(REV. DO)
2313					;IF ERROR,DVRIC WRONG!	
2314					;*****	
2315	015226	052777	000400	164126	4\$: BIS #BIT8,@DVSCR ;SET 'RECEIVER INT RESP COMP'	
2316	015234	005203			INC R3 ;UPDATE DATA IMAGE	
2317	015236	022703	000006		CMP #6,R3 ;ALL DONE?	
2318	015242	001403			BEG 70\$;BR IF YES	
2319	015244	012716	015164		MOV #3\$, (SP) ;SET RETURN	
2320	015250	000002			RTI ;CONTINUE	
2321	015252	042777	000100	164102	70\$: BIC #BIT6,@DVSCR ;NO MORE INTERRUPTS.	
2322	015260	012716	015266		MOV #68\$, (SP) ;SET RETURN	
2323	015264	000002			RTI ;CONT.	
2324	015266	105777	164070		68\$: TSTB @DVSCR ;WAIT FOR	
2325	015272	100375			BPL .-4 ;DVSCRO7=1	
2326	015274	010005			MOV R0,R5 ;LOAD LINE NO.	
2327	015276	000305			SWAB R5 ;PUT IN HIGH BYTE	
2328	015300	052705	040005		BIS #BIT14+5,R5 ;'BYTE CNT WARNING + DATA'	
2329	015304	017704	164056		MOV @DVRIC,R4 ;READ RESULTS	
2330					;*****	
2331	015310	004737	023304		JSR PC,PAREN ;CHECK PARITY ENABLE	(REV. DO)
2332					;IF ERROR, DVRIC WRONG!	

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 47
 CZDVD.D.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0058

```

2333
2334 015314 012701 032472
2335 015320 005005
2336 015322 005004
2337 015324 112104
2338
2339 015326 004737 023304
2340
2341
2342 015332 005205
2343 015334 022705 000006
2344 015340 001371
2345 015342 104412
2346 015344 104401
2347 015346 005200
2348 015350 005302
2349 015352 001252
2350 015354 000207
2351 015356 000000
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373 015360 012737 000013 001226
2374 015366 012737 016254 001216
2375 015374 012700 000000
2376 015400 013737 001406 001244
2377 015406 013737 001422 001236
2378 015414 100402
2379 015416 004737 015526
2380 015422 012700 000004
2381 015426 013737 001410 001244
2382 015434 013737 001424 001236
2383 015442 100402
2384 015444 004737 015526
2385 015450 012700 000010
2386 015454 013737 001412 001244
2387 015462 013737 001426 001236
2388 015470 100402

*****
5$: MOV #RXBA,R1 ;GET RX POINTER
   CLR R5
   CLR R4
6$: MOVB (R1)+,R4 ;GET RX DATA (INCORE)
*****
   JSR PC,PAREN ;CHECK PARITY ENABLE (REV. DO)
   ;IF ERROR, RECVR PLACED DATA IN CORE INCORRECTLY.
*****
7$: INC R5 ;UPDATE DATA IMAGE
   CMP #6,R5 ;ALL DONE?
   BNE 6$ ;BR IF NO
   MSTCLR ;INIT DV11
   SCOP1 ;LOCK ON CURRENT LINE?
   INC R0 ;UPDATE LINE POINTER
   DEC R2 ;4 LINE GROUP DONE?
   BNE 66$ ;BR IF NO
   RTS PC ;EXIT FOR NEXT GROUP OF LINES
69$: 0

```

```

***** TEST 13 *****
; *TEST OF THE 'MARKED BYTE COUNT'.
; *TEST THAT WHEN BIT15=0 FOR THE RECEIVER THAT
; *BITS 13,14,15 OF LINE STATE OCCUR IN
; *THE RECEIVER MODE BITS REGISTER.
; *TEST THAT WHEN BIT15=0 FOR THE TRANSMITTER
; *THAT BITS 13,14,15 OF THE LINE PROGRESS REGISTER
; *OCCUR INT THE TRANSMITTER MODE REG.
; *ALSO VERIFY THAT BIT10=1 IN LINE STATE MAKES
; *RECEIVER 'EXPECT THE BCC'
; *AND THAT BIT10 IN LINE PROGRESS TELL TX TO SEND BCC.
; *THIS TEST USES CRC.CCITT FOR THE POLYNOMIAL
; **NOTE*: IF LINE CARD IS SET FOR OTHER THAN '8' BITS
; *THE TEST WILL *NOT* BE EXECUTED ON THAT LINE CARD!!
; *THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
*****

```

: TEST 13

```

-----
TST13: MOV #13,TSTNO
        MOV #TST14,NEXT
        MOV #0.,R0 ;PLACE LINE NUMBER INTO R0
        MOV MASK.A,MASKX ;PLACE 'MASK' FOR CHARS INTO MASKX
        MOV LOO.03,STAT ;LOAD LINE CARD STATUS INTO STAT
        BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
        JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 1
100$: MOV #4.,R0 ;PLACE LINE NUMBER INTO R0
        MOV MASK.B,MASKX ;GET MASK
        MOV LO4.07,STAT ;LOAD LINE CARD STATUS INTO STAT
        BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
        JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 2
101$: MOV #8.,R0 ;LOAD LINE NUMBER
        MOV MASK.C,MASKX ;GET MASK
        MOV LO8.11,STAT ;LOAD LINE CARD STATUS INTO STAT
        BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 48
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0059

2389	015472	004737	015526			JSR	PC,105\$;DO THE TEST FOR LINE CARD 3
2390	015476	012700	000014		102\$:	MOV	#12.,R0	;LOAD LINE NO.
2391	015502	013737	001414	001244		MOV	MASK.D,MASKX	;GET MASK
2392	015510	013737	001430	001236		MOV	L12.15,STAT	;LOAD LINE CARD STATUS
2393	015516	100402				BMI	103\$;BR IF LINE CARD NOT TO BE TESTED
2394	015520	004737	015526			JSR	PC,105\$;DO THE TESTS FOR LINE CARD 4
2395	015524	104400			103\$:	SCOPE		;SCOPE THIS TEST.
2396	015526				105\$:			;TEST ENTRANCE.
2397	015526	012737	015714	001220		MOV	#65\$,LOCK	;SET RETURN IF SW09=1
2398	015534	032737	001400	001236		BIT	#BIT9+BIT8,STAT	; '8 BITS/PER/CHAR ?'
2399	015542	001401				BEQ	.+4	;BR IF YES
2400	015544	000207				RTS	PC	;EXIT TEST FOR THIS LINE CARD!
2401	015546	104413				RAMCLR		;CLEAR ALL SECONDARY REGISTERS
2402	015550	012702	000004			MOV	#4,R2	;SET FOR 4 LINE GROUP
2403	015554	012704	000012			MOV	#10.,R4	;LOAD 10 BYTES
2404	015560	012705	026472			MOV	#TXTAB,R5	;WITH
2405	015564	112725	000010		1\$:	MOVB	#BIT3,(R5)+	;INC/BCC
2406	015570	005304				DEC	R4	
2407	015572	001374				BNE	1\$	
2408	015574	012705	026472			MOV	#TXTAB,R5	;CLEAR
2409	015600	013704	001236			MOV	STAT,R4	;SYNC
2410	015604	042704	177400			BIC	#^C<377>,R4	;CONTROL
2411	015610	060405				ADD	R4,R5	;BYTE
2412	015612	105015				CLRB	(R5)	
2413	015614	012705	025472			MOV	#TXBAP,R5	;LOAD
2414	015620	005004				CLR	R4	;DATA
2415	015622	110425			2\$:	MOVB	R4,(R5)+	;INTO
2416	015624	005204				INC	R4	;TRANSMITTER BUFFER
2417	015626	022704	000013			CMP	#11.,R4	
2418	015632	001373				BNE	2\$	
2419						;*****		
2420	015634	005003				CLR	R3	
2421	015636	005005				CLR	R5	
2422	015640	012704	033072			MOV	#RXTAB,R4	;POINT TO RX TABLE BASE ADDR.
2423	015644	013703	001244			MOV	MASKX,R3	;GET BITS/CHAR DATA
2424	015650	032703	000400			BIT	#400,R3	;8 BITS/CHAR?
2425	015654	001004				BNE	40\$;BR IF YES
2426	015656	032737	040000	001236		BIT	#PARBIT,STAT	;PAR. EN?
2427	015664	001007				BNE	41\$;BR IF YES
2428	015666	012705	000012		40\$:	MOV	#10.,R5	;LOAD 10
2429	015672	112724	000010		3\$:	MOVB	#BIT3,(R4)+	;RECEIVER
2430	015676	005305				DEC	R5	;CONTROL BYTES
2431	015700	001374				BNE	3\$;WITH 'INC/BCC'
2432	015702	000404				BR	65\$;SKIP OVER PAR. EN SETUP
2433	015704	012705	000012		41\$:	MOV	#10.,R5	;SET UP TABLE LINE COUNTER
2434	015710	004737	023366			JSR	PC,BTCHAR	;GO SETUP PAR. EN. DATA
2435						;*****		
2436	015714	010077	163452		65\$:	MOV	R0,@DVSRS	;LOAD LINE NO.
2437	015720	032737	004000	001236		BIT	#ASYNC,STAT	;#IS THIS AN ASYNC LINE CARD?
2438	015726	001406				BEQ	80\$;#BR IF NOT ASYNC.
2439	015730	004537	025032			PERFORM	,SETREG	;#ADJUST FOR ASYNC LINE CARD
2440	015734	000	001			.BYTE	000,001	;#REGISTERS
2441	015736	025472				TXBAP		;#LOAD FOR ASYNC
2442	015740	077766				<-10.>-BIT15		;#LOAD FOR ASYNC
2443	015742	000405				BR	81\$;#CONTINUE TEST
2444	015744	004537	025032		80\$:	PERFORM	,SETREG	

(REV. D0)

2445	015750	000	001		.BYTE 000,001	;TX PRINCIPLE BA, BC
2446	015752	025470			SYNC	
2447	015754	077764			<-12.>-BIT15	;MARKED BC.
2448	015756	004537	025032	81\$:	PERFORM ,SETREG	
2449	015762	004	005		.BYTE 004,005	;RX BA, BC
2450	015764	032472			RXBA	
2451	015766	077766			<-10.>-BIT15	;MARKED BC
2452	015770	004537	025032		PERFORM ,SETREG	
2453	015774	010	011		.BYTE 010,011	;TX TABLE, RX TABLE
2454	015776	026472			TXTAB	
2455	016000	033072			RXTAB	
2456	016002	004537	025032		PERFORM ,SETREG	
2457	016006	012	013		.BYTE 012,013	;LINE PROTOCOL, LINE STATE
2458	016010	000031			BIT4+BIT3+BIT0	;CRC.CCITT, IDLE MARK
2459	016012	162004			BIT15+BIT14+BIT13+BIT10+BIT2	
2460	016014	004537	025032		PERFORM ,SETREG	;MODE 7, TXGO
2461	016020	016	017		.BYTE 016,017	;LINE PROGRESS REC, REC CNTR STORE
2462	016022	162000			BIT15+BIT14+BIT13+BIT10	;NEXT MODE=7
2463	016024	000000			0	;ZERO
2464	016026	032737	004000	001236	BIT #ASYNC,STAT	;IS THIS ASYNC LINE CARD?
2465	016034	001412			BEQ 60\$;#BR IF NO.
2466	016036	004537	025076		PERFORM ,LOAD.MODE	;#LOAD PARAMETERS.
2467	016042	020000			BIT13	;#RECEIVER ENABLE
2468	016044	004537	025076		PERFORM ,LOAD.MODE	;#
2469	016050	015000			<BIT12+BIT11>+BIT9	;#8 BITS/PER/CHAR
2470	016052	004537	025076		PERFORM ,LOAD.MODE	;#
2471	016056	072000			<BIT14+BIT13+BIT12>+BIT10	
2472						;#9600 BAUD.
2473	016060	000405			BR 61\$	
2474	016062	004537	025076	60\$:	PERFORM ,LOAD.MODE	;LOAD
2475	016066	034000			BIT13+BIT12+BIT11	;MODE AND RECV ENABLE
2476	016070	004537	024620		PERFORM ,SETSYNC	;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
2477	016074	005277	163262	61\$:	INC @DVSCR	;SET MICRO CPU GO
2478	016100	105777	163256		TSTB @DVSCR	;WAIT FOR
2479	016104	100375			BPL -4	;DVSCRO7=1
2480	016106	017704	163254		MOV @DVRIC,R4	;READ RESULT
2481	016112	010005			MOV R0,R5	;LOAD LINE NUMBER
2482	016114	000305			SWAB R5	;PUT IN HIGH BYTE
2483	016116	052705	050000		BIS #BIT14+BIT12,R5	;SET 'BLOCK CHECK COMPLETE'
2484					*****	
2485	016122	004737	023304		JSR PC,PAREN	;CHECK FOR PARITY ENABLE (REV. D0)
2486						;IF ERROR, DVRIC INCORRECT
2487					*****	
2488	016126	112777	000014	163240	4\$:	MOV #14,@DVSRSH ;GET TX MODE REGISTER
2489	016134	017704	163236		MOV @DVSR,R4	
2490	016140	012705	000007		MOV #BIT2+BIT1+BIT0,R5	;WAS NEXT MODE PICKED UP?
2491					*****	
2492	016144	004737	023304		JSR PC,PAREN	;CHECK PARITY ENABLE (REV. D0)
2493						;IF ERROR, NEXT MODE INCORRECT: S/B=7
2494					*****	
2495	016150	105277	163220	5\$:	INCB @DVSRSH	;SEL RX MODE REG
2496	016154	017704	163210		MOV @DVSR,R4	;READ
2497					*****	
2498	016160	004737	023304		JSR PC,PAREN	;CHECK PARITY ENABLE (REV. D0)
2499						;IF ERROR, RX MODE REG INCORRECT: S/B=7
2500					*****	

```

2501 016164 005005 6$: CLR R5 ;SET EXPECTED=0
2502 016166 112777 000006 163200 MOVB #6,@DVSRSH ;SEL TX BCC REG
2503 016174 017704 163176 MOV @DVSRSH,R4 ;READ
2504 016200 001401 BEQ 7$ ;BR IF=0
2505 016202 104001 HLT 1 ;IF BCC WAS SENT; BCC S/B=0
2506 *****
2507 016204 032737 000400 001244 7$: BIT #400,MASKX ;8 BITS/CHAR? (REV. D0)
2508 016212 001004 BNE 9$ ;BR IF YES
2509 016214 032737 040000 001236 BIT #PARBIT,STAT ;PARITY EN?
2510 016222 001006 BNE 8$ ;BR IF YES
2511 016224 105277 163144 9$: INCB @DVSRSH ;SEL RX BCC REG
2512 *****
2513 016230 017704 163142 MOV @DVSRSH,R4 ;READ IT
2514 016234 001401 BEQ 8$ ;
2515 016236 104001 HLT 1 ;IF RX REC'D GOOD BCC; BCC S/B=0
2516 016240 104413 8$: RAMCLR ;CLEAR ALL SEC REG
2517 016242 104401 SCOP1 ;LOCK ON CURRENT LINE?
2518 016244 005200 INC R0 ;UPDATE LINE POINTER
2519 016246 005302 DEC R2 ;4 LINE GROUP DONE?
2520 016250 001221 BNE 65$ ;BR IF NO
2521 016252 000207 RTS PC ;EXIT FOR NEXT 4 LINE GROUP
2522
2523
2524 ***** TEST 14 *****
2525 ;*TEST OF THE 'MARKED BYTE COUNT'.
2526 ;*TEST THAT WHEN BIT15=0 FOR THE RECEIVER THAT
2527 ;*BITS 13,14,15 OF LINE STATE OCCUR IN
2528 ;*THE RECEIVER MODE BITS REGISTER.
2529 ;*TEST THAT WHEN BIT15=0 FOR THE TRANSMITTER
2530 ;*THAT BITS 13,14,15 OF THE LINE PROGRESS REGISTER
2531 ;*OCCUR IN THE TRANSMITTER MODE REG.
2532 ;*ALSO VERIFY THAT BIT10=1 IN LINE STATE MAKES
2533 ;*RECEIVER 'EXPECT THE BCC'
2534 ;*AND THAT BIT10 IN LINE PROGRESS TELL TX TO SEND BCC.
2535 ;*THIS TEST USES LRC8 FOR THE POLYNOMIAL.
2536 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
2537 *****
2538
2539 ; TEST 14
2540 -----

```

