

IDENTIFICATION

SEQ 0001

PRODUCT CODE: AC-8740C-MC
PRODUCT NAME: CZDVDC0 DV11 RDM TST PRT2
DATE RELEASED: FEB-1978
MAINTAINER: DIAG-MK
AUTHOR: J EGOLF J VALDES

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.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OF RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975, 1978 DIGITAL EQUIPMENT CORPORATION

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 (DZDVE 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).

DZDVD LIKE DZDVC ALLOWS THE MICRO PROCESSOR TO "FREE RUN". BECAUSE OF THE LENGTH OF THE "FREE RUNNING" TEST; THIS TEST IS TO CATCH THE OVERFLOW FROM DZDVC.

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. DZDVA [REV] BASIS R/W TEST AND ROM INSTRUCTION EXERCISER.
2. DZDVB [REV] STATIC LINE CARD TESTS.
3. DZDVC [REV] 'FREE RUNNING' ROM TESTS PART 1.
4. DZDVD [REV] 'FREE RUNNING' ROM TESTS PART 2.
5. DZDVE [REV] MODEM CONTROL AND CABLE TESTS PLUS MANUAL PARAMETER INPUT. [TRIAL PROGRAM]
6. 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 ST-RT 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-DD 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 SWITCHS 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 CANN'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 INTERMITTEMT ERRORS; SW14=1 WILL LOOP ON TEST REGARDLESS OF ERROR OR NOT ERROR.
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 INTERPEDITED.

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 21524 FROM 403 TO 240
LOC 21532 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' (ADDRESS 1224) 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 COMPLETELY 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. PARITY OFF.
3. 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 DZDVD-B CSR: 175000 VEC: 300 PASSES: 000001 ERRORS: 000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE NOT NECESSARILY THE VALUES FOR THE DEVICE. THEY ARE ONLY FOR THIS EXAMPLE.

NOTE: DZDVE (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 OCCURED THE BELL MAY BE MISTAKEN FOR END PASS. THE PASS EXECUTION IS SO FAST THAT THE STANDARD END PASS WAS TOO LENGTHLY. THEREFORE EACH CHAR IS AN "END PASS AND THE ENTIRE "END" IS NOT REQUIRED FOR ACCEPTANCE.

8.4 KEY LOCATIONS

RETURN (1212) CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS ASSERTED.

NEXT (1214) CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.

TSTNO (1224) CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.

RUN (1302) 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 (1276) 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 (1356) CONTAINS THE RECEIVER CSR OF THE CURRENT DV11 UNDER TEST.

L00.03 (1412)
L04.07 (1414)
L08.11 (1416)
L12.15 (1420)

CONTAINS THE STATUS OF THE CURRENT DV11 UNDER TEST.

BIT 15 SET: LINE CARD *NOT INSTALLED (AND WONT BE TESTED)

BIT 14 SET: RESERVED

BIT 13 SET: RESERVED

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.

8.4A MORE ON THAT 'STATUS TABLE' (1500-1736)

```
'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: RESERVED
BIT 13 SET: RESERVED
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.
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 OCCURES; THE POINTER (HOLDING 175000) IS UPDATED BY 10 AND THE ABOVE IS REPEATED UNTILL 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 OCCURES (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 OCCURED; 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 (APPROIATE) LINE CARDS MISSING
- 2) TWO SYNCs.
SET BIT12 IF YOU HAVE A 4 LINE GROUP SET FOR 1 SYNC.
- 3) EIGHT BITS PER CHAR.
ADJUST BITS 9 AND BIT 8 IN STATUS MAP FOR YOUR CORRECT CONFIG.
- 4) SYNCHRONOUS LINE CARDS INSTALLED
SET BIT11 OF STATUS MAP FOR ASYNC LINE CARD AND ZERO SYNC CHARS.
- 5) SYNC "A"=226 AND SYNC "B"=062

IN ALL ADJUSTMENTS PLEASE REFER TO SECTION 8.4A FOR GRETER DETAIL.

```

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

```

```

;*AC-8740C-MC/<377>/"CZDVDC0 DV11" ROM TST PRT2
;*COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
;-----
;STARTING PROCEDURE
;LOAD PROGRAM
;LJAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE "AC-8740C-MC/<377>/"CZDVDC0 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 INITAL START UP.
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

```

```

;LOCK ON TEST SELECT
;RESTART PROGRAM AT SELECTED TEST
;RESELECT DV11 DESIRED ACTIVE
;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT

```

```

36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82

```

```

;REGISTER DEFINITIONS
;-----
R0=%0           ;GENERAL REGISTER
R1=%1           ;GENERAL REGISTER
R2=%2           ;GENERAL REGISTER
R3=%3           ;GENERAL REGISTER
R4=%4           ;GENERAL REGISTER
R5=%5           ;GENERAL REGISTER
SP=%6           ;PROCESSOR STACK POINTER
PC=%7           ;PROGRAM COUNTER

;LOCATION EQUIVALENCIES
;-----
PS=177776      ;PROCESSOR STATUS WORD
STACK=1200     ;START OF PROCESSOR STACK

BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1

;-----
ALU=BIT12
RAM=BIT13
XFR=BIT13+BIT12
NPR=BIT14
S.C=BIT14+BIT12
BCC=BIT14+BIT13
BRB=BIT14+BIT13+BIT12
;-----

```

```

83 ;*****
84 ;-----
85 ;TRAPCATCHER 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          ;STANDARD INTERRUPT VECTORS
93          ;-----
94
95
96          000024          .=24          .PFAIL          ;POWER FAIL HANDLER
97 000024 004402          340          ;SERVICE AT LEVEL 7
98 000026 000340          .HLT          ;ERROR HANDLER
99 000030 004002          340          ;SERVICE AT LEVEL 7
100 000032 000340          .TRPSRV         ;GENERAL HANDLER DISPATCH SERVICE
101 000034 003750          340          ;SERVICE AT LEVEL 7
102 000036 000340          ;SERVICE AT LEVEL 7
103          000040          .=40
104 000040 000001          .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          .=174
110 000174 000000          LIGHT: 0
111          000176          .=176
112 000176 000000          SSWR: 0
113
114          000200          .=200
115 000200 000197 001742          JMP .START          ;GO TO START OF PROGRAM
116
117
118          001000          .=1000
119 001000 005377 041501 034055          MTITLE: .ASCIZ <377><12>/AC-8740C-MC/<377>/"CZDVDC0 DV11" ROM TST PRT2/<377>
120          001200          .=1200
121 001200          LIGHTS:
122 001200 177570          177570
123 001202 177570          SWR: 177570
124          ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
125          ;-----
126          TKCSR: 177560          ;TELETYPE KEYBOARD CONTROL REGISTER
127 001204 177560          TKDBR: 177562          ;TELETYPE KEYBOARD DATA BUFFER
128 001206 177562          TPCSR: 177564          ;TELEPRINTER CONTROL REGISTER
129 001210 177564          TPDBR: 177566          ;TELEPRINTER DATA BUFFER
130 001212 177566
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

```

```

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          CLX: 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          .EVEN
171 001306 001306          CREAM: DV.MAP     ;TABLE POINTER.

```

```

172                                     ;PROGRAM CONTROL FLAGS
173                                     ;-----
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                                     SY=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 001314      104401     .SCOPE
194 001316      003020     SCOP1=TRAP+1      ;CALL TO LOOP ON CURRENT DATA HANDLER
195 001316      104402     .SCOP1
196 001320      003044     TYPE=TRAP+2      ;CALL TO TELETYPE OUTPUT ROUTINE
197 001320      104403     .TYPE
198 001322      003120     INSTR=TRAP+3      ;CALL TO ASCII STRING INPUT ROUTINE
199 001322      104404     .INSTR
200 001324      003224     INSTER=TRAP+4     ;CALL TO INPUT ERROR HANDLER
201 001324      104405     .INSTER
202 001326      003244     PARAM=TRAP+5     ;CALL TO NUMERICAL DATA INPUT ROUTINE
203 001326      104406     .PARAM
204 001330      003444     SAV05=TRAP+6     ;CALL TO REGISTER SAVE ROUTINE
205 001330      104407     .SAV05
206 001332      003504     RES05=TRAP+7     ;CALL TO REGISTER RESTORE ROUTINE
207 001332      104410     .RES05
208 001334      003536     CONVRT=TRAP+10   ;CALL TO DATA OUTPUT ROUTINE
209 001334      104411     .CONVRT
210 001336      003542     CNVRT=TRAP+11   ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
211 001336      104412     .CNVRT
212 001340      004556     MSTCLR=TRAP+12   ;CALL TO ISUE A MASTER CLEAR
213 001340      104413     .MSTCLR
214 001342      004516     RAMCLR=TRAP+13   ;CALL TO CLEAR THE RAMS
215 001342      104414     .RAMCLR
216 001344      004476     DELAY=TRAP+14   ;CALL TO VARIABLE DELAY COUNTER
217 001344      104415     .DELAY
218 001346      004566     ROMCLK=TRAP+15   ;CALL TO CLOCK ROM ONCE
219 001346      104416     .ROMCLK
220 001350      004576     DATACLK=TRAP+16 ;CALL TO CLK DATA
221 001350      104417     .DATACLK
222
223                                     ;*****

```