```

2541 016254 012737 000014 001226 TST14: MOV #14,TSTNO
2542 016262 012737 017136 001216 MOV #TST15,NEXT
2543 016270 012700 000000 MOV #0.,R0 ;PLACE LINE NUMBER INTO R0
2544 016274 013737 001406 001244 MOV MASK.A,MASKX ;PLACE 'MASK' FOR CHARS INTO MASKX
2545 016302 013737 001422 001236 MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
2546 016310 100402 BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
2547 016312 004737 016422 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 1
2548 016316 012700 000004 100$: MOV #4.,R0 ;PLACE LINE NUMBER INTO R0
2549 016322 013737 001410 001244 MOV MASK.B,MASKX ;GET MASK
2550 016330 013737 001424 001236 MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
2551 016336 100402 BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
2552 016340 004737 016422 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 2
2553 016344 012700 000010 101$: MOV #8.,R0 ;LOAD LINE NUMBER
2554 016350 013737 001412 001244 MOV MASK.C,MASKX ;GET MASK
2555 016356 013737 001426 001236 MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
2556 016364 100402 BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED

```

2557	016366	004737	016422		JSR	PC,105\$:DO THE TEST FOR LINE CARD 3
2558	016372	012700	000014		MOV	#12.,R0	:LOAD LINE NO.
2559	016376	013737	001414	001244	MOV	MASK.D,MASKX	:GET MASK
2560	016404	013737	001430	001236	MOV	L12.15,STAT	:LOAD LINE CARD STATUS
2561	016412	100402			BMI	103\$:BR IF LINE CARD NOT TO BE TESTED
2562	016414	004737	016422		JSR	PC,105\$:DO THE TESTS FOR LINE CARD 4
2563	016420	104400			SCOPE		:SCOPE THIS TEST.
2564	016422						:TEST ENTRANCE.
2565	016422	012737	016576	001220	MOV	#65\$,LOCK	:SET RETURN IF SW09=1
2566	016430	104413			RAMCLR		:CLEAR ALL SECONDARY REGISTERS
2567	016432	012702	000004		MOV	#4,R2	:SET FOR 4 LINE GROUP
2568	016436	012704	000012		MOV	#10.,R4	:LOAD 10 BYTES
2569	016442	012705	026472		MOV	#TXTAB,R5	:WITH
2570	016446	112725	000010		MOVB	#BIT3,(R5)+	:INC/BCC
2571	016452	005304			DEC	R4	
2572	016454	001374			BNE	1\$	
2573	016456	012705	026472		MOV	#TXTAB,R5	:CLEAR
2574	016462	013704	001236		MOV	STAT,R4	:SYNC
2575	016466	042704	177400		BIC	#^C<377>,R4	:CONTROL
2576	016472	060405			ADD	R4,R5	:BYTE
2577	016474	105015			CLRB	(R5)	
2578	016476	012705	025472		MOV	#TXBAP,R5	:LOAD
2579	016502	005004			CLR	R4	:DATA
2580	016504	110425			MOVB	R4,(R5)+	:INTO
2581	016506	005204			INC	R4	:TRANSMITTER BUFFER
2582	016510	022704	000013		CMP	#11.,R4	
2583	016514	001373			BNE	2\$	
2584					;*****		
2585	016516	005003			CLR	R3	
2586	016520	005005			CLR	R5	
2587	016522	012704	033072		MOV	#RXTAB,R4	:POINT TO RX TABLE BASE ADDR.
2588	016526	013703	001244		MOV	MASKX,R3	:GET BITS/CHAR DATA
2589	016532	032703	000400		BIT	#400,R3	:8 BITS/CHAR?
2590	016536	001004			BNE	40\$:BR IF YES
2591	016540	032737	040000	001236	BIT	#PARBIT,STAT	:PAR. EN?
2592	016546	001007			BNE	41\$:BR IF YES
2593	016550	012705	000012		MOV	#10.,R5	:LOAD 10
2594	016554	112724	000010		MOVB	#BIT3,(R4)+	:RECEIVER
2595	016560	005305			DEC	R5	:CONTROL BYTES
2596	016562	001374			BNE	3\$:WITH 'INC/BCC'
2597	016564	000404			BR	65\$:SKIP OVER PAR. EN SETUP
2598	016566	012705	000012		MOV	#10.,R5	:SET UP TABLE LINE COUNTER
2599	016572	004737	023366		JSR	PC,BTCHAR	:GO SETUP PAR. EN. DATA
2600					;*****		
2601	016576	010077	162570		MOV	R0,ADVSR5	:LOAD LINE NO.
2602	016602	032737	004000	001236	BIT	#ASYNC,STAT	:#IS THIS AN ASYNC LINE CARD?
2603	016610	001406			BEQ	80\$:#BR IF NOT ASYNC.
2604	016612	004537	025032		PERFORM	SETREG	:#ADJUST FOR ASYNC LINE CARD
2605	016616	000	001		.BYTE	000,001	:#REGISTERS
2606	016620	025472			TXBAP		:#LOAD FOR ASYNC
2607	016622	077766			<-10.>-BIT15		:#LOAD FOR ASYNC
2608	016624	000405			BR	81\$:#CONTINUE TEST
2609	016626	004537	025032		PERFORM	SETREG	
2610	016632	000	001		.BYTE	000,001	:TX PRINCIPLE BA, BC
2611	016634	025470			SYNC		
2612	016636	077764			<-12.>-BIT15		:MARKED BC!

(REV. D0)

2613	016640	004537	025032	81\$:	PERFORM ,SETREG	:
2614	016644	004	005		.BYTE 004,005	:RX BA, BC
2615	016646	032472			RXBA	:
2616	016650	077766			<-10.>-BIT15	:MARKED BC!
2617	016652	004537	025032		PERFORM ,SETREG	:
2618	016656	010	011		.BYTE 010,011	:TX TABLE, RX TABLE
2619	016660	026472			TXTAB	:
2620	016662	033072			RXTAB	:
2621	016664	004537	025032		PERFORM ,SETREG	:
2622	016670	012	013		.BYTE 012,013	:LINE PROTOCOL, LINE STATE
2623	016672	000001			BIT0	:LRCC, IDLE MARK
2624	016674	162004			BIT15+BIT14+BIT13+BIT10+BIT2	:
2625	016676	004537	025032		PERFORM ,SETREG	:MODE 7, TXGO
2626	016702	016	017		.BYTE 016,017	:LINE PROGRESS REC, REC CNTR STORE
2627	016704	162000			BIT15+BIT14+BIT13+BIT10	:NEXT MODE=7
2628	016706	000000			0	:ZERO
2629	016710	032737	004000 001236		BIT #ASYNC,STAT	:IS THIS ASYNC LINE CARD?
2630	016716	001412			BEQ 60\$:#BR IF NO.
2631	016720	004537	025076		PERFORM ,LOAD.MODE	:#LOAD PARAMETERS.
2632	016724	020000			BIT13	:#RECEIVER ENABLE
2633	016726	004537	025076		PERFORM ,LOAD.MODE	:
2634	016732	015000			<BIT12+BIT11>+BIT9	:#8 BITS/PER/CHAR
2635	016734	004537	025076		PERFORM ,LOAD.MODE	:
2636	016740	072000			<BIT14+BIT13+BIT12>+BIT10	:#9600 BAUD.
2637						
2638	016742	000405			BR 61\$	
2639	016744	004537	025076	60\$:	PERFORM ,LOAD.MODE	:LOAD
2640	016750	034000			BIT13+BIT12+BIT11	:MODE AND RECV ENABLE
2641	016752	004537	024620		PERFORM ,SETSYNC	:GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
2642	016756	005277	162400	61\$:	INC @DVSCR	:SET MICRO CPU GO
2643	016762	105777	162374		TSTB @DVSCR	:WAIT FOR
2644	016766	100375			BPL -4	:DVSCRO7=1
2645	016770	017704	162372		MOV @DVRIC,R4	:READ RESULT
2646	016774	010005			MOV R0,R5	:LOAD LINE NUMBER
2647	016776	000305			SWAB R5	:PUT IN HIGH BYTE
2648	017000	052705	050000		BIS #BIT14+BIT12,R5	:SET 'BLOCK CHECK COMPLETE'
2649					*****	
2650	017004	004737	023304		JSR PC,PAREN	:CHECK FOR PARITY ENABLE (REV. D0)
2651						:IF ERROR, DVRIC INCORRECT
2652					*****	
2653	017010	112777	000014 162356	4\$:	MOVB #14,@DVSRSH	:GET TX MODE REGISTER
2654	017016	017704	162354		MOV @DVSRSH,R4	:
2655	017022	012705	000007		MOV #BIT2+BIT1+BIT0,R5	:WAS NEXT MODE PICKED UP?
2656					*****	
2657	017026	004737	023304		JSR PC,PAREN	:CHECK PARITY ENABLE (REV. D0)
2658						:IF ERROR, NEXT MODE INCORRECT: S/B=7
2659					*****	
2660	017032	105277	162336	5\$:	INCB @DVSRSH	:SEL RX MODE REG
2661	017036	017704	162334		MOV @DVSRSH,R4	:READ
2662					*****	
2663	017042	004737	023304		JSR PC,PAREN	:CHECK PARITY ENABLE (REV. D0)
2664						:IF ERROR, RX MODE REG INCORRECT: S/B=7
2665					*****	
2666	017046	005005		6\$:	CLR R5	:SET EXPECTED=0
2667	017050	112777	000006 162316		MOVB #6,@DVSRSH	:SEL TX BCC REG
2668	017056	017704	162314		MOV @DVSRSH,R4	:READ

```

2669 017062 001401      BEQ      7$      ;BR IF=0
2670 017064 104001      HLT      1      ;IF BCC WAS SENT; BCC S/B=0
2671 *****
2672 017066 032737 000400 001244 7$: BIT      #400,MASKX ;8 BITS/CHAR? (REV. DO)
2673 017074 001004      BNE      9$      ;BR IF YES
2674 017076 032737 040000 001236      BIT      #PARBIT,STAT ;PARITY EN?
2675 017104 001006      BNE      8$      ;BR IF YES
2676 017106 105277 162262      9$: INCB     @DVSRSH ;SEL RX BCC REG
2677 *****
2678 017112 017704 162260      MOV      @DVSR,R4 ;READ IT
2679 017116 001401      BEQ      8$      ;
2680 017120 104001      HLT      1      ;IF RX RECVD GOOD BCC; BCC S/B=0
2681 017122 104413      8$: RAMCLR ;CLEAR ALL SEC REG
2682 017124 104401      SCOP1 ;LOCK ON CURRENT LINE?
2683 017126 005200      INC      R0      ;UPDATE LINE POINTER
2684 017130 005302      DEC      R2      ;4 LINE GROUP DONE?
2685 017132 001221      BNE      65$     ;BR IF NO
2686 017134 000207      RTS      PC      ;EXIT FOR NEXT 4 LINE GROUP
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

```

```

***** TEST 15 *****
*TEST OF RECIEVER AND TRANSMITTER MODE BITS.
*TEST TO TRANSMIT AND RECEIVE
*A DIFFERENT CHAR FROM EACH
*MODE. THE TX TABLE WILL BE
*FILLED WITH "SEND DLE" SO IF CHAR
*GOES TO WRONG TABLE RX WILL
*RECEIVE A DLE CHAR(31). THE RX
*FILLS TABLE WITH "INCLUDE IN BCC"
*SO THAT IF RECV GOES TO WRONG
*TABLE THE RX BCC REG WILL
*BE NON-ZERO!
*CHAR CURRENT MODE NEXT MODE
* 15 0 1
* 16 1 2
* 21 2 3
* 23 3 4
* 25 4 5
* 7 5 6
* 34 6 7
* 32 7 7
* 36 7 7
*
*
*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
*****

```