```

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     DVLCR: 0     ;POINTER TO DV11 NEXT RECEIVED CHARACTER REGISTER
233 001370      000000     DVLSR: 0     ;POINTER TO DV11 LINE PARAMETER REGISTER
234 001372      000000     DVRSR: 0     ;POINTER TO DV11 SECONDARY REGISTER SELECT REGISTER
235 001374      000000     DVRSRH: 0    ;POINTER TO DV11 SECONDARY REGISTER SELECT HIGH BYTE.
236 001376      000000     DVSRAL: 0   ;POINTER TO DV11 SECONDARY REGISTER ACCESS REGISTER
237 001400      000000     DVSEFR: 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      000      MASK.A: .BYTE 000      ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 00-03
246 001407      000      MASK.B: .BYTE 000      ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 04-07
247 001410      00J      MASK.C: .BYTE 000      ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 08-11
248 001411      000      MASK.D: .BYTE 000      ;LAST CHAR TO TEST AND PARITY MASK FOR LINES 12-15
249
250 001412      010      CLK.A: .BYTE B.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 00-03
251 001413      010      CLK.B: .BYTE B.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 04-07
252 001414      010      CLK.C: .BYTE B.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 08-11
253 001415      010      CLK.D: .BYTE B.      ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR FOR LINES 12-15
254
255 001416      000000     L00.03: 000000     ;PARAMETERS FOR LINES 00-03
256 001420      000000     L04.07: 000000     ;PARAMETERS FOR LINES 04-07
257 001422      000000     L08.11: 000000     ;PARAMETERS FOR LINES 08-11
258 001424      000000     L12.15: 000000     ;PARAMETERS FOR LINES 12-15
259
260 001426      000000     SYNC2A: 000000     ;SYNC 2
261 001430      000000     SYNC2B: 000000     ;
262 001432      000000     SYNC2C: 000000     ;
263 001434      000000     SYNC2D: 000000     ;
264
265                                     ;SUMMARY
266                                     ;-----
267 ;      MASK.X      040      5 BITS PER CHAR.
268 ;
269 ;      100      6 BITS PER CHAR.
270 ;      200      7 BITS PER CHAR.
271 ;      000      8 BITS PER CHAR.
272
273 ;      CLK.X      005      5 BITS PER CHAR.
274 ;      006      6 BITS PER CHAR.
275 ;      007      7 BITS PER CHAR.
276 ;      010      8 BITS PER CHAR.

```

;DV11 STATUS TABLE AND ADDRESS ASSIGNMENTS

276
277
278
279 001500 001500 .=1500
280 001500 DV.MAP:
281 001500 000001 DVCRO0: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 00
282 001502 000001 DVTR00: .BLKW 1 ;VECTOR "A" FOR DV11 NUMBER 00
283 001504 000001 DV00.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 00
284 001506 000001 SYNA00: .BLKW 1 ;SYNC TWO
285 001510 000001 DV00.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 00
286 001512 000001 SYNBOJ: .BLKW 1 ;SYNC TWO
287 001514 000001 DV00.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 00
288 001516 000001 SYNC00: .BLKW 1 ;SYNC TWO
289 001520 000001 DV00.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 00
290 001522 000001 SYND00: .BLKW 1 ;SYNC TWO
291
292 001524 000001 DVCRO1: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 01
293 001526 000001 DVTR01: .BLKW 1 ;VECTOR "A" FOR DV11 NUMBER 01
294 001530 000001 DV01.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 01
295 001532 000001 SYNA01: .BLKW 1 ;SYNC TWO
296 001534 000001 DV01.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 01
297 001536 000001 SYNBO1: .BLKW 1 ;SYNC TWO
298 001540 000001 DV01.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 01
299 001542 000001 SYNC01: .BLKW 1 ;SYNC TWO
300 001544 000001 DV01.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 01
301 001546 000001 SYND01: .BLKW 1 ;SYNC TWO
302
303 001550 000001 DVCRO2: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 02
304 001552 000001 DVTR02: .BLKW 1 ;VECTOR "A" FOR DV11 NUMBER 02
305 001554 000001 DV02.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 02
306 001556 000001 SYNA02: .BLKW 1 ;SYNC TWO
307 001560 000001 DV02.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 02
308 001562 000001 SYNBO2: .BLKW 1 ;SYNC TWO
309 001564 000001 DV02.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 02
310 001566 000001 SYNC02: .BLKW 1 ;SYNC TWO
311 001570 000001 DV02.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 02
312 001572 000001 SYND02: .BLKW 1 ;SYNC TWO
313
314 001574 000001 DVCRO3: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 03
315 001576 000001 DVTR03: .BLKW 1 ;VECTOR "A" FOR DV11 NUMBER 03
316 001600 000001 DV03.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 03
317 001602 000001 SYNA03: .BLKW 1 ;SYNC TWO
318 001604 000001 DV03.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 03
319 001606 000001 SYNBO3: .BLKW 1 ;SYNC TWO
320 001610 000001 DV03.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 03
321 001612 000001 SYNC03: .BLKW 1 ;SYNC TWO
322 001614 000001 DV03.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 03
323 001616 000001 SYND03: .BLKW 1 ;SYNC TWO
324
325 001620 000001 DVCRO4: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 04
326 001622 000001 DVTR04: .BLKW 1 ;VECTOR "A" FOR DV11 NUMBER 04
327 001624 000001 DV04.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 04
328 001626 000001 SYNA04: .BLKW 1 ;SYNC TWO
329 001630 000001 DV04.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 04
330 001632 000001 SYNBO4: .BLKW 1 ;SYNC TWO
331 001634 000001 DV04.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 04

332 001636 000001 SYNC04: .BLKW 1 ;SYNC TWO
333 001640 000001 DV04.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 04
334 001642 000001 SYND04: .BLKW 1 ;SYNC TWO
335
336 001644 000001 DVCRO5: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 05
337 001646 000001 DVTR05: .BLKW 1 ;VECTOR "A" FOR DV11 NUMBER 05
338 001650 000001 DV05.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 05
339 001652 000001 SYNA05: .BLKW 1 ;SYNC TWO
340 001654 000001 DV05.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 05
341 001656 000001 SYNBO5: .BLKW 1 ;SYNC TWO
342 001660 000001 DV05.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 05
343 001662 000001 SYNC05: .BLKW 1 ;SYNC TWO
344 001664 000001 DV05.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 05
345 001666 000001 SYND05: .BLKW 1 ;SYNC TWO
346
347 001670 000001 DVCRO6: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 06
348 001672 000001 DVTR06: .BLKW 1 ;VECTOR "A" FOR DV11 NUMBER 06
349 001674 000001 DV06.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 06
350 001676 000001 SYNA06: .BLKW 1 ;SYNC TWO
351 001700 000001 DV06.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 06
352 001702 000001 SYNBO6: .BLKW 1 ;SYNC TWO
353 001704 000001 DV06.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 06
354 001706 000001 SYNC06: .BLKW 1 ;SYNC TWO
355 001710 000001 DV06.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 06
356 001712 000001 SYND06: .BLKW 1 ;SYNC TWO
357
358 001714 000001 DVCRO7: .BLKW 1 ;CONTROL STATUS REGISTER FOR DV11 NUMBER 07
359 001716 000001 DVTR07: .BLKW 1 ;VECTOR "A" FOR DV11 NUMBER 07
360 001720 000001 DV07.A: .BLKW 1 ;PARAMETER FOR LINES 00-03 FOR DV11 NUMBER 07
361 001722 000001 SYNA07: .BLKW 1 ;SYNC TWO
362 001724 000001 DV07.B: .BLKW 1 ;PARAMETER FOR LINES 04-07 FOR DV11 NUMBER 07
363 001726 000001 SYNBO7: .BLKW 1 ;SYNC TWO
364 001730 000001 DV07.C: .BLKW 1 ;PARAMETER FOR LINES 08-11 FOR DV11 NUMBER 07
365 001732 000001 SYNC07: .BLKW 1 ;SYNC TWO
366 001734 000001 DV07.D: .BLKW 1 ;PARAMETER FOR LINES 12-15 FOR DV11 NUMBER 07
367 001736 000001 SYND07: .BLKW 1 ;SYNC TWO
368
369 001740 000000 DV.END: 000000

```

370
371 ;PROGRAM INITIALIZATION
372 ;LOCK OUT INTERRUPTS
373 ;SET UP PROCESSOR STACK
374 ;SET UP POWER FAIL VECTOR
375 ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
376 ;TYPE TITLE MESSAGE
377
378 001742 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
379 001750 012708 001200 MOV #STACK,SP ;SET UP STACK
380 001754 012737 004402 000024 MOV #PFAIL,@#24 ;SET UP POWER FAIL VECTOR
381 001762 113737 001301 001303 MOV#B DVNUM,SAVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM.
382 001770 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
383 001774 105037 001311 CLR#B ERRFLG ;CLEAR ERROR FLAG
384 002000 105037 001313 CLR#B QV,FLG ;ZERO QUICK VERIFY FLAG
385 002004 012737 001500 001306 MOV #DV,MAP,CREAM ;GET MAP POINTER.
386 002012 112737 000001 001304 MOV#B #1,RUN ;POINT POINTER TO FIRST DEVICE.
387 002020 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
388 002024 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
389 002030 012737 000001 001226 MOV #1,TESTNO ;SET UP FOR TEST 1
390 002036 012737 001742 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
391 ;TESTING STARTS
392 002044 105737 001310 TST#B INIFLG ;HAS INITIALIZATION BEEN PERFORMED
393 002050 001063 BNE 1$ ;BR IF YES
394 002052 013746 000004 MOV 4,-(SP) ;
395 002056 013746 000006 MOV 6,-(SP) ;
396 002062 005037 000006 CLR 6 ;
397 002066 012737 002104 000004 MOV #80$,4 ;
398 002074 005777 177102 TST @SWR ;
399 002100 000240 NOP ;
400 002102 000407 BR 81$ ;
401 002104 022626 80$: CMP (SP)+,(SP)+ ;
402 002106 012737 000174 001200 MOV #LIGHT,LIGHTS ;
403 002114 012737 000176 001202 MOV #SSWR,SWR ;
404 002122 012637 000006 81$: MOV (SP)+,6 ;
405 002126 012637 000004 MOV (SP)+,4 ;
406 002132 104402 001000 TYPE ;TYPE TITLE MESSAGE
407 002136 105137 001310 COMB INIFLG ;IF NOT SET FLAG AND DO
408 002142 105777 177034 TST#B @SWR ;BIT7=1??
409 002146 100402 BMI 16$ ;BR IF NO AUTO SIZE
410 002150 004737 006624 JSR PC,CSRMAP ;GO DO THE AUTO SIZE
411 002154 104402 005461 16$: TYPE ;TYPE HEADER
412 002160 012737 001500 001246 MOV #DV,MAP,TEMP1 ;SET POINTER
413 002166 017737 177054 5$: MOV @TEMP1,TEMP2 ;SET DATA
414 002174 022737 177777 001250 CMP #177777,TEMP2 ;ALL DONE?
415 002202 001406 BEQ 1$ ;BR IF YES
416 002204 104410 CONVRT ;
417 002206 005037 XSTATQ ;
418 002210 062737 000002 001246 ADD #2,TEMP1 ;UPDATE POINTER
419 002216 000763 BR 5$ ;
420 002220 005737 000042 1$: TST @#42 ;IS PROGRAM RUNNING UNDER MONITOR
421 002224 001030 BNE 3$ ;BR IF YES
422 002226 032777 000001 176746 BIT #SW00,@SWR ;SELECT SPECIFIC DEVICES??
423 002234 001424 BEQ 3$ ;BR IF NO.
424 002236 104402 005402 TYPE ,MNEW ;TYPE THE MESSAGE.
425 002242 005000 CLR RO ;ZERO DATA LIGHTS
  
```

```

426 002244 000000 HALT ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
427 002246 127737 176730 001302 CMB#B @SWR,SAVACT ;IS THE NUMBER VALID?
428 002254 101404 BLOS 2$ ;BR IF NUMBER IS OK.
429 002256 104402 005243 TYPE ,MERR3 ;TELL USER OF INVALID NUMBER.
430 002262 000000 HALT ;STOP EVERY THING.
431 002264 000776 BR -2 ;RESTART THE PROGRAM AGAIN.
432 002266 117737 176710 001300 2$: MOV#B @SWR,DVACTV ;GET NEW DEVICE PATTERN
433 002274 113700 001300 MOV#B DVACTV,RO ;SHOW THE USER WHAT HE SELECTED.
434 002300 042700 177400 BIC #C<377>,RO ;USE ONLY LOW BYTE.
435 002304 000000 HALT ;CONTINUE DYNAMIC SWITCHES.
436 002306 002700 000300 3$: MOV #300,RO ;PREPARE TO CLEAR THE FLOATING
437 002312 012701 000302 MOV #302,R1 ;VECTOR AREA. 300-776
438 002316 010120 4$: MOV R1,(R0)+ ;START PUTTING "PC+2 - HALT"
439 002320 005021 CLR (R1)+ ;IN VECTOR AREA.
440 002322 022021 CMP (R0)+,(R1)+ ;POP POINTERS
441 002324 022700 001000 CMP #1000,RO ;ALL DONE??
442 002330 001372 BNE 4$ ;BR IF NO.
443
444 ;TEST START AND RESTART
445 -----
446
447 002332 012737 000340 177776 .BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
448 002340 012706 001200 MOV #STACK,SP ;SET UP STACK
449 002344 005737 000042 TST @#42 ;IS PROGRAM UNDER MONITOR CONTROL
450 002350 001023 BNE 3$ ;BR IF YES
451 002352 032777 000004 176622 BIT #BIT2,@SWR ;CHECK FOR LOCK ON TEST
452 002360 001411 BEQ 1$ ;BR IF NO LOCK DESIRED.
453 002362 104402 005301 TYPE ,MLDCK ;TYPE LOCK SELECTED.
454 002366 012737 000240 002702 MOV #NOP,TTST ;ADJUST SCOPE ROUTINE.
455 002374 012737 000240 002704 MOV #NOP,TTST+2 ;SET UP TO LOCK
456 002402 000406 BR 2$ ;CONTINUE ALONG.
457 002404 013737 003014 002702 1$: MOV BRW,TTST ;PREPARE NORMAL SCOPE ROUTINE
458 002412 013737 003016 002704 MOV BRX,TTST+2 ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
459 002420 2$:
460 002420 012737 005666 001214 3$: MOV #CYCLE,RETURN ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
461 002426 104402 005171 4$: TYPE ,MR ;TYPE R
462 002432 000177 176556 JMP @RETURN ;START TESTING
  
```