```

; TEST 15
-----
2720 017136 012737 000015 001226 TST15: MOV      #15,TSTNO
2721 017144 012737 020260 001216      MOV      #TST16,NEXT
2722 017152 012700 000000      MOV      #0.,R0 ;PLACE LINE NUMBER INTO R0
2723 017156 013737 001406 001244      MOV      MASK.A,MASKX ;PLACE 'MASK' FOR CHARS INTO MASKX
2724 017164 013737 001422 001236      MOV      L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT

```

2725	017172	100402			BMI	100\$:BR IF LINE CARD NOT TO BE TESTED
2726	017174	004737	017304		JSR	PC,105\$:GO TO THE TEST FOR LINE CARD 1
2727	017200	012700	000004	100\$:	MOV	#4.,R0	:PLACE LINE NUMBER INTO R0
2728	017204	013737	001410	001244	MOV	MASK.B,MASKX	:GET MASK
2729	017212	013737	001424	001236	MOV	L04.07,STAT	:LOAD LINE CARD STATUS INTO STAT
2730	017220	100402			BMI	101\$:BR IF LINE CARD NOT TO BE TESTED
2731	017222	004737	017304		JSR	PC,105\$:GO TO THE TEST FOR LINE CARD 2
2732	017226	012700	000010	101\$:	MOV	#8.,R0	:LOAD LINE NUMBER
2733	017232	013737	001412	001244	MOV	MASK.C,MASKX	:GET MASK
2734	017240	013737	001426	001236	MOV	L08.11,STAT	:LOAD LINE CARD STATUS INTO STAT
2735	017246	100402			BMI	102\$:BR IF LINE CARD NOT TO BE TESTED
2736	017250	004737	017304		JSR	PC,105\$:DO THE TEST FOR LINE CARD 3
2737	017254	012700	000014	102\$:	MOV	#12.,R0	:LOAD LINE NO.
2738	017260	013737	001414	001244	MOV	MASK.D,MASKX	:GET MASK
2739	017266	013737	001430	001236	MOV	L12.15,STAT	:LOAD LINE CARD STATUS
2740	017274	100402			BMI	103\$:BR IF LINE CARD NOT TO BE TESTED
2741	017276	004737	017304		JSR	PC,105\$:DO THE TESTS FOR LINE CARD 4
2742	017302	104400		103\$:	SCOPE		:SCOPE THIS TEST.
2743	017304			105\$:			:TEST ENTRANCE.
2744	017304	012737	017670	001220	MOV	#12\$,LOCK	:LOCK ON LINE RETURN
2745	017312	104413			RAMCLR		:CLEAR ALL SEC REGISTERS
2746	017314	012705	026472		MOV	#TXTAB,R5	:LOAD
2747	017320	012704	033072		MOV	#RXTAB,R4	
2748	017324	012701	004000		MOV	#4000,R1	:ALL CNTRL BYTES
2749	017330	112725	000002	1\$:	MOVB	#BIT1,(R5)+	:WITH 'SND/DLE'
2750	017334	112724	000010	2\$:	MOVB	#BIT3,(R4)+	:WITH 'INCL/BCC'
2751	017340	005301			DEC	R1	
2752	017342	001372			BNE	1\$	
2753	017344	004537	024620		PERFORM	,SETSYNC	:GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
2754	017350	012702	000004	11\$:	MOV	#4,R2	:SET FOR 4 LINE GROUP
2755	017354	010246			MOV	R2,-(SP)	:SAVE R2 (REV. DO)
2756	017356	113705	001236		MOVB	STAT,R5	:CLEAR
2757	017362	042705	177400		BIC	#^C<377>,R5	:SYNC
2758	017366	012704	026472		MOV	#TXTAB,R4	:ENTRY
2759	017372	060504			ADD	R5,R4	:IN
2760	017374	105014			CLRB	(R4)	:CONTROL TABLE
2761					:*****		
2762	017376	013703	001244		MOV	MASKX,R3	:GET BITS/CHAR DATA (REV. DO)
2763	017402	032703	000400		BIT	#400,R3	:8 BITS/CHAR?
2764	017406	001004			BNE	21\$:BR IF YES
2765	017410	032737	040000	001236	BIT	#PARBIT,STAT	:PAR. EN?
2766	017416	001003			BNE	22\$:BR IF YES
2767	017420	012702	024124	21\$:	MOV	#RXBDAT,R2	:POINT TO 8 BIT DATA
2768	017424	000443			BR	29\$:GO LOAD TABLE
2769	017426	132703	000040	22\$:	BITB	#40,R3	:5 BITS?
2770	017432	001412			BEQ	24\$:BR IF NOT
2771	017434	032737	020000	001236	BIT	#BIT13,STAT	:EVEN PAR EN?
2772	017442	001403			BEQ	23\$:BR IF NOT
2773	017444	012702	024256		MOV	#RXSEVN,R2	
2774	017450	000431			BR	29\$	
2775	017452	012702	024300	23\$:	MOV	#RX5ODD,R2	
2776	017456	000426			BR	29\$	
2777	017460	132703	000100	24\$:	BITB	#100,R3	:6 BITS?
2778	017464	001012			BNE	26\$:BR IF NOT
2779	017466	032737	020000	001236	BIT	#BIT13,STAT	:EVEN PAR EN?
2780	017474	001403			BEQ	25\$:BR IF NOT

2781	017476	012702	024212		MOV	#RX6EVN,R2	:
2782	017502	000414			BR	29\$:
2783	017504	012702	024234	25\$:	MOV	#RX6ODD,R2	:
2784	017510	C00411			BR	29\$:
2785	017512	032737	020000	001236	26\$:	BIT	#BIT13,STAT
2786	017520	001403			BEQ	27\$:EVN PAR EN?
2787	017522	012702	024146		MOV	#RX7EVN,R2	:BR IF NOT
2788	017526	000402			BR	29\$:
2789	017530	012702	024170	27\$:	MOV	#RX7ODD,R2	:
2790					:*****		
2791	017534	112737	000040	026507	29\$:	MOVB	#BIT5,TXTAB+15
2792	017542	112737	000100	027110		MOVB	#BIT6,TXTAB+BIT8+16
2793	017550	112737	000140	027513		MOVB	#BIT6+BIT5,TXTAB+BIT9+21
2794	017556	112737	000200	030115		MOVB	#BIT7,TXTAB+BIT9+BIT8+23
2795	017564	112737	000240	030517		MOVB	#BIT7+BIT5,TXTAB+BIT10+25
2796	017572	112737	000300	031101		MOVB	#BIT7+BIT6,TXTAB+BIT10+BIT8+7
2797	017600	112737	000340	031526		MOVB	#BIT7+BIT6+BIT5,TXTAB+BIT10+BIT9+34
2798	017606	112737	000340	032124		MOVB	#BIT7+BIT6+BIT5,TXTAB+BIT10+BIT9+BIT8+32
2799	017614	112737	000340	032130		MOVB	#BIT7+BIT6+BIT5,TXTAB+BIT10+BIT9+BIT8+36
2800					:*****		
2801						MOVB	#BIT5,a(R2)+
2802	017622	112732	000040			MOVB	#BIT6,a(R2)+
2803	017626	112732	000100			MOVB	#BIT6+BIT5,a(R2)+
2804	017632	112732	000140			MOVB	#BIT7,a(R2)+
2805	017636	112732	000200			MOVB	#BIT7+BIT5,a(R2)+
2806	017642	112732	000240			MOVB	#BIT7+BIT6,a(R2)+
2807	017646	112732	000300			MOVB	#BIT7+BIT6+BIT5,a(R2)+
2808	017652	112732	000340			MOVB	#BIT7+BIT6+BIT5,a(R2)+
2809	017656	112732	000340			MOVB	#BIT7+BIT6+BIT5,a(R2)+
2810	017662	112732	000340			MOVB	#BIT7+BIT6+BIT5,a(R2)+
2811	017666	012602			MOV	(SP)+,R2	:RESTORE R2
2812					:*****		
2813							
2814	017670	012705	032472		12\$:	MOV	#RXBA,R5
2815	017674	005025				CLR	(R5)+
2816	017676	005025				CLR	(R5)+
2817	017700	005025				CLR	(R5)+
2818							: R
2819	017702	005025				CLR	(R5)+
2820	017704	005025				CLR	(R5)+
2821	017706	012705	025472			MOV	#TXBAP,R5
2822							:L
2823	017712	012725				MOV	(PC)+,(R5)+
2824	017714	015	016			.BYTE	15,16
2825	017716	012725				MOV	(PC)+,(R5)+
2826	017720	021	023			.BYTE	21,23
2827	017722	012725				MOV	(PC)+,(R5)+
2828	017724	025	007			.BYTE	25,7
2829	017726	012725				MOV	(PC)+,(R5)+
2830	017730	034	032			.BYTE	34,32
2831	017732	112725	000036			MOVB	#36,(R5)+
2832	017736	010077	161430			MOV	R0,ADVSR5
2833	017742	032737	004000	001236		BIT	#ASYNC,STAT
2834	017750	001406				BEQ	80\$
2835	017752	004537	025032			PERFORM	SETREG
2836	017756	000	001			.BYTE	000,001

(REV. DC)

:SET RX POINTER
 :Z
 : E
 : R
 : O
 :BUFFER!
 :L
 : O
 : A
 : D
 : T
 : R
 : A
 : N
 : S
 :BUFFER
 :LOAD LINE NO.
 :#IS THIS AN ASYNC LINE CARD?
 :#BR IF NOT ASYNC.
 :#ADJUST FOR ASYNC LINE CARD
 :#REGISTERS

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 56
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0067

2837	017760	025472				TXBAP		:#LOAD FOR ASYNC
2838	017762	177767				-9.		:#LOAD FOR ASYNC
2839	017764	000405				BR	81\$:#CONTINUE TEST
2840	017766	004537	025032		80\$:	PERFORM	SETREG	
2841	017772	000	001			.BYTE	000,001	:PRINCIPLE BA, BC
2842	017774	025470				SYNC		
2843	017776	177765				-11.		
2844	020000	004537	025032		81\$:	PERFORM	SETREG	
2845	020004	004	005			.BYTE	004,005	:RX BA, BC
2846	020006	032472				RXBA		
2847	020010	177767				-9.		
2848	020012	004537	025032			PERFORM	SETREG	
2849	020016	010	011			.BYTE	010,011	:TX TABLE, RX TAB
2850	020020	026472				TXTAB		
2851	020022	033072				RXTAB		
2852	020024	004537	025032			PERFORM	SETREG	
2853	020030	012	013			.BYTE	012,013	:LINE PROTOCOL, LINE STATE
2854	020032	014400				31*400		:31 IN HIGH BYTE
2855	020034	000004				BIT2		:TX GO
2856	020036	032737	004000	001236		BIT	#ASYNC,STAT	:#IS THIS ASYNC LINE CARD?
2857	020044	001412				BEQ	60\$:#BR IF NO.
2858	020046	004537	025076			PERFORM	LOAD.MODE	:#LOAD PARAMETERS.
2859	020052	020000				BIT13		:#RECEIVER ENABLE
2860	020054	004537	025076			PERFORM	LOAD.MODE	
2861	020060	015000				<BIT12+BIT11>+BIT9		:#8 BITS/PER/CHAR
2862	020062	004537	025076			PERFORM	LOAD.MODE	
2863	020066	072000				<BIT14+BIT13+BIT12>+BIT10		:#9600 BAUD.
2864								
2865	020070	000403				BR	61\$	
2866	020072	004537	025076		60\$:	PERFORM	LOAD.MODE	:LOAD
2867	020076	034000				BIT13+BIT12+BIT11		:MODE AND RX ENABLE
2868	020100	005277	161256		61\$:	INC	@DVSCR	:SET MICRO CPU GO
2869	020104	105777	161252			TSTB	@DVSCR	:WAIT FOR
2870	020110	100375				BPL	-4	:DVSCR07=1
2871	020112	012701	025472			MOV	#TXBAP,R1	:SET TX POINTER
2872	020116	012703	032472			MOV	#RXBA,R3	:SET RX POINTER
2873	020122	012737	000011	001246		MOV	#9.,TEMP1	:CHECK 9. CHAR
2874	020130	005005				CLR	R5	
2875	020132	005004				CLR	R4	
2876	020134	112105			3\$:	MOVB	(R1)+,R5	:SET EXPECTED
2877	020136	112304				MOVB	(R3)+,R4	:SET FOUND
2878								
2879	020140	004737	023304			JSR	PC,PAREN	:CHECK PARITY EN. AND DATA (REV. D0)
2880								:IF ERROR, DATA DID NOT COMPARE (IS IT IDLE?)
2881								
2882	020144	001401				BEQ	4\$	
2883	020146	104001				HLT	1	:DATA COMPARE ERROR (IS IT IDLE?)
2884	020150	005337	001246		4\$:	DEC	TEMP1	:ALL CHARS DONE?
2885	020154	001367				BNE	3\$:BR IF NO
2886	020156	005005				CLR	R5	
2887								
2888	020160	032737	000400	001244		BIT	#400,MASKX	:8 BITS/CHAR? (REV. D0)
2889	020166	001004				BNE	8\$:BR IF YES
2890	020170	032737	040000	001236		BIT	#PARBIT,STAT	:PAR. EN?
2891	020176	001022				BNE	7\$:BR IF YES
2892	020200	112777	000007	161166	8\$:	MOVB	#7,@DVSRSH	:SEL RX BCC REG


```

2893 ;*****
2894 020206 017704 161164      MOV      @DVSRA,R4      ;READ IT
2895
2896 020212 001401      BEQ      5$      ;IF RX WENT TO GOOD CNTRL BYTE;
2897 020214 104001      HLT      1      ;RX BCC S/B=0
2898 020216 012705 000007      5$: MOV      #7,R5      ;SET MODE=D
2899 020222 112777 000014 161144  MOVB     #14,@DVSRSH ;SEL TX MODE REG
2900 020230 017704 161142      MOV      @DVSRA,R4      ;READ TX MODE REG
2901 020234 105277 161134      6$: INCB     @DVSRSH ;SEL RX MODE REG
2902 020240 017704 161132      MOV      @DVSRA,R4      ;READ IT
2903 020244 104412      7$: MSTCLR ;INIT DV11
2904
2905 020246 104401      SCOP1      ;LOCK ON CURRENT LINE.
2906 020250 005200      INC      R0      ;INC LINE POINTER
2907 020252 005302      DEC      R2      ;4 LINE GROUP DONE?
2908 020254 001205      BNE      12$     ;BR IF NO
2909 020256 000207      RTS      PC      ;EXIT FOR NEXT GROUP OF LINES
2910
2911 ;***** TEST 16 *****
2912 ;*TEST OF RECEIVER AND TRANSMITTER MULTIPLE FUNCTIONS.
2913 ;*TEST OF RECV BCC AND TRANS BCC.
2914
2915 ;*CHAR      RX FUNC.      TX FUNC.
2916 ;* 0          INC/BCC      INC/BCC
2917 ;* 1          INC/BCC/DSCARD INC/BCC
2918
2919 ;* 2          INC/BCC      INC/BCC/SND/DLE
2920 ;* 3          INC/BCC      INC/BCC
2921
2922 ;* 4          NO FUNC      SND/DLE
2923 ;* 5          INC/BCC/DSCARD INC/BCC
2924 ;* 6          INC/BCC/EXP/RCC INC/BCC/SND/BCC
2925 ;*          NEXT MODE =7   NEXT MODE =7
2926 ;*
2927 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
2928 ;*****
2929
2930
2931 ; TEST 16
2932 ;-----
2933 020260 012737 000016 001226 TST16: MOV      #16,TSTNO
2934 020266 012737 021256 001216      MOV      #TST17,NEXT
2935 020274 012700 000000      MOV      #0.,R0      ;PLACE LINE NUMBER INTO R0
2936 020300 013737 001406 001244      MOV      MASK.A,MASKX ;PLACE 'MASK' FOR CHARS INTO MASKX
2937 020306 013737 001422 001236      MOV      L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
2938 020314 100402      BMI      100$     ;BR IF LINE CARD NOT TO BE TESTED
2939 020316 004737 020426      JSR      PC,105$   ;GO DO THE TEST FOR LINE CARD 1
2940 020322 012700 000004 100$: MOV      #4.,R0      ;PLACE LINE NUMBER INTO R0
2941 020326 013737 001410 001244      MOV      MASK.B,MASKX ;GET MASK
2942 020334 013737 001424 001236      MOV      L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
2943 020342 100402      BMI      101$     ;BR IF LINE CARD NOT TO BE TESTED
2944 020344 004737 020426      JSR      PC,105$   ;GO DO THE TEST FOR LINE CARD 2
2945 020350 012700 000010 101$: MOV      #8.,R0      ;LOAD LINE NUMBER
2946 020354 013737 001412 001244      MOV      MASK.C,MASKX ;GET MASK
2947 020362 013737 001426 001236      MOV      L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
2948 020370 100402      BMI      102$     ;BR IF LINE CARD NOT TO BE TESTED

```

2949	020372	004737	020426			JSR	PC,105\$;DO THE TEST FOR LINE CARD 3
2950	020376	012700	000014		102\$:	MOV	#12.,R0		;LOAD LINE NO.
2951	020402	013737	001414	001244		MOV	MASK.D,MASKX		;GET MASK
2952	020410	013737	001430	001236		MOV	L12.15,STAT		;LOAD LINE CARD STATUS
2953	020416	100402				BMI	103\$;BR IF LINE CARD NOT TO BE TESTED
2954	020420	004737	020426			JSR	PC,105\$;DO THE TESTS FOR LINE CARD 4
2955	020424	104400			103\$:	SCOPE			;SCOPE THIS TEST.
2956	020426				105\$:				;TEST ENTRANCE.
2957	020426	012737	020626	001220		MOV	#3\$,LOCK		;RETURN IF SW09=1
2958	020434	032737	001400	001236		BIT	#BIT9+BIT8,STAT		; '8 BITS/PER/CHAR ?'
2959	020442	001401				BEQ	+.4		;BR IF YES
2960	020444	000207				RTS	PC		;EXIT TEST FOR THIS LINE CARD!
2961	020446	104413				RAMCLR			;CLEAR ALL SEC REGISTERS
2962	020450	012705	026472			MOV	#TXTAB,R5		;CLEAR
2963	020454	012703	033072			MOV	#RXTAB,R3		;TRANSMITTER
2964	020460	005004				CLR	R4		;AND
2965	020462	005025			1\$:	CLR	(R5)+		;RECEIVER
2966									
2967	020464	005023				CLR	(R3)+		;CONTROL
2968	020466	105204				INCB	R4		;TABLES
2969	020470	100374				BPL	1\$		
2970	020472	012705	000010			MOV	#BIT3,R5		;INC/BCC IS IN R5
2971	020476	110537	026472			MOVB	R5,TXTAB		;INC/BCC
2972	020502	110537	026473			MOVB	R5,TXTAB+1		;INC/BCC
2973	020506	110537	026474			MOVB	R5,TXTAB+2		;INC/BCC
2974									
2975	020512	152737	000002	026474		BISB	#BIT1,TXTAB+2		; SND/DLE
2976	020520	110537	026475			MOVB	R5,TXTAB+3		;INC/BCC
2977	020524	112737	000002	026476		MOVB	#BIT1,TXTAB+4		;SND/DLE
2978	020532	110537	026477			MOVB	R5,TXTAB+5		;INC/BCC
2979	020536	110537	026500			MOVB	R5,TXTAB+6		;INC/BCC
2980	020542	052737	000344	026500		BIS	#BIT7+BIT6+BIT5+BIT2,TXTAB+6		;INC/BCC SND/BCC MOD=7
2981	020550	110537	033072			MOVB	R5,RXTAB		;INC/BCC
2982									
2983	020554	110537	033073			MOVB	R5,RXTAB+1		;INC/BCC
2984	020560	152737	000020	033073		BISB	#BIT4,RXTAB+1		; DSCARD
2985	020566	110537	033074			MOVB	R5,RXTAB+2		;INC/BCC
2986	020572	110537	033075			MOVB	R5,RXTAB+3		;INC/BCC
2987	020576	105037	033076			CLRB	RXTAB+4		;NO FUNC.
2988	020602	110537	033077			MOVB	R5,RXTAB+5		;INC/BCC
2989	020606	152737	000020	033077		BISB	#BIT4,RXTAB+5		; DSCARD
2990	020614	112737	000354	033100		MOVB	#BIT7+BIT6+BIT5+BIT3+BIT2,RXTAB+6		;INC/BCC EXP/BCC MODE=7
2991	020622	012702	000004			MOV	#4,R2		;SET FOR 4 LINE GROUP
2992	020626	005037	032472		3\$:	CLR	RXBA		;ZERO
2993	020632	005037	032474			CLR	RXBA+2		;RX
2994									
2995	020636	005037	032476			CLR	RXBA+4		;BUFFER
2996	020642	005037	032500			CLR	RXBA+6		;AREA
2997	020646	010077	160520			MOV	R0,@DVSRS		;LOAD LINE NO.
2998	020652	032737	004000	001236		BIT	#ASYNC,STAT		;#IS THIS AN ASYNC LINE CARD?
2999	020660	001406				BFO	80\$;#BR IF NOT ASYNC.
3000	020662	004537	025032			PERFORM	SETREG		;#ADJUST FOR ASYNC LINE CARD
3001	020666	000	001			.BYTE	000,001		;#REGISTERS
3002	020670	025472				TXBAP			;#LOAD FOR ASYNC
3003	020672	177771				-7.			;#LOAD FOR ASYNC
3004	020674	000405				BR	81\$;#CONTINUE TEST

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 59
 CZDVD.D.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0070