```

463 ;END OF PASS
464 ;TYPE NAME OF TEST
465 ;UPDATE PASS COUNT
466 ;CHECK FOR EXIT TO ACT-11
467 ;RESTART TEST
468
469 002436 000005 .EOP: RESET ;MAKE THE WORLD CLEAN AGAIN.
470 002440 005037 001234 CLR LSTERR ;CLEAR LAST ERROR PC
471 002444 105037 001311 CLR ERRFLG ;CLEAR ERROR FLAG
472 002450 005237 001230 INC PASCNT ;UPDATE PASS COUNT
473 002454 013777 001230 176516 MOV PASCNT,@LIGHTS ;DISPLAY PASS COUNT
474 002462 104402 005145 TYPE ,MEPASS ;TYPE END PASS
475 002466 104402 005330 TYPE ,MCSRX ;TYPE CSR
476 002472 104411 002664 CNVRT ,XCSR ;SHOW IT
477 002476 104402 005336 TYPE ,MVECX ;TYPE VECTOR
478 002502 104411 002812 CNVRT ,XVEC ;SHOW IT
479 002506 104402 005344 TYPE ,MPASSX ;TYPE PASSES
480 002512 104411 002620 CNVRT ,XPASS ;SHOW IT
481 002516 104402 005355 TYPE ,MERRX ;TYPE ERRORS
482 002522 104411 002626 CNVRT ,XERR ;SHOW IT
483 002526 105337 001303 DECB SAVNUM ;ARE ALL DEVICES TESTED?
484 002532 001017 BNE RESTRT ;BR IF NO.
485 002534 112737 000377 001313 MOV #377,QV.FLG ;SET THE QUICK VERIFY FLAG.
486 002542 113737 001301 001303 MOV DVNUM,SAVNUM ;RESTORE THE COUNT
487 002550 013701 000042 MOV @#42,R1 ;CHECK FOR ACT-11 OR DDP
488 002554 001406 BEQ RESTRT ;IF NOT, CONTINUE TESTING
489 002556 000005 RESET ;STOP THE SHOW--CLEAR THE WORLD
490 002560
491 002560 004711 LOGICAL: JSR PC,(R1)
492 002562 000240 NOP
493 002564 000240 NOP
494 002566 000240 NOP
495 002570 000240 NOP
496 002572 012737 005666 001214 RESTRT: MOV #CYCLE,RETURN
497 002600 000137 005666 JMP CYCLE
498 002604 000001 XCSR: 1
499 002606 006 002 .BYTE 6,2
500 002610 001362 DVSCR
501 002612 000001 XVEC: 1
502 002614 003 002 .BYTE 3,2
503 002616 001352 DVRVEC
504 002620 000001 XPASS: 1
505 002622 006 002 .BYTE 6,2
506 002624 001270 PASCNT
507 002626 000001 XERR: 1
508 002630 006 002 .BYTE 6,2
509 002632 001232 ERRCNT
510
511 ;SCOPE LOOP AND INTERATION HANDLER
512
513
514 002634 .SCOPE:
515 002634 022737 177570 001202 CMP #177570,SWR ;IS THERE A REAL SWR?
516 002642 001411 BEQ 64$ ;BR IF YES
517 002644 017746 176336 MOV @TKDBR,-(SP) ;SAVE KEYBOARD CHAR
518 002650 042716 000200 BIC #BIT7,(SP) ;CLEAR PARITY BIT

```

```

519 002654 122726 000007 CMPB #7,(SP)+ ;WAS IT CNTRL 'G' ?
520 002660 001002 BNE +6 ;BR IF NO.
521 002662 004737 004640 JSR PC,SERV.G ;SERVICE 'CNTRL 'G''.
522 002666 005037 001234 64$: CLR LSTERR ;CLEAR LAST ERROR PC.
523 002672 010016 MOV RO,(SP) ;SAVE RO ON THE STACK
524 002674 032777 040000 176300 TTST: BIT #BIT14,@SWR ;"LOOP ON THIS TEST"?
525 002702 001407 BEQ 1$ ;BR IF NO. (IF LOCK SW01=1; THIS LOC #240)
526 002704 000437 BR 3$ ;GOTO 3$ (IF LOCK SW01=1; THIS LOC #240)
527 002708 105777 176272 TSTB @TKCSR ;KEYBOARD DONE?
528 002712 100034 BPL 3$ ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
529 002714 017700 176266 MOV @TKDBR,RO ;CLEAR DONE BIT
530 002720 000415 BR 2$ ;CONTINUE
531 002722 032777 004000 176252 1$: BIT #SW11,@SWR ;DELETE ITERATION? (QUICK PASS)
532 002730 001011 BNE 2$ ;BR IF YES
533 002732 105737 001313 TSTB QV.FLG ;HAVE PASSES BEECOMPLETED?
534 002736 001406 BEQ 2$ ;BR IF QUICK PASS.
535 002740 005237 001224 INC LPCNT ;UPDATE ITERATION COUNTER
536 002744 023737 001224 001222 CMP LPCNT,ICOUNT ;ARE ALL ITERATIONS DONE??
537 002752 001014 BNE 3$ ;BR IF NOT YET
538 002754 105037 001311 2$: CLR ERRFLG ;PREPARE FOR NEW TEST
539 002760 005037 001224 CLR LPCNT ;START ICOUNTER AT 0
540 002764 005037 001220 CLR LOCK
541 002770 012737 000024 001222 MOV #20,ICOUNT ;RESET ITERATIONS
542 002776 013737 001216 001214 MOV NEXT,RETURN ;GET NEXT TEST
543 003004 011600 3$: MOV (SP),RO ;POP RO OFF OF THE STACK
544 003006 022626 POP2SP ;FAKE AN "RTI"
545 003010 000177 176200 JMP @RETURN ;GO DO THE TEST
546 003014 001407 BRW: 1407
547 003016 000437 BRX: 437
548
549 ;CHECK FOR FREEZE ON CURRENT DATA
550
551
552 003020 032777 001000 176154 .SCOP1: BIT #SW09,@SWR ;IS SW09=1(SET)?
553 003026 001405 BEQ 1$ ;BR IF NOT SET.
554 003030 005737 001220 TST LOCK
555 003034 001402 BEQ 1$
556 003036 013716 001220 1$: MOV LOCK,(SP) ;GOTO THE ADDRESS IN LOCK.
557 003042 000002 RTI ;GO BACK.
558
559 ;TELETYPE OUTPUT ROUTINE
560
561
562 003044 010546 .TYPE: MOV R5,-(SP) ;SAVE R5 ON THE STACK.
563 003046 017605 MOV @2(SP),R5 ;GET ADDRESS OF MESSAGE.
564 003052 062766 000002 000002 ADD #2,2(SP) ;POP OVER ADDRESS.
565 003060 032777 010000 176114 1$: BIT #SW12,@SWR ;INHIBIT ALL PRINT OUT??
566 003066 001012 BNE 3$ ;BR IF NO PRINT OUT WANTED (SW12=1)
567 003070 105715 TSTB (R5) ;IS NUMBER MINUS? (MSB=1(BIT7))
568 003072 100002 BPL 2$ ;BR IF NUMBER IS PLUS
569 003074 104402 005104 TYPE ,MCRLF ;TYPE A CR/LF!
570 003100 105777 176104 2$: TSTB @TPCSR ;TTY READY?
571 003104 100375 BPL 2$ ;BR IF NO.
572 003106 112577 176100 MOVB (R5)+,@TPDBR ;PRINT CURRENT CHAR.
573 003112 001362 BNE 1$ ;IF NOT ZERO KEEP PRINTING!
574 003114 012605 3$: MOV (SP)+,R5 ;END OF OUTPUT. RESTORE R5

```

```

575 003116 000002          RTI          ;GO HOME
576          ;-----
577
578 003120 010346          .INSTR: MOV    R3,-(SP)          ;SAVE R3 ON STACK
579 003122 010446          MOV    R4,-(SP)          ;SAVE R4 ON STACK
580 003124 017637 000004 003142      MOV    #4(SP),,MSG
581 003132 062766 000002 000004      ADD    #2,4(SP)
582 003140 104402          .INST1: TYPE
583 003142 000000          .MSG: 0
584 003144 012704 005520          MOV    -#INBUF,R4
585 003150 0 2703 000007          MOV    #7,R3
586 003154 105777 176024          1$: TSTB  @TKCSR
587 003160 100375          BPL    1$
588 003162 117714 176020          MOVB  @TKDBR,(R4)
589 003166 142714 000200          BICB  #200,(R4)
590 003172 122427 000015          CMPB  (R4),#15
591 003178 001417          BEQ   INSTR2
592 003200 105777 176004          2$: TSTB  @TPCSR
593 003204 100375          BPL    2$
594 003208 017777 175774 175778      MOV    @TKDBR,@TPDBR
595 003214 005303          DEC   R3
596 003216 001356          BNE   1$
597 003220 012604          MOV    (SP)+,R4
598 003222 012603          MOV    (SP)+,R3
599 003224 104402 005100          .INSTE: TYPE
600 003230 010346          MOV    R3,-(SP)
601 003232 010446          MOV    R4,-(SP)
602 003234 000741          BR     .INST1
603 003236 012604          INSTR2: MOV   (SP)+,R4          ;RESTORE R4
604 003240 012603          MOV   (SP)+,R3          ;RESTORE R3
605 003242 000002          RTI
606
607          ;CONVERT ASCII STRING TO OCTAL
608          ;-----
609
610 003244 010548          .PARAM: MOV   R5,-(SP)
611 003246 010446          MOV   R4,-(SP)
612 003250 016605 000004          MOV   4(SP),R5
613 003254 012537 003434          MOV   (R5)+,LOLIM
614 003260 012537 003436          MOV   (R5)+,HILIM
615 003264 012537 003440          MOV   (R5)+,DEVADR
616 003270 112537 003442          MOVB  (R5)+,LOBITS
617 003274 112537 003443          MOVB  (R5)+,ADRCNT
618 003300 010566 000004          MOV   R5,4(SP)
619 003304 005005          PARAM1: CLR   R5
620 003306 012704 005520          MOV   #INBUF,R4
621 003312 122714 000015          CMPB  #15,(R4)
622 003316 001420          BEQ   PARERR
623 003320 121427 000060          1$: CMPB  (R4),#60
624 003324 002415          BLT   PARERR
625 003326 121427 000067          CMPB  (R4),#67
626 003332 003012          BGT   PARERR
627 003334 142714 000060          BICB  #60,(R4)
628 003340 152405          BISB  (R4)+,R5
629 003342 122714 000015          CMPB  #15,(R4)
630 003346 001408          BEQ   LIMITS
    
```

```

631 003350 006305          ASL   R5
632 003352 006305          ASL   R5
633 003354 006305          ASL   R5
634 003356 000760          BR    1$
635 003360 104404          PARERR: INSTER
636 003362 000750          BR    PARAM1
637
638          ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
639          ;-----
640
641 003364 020537 003436          LIMITS: CMP   R5,HILIM
642 003370 101373          BHI   PARERR
643 003372 020537 003434          CMP   R5,LOLIM
644 003376 103770          BLD   PARERR
645 003400 133705 003442          BITB  LOBITS,R5
646 003404 001365          BNE   PARERR
647
648          ;STORE NUMBER AT SPECIFIED ADDRESS
649
650 003406 013704 003440          MOV   DEVADR,R4
651 003412 010524          1$: MOV   R5,(R4)+
652 003414 062705 000002          ADD   #2,R5
653 003420 105337 003443          DECB  ADRCNT
654 003424 001372          BNE   1$
655 003426 012604          MOV   (SP)+,R4
656 003430 012605          MOV   (SP)+,R5
657 003432 000002          RTI
658 003434 000000          LOLIM: 0
659 003436 000000          HILIM: 0
660 003440 000000          DEVADR: 0
661 003442 000000          LOBITS: 0
662          ADRCNT=LOBITS+1
663
664          ;SAVE PC OF TEST THAT FAILED AND R0-R5
665          ;-----
666
667 003444 016637 000004 001276 .SAV05: MOV   4(SP),SAVPC          ;SAVE R7 (PC)
668
669          ;SAVE R0-R5
670
671 003452 010537 001272          SV05: MOV   R5,SAVR5          ;SAVE R5
672 003456 010437 001270          MOV   R4,SAVR4          ;SAVE R4
673 003462 010337 001268          MOV   R3,SAVR3          ;SAVE R3
674 003466 010237 001264          MOV   R2,SAVR2          ;SAVE R2
675 003472 010137 001262          MOV   R1,SAVR1          ;SAVE R1
676 003476 010037 001260          MOV   R0,SAVR0          ;SAVE R0
677 003502 000002          RTI          ;LEAVE.
678
679          ;RESTORE R0-R5
680
681 003504 0 3700 001260          .RES05: MOV   SAVR0,R0          ;RESTORE R0
682 003510 013701 001262          MOV   SAVR1,R1          ;RESTORE R1
683 003514 013702 001264          MOV   SAVR2,R2          ;RESTORE R2
684 003520 013703 001266          MOV   SAVR3,R3          ;RESTORE R3
685 003524 013704 001270          MOV   SAVR4,R4          ;RESTORE R4
686 003530 013705 001272          MOV   SAVR5,R5          ;RESTORE R5
    
```

```

687 003534 000002 RTI ; LEAVE
688
689 ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
690 -----
691
692 003536 104402 005104 .CONVR: TYPE ,MCRLF
693 003542 010046 .CNVRT: MOV R0,-(SP)
694 003544 010146 MOV R1,-(SP)
695 003546 010346 MOV R3,-(SP)
696 003550 010446 MOV R4,-(SP)
697 003552 010546 MOV R5,-(SP)
698 003554 017601 000012 MOV @12(SP),R1
699 003560 062766 000002 000012 ADD #2,12(SP)
700 003566 012137 003742 MOV (R1)+,WRDCNT
701 003572 112137 003744 1$: MOV (R1)+,CHRCNT
702 003576 112137 003745 MOV (R1)+,SPACNT
703 003602 013137 003746 MOV @ (R1)+,BINWRD
704 003606 013704 003746 2$: MCV BINWRD,R4
705 003612 113705 003744 MOV CHRCNT,R5
706 003616 012700 005562 MCV #TEMP,R0
707 003622 010403 3$: MOV R4,R3
708 003624 042703 177770 BIC #177770,R3
709 003630 062703 000060 ADD #060,R3
710 003634 110320 MOV R3,(R0)+
711 003636 000241 CLC
712 003640 006004 ROR R4
713 003642 000241 CLC
714 003644 006004 ROR R4
715 003646 000241 CLC
716 003650 006004 ROR R4
717 003652 053005 DEC R5
718 003654 001362 BNE 3$
719 003656 012703 005624 MOV #MDATA,R3
720 003662 114023 4$: MOV -(R0),(R3)+
721 003664 105337 003744 DECB CHRCNT
722 003670 001374 BNE 4$
723 003672 105737 003745 TSTB SPACNT
724 003676 001405 BEQ 6$
725 003700 112723 000040 5$: MOV #040,(R3)+
726 003704 105337 003745 DECB SPACNT
727 003710 001373 BNE 5$
728 003712 105013 6$: CLR B (R3)
729 003714 104402 005624 TYPE #MDATA
730 003720 005337 003742 DEC WRDCNT
731 003724 001322 BNE 1$
732 003726 012605 MOV (SP)+,R5
733 003730 012604 MOV (SP)+,R4
734 003732 012603 MOV (SP)+,R3
735 003734 012601 MOV (SP)+,R1
736 003736 012600 MOV (SP)+,R0
737 003740 000002 RTI
738 003742 000000 WRDCNT: 0
739 003744 000000 CHRCNT: 0
740 003746 003745 SPACNT=CHRCNT+1
741 003746 000000 BINWRD: 0
742
    
```

```

743
744 ;TRAP DISPATCH SERVICE
745 ;ARGUMENT OF TRAP IS EXTRACTED
746 ;AND USED AS OFFSET TO OBTAIN POINTER
747 ;TO SELECTED SUBROUTINE
748
749 003750 011646 .TRPSR: MOV (SP),-(SP) ;GET PC OF RETURN
750 003752 162716 000002 SUB #2,(SP) ;=PC OF TRAP
751 003756 017616 000000 MOV @ (SP),(SP) ;GET TRP
752 003762 006316 TRPK: ASL (SP) ;MULTIPLY TRAP ARG BY 2
753 003764 027116 177001 BIC #177001,(SP) ;CLEAR UNWANTED BITS
754 003770 062716 001314 ADD #,TRPTAB,(SP) ;POINTER TO SUBROUTINE ADDRESS
755 003774 017616 000000 MOV @ (SP),(SP) ;SUBROUTINE ADDRESS
756 004000 000136 JMP @ (SP)+ ;GO TO SUBROUTINE
757
758 ;ERROR HANDLER
759 -----
760
761 004002 .HLT:
762 004002 022737 177570 001202 CMP #177570,SWR ;IS THERE A REAL SWR?
763 004010 001411 BEQ 64$ ;BR IF YES
764 004012 017746 175170 MOV @TKDBR,-(SP) ;SAVE KEYBOARD CHAR
765 004016 042716 000200 BIC #BIT7,(SP) ;CLEAR PARITY BIT
766 004022 122726 000007 CMPSB #7,(SP)+ ;WAS IT CNTRL 'G' ?
767 004026 001002 BNE +6 ;BR IF NO.
768 004030 004737 004640 JSR PC,SERV.G ;SERVICE 'CNTRL 'G''.
769 004034 032777 010000 175140 64$: BEQ #SW12,@SWR ;BELL ON ERROR?
770 004042 001406 BEQ XBX ;BR IF NO BELL
771 004044 105777 175140 TSTB @TPCSR ;TTY READY.
772 004050 100003 BPL XBX ;DON'T WAIT IF TTY NOT READY.
773 004052 112777 000207 175132 MOV #207,@TPDBR ;PUSH A BELL AT THE TTY.
774 004060 032777 020000 175114 XBX: BIT #SW13,@SWR ;DELETE ERROR PRINT OUT?
775 004066 001105 BNE HALTS ;BR IF NO PRINT OUT WANTED.
776 004070 021637 001234 CMP (SP),LSTERR ;WAS THIS ERROR FOUND LAST TIME?
777 004074 001404 BEQ 1$ ;BR IF YES
778 004076 011637 001234 MOV (SP),LSTERR ;RECORD BEING HERE
779 004102 105037 001311 CLR B ERRFLG ;PREPARE HEADER
780 004106 104406 1$: SAVO5 ;SAVE ALL PROC REGISTERS
781 004110 011605 MOV (SP),R5 ;GET THE PC OF ERROR
782 004112 162705 000002 SUB #2,R5 ;GET ADDRESS OF TRAP CALL
783 004116 011504 MOV (R5),R4 ;GET HLT INSTRUCTION
784 004120 006304 ASL R4 ;MULT BY TWO
785 004122 061504 ADD (R5),R4 ;DOUBLE IT
786 004124 006304 ASL R4 ;MULT AGAIN
787 004126 042704 177001 BIC #177001,R4 ;CLEAR JUNK
788 004132 062704 034404 ADD #,ERRTAB,R4 ;GET POINTER
789 004136 012437 004252 MOV (R4)+,ERRMSG ;GET ERROR MESSAGE
790 004142 012437 004264 MOV (R4)+,DATAHD ;GET DATA HEADRER
791 004146 011437 004276 MOV (R4),DATABP ;GET DATA TABLE
792 004152 105737 001311 TSTB ERRFLG ;TYPE HEADREER
793 004156 001403 BEQ TYPMSG ;BR IF YES
794 004160 005737 004276 TST DATABP ;DOES DATA TABLE EXIST?
795 004164 001040 BNE TYPDAT ;BR IF YES.
796 004166 104402 005104 TYPMSG: TYPE ,MCRLF
797 004172 104402 005104 TYPE ,MCRLF
798 004176 005737 001220 TST LOCK
    