```

3005 020676 004537 025032      80$:  PERFORM ,SETREG      ;
3006 020702      000      001      .BYTE 000,001      ;PRINCIPLE BA, BC
3007 020704 025470      SYNC      ;
3008 020706 177767      -9.      ;
3009 020710 004537 025032      81$:  PERFORM ,SETREG      ;
3010 020714      004      005      .BYTE 004,005      ;RX BA, BC
3011 020716 032472      RXBA      ;
3012 020720 177766      -10.      ;
3013
3014 020722 004537 025032      PERFORM ,SETREG      ;
3015 020726      010      011      .BYTE 010,011      ;TX TAB, RXTAB
3016 020730 026472      TXTAB      ;
3017 020732 033072      RXTAB      ;
3018 020734 004537 025032      PERFORM ,SETREG      ;
3019 020740      013      012      .BYTE 013,012      ;LINE STATE, LINE PROTOCOL
3020 020742 000004      BIT2      ;TX GO
3021 020744 010031      <20*400>+BIT4+BIT3+BIT0 ;DLE(20 HIGH BYTE),CRC.CCITT, IDLE MARK
3022 020746 032737 004000 001236  BIT      #ASYNC,STAT ;IS THIS ASYNC LINE CARD?
3023 020754 001412      BEQ      60$      ;#BR IF NO.
3024 020756 004537 025076      PERFORM ,LOAD.MODE      ;#LOAD PARAMETERS.
3025 020762 020000      BIT13      ;#RECEIVER ENABLE
3026 020764 004537 025076      PERFORM ,LOAD.MODE      ;#
3027 020770 015000      <BIT12+BIT11>+BIT9 ;#8 BITS/PER/CHAR
3028 020772 004537 025076      PERFORM ,LOAD.MODE      ;#
3029 020776 072000      <BIT14+BIT13+BIT12>+BIT10 ;#9600 BAUD.
3030
3031 021000 000405      BR      4$
3032 021002 004537 025076      60$:  PERFORM ,LOAD.MODE      ;LOAD
3033 021006 034000      BIT13+BIT12+BIT11 ;MODE AND RX ENABLE
3034 021010 004537 024620      PERFORM ,SETSYNC      ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
3035 021014 005004      4$:  CLR      R4      ;LOAD
3036 021016 012705 025472      MOV      #TXBAP,R5 ;TX
3037 021022 110425      5$:  MOVB      R4,(R5)+ ;DATA
3038 021024 005204      INC      R4      ;
3039 021026 020427 000007      CMP      R4,#7 ;
3040 021032 001373      BNE      5$      ;
3041 021034 005277 160322      INC      @DVSCR ;SET MICRO CODE GO
3042 021040 105777 160316      TSTB      @DVSCR ;WAIT FOR
3043 021044 100375      BPL      -4 ;DVSCR07=1
3044 021046 012701 032472      MOV      #RXBA,R1 ;GET RX POINTER
3045 021052 012703 021246      MOV      #50$,R3 ;GET DATA EXPECTED POINTER
3046 021056 012737 000007 001252  MOV      #7.,TEMP3 ;CHECK 7 CHARS
3047 021064 112104      6$:  MOVB      (R1)+,R4 ;GET RECEIVED CHAR
3048
3049 021066 112305      MOVB      (R3)+,R5 ;GET EXPECTED CHAR
3050      ;*****8**
3051 021070 004737 023304      JSR      PC,PAREN ;CHECK PARITY EN. & DATA (REV. DO)
3052      ;IF ERROR, RX DATA BAD
3053      ;*****
3054 021074 005337 001252      7$:  DEC      TEMP3 ;ALL CHARS DONE?
3055 021100 001371      BNE      6$      ;BR IF NO
3056
3057 021102 112777 000014 160264  MOVB      #14,@DVSRSH ;GET TX MODE REG.
3058 021110 017704 160262      MOV      @DVSR,R4 ;
3059 021114 042704 177770      BIC      #^C<BIT2+BIT1+BIT0>,R4 ;CLEAR JUNK
3060 021120 012705 000007      MOV      #7,R5 ;SET EXPECTED=7

```

```

3061
3062 021124 004737 023304      ;*****
3063                               JSR      PC,PAREN      ;CHECK PARITY EN. & DATA (REV. DO)
3064                               ;IF ERROR, TX MODE REG NOT = 7!
3065 021130 105277 160240      8$: INCB     @DVSRSH      ;RX MODE REG
3066 021134 017704 160236      MOV      @DVSRRA,R4
3067 021140 042704 177770      BIC      #^C<BIT2+BIT1+BIT0>,R4
3068                               ;*****
3069 021144 004737 023304      JSR      PC,PAREN      ;CHECK PARITY EN. & DATA (REV. DO)
3070                               ;IF ERROR, RX MODE REG NOT = 7!
3071                               ;*****
3072 021150 112777 000006 160216 9$: MOV      #6,@DVSRSH      ;TX BCC REG
3073 021156 017704 160214      MOV      @DVSRRA,R4
3074 021162 001402              BEQ      10$
3075 021164 005005              CLR      R5
3076 021166 104001              HLT      1
3077 021170 105277 160200      10$: INCB     @DVSRSH      ;TX BCC REG S/B=0
3078 021174 017704 160176      MOV      @DVSRRA,R4
3079 021200 001402              BEQ      11$
3080 021202 005005              CLR      R5
3081 021204 104001              HLT      1
3082 021206 010005      11$: MOV      R0,R5      ;RX BCC REG S/B=0
3083 021210 000305              SWAB     R5      ;LOAD LINE NO.
3084 021212 052705 050000      BIS      #BIT14+BIT12,R5 ;PUT IN HIGH BYTE
3085 021216 017704 160144      MOV      @DVRRIC,R4      ;SET BCC COMPLETE
3086                               ;*****
3087 021222 004737 023304      JSR      PC,PAREN      ;CHECK PARITY EN. & DATA (REV. DO)
3088                               ;IF ERROR, DVRIC INCORRECT
3089                               ;*****
3090 021226 104413      12$: RAMCLR      ;CLEAR ALL SEC REGS
3091 021230 104401      SCOP1          ;RETURN WITH SAME LINE
3092 021232 005200      INC      R0      ;UPDATE LINE POINTER
3093 021234 005302      DEC      R2      ;4 LINES DONE?
3094 021236 001402      BEQ      .+6     ;BR IF NO
3095 021240 000137 020626      JMP      3$      ;JMP IF YES
3096 021244 000207      RTS      PC      ;EXIT
3097 021246 000 020      50$: .BYTE 0,20
3098 021250 002 003      .BYTE 2,3
3099 021252 020 004      .BYTE 20,4
3100 021254 006 000      .BYTE 6,0

```

```

3101
3102      ;***** TEST 17 *****
3103      ;*TEST OF RECEIVER RESYNC
3104      ;*TEST TO TRANSMIT A BLOCK OD
3105      ;*DATA (SYN,SYN,1,2,3,4,5)
3106      ;*HAVING CHAR '1' BEING A 'SPECIAL CHAR' TO THE RECEIVER
3107      ;*AT WHICH TIME A 'RE-SYNC' PULSE WILL BE ISSUED
3108      ;*AND A RESTART CHAR PROC. (DVSCRO8=1) WILL BE DONE.
3109      ;*WHEN THE TRANSMITTER IS DONE (DVSCR15=1) A SECOND
3110      ;*BLOCK OF DATA (SYN,SYN,SYN,SYN,6,7,10)
3111      ;*WILL BE SENT EXPECTING THAT THE NEXT TIME DVSCRO7=1
3112      ;*THAT THE DVRIC WILL HAVE:
3113      ;*14=1 11:08=LINE NO. 07:00= '10'
3114      ;*RXBUFFER (CORE) S/B= 1,6,7,10.
3115      ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
3116      ;*****

```

```

3117
3118
3119      : TEST 17
3120      :-----
3120 021256 012737 000017 001226 1ST17: MOV #17,TSTNO
3121 021264 012737 022506 001216      MOV #TST20,NEXT
3122 021272 012700 000000      MOV #0.,R0      ;PLACE LINE NUMBER INTO R0
3123 021276 013737 001406 001244      MOV MASK.A,MASKX ;PLACE 'MASK'FOR CHARS INTO MASKX
3124 021304 013737 001422 001236      MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
3125 021312 100402      BMI 100$      ;BR IF LINE CARD NOT TO BE TESTED
3126 021314 004737 021424      JSR PC,105$      ;GO DO THE TEST FOR LINE CARD 1
3127 021320 012700 000004      100$: MOV #4.,R0      ;PLACE LINE NUMBER INTO R0
3128 021324 013737 001410 001244      MOV MASK.B,MASKX ;GET MASK
3129 021332 013737 001424 001236      MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
3130 021340 100402      BMI 101$      ;BR IF LINE CARD NOT TO BE TESTED
3131 021342 004737 021424      JSR PC,105$      ;GO DO THE TEST FOR LINE CARD 2
3132 021346 012700 000010      101$: MOV #8.,R0      ;LOAD LINE NUMBER
3133 021352 013737 001412 001244      MOV MASK.C,MASKX ;GET MASK
3134 021360 013737 001426 001236      MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
3135 021366 100402      BMI 102$      ;BR IF LINE CARD NOT TO BE TESTED
3136 021370 004737 021424      JSR PC,105$      ;DO THE TEST FOR LINE CARD 3
3137 021374 012700 000014      102$: MOV #12.,R0     ;LOAD LINE NO.
3138 021400 013737 001414 001244      MOV MASK.D,MASKX ;GET MASK
3139 021406 013737 001430 001236      MOV L12.15,STAT ;LOAD LINE CARD STATUS
3140 021414 100402      BMI 103$      ;BR IF LINE CARD NOT TO BE TESTED
3141 021416 004737 021424      JSR PC,105$      ;DO THE TESTS FOR LINE CARD 4
3142 021422 104400      103$: SCOPE      ;SCOPE THIS TEST.
3143 021424      105$:      ;TEST ENTRANCE.
3144 021424 012737 021516 001220      MOV #1$,LOCK      ;SET RETURN
3145 021432 104413      RAMCLR      ;CLEAR ALL SEC. REGS
3146      ;*****
3147 021434 012705 026472      MOV #TXTAB,R5      ;CLEAR (REV. DO)
3148 021440 012703 033072      MOV #RXTAB,R3      ;TRANSMITTER
3149 021444 005004      CLR R4      ;AND
3150 021446 005025      45$: CLR (R5)+      ;RECEIVER
3151 021450 005023      CLR (R3)+      ;CONTROL
3152 021452 105204      INCB R4      ;TABLES
3153 021454 100374      BPL 45$      ;
3154 021456 032737 020000 001236      BIT #20000,STAT ;EVN PAR. EN?
3155 021464 001010      BNE 20$      ;BR IF TRUE
3156      ;*****
3157 021466 112737 000001 033073      MOVB #BIT0,RXTAB+1 ;SET 'SPECIAL CHAR' CNTRL BYTE
3158 021474 005037 033100      CLR RXTAB+6      ;CLEAR
3159 021500 005037 033102      CLR RXTAB+10     ;OTHER CNTRL BYTES
3160 021504 000402      BR 21$      ;SKIP OVER EVEN SETUP (REV. DO)
3161 021506 004737 024322      20$: JSR PC,EVDAT      ;GO GET EVEN DATA (REV. DO)
3162 021512 012702 000004      21$: MOV #4,R2      ;SET FOR 4 LINE GROUP (REV. DO)
3163 021516 010077 157650      1$: MOV R0,#DVSRS      ;LOAD LINE NUMBER
3164 021522 032737 004000 001236      BIT #ASYNC,STAT ;#IS THIS AN ASYNC LINE CARD?
3165 021530 001406      BEQ 80$      ;#BR IF NOT ASYNC.
3166 021532 004537 025032      PERFORM ,SETREG ;#ADJUST FOR ASYNC LINE CARD
3167 021536 000 001      .BYTE 000,001 ;#REGISTERS
3168 021540 025472      TXBAP      ;#LOAD FOR ASYNC
3169 021542 177773      -5      ;#LOAD FOR ASYNC
3170 021544 000405      BR 81$      ;#CONTINUE TEST
3171 021546 004537 025032      80$: PERFORM ,SETREG ;
3172 021552 000 001      .BYTE 000,001 ;RX BA P, RX BC P

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 62
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0073

3173	021554	025470			SYNC	:
3174	021556	177771			-7	:
3175	021560	004537	025032	81\$:	PERFORM SETREG	:
3176	021564	004	005		.BYTE 004,005	:RX BA RX BC
3177	021566	032472			RXBA	:
3178	021570	177774			-4	:
3179	021572	004537	025032		PERFORM SETREG	:
3180	021576	010	011		.BYTE 010,011	:RX CNTRL TAB, RX CNTRL TAB
3181	021600	026472			TXTAB	:
3182	021602	033072			RXTAB	:
3183	021604	004537	025032		PERFORM SETREG	:
3184	021610	013	012		.BYTE 013,012	:LINE STATE, LINE PROTOCOL PARAMS.
3185	021612	000004			BIT2	:TX GO
3186	021614	000101			BIT6+BIT0	:TX,DDCMP, IDLE MARK
3187	021616	032737	004000 001236		BIT #ASYNC,STAT	:#IS THIS ASYNC LINE CARD?
3188	021624	001412			BEQ 60\$:#BR IF NO.
3189	021626	004537	025076		PERFORM LOAD.MODE	:#LOAD PARAMETERS.
3190	021632	020000			BIT13	:#RECEIVER ENABLE
3191	021634	004537	025076		PERFORM LOAD.MODE	:
3192	021640	015000			<BIT12+BIT11>+B'	:#8 BITS/PER/CHAR
3193	021642	004537	025076		PERFORM LOAD.MODE	:
3194	021646	072000			<BIT14+BIT13+BIT12>+BIT10	:#9600 BAUD.
3195						
3196	021650	000403			BR 61\$	
3197	021652	004537	025076	60\$:	PERFORM LOAD.MODE	:LOAD
3198	021656	034000			BIT13+BIT12+BIT11	:MODE + RX ENABLE
3199	021660	005037	032472	61\$:	CLR RXBA	:CLEAR
3200	021664	005037	032474		CLR RXBA+2	:RX BUFFER
3201	021670	012705	025472		MOV #TXBAP,R5	:SET TX POINTER
3202	021674	005004			CLR R4	:LOAD
3203	021676	005204		2\$:	INC R4	:DATA
3204	021700	110425			MOVB R4,(R5)+	:INTO
3205	021702	022704	000005		CMP #5,R4	:TX BUFFER
3206	021706	001373			BNE 2\$:(1-5)
3207	021710	004537	024620		PERFORM SETSYNC	:GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
3208	021714	005277	157442		INC @DVSCR	:SET UCPU GO
3209	021720	005005			CLR R5	:SET COUNTER TO 0
3210	021722	005777	157434		TSTB @DVSCR	:WAIT FOR DVSCR07=1
3211	021726	100404			BMI .+12	:BR IF SET.
3212	021730	104414			DELAY	:STALL TIME
3213	021732	005205			INC R5	:UPDATE
3214	021734	001372			BNE .-12	:WAIT
3215	021736	104000			HLT	:DVSCR07 NOT SET.
3216	021740	005005			CLR R5	:SET COUNTER TO 0
3217	021742	005777	157414		TST @DVSCR	:TX DONE?
3218	021746	100404			BMI .+12	:BR IF DVSCR15=1
3219	021750	104414			DELAY	:STALL TIME
3220	021752	005205			INC R5	:UPDATE
3221	021754	001372			BNE .-12	:
3222	021756	104000			HLT	:DVSCR15 NOT SET.
3223	021760	012705	025472		MOV #TXBAP,R5	:SET TX POINTER
3224	021764	113725	001236		MOVB STAT,(R5)+	:SYNC
3225	021770	113725	001236		MOVB STAT,(R5)+	:SYNC
3226	021774	012704	000006		MOV #6,R4	:SET 1ST DATA TO 6
3227	022000	110425		3\$:	MOVB R4,(R5)+	:LOAD
3228	022002	005204			INC R4	:DATA