```

```

799 004202 001402          BEQ      1$
800 004204 104402 005400    TYPE     ,MASTEK
801 004210 104402 005366    1$:     TYPE     ,MTSTN
802 004214 104411 004374    CNVRT    ,XTSTN      ;SHOW IT
803 004220 104402 005454    TYPE     ,MERRPC    ;TYPE PC.
804 004224 104411 004366    CNVRT    ,ERTAB0    ;SHOW IT
805 004230 104402 005104    TYPE     ,MCRLF     ;GIVE A CR/LF
806 004234 112737 177777 001311  MOV     #-1,ERRFLG  ;NO MORE HEADER UNLESS NO DATA TABLE.
807 004242 005737 004252    TST     ERRMSG     ;IS THERE AN ERROR MESSAGE?
808 004246 001402          BEQ      WRKO.FM    ;BR IF NO.
809 004250 104402          TYPE     ;TYPE
810 004252 000000          ERRMSG: 0          ; ERROR MESSAGE
811 004254          WRKO.FM: ;
812 004254 005737 004264    TST     DATAHD   ; DATA HEADER?
813 004260 001402          BEQ     TYPDAT   ;BR IF NO
814 004262 104402          TYPE     ;TYPE
815 004264 000000          DATAHD: 0       ; DATA HEADER
816 004266 005737 004276    TYPDAT: TST     DATABP ; DATA TABLE?
817 004272 001402          BEQ     RESREG   ;BR IF NO.
818 004274 104410          CONVRT  ;SHOW
819 004276 000000          DATABP: 0      ; DATA TABLE
820 004300 104407          RESREG: RES05    ;RESTORE PROC REGISTERS
821 004302 005777 174674    HALTS:  TST     @SWR  ;HALT ON ERROR?
822 004306 100005          BPL     EXITER   ;BR IF NO HALT ON ERROR
823 004310 010046          PUSHRO ;SAVE RD
824 004312 016600 000002    MOV     2(SP),R0  ;SHOW ERROR PC IN DATA LIGHTS
825 004316 000000          HALT
826 004320 012600          POPRO  ;GET RD
827 004322 005237 001232    EXITER: INC     ERRCNT ;UPDATE ERROR COUNT
828 004326 032777 000400 174646  BIT    #SW0B,@SWR  ;GOTO TOP OF TEST?
829 004334 001007          BNE     1$       ;BR IF YES
830 004336 032777 002000 174636  BIT    #SW10,@SWR  ;GOTO NEXT TEST?
831 004344 001407          BEQ     2$       ;BR IF NO
832 004346 013737 001216 001214  MOV     NEXT,RETURN ;SET FOR NEXT TEST
833 004354 012706 001200          1$:  MOV     #STACK,SP ;RESET SP
834 004360 000177 174630          JMP     @RETURN   ;GOTO SPECIFIED TEST
835 004364 000002          2$:  RTI
836 004366 000001          ERTAB0: 1
837 004370 006          .BYTE  6,2
838 004372 001276          SAVPC
839 004374 000001          XTSTN: 1
840 004376 003          .BYTE  3,2
841 004400 001226          TSTNO  ;ENTER HERE ON POWER FAILURE
842
843
844
845
846 004402          .PFAIL:
847 004402 012737 004414 000024  MOV     #RESTART,24 ;SET UP FOR POWER UP TRAP
848 004410 000000          HALT      ;HALT ON POWER DOWN NORMAL
849 004412 000777          BR
850
851          ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
852
853 004414          RESTAR:
854 004414 012737 004402 000024  MOV     #.PFAIL,24  ;SET UP FOR POWER FAILURE

```

```

855 004422 012706 001200          MOV     #STACK,SP  ;RESET THE STACK POINTER
856 004426 005037 005562          CLR     TEMP       ;READY FOR TIMER
857 004432 005237 005562          INC     TEMP       ;PLUS ONE TO THE TIMER!
858 004436 001375          BNE     -4         ;BR IF MORE TO GO
859 004440 104402 005107          TYPE     ,MPFAIL   ;TYPE THE MESSAGE
860 004444 104411 004470          CNVRT    ,PFTAB    ;TELL WHAT TEST TO RETURN TO.
861 004450 105037 001311          CLRB   ERRFLG    ;START CLEAN
862 004454 005037 001234          CLR     LSTERR    ;*****
863 004460 104412          MSTCLR  ;START CLEAN UP OF DEVICE
864 004462 104413          RAMCLR  ;CLEAR IT ALL!
865 004464 000177 174524          JMP     @RETURN    ;START DOING THAT TEST AGAIN.
866 004470 000001          PFTAB: 1
867 004472 003          .BYTE  3,2
868 004474 001226          TSTNO
869 004476 010046          .DELAY: MOV     R0,-(SP)
870 004500 013700 004514          MOV     1$,R0
871 004504 005300          DEC     R0
872 004506 001376          BNE     -2
873 004510 012600          MOV     (SP)+,R0
874 004512 000002          RTI
875 004514 000036          1$:  30.
876
877 004516          .RAMCLR:
878 004516 012777 004000 174636  MOV     #MRESET,@DVSCR ;ISSUE A MASTER CLEAR
879 004524 010146          MOV     R1,-(SP)   ;SAVE R1 ON THE STACK
880 004526 010446          MOV     R4,-(SP)   ;SAVE R4 ON THE STACK
881 004530 013701 001372          MOV     DVSR5,R1   ;GET SECONDARY SEL. REG.
882 004534 013704 001376          MOV     DVSR4,R4   ;GET SECONDARY REGISTER ACCESS REG.
883 004540 005014          CLR     (R4)       ;ZERO THE SECONDARY REGISTER
884 004542 062711 170361          ADD     #'C<BIT11+BIT10+BIT9+BIT8+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT1+BIT0>+BIT0,(R1)
885 004546 01374          BNE     1$
886 004550 012604          MOV     (SP)+,R4  ;RESTORE R4
887 004552 012601          MOV     (SP)+,R1  ;RESTORE R1
888 004554 000002          RTI
889
890 004556          .MSTCLR:
891 004556 012777 004000 174576  MOV     #MRESET,@DVSCR ;ISSUE MASTER CLEAR.
892 004564 000002          RTI
893
894 004566          .ROMCLK:
895 004566 052777 000002 174566  BIS     #BIT1,@DVSCR
896 004574 000002          RTI
897
898 004576          .DATACLK:
899 004576 010046          MOV     R0,-(SP)
900 004600 005000          CLR     R0
901 004602 052777 000400 174560  BIS     #BIT8,@DVLCR
902 004610 017737 174554 004636  1$:  MOV     @DVLCR,3$
903 004616 106037 004637          RORB   3$+1
904 004622 103003          BCC     2$
905 004624 005200          INC     R0
906 004626 001370          BNE     1$
907 004630 104000          HLT     0
908 004632 012600          2$:  MOV     (SP)+,R0
909 004634 000002          RTI
910 004636 000001          3$:  .BLKW 1

```

```

911
912 004640 032777 004000 174338 SERV.G: BIT #4000,@TKCSR ;RX BUSY?
913 004646 001374 BNE SERV.G ;BR IF YES
914 004650 017777 174328 005072 MOV @SWR,90$ ;SAVE (SWR).
915 004656 013777 005072 174316 1$: MOV 90$,@SWR ;
916 004664 104402 005052 TYPE ,89$ ;
917 004670 104411 005064 CNVRT ,89$ ;
918 004674 104402 005074 TYPE ,91$ ;
919 004700 105777 174300 TSTB @TKCSR ;WAIT FOR DONE.
920 004704 100375 BPL -4 ;
921 004708 017748 174274 MOV @TKDBR,-(SP) ;
922 004712 042718 000200 BIC #BIT7,(SP) ;
923 004716 122728 000018 CMPB #15,(SP)+ ;
924 004722 001450 BEQ 5$ ;
925 004724 005077 174252 CLR @SWR ;
926 004730 105777 174254 2$: TSTB @TPCSR ;
927 004734 100375 BPL -4 ;
928 004736 018677 177776 174246 MOV -2(SP),@TPDBR ;
929 004744 000241 CLC ;
930 004746 006177 174230 ROL @SWR ;
931 004752 006177 174224 ROL @SWR ;
932 004756 006177 174220 ROL @SWR ;
933 004762 103735 BCS 1$ ;ERROR
934 004764 026627 177776 000060 CMP -2(SP),#60 ;
935 004772 002731 BLT 1$ ;
936 004774 026627 177776 000067 CMP -2(SP),#67 ;
937 005002 003325 BGT 1$ ;
938 005004 042766 177770 177776 BIC #C<7>,-2(SP) ;
939 005012 056677 177776 174182 BIS -2(SP),@SWR ;
940 005020 105777 174160 TSTB @TKCSR ;
941 005024 100375 BPL -4 ;
942 005026 017746 174154 MOV @TKDBR,-(SP) ;
943 005032 042718 000200 BIC #BIT7,(SP) ;
944 005036 122728 000018 CMPB #15,(SP)+ ;
945 005042 001332 BNE 2$ ;
946 005044 104402 005104 5$: TYPE ,MCRLF ;
947 005050 000207 RTS PC ;
948
949 005052 020377 051450 051127 89$: .ASCIZ <377>? (SWR)=/?
950 005060 036451 000057 .EVEN
951 89$: 1
952 005064 000001 .BYTE 6,0
953 005066 006 000 90$
954 005070 005072 90$
955 005072 000000 90$: .WORD 0
956 005074 036457 000057 91$: .ASCIZ ?/=/?
957 .EVEN
958 005100 020040 000077 MCM: .ASCIZ / ?/
959 005104 005015 000 MCRLF: .ASCIZ <15><12>
(2) 005107 377 053520 020122 MPFAIL: .ASCIZ <377>/PWR FAILED. RESTART AT TEST /
(2) 005145 377 047105 020104 MEPASS: .ASCIZ <377>/END PASS CZDVDCO /
(2) 005171 377 000122 MR: .ASCIZ <377>/R/
(2) 005174 050377 047522 051107 MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 005243 377 047111 052523 MERR3: .ASCIZ <377>/INSUFFICIENT DATA/
(2) 005267 377 042524 052123 MTSTPC: .ASCIZ <377>/TEST PC-/
(2) 005301 377 047514 045503 MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/

```

```

(2) 005330 051503 035122 000040 MCSRX: .ASCIZ /CSR: /
(2) 005336 042526 035103 000040 MVECX: .ASCIZ /VEC: /
(2) 005344 040520 051523 051505 MPASSX: .ASCIZ /PASSES: /
(2) 005355 105 051122 051117 MERRX: .ASCIZ /ERRORS: /
(2) 005366 042524 052123 047040 MTSTN: .ASCIZ /TEST NO: /
(2) 005400 000052 MASTEK: .ASCIZ /*/
(2) 005402 051777 052105 051440 MNEW: .ASCIZ <377>/SET SWITCH REG TO DV11'S DESIRED ACTIVE./
(2) 005454 041520 020072 000 MERRPC: .ASCIZ /PC: /
(2) 005461 377 040515 020120 XHEAD: .ASCIZ <377>/MAP OF DV11 STATUS/<377>
(2) .EVEN
(2) 005506 000002 XSTATQ: 2
959 005510 006 003 .BYTE 6,3
960 005512 001246 TEMP1
961 005514 006 002 .BYTE 6,2
962 005516 001250 TEMP2
963 .EVEN
964
965 ;BUFFERS FOR INPUT-OUTPUT
966
967 005520 000000 INBUF: 0
968 005562 .=-+40
969 005562 000000 TEMP: 0
970 005624 .=-+40
971 005624 000000 MDATA: 0
972 005666 .=-+40

```