3229	022004	022704	000011			CMP	#11,R4	: ALL DONE?
3230	022010	001373				BNE	3\$: BR IF NO
3231	022012	032737	004000	001236		BIT	#ASYNC,STAT	: #IS THIS AN ASYNC LINE CARD?
3232	022020	001406				BEQ	82\$: #BR IF NOT ASYNC.
3233	022022	004537	025032			PERFORM	SETREG	: #ADJUST FOR ASYNC LINE CARD
3234	022026	000	001			.BYTE	000,001	: #REGISTERS
3235	022030	025474				TXBAP+2		: #LOAD FOR ASYNC
3236	022032	177775				-3		: #LOAD FOR ASYNC
3237	022034	000405				BR	83\$: #CONTINUE TEST
3238	022036	004537	025032		82\$:	PERFORM	SETREG	
3239	022042	000	001			.BYTE	000,001	: TX BA P, TX BC P
3240	022044	025470				SYNC		
3241	022046	177771				-7		
3242	022050	032737	004000	001236	83\$:	BIT	#ASYNC,STAT	: #ASYNC LINE CARD?
3243	022056	001403				BEQ	.+10	: #BR IF NO
3244	022060	004537	025076			PERFORM	LOAD.MODE	: #CLEAR RX ENABLE
3245	022064	000000				0		: #
3246	022066	112777	000013	157300		MOVB	#13,@DVSRSH	: LINE STATE
3247	022074	042777	000200	157274		BIC	#BIT7,@DVSRA	: CLEAR 'USE SEC TABLES'
3248	022102	052777	000002	157266		BIS	#BIT1,@DVSRA	: SET RE-SYNC
3249	022110	112777	000012	157256		MOVB	#12,@DVSRSH	: SEL LINE PROTOCOL PARAM.
3250	022116	052777	000002	157252		BIS	#BIT1,@DVSRA	: SET STRIP LEADING SYNC
3251	022124	012737	006000	022134		MOV	#6000,84\$: GIVE UCPU TIME
3252	022132	005327				DEC	(PC)+	: TO RESYNC SILO
3253	022134	000000			84\$:	0		
3254	022136	001375				BNE	.-4	
3255	022140	032737	004000	00 236		BIT	#ASYNC,STAT	: #ASYNC LINE CARD?
3256	022146	001403				BEQ	.+10	: #BR IF NOT ASYNC LINE CARD.
3257	022150	004537	025076			PERFORM	LOAD.MODE	: #SET RX ENABLE FOR ASYNC LINE CARD
3258	022154	020000				BIT13		: #RX ENABLE
3259	022156	112777	000013	157210		MOVB	#13,@DVSRSH	: SEL LINE STATE.
3260	022164	052777	000004	157204		BIS	#BIT2,@DVSRA	: SET TX GO.
3261	022172	052777	000400	157162		BIS	#BIT8,@DVSCR	: RESTART CPU
3262	022200	005004				CLR	R4	: SET FOR TIME OUT.
3263	022202	105777	157154			TSTB	@DVSCR	: RX DONE?
3264	022206	100404				BMI	.+12	: BR IF YES
3265	022210	104414				DELAY		: WASTE TIME
3266	022212	005204				INC	R4	: LOOP DONE?
3267	022214	001372				BNE	.-12	: BR IF NO
3268	022216	104000				HLT		: DVSCRO7 NOT SET AFTER RESYNC.
3269	022220	017704	157142			MOV	@DVRIC,R4	: READ DVRIC
3270	022224	010005				MOV	R0,R5	: LOAD LINE NO
3271	022226	000305				SWAB	R5	: PLACE IN HIGH BYTE
3272	022230	052705	040010			BIS	#BIT14+10,R5	: SET BC WARNING + CHAR 10
3273						: *****		
3274	022234	010246				MOV	R2,-(SP)	: SAVE R2
3275	022236	013702	001244			MOV	MASKX,R2	: GET BITS/CHAR. DATA
3276	022242	032737	040000	001236		BIT	#PARBIT,STAT	: PARITY EN?
3277	022250	001405				BEQ	40\$: IF NO, BRANCH
3278	022252	032702	000400			BIT	#400,R2	: 8 BITS/CHAR?
3279	022256	001002				BNE	40\$: BR IF YES
3280	022260	043704	001244			BIC	MASKX,R4	: ELSE CLEAR BIT
3281	022264	020504			40\$:	CMP	R5,R4	: RIC OK
3282						: *****		
3283	022266	001401				BEQ	4\$	
3284	022270	104001				HLT	1	: DVRIC WRONG

(REV. DO)

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 64
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0075

3285	022272	012703	032472		4\$:	MOV	#RXBA,R3	;CHECK RX DATA	
3286	022276	112304				MOVB	(R3)+,R4		
3287	022300	012705	000001			MOV	#1,R5		
3288						;*****			
3289	022304	032737	040000	001236		BIT	#PARBIT,STAT	;PARITY ENABLED?	(REV. DO)
3290	022312	001405				BEQ	41\$;IF NO, BRANCH	
3291	022314	032702	000400			BIT	#400,R2	;8 BITS?	
3292	022320	001002				BNE	41\$;BR IF YES	
3293	022322	043704	001244			BIC	MASKX,R4	;ELSE CLEAR BIT	
3294	022326	120504			41\$:	CMPB	R5,R4		
3295						;*****			
3296	022330	001401				BEQ	5\$		
3297	022332	104001				HLT	1	;1ST CHAR NOT '1'!	
3298	022334	112304			5\$:	MOVB	(R3)+,R4		
3299	022336	012705	000006			MOV	#6,R5		
3300						;*****			
3301	022342	032737	040000	001236		BIT	#PARBIT,STAT	;PARITY ENABLED?	(REV. DO)
3302	022350	001405				BEQ	42\$;IF NO, BRANCH	
3303	022352	032702	000400			BIT	#400,R2	;8 BITS?	
3304	022356	001002				BNE	42\$;BR IF YES	
3305	022360	043704	001244			BIC	MASKX,R4	;ELSE CLEAR BIT	
3306	022364	120504			42\$:	CMPB	R5,R4		
3307						;*****			
3308	022366	001401				BEQ	6\$		
3309	022370	104001				HLT	1	;2ND CHAR NOT '6'!	
3310	022372	112304			6\$:	MOVB	(R3)+,R4		
3311	022374	012705	000007			MOV	#7,R5		
3312						;*****			
3313	022400	032737	040000	001236		BIT	#PARBIT,STAT	;PARITY ENABLED?	(REV. DO)
3314	022406	001405				BEQ	43\$;IF NO, BRANCH	
3315	022410	032702	000400			BIT	#400,R2	;8 BITS?	
3316	022414	001002				BNE	43\$;BR IF YES	
3317	022416	043704	001244			BIC	MASKX,R4	;ELSE CLEAR BIT	
3318	022422	120504			43\$:	CMPB	R5,R4		
3319						;*****			
3320	022424	001401				BEQ	7\$		
3321	022426	104001				HLT	1	;3RD CHAR NOT '7'!	
3322	022430	005205			7\$:	INC	R5		
3323	022432	112304				MOVB	(R3)+,R4		
3324						;*****			
3325	022434	032737	040000	001236		BIT	#PARBIT,STAT	;PARITY ENABLED?	(REV. DO)
3326	022442	001405				BEQ	44\$;IF NO, BRANCH	
3327	022444	032702	000400			BIT	#400,R2	;8 BITS?	
3328	022450	001002				BNE	44\$;BR IF YES	
3329	022452	043704	001244			BIC	MASKX,R4	;ELSE CLEAR BIT	
3330	022456	120504			44\$:	CMPB	R5,R4		
3331						;*****			
3332	022460	001401				BEQ	8\$		
3333	022462	104001				HLT	1	;4TH CHAR NOT '10'!	
3334	022464	104412			8\$:	MSTCLR		;RESET DV11	
3335	022466	104401				SCOP1		;LOCK ON CURRENT LINE?	
3336	022470	012602				MOV	(SP)+,R2	;RESTORE R2	(REV. DO)
3337	022472	005200				INC	R0	;UPDATE LINE NO.	
3338	022474	005302				DEC	R2	;4 LINES DONE	
3339	022476	001402				BEQ	.+6	;BR IF YES	
3340	022500	000137	021516			JMP	1\$;JMP IF NO	

C2DVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 65
 C2DVD.D.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0076

```

3341 022504 000207          RTS      PC          ;EXIT FOR NEXT 4 LINE GROUP
3342
3343          ;***** TEST 20 *****
3344          ;*TEST OF RECEIVER OVERRUN.
3345          ;*TEST TO TXMIT 134. CHARS AND RECV 129
3346          ;*SERVICEING THE FIRST CHAR AS A SPECIAL CHAR
3347          ;*AND STOPING THE CHAR PROCESSOR.
3348          ;*WHEN THE TRANSMITTER FINISHES ALL 134. CHARS
3349          ;*THE RECEIVER IS RESTARTED AND THE NEXT ENTRY
3350          ;*IN THE RIC REG S/B OVER RUN ON CHAR 202(8).
3351          ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
3352          ;*****
3353
3354          ; TEST 20
3355          ;-----
3356 022506 012737 000020 001226 1ST20: MOV      #20,TSTNO
3357 022514 012737 002436 001216      MOV      #.EOP,NEXT
3358 022522 012700 000000          MOV      #0.,RO
3359 022526 013737 001406 001244      MOV      MASK.A,MASKX
3360 022534 013737 001422 001236      MOV      L00.03,STAT
3361 022542 100402          BMI      100$
3362 022544 004737 022654          JSR      PC,105$
3363 022550 012700 000004          100$: MOV      #4.,RO
3364 022554 013737 001410 001244      MOV      MASK.B,MASKX
3365 022562 013737 001424 001236      MOV      L04.07,STAT
3366 022570 100402          BMI      101$
3367 022572 004737 022654          JSR      PC,105$
3368 022576 012700 000010          101$: MOV      #8.,RO
3369 022602 013737 001412 001244      MOV      MASK.C,MASKX
3370 022610 013737 001426 001236      MOV      L08.11,STAT
3371 022616 100402          BMI      102$
3372 022620 004737 022654          JSR      PC,105$
3373 022624 012700 000014          102$: MOV      #12.,RO
3374 022630 013737 001414 001244      MOV      MASK.D,MASKX
3375 022636 013737 001430 001236      MOV      L12.15,STAT
3376 022644 100402          BMI      103$
3377 022646 004737 022654          JSR      PC,105$
3378 022652 104400          103$: SCOPE
3379 022654          105$:
3380 022654 012737 022730 001220      MOV      #1$,LOCK
3381 022662 104413          RAMCLR
3382 022664 005004          CLR      R4
3383 022666 012705 033072          MOV      #RXTAB,R5
3384 022672 005025          CLR      (R5)+
3385 022674 105204          INCB     R4
3386 022676 100375          BPL      #-4
3387 022700 032737 020000 001236      BIT      #20000,STAT
3388 022706 001004          BNE      31$
3389 022710 112737 000001 033073      MOVB     #BIT0,RXTAB+1
3390          ;*****
3391 022716 000402          BR      30$
3392 022720 004737 024322          31$: JSR      PC,EVNDAT
3393 022724 012702 000004          30$: MOV      #4,R2
3394          ;*****
3395 022730 010077 156436          1$: MOV      RO,ADVSR5
3396 022734 032737 004000 001236      BIT      #ASYNC,STAT

```

```

;PLACE LINE NUMBER INTO RO
;PLACE 'MASK' FOR CHARS INTO MASKX
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;GO DO THE TEST FOR LINE CARD 1
;PLACE LINE NUMBER INTO RO
;GET MASK
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;GO DO THE TEST FOR LINE CARD 2
;LOAD LINE NUMBER
;GET MASK
;LOAD LINE CARD STATUS INTO STAT
;BR IF LINE CARD NOT TO BE TESTED
;DO THE TEST FOR LINE CARD 3
;LOAD LINE NO.
;GET MASK
;LOAD LINE CARD STATUS
;BR IF LINE CARD NOT TO BE TESTED
;DO THE TESTS FOR LINE CARD 4
;SCOPE THIS TEST.
;TEST ENTRANCE.
;RETURN FOR SW09
;CLEAR ALL SEC REGISTERS
;CLEAR
;THE
;RECEIVER
;CONTROL
;TABLE
;EVEN PAR. EN? (REV. DO)
;BR IF TRUE (REV. DO)
;SET 'SPECIAL CHAR'(1)
;*****
;SKIP EVEN SETUP (REV. DO)
;GO GET EVEN DATA
;4 LINE GROUP
;*****
;LOAD LINE NO.
;IS THIS AN ASYNC LINE CARD?

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 66
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0077

3397	022742	001406			BEQ	80\$;#BR IF NOT ASYNC.
3398	022744	004537	025032		PERFORM	SETREG		;#ADJUST FOR ASYNC LINE CARD
3399	022750	000	001		.BYTE	000,001		;#REGISTERS
3400	022752	025472			TXBAP			;#LOAD FOR ASYNC
3401	022754	177573			-133.			;#LOAD FOR ASYNC
3402	022756	000405			BR	81\$;#CONTINUE TEST
3403	022760	004537	025032		PERFORM	SETREG		
3404	022764	000	001	80\$:	.BYTE	000,001		;TX BA P, TX BC P
3405	022766	025470			SYNC			
3406	022770	177572			-134.			
3407	022772			81\$:				
3408	022772	032737	004000	001236	BIT	#ASYNC,STAT		;#IS THIS AN ASYNC LINE CARD?
3409	023000	001406			BEQ	82\$;#BR IF NOT ASYNC.
3410	023002	004537	025032		PERFORM	SETREG		;#ADJUST FOR ASYNC LINE CARD
3411	023006	004	005		.BYTE	004,005		;#REGISTERS
3412	023010	032472			RXBA			;#LOAD FOR ASYNC
3413	023012	177576			-130.			;#LOAD FOR ASYNC
3414	023014	000405			BR	83\$;#CONTINUE TEST
3415	023016	004537	025032		PERFORM	SETREG		
3416	023022	004	005	82\$:	.BYTE	004,005		;RX BA, RX BC
3417	023024	032472			RXBA			
3418	023026	177577			-129.			
3419	023030	004537	025032		PERFORM	SETREG		
3420	023034	010	011	83\$:	.BYTE	010,011		;TX TAB, RX TAB
3421	023036	02647			TXTAB			
3422	023040	033072			RXTAB			
3423	023042	004537	025032		PERFORM	SETREG		
3424	023046	013	012		.BYTE	013,012		;LINE STATE, LINE PROTOCOL PARAM
3425	023050	000004			BIT2			;TX GO
3426	023052	000101			BIT6+BIT0			;TX DDCMP + IDLE MARK
3427	023054	032737	004000	001236	BIT	#ASYNC,STAT		;#IS THIS ASYNC LINE CARD?
3428	023062	001412			BEQ	50\$;#BR IF NO.
3429	023064	004537	025076		PERFORM	LOAD.MODE		;#LOAD PARAMETERS.
3430	023070	020000			BIT13			;#RECEIVER ENABLE
3431	023072	004537	025076		PERFORM	LOAD.MODE		
3432	023076	015000			<BIT12+BIT11>+BIT9			;#8 BITS/PER/CHAR
3433	023100	004537	025076		PERFORM	LOAD.MODE		
3434	023104	072000			<BIT14+BIT13+BIT12>+BIT10			;#9600 BAUD.
3435								
3436	023106	000403			BR	61\$		
3437	023110	004537	025076	60\$:	PERFORM	LOAD.MODE		;LOAD
3438	023114	034000			BIT13+BIT12+BIT11			;MODE
3439	023116	012705	025472	61\$:	MOV	#TXBAP,R5		;LOAD
3440	023122	005004			CLR	R4		;TX
3441	023124	105204		2\$:	INCB	R4		;DATA
3442	023126	001402			BEQ	21\$;BUFFER
3443	023130	110425			MOVB	R4,(R5)+		
3444	023132	000774			BR	2\$		
3445	023134	004537	024620	21\$:	PERFORM	SETSYNC		;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
3446	023140	005277	156216		INC	@DVSCR		;SET UCPU GO
3447	023144	105777	156212		TSTB	@DVSCR		;DVSCR07=1?
3448	023150	100375			BPL	-4		;BR IF NO
3449	023152	005777	156204		TST	@DVSCR		;DVSCR15=1?
3450	023156	100375			BPL	-4		;BR IF NO
3451	023160	112777	000012	156206	MOVB	#12,@DVSRSH		;LINE PROTOCOL PARAM.
3452	023166	052777	000040	156202	BIS	#BIT5,@DVSRSA		;SET RX DDCMP

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 67
 CZDVD.D.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0078

```

3453 023174 052777 000400 156160      BIS      #BIT8,ADVSCR      ;RESTART
3454 023202 105777 156154      TSTB     ADVSCR          ;DVSCR07=1?
3455 023206 100375              BPL       -4              ;BR IF NO
3456 023210 017704 156152      MOV      ADVRIC,R4      ;READ RIC
3457 023214 010005              MOV      R0,R5          ;LINE
3458 023216 000305              SWAB     R5              ;HIGH BYTE
3459 023220 052705 020202      BIS      #BIT13+202,R5      ;130.
3460 023224 032737 004000 001236    BIT      #ASYNC,STAT      ;IS THIS AN ASYNC LINE CARD?
3461 023232 001401              BEQ      +4              ;BR IF NOT ASYNC
3462 023234 005205              INC      R5              ;ADJUST FOR ASYNC. DOUBLE BUFFER CAUSES
3463                                ;CHAR TO BE ONE MORE THAN SYNC LINE CARD.
3464                                ;*****
3465 023236 032737 000400 001244    BIT      #400,MASKX      ;8 BITS/CHAR? (REV. D0)
3466 023244 001004              BNE      6$              ;BR IF YES
3467 023246 043705 001244      BIC      MASKX,R5          ;CLEAR UNUSED BITS
3468 023252 043704 001244      BIC      MASKX,R4          ;
3469 023256 020504 6$:          CMP      R5,R4          ;OK?
3470 023260 001401              BEQ      3$              ;BR IF YES
3471 023262 104001              HLT      1              ;OVER-RUN OR ON WRONG CHARACTER.
3472                                ;*****
3473 023264 104412 3$:          MSTCLR          ;RESET DVA
3474 023266 104401              SCOP1          ;LOCK ON CURRENT LINE?
3475 023270 005200              INC      R0              ;UPDATE LINE NO.
3476 023272 005302              DEC      R2              ;4 LINES DONE
3477 023274 001402              BEQ      +6              ;BR IF YES
3478 023276 000137 022730      JMP      1$              ;JMP IF YES
3479 023302 000207              RTS      PC              ;EXIT
3480
3481                                ;*****
3482                                ;SUBROUTINE TO CLEAR MASK BIT IF PARITY ENABLED AND (REV. D0)
3483                                ;TO COMPARE DATA.
3484
3485 023304 032737 040000 001236    PAREN: BIT      #PARBIT,STAT      ;PARITY ENABLED?
3486 023312 001402              BEQ      4$              ;IF NO, BRANCH
3487 023314 043704 001244      BIC      MASKX,R4          ;ELSE CLEAR BIT
3488 023320 120504 4$:          CMPB     R5,R4          ;COMPARE DATA
3489 023322 001420              BEQ      5$              ;BRANCH IF OK
3490 023324 022737 017136 001216    CMP      #TST15,NEXT      ;CALLED FROM TST 14?
3491 023332 001013              BNE      6$              ;BR IF NOT
3492 023334 032737 040000 001236    BIT      #PARBIT,STAT      ;PARITY EN?
3493 023342 001407              BEQ      6$              ;BR IF NOT
3494 023344 032737 020000 001236    BIT      #20000,STAT      ;EVN. PAR?
3495 023352 001003              BNE      6$              ;BR IF YES
3496 023354 122704 000011      CMPB     #BIT3+BIT0,R4      ;DATA OK?
3497 023360 001401              BEQ      5$              ;BR IF YES
3498 023362 104001 6$:          HLT      1              ;REPORT ERROR
3499 023364 000207 5$:          RTS      PC              ;RETURN TO MAIN PROGRAM
3500
3501                                ;SUBROUTINE TO SELECT CORRECT DATA TABLE
3502                                ;TO USE IN TESTS 12 AND 14, DEPENDING ON
3503                                ;BITS/CHAR. AND PARITY STATUS.
3504
3505
3506
3507 023366 BTCHAR:              MOV      R5,TEMP1          ;SET UP COUNTER
3508 023366 010537 001246

```

3509	023372	132703	000040		BITB	#40,R3	:5 BITS/CHAR?
3510	023376	001412			BEQ	2\$:BR IF NOT
3511	023400	032737	020000	001236	BIT	#20000,STAT	:EVN PAR?
3512	023406	001403			BEQ	1\$:BR IF NOT
3513	023410	012701	023660		MOV	#EVEN5,R1	:POINT TO DATA TABLE
3514	023414	000431			BR	6\$:DO IT
3515	023416	012701	023706	1\$:	MOV	#ODD5,R1	:
3516	023422	000426			BR	6\$:
3517	023424	132703	000100	2\$:	BITB	#100,R3	:6 BITS?
3518	023430	001412			BEQ	4\$:BR IF NOT
3519	023432	032737	020000	001236	BIT	#20000,STAT	:EVN PAR?
3520	023440	001403			BEQ	3\$:BR IF NOT
3521	023442	012701	023604		MOV	#EVEN6,R1	:POINT TO DATA TABLE
3522	023446	000414			BR	6\$:DO IT
3523	023450	012701	023632	3\$:	MOV	#ODD6,R1	:
3524	023454	000411			BR	6\$:
3525	023456	032737	020000	001236	4\$:	BIT	#20000,STAT
3526	023464	001403			BEQ	5\$:BR IF NOT
3527	023466	012701	023530		MOV	#EVEN7,R1	:POINT TO DATA TABLE
3528	023472	000402			BR	6\$:DO IT
3529	023474	012701	023556	5\$:	MOV	#ODD7,R1	:
3530	023500	022705	000006	6\$:	CMP	#6,R5	:CALLED FROM TEST 12:
3531	023504	001403			BEQ	7\$:YES,SKIP TST 14 SETUP
3532	023506	112731	000010		MOVB	#BIT3,a(R1)+	:
3533							:LOAD "INC BCC"
3534	023512	000402			BR	8\$:SKIP TST 12 SETUP
3535	023514	112731	000001	7\$:	MOVB	#BIT0,a(R1)+	:
3536	023520	005337	001246	8\$:	DEC	TEMP1	:REDUCE COUNT
3537	023524	001365			BNE	6\$:GO BACK IF NOT FINISHED
3538	023526	000207			RTS	PC	:RETURN TO MAIN PROGRAM
3539							
3540							
3541							:DATA TABLES FOR TESTS 12 AND 14
3542							
3543	023530	033072			EVEN7:	RXTAB+0	
3544	023532	033273				RXTAB+201	
3545	023534	033274				RXTAB+202	
3546	023536	033075				RXTAB+3	
3547	023540	033276				RXTAB+204	
3548	023542	033077				RXTAB+5	
3549	023544	033100				RXTAB+6	
3550	023546	033301				RXTAB+207	
3551	023550	033302				RXTAB+210	
3552	023552	033103				RXTAB+11	
3553	023554	033104				RXTAB+12	
3554							
3555	023556	033272			ODD7:	RXTAB+200	
3556	023560	033073				RXTAB+1	
3557	023562	033074				RXTAB+2	
3558	023564	033275				RXTAB+203	
3559	023566	033076				RXTAB+4	
3560	023570	033277				RXTAB+205	
3561	023572	033300				RXTAB+206	
3562	023574	033101				RXTAB+7	
3563	023576	033102				RXTAB+10	
3564	023600	033303				RXTAB+211	

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 69
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0080

3565	023602	033304		RXTAB+212
3566				
3567	023604	033072	EVEN6:	RXTAB+0
3568	023606	033173		RXTAB+101
3569	023610	033174		RXTAB+102
3570	023612	033075		RXTAB+3
3571	023614	033176		RXTAB+104
3572	023616	033077		RXTAB+5
3573	023620	033100		RXTAB+6
3574	023622	033201		RXTAB+107
3575	023624	033202		RXTAB+110
3576	023626	033103		RXTAB+11
3577	023630	033104		RXTAB+12
3578				
3579	023632	033172	ODD6:	RXTAB+100
3580	023634	033073		RXTAB+1
3581	023636	033074		RXTAB+2
3582	023640	033175		RXTAB+103
3583	023642	033076		RXTAB+4
3584	023644	033177		RXTAB+105
3585	023646	033200		RXTAB+106
3586	023650	033101		RXTAB+7
3587	023652	033102		RXTAB+10
3588	023654	033203		RXTAB+111
3589	023656	033204		RXTAB+112
3590				
3591	023660	033072	EVEN5:	RXTAB+0
3592	023662	033133		RXTAB+41
3593	023664	033134		RXTAB+42
3594	023666	033075		RXTAB+3
3595	023670	033136		RXTAB+44
3596	023672	033077		RXTAB+5
3597	023674	033100		RXTAB+6
3598	023676	033141		RXTAB+47
3599	023700	033142		RXTAB+50
3600	023702	033103		RXTAB+11
3601	023704	033104		RXTAB+12
3602				
3603	023706	033132	ODD5:	RXTAB+40
3604	023710	033073		RXTAB+1
3605	023712	033074		RXTAB+2
3606	023714	033135		RXTAB+43
3607	023716	033076		RXTAB+4
3608	023720	033137		RXTAB+45
3609	023722	033140		RXTAB+46
3610	023724	033101		RXTAB+7
3611	023726	033102		RXTAB+10
3612	023730	033133		RXTAB+41
3613	023732	033134		RXTAB+42
3614				
3615				
3616				
3617				
3618				
3619	023734	010346	MODDAT:	MOV R3, -(SP) ; SAVE R3
3620	023736	005003		CLR R3 ;

; SUBROUTINE TO DETERMINE VALUES TO BE USED
 ; AS CONTROL BYTE OFFSETS.

3621	023740	005037	001252		CLR	TEMP3	;CLEAR 8 BIT /CHAR FLAG
3622	023744	005037	001250		CLR	TEMP2	;CLEAR CHAR DATA STORE
3623	023750	013703	001244		MOV	MASKX,R3	;GET BITS/CHAR DATA
3624	023754	032737	040000	001236	BIT	#PARBIT,STAT	;PARITY ENABLED?
3625	023762	001435			BEQ	4\$;BR IF NOT
3626	023764	032737	020000	001236	BIT	#20000,STAT	;EVN PAR?
3627	023772	001431			BEQ	4\$;BR IF NOT
3628	023774	032703	000400		BIT	#400,R3	;8 BITS?
3629	024000	001404			BEQ	1\$;BR IF NOT
3630	024002	112737	000001	001252	MOVB	#1,TEMP3	;SET 8 BIT FLAG
3631	024010	000422			BR	4\$;GO TO LOAD 8 BIT DATA
3632	024012	132703	000040	1\$:	BITB	#40,R3	;5 BIT, EVEN PAR?
3633	024016	001404			BEQ	2\$;BR IF NOT
3634	024020	112737	000071	001250	MOVB	#71,TEMP2	;LOAD 5 BIT DATA
3635	024026	000416			BR	EXIT2	;GET OUT
3636	024030	132703	000100	2\$:	BITB	#100,R3	;6 BITS?
3637	024034	001404			BEQ	3\$;BR IF NOT
3638	024036	112737	000131	001250	MOVB	#131,TEMP2	;LOAD DATA
3639	024044	000407			BR	EXIT2	;GET OUT
3640	024046	112737	000231	001250	MOVB	#231,TEMP2	;LOAD 7 BIT DATA
3641	024054	000403			BR	EXIT2	;GET OUT
3642	024056	112737	000031	001250	MOVB	#31,TEMP2	;SET UP FOR NO PARITY, ODD
3643							;PAR. OR 8 BIT EVEN PAR.
3644	024064	013704	001250	EXIT2:	MOV	#TEMP2,R4	
3645	024070	012603			MOV	(SP)+,R3	;RESTORE R3
3646	024072	000207			RTS	PC	;RETURN TO CALLING PROGRAM

;*TABLES OF OFFSETS FOR RXTAB IN TEST 4

3651	024074			SEVTAB:	
3652	024074	033314			RXTAB+222
3653	024076	033324			RXTAB+232
3654	024100	033316			RXTAB+224
3655	024102	033322			RXTAB+230

3658	024104			SIXTAB:	
3659	024104	033214			RXTAB+122
3660	024106	033224			RXTAB+132
3661	024110	033216			RXTAB+124
3662	024112	033222			RXTAB+130

3665	024114			FIVTAB:	
3666	024114	033154			RXTAB+62
3667	024116	033164			RXTAB+72
3668	024120	033156			RXTAB+64
3669	024122	033162			RXTAB+70

;DATA TABLES FOR TEST 15

3676	024124	033107		RX8DAT:	RXTAB+15
------	--------	--------	--	---------	----------

3677	024126	033510	RXTAB+BIT8+16
3678	024130	034113	RXTAB+BIT9+21
3679	024132	034515	RXTAB+BIT9+BIT8+23
3680	024134	035117	RXTAB+BIT10+25
3681	024136	035501	RXTAB+BIT10+BIT8+7
3682	024140	036126	RXTAB+BIT10+BIT9+34
3683	024142	036524	RXTAB+BIT10+BIT9+BIT8+32
3684	024144	036530	RXTAB+BIT10+BIT9+BIT8+36
3685			
3686			
3687			
3688			
3689	024146	033307	RX7EVN: RXTAB+215
3690	024150	033710	RXTAB+BIT8+216
3691	024152	034113	RXTAB+BIT9+21
3692	024154	034715	RXTAB+BIT9+BIT8+223
3693	024156	035317	RXTAB+BIT10+225
3694	024160	035701	RXTAB+BIT10+BIT8+207
3695	024162	036326	RXTAB+BIT10+BIT9+234
3696	024164	036724	RXTAB+BIT10+BIT9+BIT8+232
3697	024166	036530	RXTAB+BIT10+BIT9+BIT8+36
3698			
3699			
3700			
3701			
3702			
3703	024170	033107	RX7ODD: RXTAB+15
3704	024172	033510	RXTAB+BIT8+16
3705	024174	034313	RXTAB+BIT9+221
3706	024176	034515	RXTAB+BIT9+BIT8+23
3707	024200	035117	RXTAB+BIT10+25
3708	024202	035501	RXTAB+BIT10+BIT8+7
3709	024204	036126	RXTAB+BIT10+BIT9+34
3710	024206	036524	RXTAB+BIT10+BIT9+BIT8+32
3711	024210	036730	RXTAB+BIT10+BIT9+BIT8+236
3712			
3713			
3714			
3715			
3716	024212	033207	RX6EVN: RXTAB+115
3717	024214	033610	RXTAB+BIT8+116
3718	024216	034113	RXTAB+BIT9+21
3719	024220	034615	RXTAB+BIT9+BIT8+123
3720	024222	035217	RXTAB+BIT10+125
3721	024224	035601	RXTAB+BIT10+BIT8+107
3722	024226	036226	RXTAB+BIT10+BIT9+134
3723	024230	036624	RXTAB+BIT10+BIT9+BIT8+132
3724	024232	036530	RXTAB+BIT10+BIT9+BIT8+36
3725			
3726			
3727			
3728			
3729	024234	033107	RX6ODD: RXTAB+15
3730	024236	033510	RXTAB+BIT8+16
3731	024240	034213	RXTAB+BIT9+121
3732	024242	034515	RXTAB+BIT9+BIT8+23

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 72
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0083

3733	024244	035117		RXTAB+BIT10+25
3734	024246	035501		RXTAB+BIT10+BIT8+7
3735	024250	036126		RXTAB+BIT10+BIT9+34
3736	024252	036524		RXTAB+BIT10+BIT9+BIT8+32
3737	024254	036630		RXTAB+BIT10+BIT9+BIT8+136

3738
3739

3740

3741

3742	024256	033147		
------	--------	--------	--	--

3743	024260	033550		
------	--------	--------	--	--

3744	024262	034113		
------	--------	--------	--	--

3745	024264	034555		
------	--------	--------	--	--

3746	024266	035157		
------	--------	--------	--	--

3747	024270	035541		
------	--------	--------	--	--

3748	024272	036166		
------	--------	--------	--	--

3749	024274	036564		
------	--------	--------	--	--

3750	024276	036530		
------	--------	--------	--	--

3751

3752

3753

3754

3755	024300	033107		
------	--------	--------	--	--

3756	024302	033510		
------	--------	--------	--	--

3757	024304	034153		
------	--------	--------	--	--

3758	024306	034515		
------	--------	--------	--	--

3759	024310	035117		
------	--------	--------	--	--

3760	024312	035501		
------	--------	--------	--	--

3761	024314	036126		
------	--------	--------	--	--

3762	024316	036524		
------	--------	--------	--	--

3763	024320	036570		
------	--------	--------	--	--

3764

3765

3766

3767

3768

3769	024322	010346		
------	--------	--------	--	--

3770	024324	013703	001244	
------	--------	--------	--------	--

3771	024330	032703	000400	
------	--------	--------	--------	--

3772	024334	001404		
------	--------	--------	--	--

3773	024336	112737	000001	033073
------	--------	--------	--------	--------

3774	024344	000421		
------	--------	--------	--	--

3775	024346	132703	000040	
------	--------	--------	--------	--

3776	024352	001404		
------	--------	--------	--	--

3777	024354	112737	000001	033133
------	--------	--------	--------	--------

3778	024362	000412		
------	--------	--------	--	--

3779	024364	132703	000100	
------	--------	--------	--------	--

3780	024370	001404		
------	--------	--------	--	--

3781	024372	112737	000001	033173
------	--------	--------	--------	--------

3782	024400	000403		
------	--------	--------	--	--

3783	024402	112737	000001	033273
------	--------	--------	--------	--------

3784	024410	012603		
------	--------	--------	--	--

3785	024412	000207		
------	--------	--------	--	--

3786

3787

3788

RY5EVN: RXTAB+55
 RXTAB+BIT8+56
 RXTAB+BIT9+21
 RXTAB+BIT9+BIT8+63
 RXTAB+BIT10+65
 RXTAB+BIT10+BIT8+47
 RXTAB+BIT10+BIT9+74
 RXTAB+BIT10+BIT9+BIT8+72
 RXTAB+BIT10+BIT9+BIT8+36

RX50DD: RXTAB+15
 RXTAB+BIT8+16
 RXTAB+BIT9+61
 RXTAB+BIT9+BIT8+23
 RXTAB+BIT10+25
 RXTAB+BIT10+BIT8+7
 RXTAB+BIT10+BIT9+34
 RXTAB+BIT10+BIT9+BIT8+32
 RXTAB+BIT10+BIT9+BIT8+76

:EVNDAT - SUBROUTINE TO LOAD EVEN PARITY ENABLED
 :DATA FOR TESTS 17 AND 20.

EVNDAT:	MOV	R3, -(SP)	:SAVE R3
	MOV	MASKX, R3	:GET BITS/CHAR. DATA
	BIT	#400, R3	:8 BITS/CHAR?
	BEQ	22\$:BR IF NOT
	MOVB	#BIT0, RXTAB+1	:LOAD 8 BIT DATA
	BR	29\$:GET OUT
22\$:	BITB	#40, R3	:5 BITS?
	BEQ	23\$:BR IF NOT
	MOVB	#BIT0, RXTAB+41	:LOAD 5 BIT DATA
	BR	29\$:GET OUT
23\$:	BITB	#100, R3	:6 BITS?
	BEQ	24\$:BR IF NOT
	MOVB	#BIT0, RXTAB+101	:LOAD 6 BIT DATA
	BR	29\$:GET OUT
24\$:	MOVB	#BIT0, RXTAB+201	:LOAD 7 BIT DATA
29\$:	MOV	(SP)+, R3	:RESTORE R3
	RTS	PC	:RETURN TO MAIN PROGRAM

:SUBROUTINE TO CHECK DATA FOR TESTS 2,3, AND 4.