```

973 ;
974 ;ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DV11'S
975 ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
976 ;AND RUNS THE SPECIFIED DV11'S. THIS ROUTINE *MUST*
977 ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
978 ;SETUP NECESSARY.
979 ;
980 ;
981
982 005666 105737 001300 CYCLE: TSTB DVACTV ;ARE ANY DV11'S TO BE TESTED?
983 005672 001004 BNE 1$ ;BR IF DK.
984 005674 104402 005174 TYPE ,MERR2 ;NO DV11'S SELECTED!!
985 005700 000000 HALT ;STOP THE SHOW.
986 005702 000776 BR -2 ;DISQUALIFY CONT. SW.
987 005704 133737 001304 001300 1$: BITB RUN,DVACTV ;IS THIS ONE "ACTIVE"
988 005712 001020 BNE 2$ ;BR IF GOOD ONE FOUND.
989 005714 000241 CLC ;CLEAR PROC. CARRY BIT.
990 005716 106137 001304 ROLB RUN ;UPDATE POINTER
991 005722 105537 001304 ADCB RUN ;CATCH CARRY FROM RUN
992 005726 062737 000024 001306 ADD #24,CREAM ;UPDATE ADDRESS POINTER.
993 005734 022737 001740 001306 CMP #DV.END,CREAM
994 005742 001360 BNE 1$ ;KEEP GOING; NOT ALL TESTED FOR.
995 005744 012737 001500 001306 MOV #DV.MAP,CREAM ;RESET ADDRESS POINTER.
996 005752 000754 BR 1$ ;KEEP LOOKING FOR ACTIVE DV11
997 005754 000241 2$: CLC ;CLEAR PROC. CARRY.
998 005756 106137 001304 ROLB RUN ;UPDATE POINTER.
999 005762 105537 001304 ADCB RUN ;CATCH CARRY.
1000 005766 013700 001306 MOV CREAM,RO ;GET ADDRESS POINTER.
1001 005772 062737 000024 001306 ADD #24,CREAM ;UPDATE.
1002 006000 022737 001740 001306 CMP #DV.END,CREAM
1003
1004 006006 001003 BNE 3$ ;ALL DONE?
1005 006010 012737 001500 001306 MOV #DV.MAP,CREAM ;BR IF NO.
1006 006016 012037 001362 3$: MOV (RO)+,DVSCRH ;RESTORE POINTER.
1007 006022 012037 001352 MOV (RO)+,DVSRV ;LOAD SYSTEM CTRL. REG
1008 006025 012037 001416 MOV (RO)+,L00.03 ;LOAD VECTOR
1009 006032 012037 001426 MOV (RO)+,SYNCA ;GET LINE PARAMETERS. 00-03
1010 006036 012037 001420 MOV (RO)+,L04.07 ;
1011 006042 012037 001430 MOV (RO)+,SYNCA ; 04-07
1012 006046 012037 001422 MOV (RO)+,L08.11 ;
1013 006052 012037 001432 MOV (RO)+,SYNCA ; 08-11
1014 006056 012037 001424 MOV (RO)+,L12.15 ;
1015 006062 012037 001434 MOV (RO)+,SYNCA ; 12-15
1016 006066 012700 000002 MOV #2,RO ;SAVE CORE THIS WAY!
1017 006072 013737 001362 001364 MOV DVSCRH,DVSCRH ;GET SYS CTRL. REG HIGH BYTE.
1018 006100 005237 001364 INC DVSCRH ;GOT IT.
1019 006104 013737 001364 001366 MOV DVSCRH,DVRIC ;GET NXT REC. CHAR REG.
1020 006112 005237 001366 INC DVRIC ;GOT IT
1021 006116 013737 001366 001370 MOV DVRIC,DVLCR ;GET LN. PAR.REG.
1022 006124 060037 001370 ADD RO,DVLCR ;GOT IT
1023 006130 013737 001370 001372 MOV DVLCR,DVSR ;GET SEC. REG. SEL. REG.
1024 006136 060037 001372 ADD RO,DVSR ;GOT IT
1025 006142 013737 001372 001374 MOV DVSR,DVSRSH ;GET HIGH BYTE.
1026 006150 005237 001374 INC DVSRSH ;GOT IT
1027 006154 013737 001374 001376 MOV DVSRSH,DVSR ;SEC. REG. ACCESS.
1028 006162 005237 001376 INC DVSR ;GOT IT

```

```

1029 006166 013737 001376 001400 MOV DVSR,DVSR ;SPEC. FUN. REG.
1030 006174 060037 001400 ADD RO,DVSR ;
1031 006200 013737 001400 001402 MOV DVSR,DVNSR ;NPR STAT. REG.
1032 006206 060037 001402 ADD RO,DVNSR ;
1033 006212 013737 001402 001404 MOV DVNSR,RESV16 ;RESERVED REG
1034 006220 060037 001404 ADD RO,RESV16 ;
1035
1036 006224 013737 001352 001354 MOV DVSR,DVRLVL ;PTY LVL
1037 006232 060037 001354 ADD RO,DVRLVL ;
1038 006236 013737 001354 001356 MOV DVRLVL,DVTRVEC ;TX VEC
1039 006244 060037 001356 ADD RO,DVTRVEC ;
1040 006250 013737 001356 001360 MOV DVTRVEC,DVTLVL ;TX LVL
1041 006256 060037 001360 ADD RO,DVTLVL ;
1042
1043 006262 012700 001416 MOV #L00.03,RO ;LOAD STAUS 00-03
1044 006266 012701 001406 MOV #MASK.A,R1 ;PREPARE MASK.
1045 006272 012702 001412 MOV #CLK.A,R2 ;PREPARE CLOCKS
1046 006276 004737 006516 JSR PC,FIX.00 ;GO AND CALCULATE CONFIGURATION.
1047
1048 006302 012700 001420 MOV #L04.07,RO ;LOAD STAUS 00-03
1049 006306 012701 001407 MOV #MASK.B,R1 ;PREPARE MASK.
1050 006312 012702 001413 MOV #CLK.B,R2 ;PREPARE CLOCKS
1051 006316 004737 006516 JSR PC,FIX.00 ;GO AND CALCULATE CONFIGURATION.
1052
1053 006322 012700 001422 MOV #L08.11,RO ;LOAD STAUS 00-03
1054 006326 012701 001410 MOV #MASK.C,R1 ;PREPARE MASK.
1055 006332 012702 001414 MOV #CLK.C,R2 ;PREPARE CLOCKS
1056 006336 004737 006516 JSR PC,FIX.00 ;GO AND CALCULATE CONFIGURATION.
1057
1058 006342 012700 001424 MOV #L12.15,RO ;LOAD STAUS 00-03
1059 006346 012701 001411 MOV #MASK.D,R1 ;PREPARE MASK.
1060 006352 012702 001415 MOV #CLK.D,R2 ;PREPARE CLOCKS
1061 006356 004737 006516 JSR PC,FIX.00 ;GO AND CALCULATE CONFIGURATION.
1062 006362 032777 000002 172612 BIT #SW01,@SWR
1063 006370 001445 BEQ 7$
1064
1065 006372 005737 000042 4$: TST @#42
1066 006376 001042 BNE 7$
1067 006400 104402 005104 TYPE ,MCLRF
1068 006404 104403 INSTR
1069 006406 005366 MTSTN
1070 006410 104405 PARAM
1071 006412 000001 1
1072 006414 001000 1000
1073 006416 001226 TSTNO
1074 006420 000 .BYTE 0
1075 006421 001 .BYTE 1
1076 006422 012700 007256 MOV #TST1,RO
1077 006426 022710 5$: CMP (PC)+,(RO)
1078 006430 012737 MOV (PC)+,@(PC)+
1079 006432 001015 BNE 6$
1080 006434 023760 001226 000002 CMP TSTNO,2(RO)
1081 006442 001011 BNE 6$
1082 006444 022760 001226 000004 CMP #TSTNO,4(RO)
1083 006452 001005 BNE 6$
1084 006454 010037 001214 MOV RO,RETURN

```

```

1085 006460 104402 005104      TYPE      ,MCRLF
1086 006464 000412      BR        8$
1087 006466 005720      6$:      TST      (R0)+
1088 006470 020027      CMP      R0,#TLAST+10
1089 006474 001354      BNE     5$
1090 006476 104402 005100      TYPE      ,MQM
1091 006502 000733      BR        4$
1092 006504 012737      7$:      MOV      #TST1,RETURN ;PREPARE RETURN ADDRESS
1093 006512 000177 172476      8$:      JMP      @RETURN ;GO START TESTING.
1094
1095 006516 011003      FIX.00:  MOV      (R0),R3 ;GET PARAMETERS.
1096 006520 042703 176377      BIC      #*C<1400>,R3 ;CLEAR JUNK.
1097 006524 005703      TST      R3 ;TEST FOR EIGHT BITS.
1098 006526 001004      BNE     1$ ;BR IF NOT 8 BITS.
1099 006530 108011      CLR     (R1) ;SET
1100 006532 112712 000010      MOV     #8,(R2) ;
1101 006536 000424      BR      4$
1102 006540 022703 000400      1$:      CMP      #400,R3 ;CHECK FOR SEVEN BITS.
1103 006544 001005      BNE     2$ ;BR IF NOT 7 BITS.
1104 006546 112711 000200      MOV     #200,(R1) ;
1105 006552 112712 000007      MOV     #7,(R2) ;
1106 006556 000414      BR      4$
1107 006560 022703 001000      2$:      CMP      #1000,R3 ;CHECK FOR SIX BITS.
1108 006564 001005      BNE     3$ ;BR IF NOT SIX BITS.
1109 006566 112711 000300      MOV     #300,(R1) ;
1110 006572 112712 000006      MOV     #6,(R2) ;
1111 006576 000404      BR      4$
1112 006600 112711 000340      3$:      MOV     #340,(R1) ;IF NONE OF THE ABOVE; MUST BE 5 BITS.
1113 006604 112712 000005      MOV     #5,(R2) ;
1114 006610 032710 040000      4$:      BIT      #PARBIT,(R0) ;PARITY ENABLED?
1115 006614 001401      BEQ     5$ ;IF #0; THEN NO PARITY.
1116 006616 105212      INCB   (R2) ;PLUS ONE TO THE CLOCK!
1117 006620 000207      5$:      RTS      PC ;
1118
1119 ;*ROUTINE USED TO "AUTO SIZE" THE DV11
1120 ;*CSR AND VECTOR.
1121 ;*NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
1122 ;* ADDRESS RANGE (175000:175400)
1123 ;* AND THE VECTOR MAY BE ANY WHERE IN THE
1124 ;* FLOATING VECTOR RANGE (300:770).
1125 ;*
1126
1127 006622      AUTO.SIZE:
1128 006622 000025      RESET
1129 006624 012702 001500      CSRMAP: MOV #DV.MAP,R2 ;INSURE A BUS INIT.
1130 006630 005022      1$:      CLR      (R2)+ ;LOAD MAP POINTER.
1131 006632 022702 001740      CMP      #DV.END,R2 ;ZERO ENTIRE MAP
1132 006636 001374      BNE     1$ ;ALL DONE?
1133 006640 105037 001301      CLR     DVNUM ;BR IF NO
1134 006644 012702 001500      MOV     #DV.MAP,R2 ;SET OCTAL NUMBER OF DV11'S TO 0
1135 006650 012701 175000      MOV     #175000,R1 ;SET FOR FIRST ADDRESS TO BE TESTED
1136 006654 012737 007074 000004      MOV     #6$,@#4 ;SET FOR NON-EXISTANT DEVICE TIME OUT
1137 006662 005711      TST     (R1) ;IF DV11 DVSCR S/B 0
1138 006664 001037      BNE     3$ ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO DV11
1139 006666 022761 177777 000012      CMP     #177777,12(R1) ;IF DV11 THEN DVSCR S/B ALL 1'S ON INIT!
1140 006674 001033      BNE     3$ ;BR IF NOT DV11
    