```

3789
3790 024414
3791 024414 032737 040000 001236 MRKCK:
3792 024422 001402
3793 024424 043704 001244
3794 024430 120504
3795 024432 001471
3796 024434 032737 040000 001236
3797 024442 001451
3798 024444 017704 154716
3799 024450 010146
3800 024452 005001
3801 024454 032737 020000 001236
3802 024462 001416
3803 024464 032737 000400 001244
3804 024472 001403
3805 024474 052701 010377
3806 024500 000424
3807 024502 032737 000100 001244 31$:
3808 024510 001426
3809 024512 052701 010177
3810 024516 000415
3811 024520 032737 000040 001244 32$:
3812 024526 001403
3813 024530 052701 010077
3814 024534 000406
3815 024536 032737 000200 001244 33$:
3816 024544 001410
3817 024546 052701 010377
3818 024552 010046
3819 024554 000300
3820 024556 050001
3821 024560 012600
3822 024562 020104
3823 024564 001413
3824 024566 022737 010422 001216 35$:
3825 024574 001001
3826 024576 104001
3827 024600 022737 011160 001216 40$:
3828 024606 001001
3829 024610 104005
3830 024612 104002
3831 024614 012601
3832 024616 000207
3833
3834
3835
3836
3837 024620
3838 024620 113737 001236 025470
3839 024626 113737 025470 025471
3840 024634 032737 010000 001236
3841 024642 001402
3842 024644 105037 025470
3843 024650 000205
3844 024652 010046

;*****
SETSYNC:
      MOVB  STAT,SYNC      ;SET SYNC FOR THIS LINE.
      MOVB  SYNC,SYNC+1    ;PLACE SYNC IN HIGH BYTE
      BIT    #TWO5YN,STAT  ;ONE SYNC OR TWO?
      BEQ    1$            ;BR IF JUMPED FOR TWO.
      CLRB   SYNC          ;SET FIRST SYNC TO NON-SYNC
1$:    EXIT
SIMBCC: MOV  RO,-(SP)
  
```

3845	024654	010146		MOV	R1,-(SP)
3846	024656	010246		MOV	R2,-(SP)
3847	024660	012537	001246	MOV	(R5)+,TEMP1
3848	024664	012537	001250	MOV	(R5)+,TEMP2
3849	024670	012537	001252	MOV	(R5)+,TEMP3
3850	024674	005037	025026	1\$: CLR	BCCFBK
3851	024700	013700	001252	MOV	TEMP3,R0
3852	024704	006037	001250	ROR	TEMP2
3853	024710	005500		ADC	R0
3854	024712	032700	000001	BIT	#BIT0,P0
3855	024716	001402		BEQ	2\$
3856	024720	005137	025026	COM	BCCFBK
3857	024724	013700	025024	2\$: MOV	XPOLY,R0
3858	024730	005100		COM	R0
3859	024732	040037	025026	BIC	R0,BCCFBK
3860	024736	000241		CLC	
3861	024740	006037	001252	ROR	TEMP3
3862	024744	013700	025026	MOV	BCCFBK,R0
3863	024750	013701	001252	MOV	TEMP3,R1
3864	024754	010102		MOV	R1,R2
3865	024756	040100		BIC	R1,R0
3866	024760	043702	025026	BIC	BCCFBK,R2
3867	024764	050200		BIS	R2,R0
3868	024766	043737	025024 001252	BIC	XPOLY,TEMP3
3869	024774	050037	001252	BIS	R0,TEMP3
3870	025000	005337	001246	DEC	TEMP1
3871	025004	001333		BNE	1\$
3872	025006	013737	001252 025030	MOV	TEMP3,CALBCC
3873	025014	012602		MOV	(SP)+,R2
3874	025016	012601		MOV	(SP)+,R1
3875	025020	012600		MOV	(SP)+,R0
3876	025022	000205		RTS	R5
3877	025024	000000		XPOLY:	0
3878	025026	000000		BCCFBK:	0
3879	025030	000000		CALBCC:	0
3880		000200		LRC8=	200
3881		120001		CRC16=	120001
3882		102010		CRC.CCITT=	102010
3883					
3884					
3885	025032	010046		SETREG: MOV	R0,-(SP)
3886	025034	010146		MOV	R1,-(SP)
3887	025036	112500		MOVB	(R5)+,R0
3888	025040	112501		MOVB	(R5)+,R1
3889	025042	110077	154326	MOVB	R0,@DVSRSH
3890	025046	012577	154324	MOV	(R5)+,@DVSR
3891	025052	042777	000060 154302	BIC	#BIT5+BIT4,@DVSCR
3892	025060	110177	154310	MOVB	R1,@DVSRSH
3893	025064	012577	154306	MOV	(R5)+,@DVSR
3894	025070	012601		MOV	(SP)+,R1
3895	025072	012600		MOV	(SP)+,R0
3896	025074	000205		EXIT	
3897					
3898	025076			LOAD.MODE:	
3899	025076	012577	154266	MOV	(R5)+,@DVLCR
3900	025102	052777	100000 154260	BIS	#BIT15,@DVLCR

3901	025110	010046			MOV	RO,-(SP)	
3902	025112	005000			CLR	RO	
3903	025114	005777	15425C		TST	@DVLCR	1\$:
3904	025120	100004			BPL	2\$	
3905	025122	104414			DELAY		
3906	025124	005200			INC	RO	
3907	025126	001372			BNE	1\$	
3908	025130	104000			HLT	0	:BIT 15 FAILED TO CLEAR
3909	025132	012600			MOV	(SP)+,RO	2\$:
3910	025134	000205			EXIT		
3911							
3912					:SUBROUTINE.		
3913					:CORE TABLES ALREAY SET UP		
3914					:XMIT 3 CHARS 2SYNC+ 1 DATA		
3915					:RCV 1 CHAR		
3916	025136	010077	154230		DV11ON: MOV	RO,@DVSR	
3917	025142	032737	004000	001236	BIT	#ASYNC,STAT	:#IS THIS AN ASYNC LINE CARD?
3918	025150	001406			BEQ	80\$:#BR IF NOT ASYNC.
3919	025152	004537	025032		PERFORM	,SETREG	:#ADJUST FOR ASYNC LINE CARD
3920	025156	000	001		.BYTE	000,001	:#REGISTERS
3921	025160	025472			TXBAP		:#LOAD FOR ASYNC
3922	025162	177777			-1		:#LOAD FOR ASYNC
3923	025164	000405			BR	81\$:#CONTINUE TEST
3924	025166	004537	025032		80\$: PERFORM	,SETREG	
3925	025172	000	001		.BYTE	000,001	
3926	025174	025470			SYNC		
3927	025176	177775			-3		
3928	025200	004537	025032		81\$: PERFORM	,SETREG	
3929	025204	004	005		.BYTE	004,005	
3930	025206	032472			RXBA		
3931	025210	177777			-1		
3932	025212	004537	025032		PERFORM	,SETREG	
3933	025216	010	011		.BYTE	010,011	
3934	025220	026472			TXTAB		
3935	025222	033072			RXTAB		
3936	025224	004537	025032		PERFORM	,SETREG	
3937	025230	013	012		.BYTE	013,012	
3938	025232	000004			BIT2		
3939	025234	000001			BIT0		
3940	025236	032737	004000	001236	BIT	#ASYNC,STAT	:#IS THIS ASYNC LINE CARD?
3941	025244	001412			BEQ	60\$:#BR IF NO.
3942	025246	004537	025076		PERFORM	,LOAD.MODE	:#LOAD PARAMETERS.
3943	025252	020000			BIT13		:#RECEIVER ENABLE
3944	025254	004537	025076		PERFORM	,LOAD.MODE	:#
3945	025260	015000			<BIT12+BIT11>+BIT9		:#8 BITS/PER/CHAR
3946	025262	004537	025076		PERFORM	,LOAD.MODE	:#
3947	025266	072000			<BIT14+BIT13+BIT12>+BIT10		:#9600 BAUD.
3948							
3949	025270	000405			BR	61\$	
3950	025272	004537	025076		60\$: PERFORM	,LOAD.MODE	
3951	025276	034000			BIT13+BIT12+BIT11		
3952	025300	004537	024620		PERFORM	,SETSYNC	:GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
3953	025304	000207			61\$: RTS	PC	
3954							
3955							
3956	025306				SETSCAN:		

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 76
 CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0087