```

```

1141 006676 005761 000016      TST     16(R1) ;IF DV11 THEN RESV16 S/B ALL 0'S
1142 006702 001030      BNE     3$ ;BR IF NOT DV11
1143 ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DV11 CSR ADDRESS.
1144 006704 010122      MOV     R1,(R2)+ ;STORE CSR IN CORE TABLE.
1145 006706 005722      TST     (R2)+ ;POP OVER VECTOR STORE AREA
1146 006710 052722 000226      BIS     #226,(R2)+ ;SET LINE CARD 1 STAT AND SYNC
1147 006714 052722 000062      BIS     #62,(R2)+ ;
1148 006720 052722 000226      BIS     #226,(R2)+ ;SET LINE CARD 2 STAT AND SYNC
1149 006724 052722 000062      BIS     #62,(R2)+ ;
1150 006730 052722 000226      BIS     #226,(R2)+ ;SET LINE CARD 3 STAT AND SYNC
1151 006734 052722 000062      BIS     #62,(R2)+ ;
1152 006740 052722 000226      BIS     #226,(R2)+ ;SET LINE CARD 4 STAT AND SYNC
1153 006744 052722 000062      BIS     #62,(R2)+ ;
1154 006750 105237 001001      INCB   DVNUM ;UPDATE DEVICE COUNTER
1155 006754 122737 000010 001301      CMP     #10,DVNUM ;ARE MAX. NO. OF DEV FOUND?
1156 006762 001405      BEQ     100$ ;YES DON'T LOOK FOR ANY MORE.
1157 006764 062701 000010      3$:      ADD     #10,R1 ;UPDATE CSR POINTER ADDRESS
1158 006770 022701 175400      CMP     #175400,R1
1159 006774 001332      BNE     2$ ;BR IF MORE ADDRESS TO CHECK.
1160 006776 012722 177777      100$:  MOV     #177777,(R2)+ ;TERMINATER.
1161 007002 105037 001300      DVACTV CLR     DVACTV
1162 007006 105737 001301      TST     DVNUM ;WERE ANY DV11'S FOUND AT ALL?
1163 007012 001423      BEQ     5$ ;ERROR AUTO SIZER FOUND NO DV11'S IN THIS SYS.
1164 007014 113701 001301      MOV     DVNUM,R1
1165 007020 110137 001303      MOV     R1,SAVNUM ;SAVE NUMBER OF DEVICES
1166 007024 000241      4$:      CLC
1167 007026 106137      ROLB   DVACTV ;GENERATE ACTIVE REGISTER OF DEVICES.
1168 007032 105237 001300      INCB   DVACTV ;SET THE BIT
1169 007036 005301      DEC     R1
1170 007040 001371      BNE     4$ ;BR IF MORE TO GENERATE
1171 007042 012737 000008 000004      MOV     #6,@#4 ;RESTORE TRAP VECTOR
1172 007050 113737 001300 001302      DVACTV MOV     DVACTV,SAVACT ;SAVE ACTIVE REGISTER
1173 007056 000137 007102      JMP     VECMAP ;GO FIND THE VECTOR NOW.
1174 007062 104402 005174      5$:      TYPE   #MERR2 ;NOTIFY OPR THAT NO DV11'S FOUND.
1175 007066 005000      CLR     RO ;MAKE DATA LIGHTS ZERO
1176 007070 000000      HALT   ;STOP THE SHOW
1177 007072 000776      BR      -2 ;DISABLE CNTV. SW.
1178 007074 012718 006764      6$:      MOV     #35,(SP) ;ENTERED BY NON-EXISTANT TIME-OUT.
1179 007100 000002      RTI
1180
1181 007102 012737 000340 000022      VECMAP: MOV #340,@#22 ;SET IOT TRAP Prio TO 7
1182 007110 012737 007232 000020      MOV     #45,@#20 ;SET IOT TRAP VECTOR
1183 007116 012702 001500      MOV     #DV.MAP,R2 ;SET SOFTWARE POINTER
1184 007122 012700 000300      MOV     #300,RO ;FLOATING VECTORS START HERE.
1185 007126 012701 000302      MOV     #302,R1 ;PC OF IOT INSTR.
1186 007132 010120 000004      1$:      MOV     R1,(R0)+ ;START FILLING VECTOR AREA
1187 007134 012721 000004      MOV     #4,(R1)+ ;WITH .+2; IOT
1188 007140 022021      CMP     (R0)+,(R1)+ ;ADD 2 TO RO +R1
1189 007142 020127 001000      CMP     R1,#1000
1190 007146 101771      BLOS   1$ ;BR IF MORE TO FILL
1191 007150 113737 001300 001246      2$:      MOV     DVACTV,TEMP1 ;STORE TEMPORALLY
1192 007156 006037 001246      ROR     TEMP1 ;BRING OUT A BIT
1193 007162 103034      BCC     5$ ;BR IF ALL DONE
1194 007164 005037 177776      CLR     PS ;ZERO CPU Prio
1195 007170 012772 001300 000000      MOV     #BIT9+BIT7+BIT6,@(R2)
1196 007176 005000      CLR     RO ;ATTEMPT TO FORCE AN INTERRUPT
    
```

```

1197 007200 005200          INC R0          ;STALL
1198 007202 001376          BNE -2         ;
1199 007204 052762 000300 000002 BIS #300,2(R2) ; FOR TIME TO INTERRUPT
1200 007212 042772 176777 000000 3$: BIC #<C<BIT9>,@(R2) ;NO INTERRUPT ASSUME 300 AND FIX DV11 LATER
1201 007220 005072 000000          CLR @(R2)
1202 007224 062702 000024          ADD #24,R2
1203 007230 000752          BR 25
1204 007232 051662 000002 4$: BIS (SP),2(R2) ;POP SOFTWARE POINTER
1205 007236 042762 000007 000002 BIC #7,2(R2) ;KEEP GOING
1206 007244 022626          CMP (SP)+,(SP)+ ;GET VECTOR ADDRESS
1207 007246 027116 007212          MOV #3$,(SP) ;CLEAR JUNK
1208 007252 000002          RTI ;POP IOT JUNK OFF STACK
1209 007254 000207 5$: RTS PC ;SET FOR RETURN
1210

```

```

1211
1212 ;***** TEST 1 *****
1213 ;*TEST OF TRANSMITTER CONTROL BYTES.
1214 ;*TEST OF "NEXT MODE" FOR TRANSMITTER.
1215 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
1216 ;*****
1217
1218 ; TEST 1
1219 ;-----
1220 007256 012737 000001 001226 TST1: MOV #1,TSTNO
1221 007264 012737 007664 001216 MOV #TST2,NEXT
1222 007272 012700 000000          MOV #0.,R0 ;PLACE LINE NUMBER INTO R0
1223 007276 013737 001416 001236 MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
1224 007304 100402          BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
1225 007306 004737 007374          JSR PC,105$ ;GO DD THE TEST FOR LINE CARD 1
1226 007312 012700 000004 100$: MOV #4.,R0 ;PLACE LINE NUMBER INTO R0
1227 007316 013737 001420 001236 MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
1228 007324 100402          BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
1229 007326 004737 007374          JSR PC,105$ ;GO DD THE TEST FOR LINE CARD 2
1230 007332 012700 000010 101$: MOV #8.,R0 ;LOAD LINE NUMBER
1231 007336 013737 001422 001236 MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
1232 007344 100402          BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED
1233 007346 004737 007374          JSR PC,105$ ;DD THE TEST FOR LINE CARD 3
1234 007352 012700 000014 102$: MOV #12.,R0 ;LOAD LINE NO.
1235 007356 013737 001424 001236 MOV L12.15,STAT ;LOAD LINE CARD STATUS
1236 007364 100402          BMI 103$ ;BR IF LINE CARD NOT TO BE TESTED
1237 007366 004737 007374          JSR PC,105$ ;DD THE TESTS FOR LINE CARD 4
1238 007372 104400 103$: SCOPE ;SCOPE THIS TEST.
1239 007374 105$: ;TEST ENTRANCE.
1240 007374 012737 007422 001220 MOV #1$,LOCK ;SET IF SW09=1 (LOCK)
1241 007402 104413          RAMCLR ;CLEAR ALL SEC REGISTERS
1242 007404 005003          CLR R3 ;SET IMAGE EXPECTED MODE=0
1243 007406 005001          CLR R1 ;SET IMAGE "NEXT MODE"=0
1244 007410 112737 000025 022560 MOV #25,TXBAP ;SET TX DATA CHAR
1245 007416 012702 000004          MOV #4,R2 ;SET FOR 4 LINE GROUP
1246 007422 110137 023605 1$: MOV R1,TXTAB+25 ;LOAD CONTROL BYTE(MODE)
1247 007426 010077 171740          MOV R0,@DVSRS ;LOAD LINE NUMBER
1248 007432 004537 022120          PERFORM ,SETREG ;
1249 007436 000 001          .BYTE 000,001 ;TX PRINCIPLE BA, PRINCIPLE BC
1250 007440 022560          TXBAP ;
1251 007442 177777          -1 ;
1252 007444 004537 022120          PERFORM ,SETREG ;
1253 007450 013 010          .BYTE 013,010 ;LINE STATE. CNTRL TABLE
1254 007452 000004          BIT2 ;TXGO
1255 007454 023560          TXTAB ;
1256 007456 004537 022120          PERFORM ,SETREG ;
1257 007462 014 014          .BYTE 014,014 ;TX MODE REG
1258 007464 000000          0 ;MAKE
1259 007466 000000          0 ;IT=0
1260 007470 032737 004000 001236 BIT #ASYNC,STAT ;IS THIS ASYNC LINE CARD?
1261 007476 021407          BEQ 60$ ;BR IF NO.
1262 007500 004537 022164          PERFORM ,LOAD.MODE ;#
1263 007504 015000          <BIT12+BIT11>+BIT9 ;#8 BITS/PER/CHAR
1264 007506 004537 022164          PERFORM ,LOAD.MODE ;#
1265 007512 072000          <BIT14+BIT13+BIT12>+BIT10 ;#9600 BAUD.
1266

```