```

3957 025306 010346      MOV      R3,-(SP)
3958 025310 052777 000010 154044      BIS      #BIT3,@DVSCR
3959 025316 012503      MOV      (R5)+,R3
3960 025320 001414      BEQ      2$
3961 025322 012777 050102 154050 1$:      MOV      #BIT14+BIT12+BIT6+BIT1,@DVSFR
3962 025330 104415      ROMCLK
3963 025332 005201      INC      R1
3964 025334 012777 050102 154036      MOV      #BIT14+BIT12+BIT6+BIT1,@DVSFR
3965 025342 104415      ROMCLK
3966 025344 005201      INC      R1
3967 025346 005303      DEC      R3
3968 025350 001364      BNE      1$
3969 025352 012603      2$:      MOV      (SP)+,R3
3970 025354 010100      MOV      R1,R0
3971 025356 000241      CLC
3972 025360 006000      ROR
3973 025362 000205      EXIT
3974
3975 025364 000042      REGBUF: .BLKW 34.
3976 025470 000001      SYNC:  .BLKW 1
3977 025472 000400      TXBAP:  .BLKB 400
3978 026072 000400      TXBAS:  .BLKB 400
3979 026472 000400      TXTAB:  .BLKB 400
3980 027072 000400      .BLKB 400
3981 027472 000400      .BLKB 400
3982 030072 000400      .BLKB 400
3983 030472 000400      .BLKB 400
3984 031072 000400      .BLKB 400
3985 031472 000400      .BLKB 400
3986 032072 000400      .BLKB 400
3987 032472 000400      RXBA:   .BLKB 400
3988 033072 000400      RXTAB:  .BLKB 400
3989 033472 000400      .BLKB 400
3990 034072 000400      .BLKB 400
3991 034472 000400      .BLKB 400
3992 035072 000400      .BLKB 400
3993 035472 000400      .BLKB 400
3994 036072 000400      .BLKB 400
3995 036472 000400      .BLKB 400
3996 037072 000000      DATA: 0
3997 037074 043377 042522 020105 EM1:  .ASCIIZ <377>/FREE RUNNING ROM TESTS/
      037124 051377 041505 044505 EM2:  .ASCIIZ <377>/RECEIVER CONTROL BYTE TEST./
      037161 377 051124 047101 EM3:  .ASCIIZ <377>/TRANSMITTER CONTROL BYTE TEST./
      037221 377 042522 042503 EM4:  .ASCIIZ <377>/RECEIVER BCC ERROR/
      037245 377 051124 047101 EM5:  .ASCIIZ <377>/TRANSMITTER IDLE FUNCTION TESTS/
      037306 042777 050130 041505 DH1:  .ASCIIZ <377>/EXPECTED FOUND LINE(8)/
      037342      .EVEN
      037342 000003 DT1:  3
3998 037344 006 004      .BYTE 6,4
3999 037346 001272      SAVR5
4000 037350 006 002      .BYTE 6,2
4001 037352 001270      SAVR4
4002 037354 002 001      .BYTE 2,1
4003 037356 001260      SAVR0
4004
4005 037360      .ERRTAB:

```

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 77
CZDVDD.P11 02-JUN-80 09:32 DV11 DEVICE DIAGNOSTICS.

SEQ 0088

4006	037360	000000	0		
4007	037362	000000	0		
4008	037364	000000	0		
4009	037366	037074	EM1		
4010	037370	037306	DH1	;HALT 1	
4011	037372	037342	DT1		
4012	037374	037124	EM2		
4013	037376	037306	DH1	;HALT 2	
4014	037400	037342	DT1		
4015	037402	037161	EM3		
4016	037404	037306	DH1	;HALT 3	
4017	037406	037342	DT1		
4018	037410	037221	EM4		
4019	037412	037306	DH1	;HALT 4	
4020	037414	037342	DT1		
4021			;*****		
4022	037416	037245	EM5		(REV. D0)
4023	037420	037306	DH1	;HALT 5	
4024	037422	037342	DT1		
4025			;*****		
4026	037424		CORMAX:		
4027		000001	.END		

EVNDAT	024322	3161	3392	3769#										
EXIT =	000205	81#	3843	3896	3910	3973								
EXITER	004322	824	829#											
EXIT2	024064	3635	3639	3641	3644#									
FIVTAB	024114	1638	3665#											
FIX.OO	006514	1048	1053	1058	1063	1097#								
HALTS	004302	777	823#											
HILIM	003436	616*	643	661#										
ICOUNT	001222	138#	538	543*										
INBUF	005516	586	622	969#										
INIFLG	001310	176#	394	409*										
INSTER=	104404	199#	637											
INSTR =	104403	197#	1070											
INSTR2	003236	593	605#											
LIGHT	000174	110#	404											
LIGHTS	001200	121#	404*	475*										
LIMITS	003364	632	643#											
LINES	011524	1607	1650#	1749										
LOADIT	012300	1798#	1904											
LOAD.M	025076	1264	1266	1270	1384	1386	1388	1392	1513	1515	1517	1521	1687	1689
		1691	1695	2466	2468	2470	2474	2631	2633	2635	2639	2858	2860	2862
		2866	3024	3026	3028	3032	3189	3191	3193	3197	3244	3257	3429	3431
		3433	3437	3898#	3942	3944	3946	3950						
LOBITS	003442	618*	647	663#	664									
LOCK	001220	137#	542*	556	558	800	1242*	1343*	1472*	1607*	1793*	1942*	2016*	2084*
		2161*	2258*	2397*	2565*	2744*	2957*	3144*	3380*					
		107	492#											
LOGICA	002560	178#												
LOKFLG	001312	615*	645	660#										
LOLIM	003434	139#	537*	538	541*									
LPCNT	001224	1816	3880#											
LRCB =	000200	143#	390*	472*	524*	778	780*	864*						
LSTERR	001234	255#	1010*	1045	1225	1320	1449	1586	1773	1922	1996	2067	2138	2238
LOO.03	001422	2377	2545	2724	2937	3124	3360							
		256#	1012*	1050	1229	1325	1454	1591	1778	1927	2001	2071	2143	2243
L04.07	001424	2382	2550	2729	2942	3129	3365							
		257#	1014*	1055	1233	1330	1459	1596	1783	1932	2006	2075	2148	2248
L08.11	001426	2387	2555	2734	2947	3134	3370							
		258#	1016*	1060	1237	1335	1464	1601	1788	1937	2011	2079	2153	2253
L12.15	001430	2392	2560	2739	2952	3139	3375							
MASKX	001244	151#	1319*	1324*	1329*	1334*	1415	1448*	1453*	1458*	1463*	1544	1585*	1590*
		1595*	1600*	1621	1712	1722	1733	1772*	1777*	1782*	1787*	1921*	1926*	1931*
		1936*	1995*	2000*	2005*	2010*	2039	2137*	2142*	2147*	2152*	2199	2237*	2242*
		2247*	2252*	2272	2376*	2381*	2386*	2391*	2423	2507	2544*	2549*	2554*	2559*
		2588	2672	2723*	2728*	2733*	2738*	2762	2888	2936*	2941*	2946*	2951*	3123*
		3128*	3133*	3138*	3275	3280	3293	3305	3317	3329	3359*	3364*	3369*	3374*
		3465	3467	3468	3487	3623	3770	3793	3803	3807	3811	3815		
MASK.A	001406	245#	1046	1319	1448	1585	1772	1921	1995	2137	2237	2376	2544	2723
		2936	3123	3359										
MASK.B	001410	246#	1051	1324	1453	1590	1777	1926	2000	2142	2242	2381	2549	2728
		2941	3128	3364										
MASK.C	001412	247#	1056	1329	1458	1595	1782	1931	2005	2147	2247	2386	2554	2733
		2946	3133	3369										
MASK.D	001414	248#	1061	1334	1463	1600	1787	1936	2010	2152	2252	2391	2559	2738
		2951	3138	3374										
MASTEK	005376	802	960#											

MCRLF	005104	571	694	798	799	807	948	960#	1069	1087										
MCSRX	005326	477	960#																	
MDATA	005622	721	731	973#																
MEPASS	005145	476	960#																	
MERRPC	005452	805	960#																	
MERRX	005353	483	960#																	
MERR2	005172	960#	986	1176																
MERR3	005241	431	960#																	
MLOCK	005277	455	960#																	
MNEW	005400	426	960#																	
MODDAT	023734	1798	1948	2019	3619#															
MPASSX	005342	481	960#																	
MPFAIL	005107	861	960#																	
MQM	005100	601	960#	1092																
MR	005167	463	960#																	
MRESET=	004000	81#	880	893																
MRKCK	024414	1422	1551	1737	1741	3790#														
MSTCLR=	104412	211#	865	1292	1427	1556	1743	1802	1856	1878	1967	2044	2113	2214						
		2345	2903	3334	3473															
MTITLE	001000	119#	408																	
MTSTN	005364	803	960#	1071																
MTSTPC	005265	960#																		
MVECX	005334	479	960#																	
NEXT	001216	136#	544	834	1223*	1317*	1446*	1583*	1770*	1919*	1993*	2065*	2135*	2235*						
		2374*	2542*	2721*	2934*	3121*	3357*	3490	3824	3827										
NOLIST=	***** U	1																		
NPR	= 040000	76#																		
ODD5	023706	3515	3603#																	
ODD6	023632	3523	3579#																	
ODD7	023556	3529	3555#																	
PARAM	= 104405	201#	1072																	
PARAM1	003304	621#	638																	
PARBIT=	040000	81#	1116	1710	1720	2037	2275	2426	2509	2591	2674	2765	2890	3276						
		3289	3301	3313	3325	3485	3492	3624	3791	3796										
PAREN	023304	1402	1408	1531	1537	2188	2204	2210	2312	2331	2339	2485	2492	2498						

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 84

CZDVDD.P11 02-JUN-80 09:32

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0094

QV.FLG	001313	179#	386*	487*	535									
RAM	= 020000	74#												
RAMCLR=	104413	213#	866	1243	1344	1473	1609	1794	1943	2017	2085	2162	2259	2401
		2516	2566	2681	2745	2961	3090	3145	3381					
REGBUF	025364	3975#												
RESREG	004300	819	822#											
RESTAR	004414	849	855#											
RESTR	002572	486	490	498#										
RESV16	001404	239#	1035*	1036*										
RES05 =	104407	205#	822											
RETURN	001214	135#	392*	462*	464	498*	544*	547	834*	836	867	1086*	1094*	1095
ROMCLK=	104415	217#	3962	3965										
RUN	001304	169#	388*	989	992*	993*	1000*	1001*						
RXBA	032472	1356*	1357*	1358*	1372	1400	1406	1418	1485*	1486*	1487*	1501	1529	1535
		1547	1652*	1653*	1670	1675	1708	1718	1735	1739	2030*	2034	2173*	2174*
		2185	2200	2207	2334	2450	2615	2814	2846	2872	2992*	2993*	2995*	2996*
		3011	3044	3177	3199*	3200*	3285	3412	3417	3930	3987#			
RXTAB	033072	1346	1377	1475	1506	1626*	1628*	1629*	1630*	1680	1800*	1951*	2022*	2088*
		2271	2422	2455	2587	2620	2747	2851	2963	2981*	2983*	2984*	2985*	2986*
		2987*	2988*	2989*	2990*	3017	3148	3157*	3158*	3159*	3182	3383	3389*	3422
		3543	3544	3545	3546	3547	3548	3549	3550	3551	3552	3553	3555	3556
		3557	3558	3559	3560	3561	3562	3563	3564	3565	3567	3568	3569	3570
		3571	3572	3573	3574	3575	3576	3577	3579	3580	3581	3582	3583	3584
		3585	3586	3587	3588	3589	3591	3592	3593	3594	3595	3596	3597	3598
		3599	3600	3601	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612
		3613	3652	3653	3654	3655	3659	3660	3661	3662	3666	3667	3668	3669
		3676	3677	3678	3679	3680	3681	3682	3683	3684	3689	3690	3691	3692
		3693	3694	3695	3696	3697	3703	3704	3705	3706	3707	3708	3709	3710
		3711	3716	3717	3718	3719	3720	3721	3722	3723	3724	3729	3730	3731
		3732	3733	3734	3735	3736	3737	3742	3743	3744	3745	3746	3747	3748
		3749	3750	3755	3756	3757	3758	3759	3760	3761	3762	3763	3773*	3777*
		3781*	3783*	3935	3988#									
RX5EVN	024256	2773	3742#											
RX5ODD	024300	2775	3755#											
RX6EVN	024212	2781	3716#											
RX6ODD	024234	2783	3729#											
RX7EVN	024146	2787	3689#											
RX7ODD	024170	2789	3703#											
RX8DAT	024124	2767	3676#											
SAVACT	001302	167#	429	1174*										
SAVNUM	001303	168#	383*	485*	488*	1167*								
SAVPC	001276	164#	669*	840										
SAVR0	001260	157#	678*	683	4003									
SAVR1	001262	158#	677*	684										
SAVR2	001264	159#	676*	685										
SAVR3	001266	160#	675*	686										
SAVR4	001270	161#	674*	687	4001									
SAVR5	001272	162#	673*	688	3999									
SAVSP	001274	163#												
SAV05 =	104406	203#	782											
SCOPE =	104400	191#	1240	1338	1467	1604	1791	1940	2014	2082	2156	2256	2395	2563
		2742	2955	3142	3378									
SCOPI =	104401	193#	1293	1428	1557	1744	1900	1974	2047	2114	2215	2346	2517	2682
		2905	3091	3335	3474									
SERV.G	004640	523	770	914#	915									
SETREG	025032	1250	1254	1258	1361	1366	1370	1374	1378	1490	1495	1499	1503	1507

SYNB04	001632	332#															
SYNB05	001656	343#															
SYNB06	001702	354#															
SYNB07	001726	365#															
SYNC	025470	1368	1497	1663	2446	2611	2842	3007	3173	3240	3405	3838*	3839*	3842*			
		3926	3976#														
SYNCX	001240	149#															
SYNC00	001516	290#															
SYNC01	001542	301#															
SYNC02	001566	312#															
SYNC03	001612	323#															
SYNC04	001636	334#															
SYNC05	001662	345#															
SYNC06	001706	356#															
SYNC07	001732	367#															
SYNC2A	001432	260#	1011*														
SYNC2B	001434	261#	1013*														
SYNC2C	001436	262#	1015*														
SYNC2D	001440	263#	1017*														
SYND00	001522	292#															
SYND01	001546	303#															
SYND02	001572	314#															
SYND03	001616	325#															
SYND04	001642	336#															
SYND05	001666	347#															
SYND06	001712	358#															
SYND07	001736	369#															
S.C	= 050000	77#															
TEMP	005560	708	858*	859*	971#												
TEMP1	001246	152#	414*	415	420*	962	1193*	1194*	1818*	1843*	2873*	2884*	3508*	3536*			
		3847*	3870*														
TEMP2	001250	153#	415*	416	964	1825*	3622*	3634*	3638*	3640*	3642*	3644	3848*	3852*			
TEMP3	001252	154#	1819*	1824	1834*	1836	1841*	1842*	1845	1854	3046*	3054*	3621*	3630*			
		3849*	3851	3861*	3863	3868*	3869*	3872									
TEMP4	001254	155#															
TEMP5	001256	156#															
TKCSR	001204	127#	529	588	914	921	942										
TKDBR	001206	128#	519	531	590	596	766	923	944								
TLAST	= 022506	1090	3997#														
TPCSI	001210	129#	572	594	773	928											
TPDBK	001212	130#	574*	596*	775*	930											

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 87

CZDVDD.P11 02-JUN-80 09:32

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0097

TST3	010422	1317	1445#	3824										
TST4	011160	1446	1582#	3827										
TST5	012116	1583	1769#											
TST6	013032	1770	1918#											
TST7	013346	1919	1992#											
TTST	002702	456*	457*	459*	460*	527#								
TWOSYN=	010000	81#	3840											
TXBAP	025472	1246*	1252	1352*	1353*	1363	1481*	1482*	1492	1615	1658	1801*	1949*	2021*
		2092*	2165	2263	2413	2441	2578	2606	2821	2837	2871	3002	3036	3168
		3201	3223	3235	3400	3439	3921	3977#						
TXBAS	026072	3978#												
TXTAB	026472	1248*	1257	1345	1376	1474	1505	1610*	1611*	1612*	1613*	1614*	1679	1799*
		1950*	2020*	2087*	2088	2404	2408	2454	2569	2573	2619	2746	2758	2791*
		2792*	2793*	2794*	2795*	2796*	2797*	2798*	2799*	2850	2962	2971*	2972*	2973*
		2975*	2976*	2977*	2978*	2979*	2980*	3016	3147	3181	3421	3934	3979#	
TYPDAT	004266	797	815	818#										
TYPE =	104402	195#	408	413	426	431	455	463	476	477	479	481	483	571
		584	601	694	731	798	799	802	803	805	807	811	816	861
		918	920	948	986	1069	1087	1092	1176					
		795	798#											
TYPMSG	004166	1175	1183#											
VECMAP	007102	702*	732*	740#										
WRDCNT	003742	810	813#											
WRKO.F	004254	772	774	776#										
XBX	004060	478	500#											
XCSR	002604	484	509#											
XERR	002626	75#												
XFR =	030000	413	960#											
XHEAD	005457	482	506#											
XPASS	002620	1816*	1830	1841	1869*	1891*	3857	3868	3877#					
XPOLY	025024	419	960#											
XSTATQ	005504	804	841#											
XTSTN	004374	480	503#											
XVEC	002612	1#	1214#	1218#	1306#	1312#	1435#	1441#	1564#	1578#	1754#	1765#	1908#	1914#
SCRAP =	177777	1981#	1988#	2054#	2060#	2121#	2130#	2222#	2230#	2354#	2369#	2524#	2537#	2689#
		2714#	2911#	2928#	3102#	3116#	3343#	3352#						
SE =	000022	1#	1223	1224#	1317	1318#	1446	1447#	1583	1584#	1770	1771#	1919	1920#
		1993	1994#	2065	2066#	2135	2136#	2235	2236#	2374	2375#	2542	2543#	2721
		2722#	2934	2935#	3121	3122#	3357	3358#						
SN =	000020	1#	1214	1220	1224#	1306	1314	1318#	1435	1443	1447#	1564	1580	1584#
		1754	1767	1771#	1908	1916	1920#	1981	1990	1994#	2054	2062	2066#	2121
		2132	2136#	2222	2232	2236#	2354	2371	2375#	2524	2539	2543#	2689	2718
		2722#	2911	2931	2935#	3102	3118	3122#	3343	3354	3358#	3997#		
SY =	000017	1#	182#	191	193#	195#	197#	199#	201#	203#	205#	207#	209#	211#
		213#	215#	217#	219#	221#								
.	= 037424	92#	93	96#	103#	104#	105#	106#	109#	111#	114#	118#	120#	165#
		166#	167#	168#	169#	170#	281#	283#	284#	285#	286#	287#	288#	289#
		290#	291#	292#	294#	295#	296#	297#	298#	299#	300#	301#	302#	303#
		305#	306#	307#	308#	309#	310#	311#	312#	313#	314#	316#	317#	318#
		319#	320#	321#	322#	323#	324#	325#	327#	328#	329#	330#	331#	332#
		333#	334#	335#	336#	338#	339#	340#	341#	342#	343#	344#	345#	346#
		347#	349#	350#	351#	352#	353#	354#	355#	356#	357#	358#	360#	361#
		362#	363#	364#	365#	366#	367#	368#	369#	433	522	769	851	860
		874	912#	922	929	943	970#	972#	974#	988	1179	1200	1341	1397
		1470	1526	1852	1876	1898	1960	2033	2095	2099	2159	2183	2304	2325
		2399	2479	2644	2870	2959	3043	3094	3211	3214	3218	3221	3243	3254

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 90

CZDVDD.P11 02-JUN-80 09:32

CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0099

DVEND	1#	465													
DVFRNT	1#														
HLT	55#	909	1281	1291	1705	1716	1726	1853	1877	1899	1966	2043	2100	2104	2108
	2112	2305	2505	2515	2670	2680	2883	2897	3076	3081	3215	3222	3268	3284	3297
	3309	3321	3333	3471	3498	3826	3829	3830	3908						
SADJUS	1#	1359	1488	1654	1666	2437	2602	2833	2998	3164	3231	3396	3408	3917	
SBEAK	1#	1856	1878												
SBUFFE	1#	966													
SCK15	1#														
SCK150	1#														
SCLR.T	1#														
SCYCLE	1#	975													
SEOP	1#	465													
SFINI	1#	3997													
SGETFL	1#														
SGETPA	1#	1070													
SHEADE	1#														
SLC16	1#	1213													
SLC16A	1#	1305	1434												
SLC17	1#	1563													
SLC18	1#	1753													
SLC19	1#	2053													
SLC20	1#	2120													
SLC21	1#	2221													
SLC22	1#	2353	2523												
SLC30	1#	2688													
SLC31	1#	2911													
SLC32	1#	3101													
SLC33	1#	3342													
SMSG	1#	960													
SPFAIL	1#	844													
SRAMCL	1#	871													
SRXSHI	1#														
SSCOPE	1#	512													
SSETAS	1#	1262	1382	1511	2464	2629	2856	3022	3187	3427	3940				
SSETLI	1#	1220	1314	1443	1580	1767	1916	1990	2062	2132	2232	2371	2539	2718	2931
	3118	3354													
SSETSC	1#	3955													
SSETSY	1#	3837													
SSET.T	1#														
SSILOI	1#														
SSIMBC	1#	3844													
STRPDE	1#	191	193	195	197	199	201	203	205	207	209	211	213	215	217
	219														
STSTN	1#	1220	1314	1443	1580	1767	1916	1990	2062	2132	2232	2371	2539	2718	2931
	3118	3354													
STXSHI	1#														
SVARIA	1#	117													
SXZ	1#	1214	1218	1306	1312	1435	1441	1564	1578	1754	1765	1908	1914	1981	1988
	2054	2060	2121	2130	2222	2230	2354	2369	2524	2537	2689	2714	2911	2928	3102
	3116	3343	3352												

. ABS. 037424 000

CZDVD-D MACY11 30A(1052) 17-JUN-80 15:00 PAGE 91
CZDVDD.P11 02-JUN-80 09:32 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0100

ERRORS DETECTED: 0

CZDVDD.BIN,CZDVDD.LST/CRF/SOL/NL:TOC=CZDVDD.MAC,CZDVDD.P11

RUN-TIME: 33 48 4 SECONDS

RUN-TIME RATIO: 281/86=3.2

CORE USED: 29K (57 PAGES)