```

1267 007514 000403          BR      61$          ;LOAD
1268 007516 004537 022164 60$:  PERFORM ,LOAD.MODE ;MODE
1269 007522 014000          BIT12+BIT11        ;LOCK OUT INTERRUPTS
1270 007524 012737 000340 177776 61$:  MOV      #340,PS    ;SET TRANS VECTOR
1271 007532 012777 007574 171616      MOV      #3$,@DVIVEC ;LOAD PRIO.
1272 007540 012777 000340 171616      MOV      #340,@DVLVL ;SET STATUS IE AND UCPU GO.
1273 007546 052777 020001 171606      BIS      #BIT13+BIT0,@DVSCR
1274 007554 005005          CLR      R5         ;WAIT
1275 007556 104414          DELAY          ;STALL FOR TIME
1276 007560 005037 177776      CLR      PS         ;ALLOW ITERUPTS (NSR ENTRY)
1277 007564 005205          INC      R5         ;ENTRY
1278 007568 001373          BNE      2$         ;
1279 007570 104000          HLT           ;NO SILO ENTRY (DVSCR 15 NOT=1)
1280 007572 024646          CMP      -(SP),-(SP) ;FAKE INTERUPT, BECAUSE NO REAL ONE HAPPENED.
1281 007574 042777 020000 171560 3$:  BIC      #BIT13,@DVSCR ;CLR IE
1282 007602 005037 177776      CLR      PS         ;ZERO PSW
1283 007606 022626          CMP      (SP)+,(SP)+ ;FAKE AN RTI
1284 007610 112777 000014 171556      MOV      #14,@DVSRSR ;SEL TX MODE REGISTER
1285 007616 017704 171554      MOV      @DVSRA,R4   ;READ MODE REG.
1286 007622 010305          MOV      R3,R5       ;SET EXPECTED
1287 007624 020504          CMP      R5,R4       ;WAS "NEXT MODE" LOADED CORRECTLY?
1288 007626 001401          BEQ      4$         ;BR IF YES
1289 007630 104003          HLT           ;TX MODE REGISTER WRONG
1290 007632 104412          4$:  MSTCLR          ;INIT DV11
1291 007634 104401          SCOP1        ;LOCK ON MODE, LOCK ON LINE?
1292 007636 005203          INC      R3         ;UPDATE EXPECTED MODE
1293 007640 062701 000040          ADD      #BITS,R1   ;UPDATE CNTRL BYTE IMAGE
1294 007644 105701          TSTB      R1        ;ALL DONE??
1295 007646 001665          BEQ      1$         ;BR IF NO
1296 007650 005001          CLR      R1         ;ZERO EXPECTE MODE
1297 007652 015003          CLR      R3         ;ZERO CNTRL BYTE MODE
1298 007654 005200          INC      R0         ;UPDATE LINE NO POINTER
1299 007656 005302          DEC      R2         ;4 LINES DONE
1300 007860 001260          BNE      1$         ;BR IF YES
1301 007862 000207          RTS          ;EXIT FOR NEXT GROUP OF LINES
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322

```

```

;***** TEST 2 *****
;TEST OF TRANSMITTER IDLE FUNCTIONS.
;TEST THAT THE TRANSMITTER WILL IDLE
;SYNC (IDLE) CHARS WHEN BIT 0 OF
;IDLE/PROTOCOL REGISTER IS CLEARED.
;THIS TEST IS DONE FOR SYNC LINE CARDS ONLY.
;*****

```

TEST 2

```

1313
1314 007664 012737 000002 001226 TST2:  MOV      #2,TSTNO
1315 007672 012737 010430 001216      MOV      #TST3,NEXT
1316 007700 012700 000000          MOV      #0,,RO    ;PLACE LINE NUMBER INTO RO
1317 007704 113737 001406 001244      MOV      MASK,A,MASKX ;PLACE "MASK" FOR CHARS INTO MASKX
1318 007712 013737 001416 001236      MOV      L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
1319 007720 100402          BMI      100$      ;BR IF LINE CARD NOT TO BE TESTED
1320 007722 004737 010032          JSR      PC,105$   ;GO DO THE TEST FOR LINE CARD 1
1321 007726 012700 000004          MOV      #4,,RO    ;PLACE LINE NUMBER INTO RO
1322 007732 113737 001407 001244      MOV      MASK,B,MASKX ;GET MASK

```

```

1323 007740 013737 001420 001236      MOV      L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
1324 007746 100402          BMI      101$      ;BR IF LINE CARD NOT TO BE TESTED
1325 007750 004737 010032          JSR      PC,105$   ;GO DO THE TEST FOR LINE CARD 2
1326 007754 012763 000010          MOV      #8,,RO    ;LOAD LINE NUMBER
1327 007760 113737 001410 001244      MOV      MASK,C,MASKX ;GET MASK
1328 007766 013737 001422 001236      MOV      L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
1329 007774 100402          BMI      102$      ;BR IF LINE CARD NOT TO BE TESTED
1330 007776 004737 010032          JSR      PC,105$   ;DO THE TEST FOR LINE CARD 3
1331 010002 012700 000014          MOV      #12,,RO   ;LOAD LINE NO.
1332 010006 113737 001411 001244      MOV      MASK,D,MASKX ;GET MASKX
1333 010014 013737 001424 001236      MOV      L12.15,STAT ;LOAD LINE CARD STATUS
1334 010022 100402          BMI      103$      ;BR IF LINE CARD NOT TO BE TESTED
1335 010024 004737 010032          JSR      PC,105$   ;DO THE TESTS FOR LINE CARD 4
1336 010030 104400          SCOPE          ;SCOPE THIS TEST.
1337 010032          TEST ENTRANCE.
1338 010032 032737 004000 001236      BIT      #ASYNC,STAT ;IS THIS AN ASYNC LINE CARD?
1339 010040 001401          BEQ      +4        ;BR IF NOT ASYNC.
1340 010042 000207          RTS          ;EXIT TEST. (ASYNC LINE CARD NOT TESTED)
1341 010044 012737 010116 001220      MOV      #3$,LOCK  ;SET FOR RETURN IF SW09=1
1342 010052 104413          RAMCLR        ;CLEAR ALL SEC REGISTERS
1343 010054 012705 023560          MOV      #TXTAB,R5 ;CLEAR
1344 010060 012704 030160          MOV      #RXTAB,R4 ;RECEIVER
1345 010064 005001          CLR      R1        ;AND
1346 010066 005025          CLR      (R5)+     ;TRANSMITTER
1347 010070 005024          CLR      (R4)+     ;CONTROL
1348 010072 105201          INCB      R1       ;TABLES
1349 010074 100374          BPL      1$         ;
1350 010076 012737 000001 022560      MOV      #1,TXBAP  ;LOAD TX
1351 010104 112737 000015 022561      MOV      #15,TXBAP+1 ;DTA
1352 010112 012702 000004          MOV      #4,R2     ;SET FOR 4 LINE GROUP
1353 010116 010077 171250          MOV      R0,@DVSRS ;LOAD LINE NUMBER
1354 010122 005037 027560          CLR      RXBA      ;CLEAR
1355 010126 005037 027562          CLR      RXBA+2    ;RECEIVER
1356 010132 005037 027564          CLR      RXBA+4    ;BUFFER
1357 010136 032737 004000 001236      BIT      #ASYNC,STAT ;IS THIS AN ASYNC LINE CARD?
1358 010144 001406          BEQ      80$       ;BR IF NOT ASYNC.
1359 010146 004537 022120          PERFORM ,SETREG   ;ADJUST FOR ASYNC LINE CARD
1360 010152 000000 001         .BYTE 000,001    ;REGISTERS
1361 010154 022560          TXBAP          ;LOAD FOR ASYNC
1362 010156 177776          -2           ;LOAD FOR ASYNC
1363 010160 004045          BR      81$       ;CONTINUE TEST
1364 010162 004537 022120      80$:  PERFORM ,SETREG ;TX PRINCIPLE BA, PRINCIPLE BC
1365 010166 000000 001         .BYTE 000,001    ;
1366 010170 022556          SYNC          ;
1367 010172 177774          -4           ;
1368 010174 004537 022120      81$:  PERFORM ,SETREG ;RX BA, RX BC
1369 010200 004000 005         .BYTE 004,005    ;
1370 010202 027560          RXDA          ;
1371 010204 177772          -6           ;
1372 010206 004537 022120          PERFORM ,SETREG ;TX TABLE, RXTABLE
1373 010212 010000 011         .BYTE 010,011    ;
1374 010214 023560          TXTAB        ;
1375 010216 030160          RXTAB        ;
1376 010220 004537 022120          PERFORM ,SETREG ;LINE STATE, LINE PROTOCOL
1377 010224 010000 012         .BYTE 013,012    ;
1378 010226 000004          BIT2         ;TX GOOD

```

```

1379 010230 000000          0
1380 010232 032737 004000 GJ1236 BIT #ASYNC,STAT ;DEFAULT-IDLE SYNC
1381 010240 001412          BEQ 60$ ;#IS THIS ASYNC LINE CARD?
1382 010242 004537 022164 PERFORM ,LOAD.MODE ;#BR IF NO.
1383 010246 020000          BIT13 ;LOAD PARAMETERS.
1384 010250 004537 022164 PERFORM ,LOAD.MODE ;#RECEIVER ENABLE
1385 010254 015000          <BIT12+BIT11>+BIT9 ;#
1386 010256 004537 022164 PERFORM ,LOAD.MODE ;#8 BITS/PER/CHAR
1387 010262 072000          <BIT14+BIT13+BIT12>+BIT10 ;#
1388          ;#9600 BAUD.
1389 010264 000405          BR 4$
1390 010266 004537 022164 60$: PERFORM ,LOAD.MODE ;LOAD
1391 010272 034000          BIT13+BIT12+BIT11 ;MODE AND RX ENABLE
1392 010274 004537 021700 PERFORM ,SETSYNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
1393 010300 005277 171056 4$: INC @DVSCR ;SET MICRO CPU GO
1394 010304 105777 171052 TSTB @DVSCR ;WAIT FOR
1395 010310 100375          BPL -4 ;DVSCRO7=1
1396 010312 005004          CLR R4 ;
1397 010314 012705 000001 MOV #1,R5 ;SET EXPECTED
1398 010320 113704 027560 MOVB RXBA,R4 ;READ 1ST CHAR
1399 010324 020504          CMP R5,R4 ;OK?
1400 010326 001401          BEQ 5$ ;
1401 010330 104001          HLT 1 ;1ST CHAR S/B=11
1402 010332 112705 000015 5$: MOVB #15,R5 ;SET EXPECTED
1403 010336 113704 027561 MOVB RXBA+1,R4 ;GET 2ND CHAR
1404 010342 020504          CMP R5,R4 ;
1405 010344 001401          BEQ 6$ ;
1406 010346 104001          HLT 1 ;2ND CHAR S/B=15
1407 010350          ;
1408 010350 113705 001236 6$: MOVB STAT,R5 ;SET EXPECTED=SYNC CHAR
1409 010354 042705 177400 BIC #<C377>,R5 ;CLEAR HIGH BYTE
1410 010360 143705 001244 BITCB MASKX,R5 ;CLEAR BITS/PER/CHAR MASK.
1411 010364 012703 000004 MOV #4,R3 ;SET TO LOOK AT 4 CHARS
1412 010370 012701 027562 MOV #RXBA+2,R1 ;GET RX DATA POINTER
1413 010374 112104          MOVB (R1)+,R4 ;GET FOUNDED DATA
1414 010376 042704 177400 7$: BIC #<C377>,R4 ;CLEAR HIGH BYTE
1415 010402 020504          CMP R5,R4 ;
1416 010404 001401          BEQ 8$ ;
1417 010406 104001          HLT 1 ;TRANSMITTER IDLED WRONG
1418 010410 005303 8$: DEC R3 ;4 CHARS CHECKED?
1419 010412 001370          BNE 7$ ;BR IF NO
1420 010414 104412          MSTCLR ;INIT DV11
1421 010416 104401          SCOPI ;LOCK ON LINE?
1422 010420 005270          INC R0 ;UPDATE LINE POINTER
1423 010422 005302          DEC R2 ;4 LINE GROUP DONE?
1424 010424 001234          BNE 3$ ;BR IF NO
1425 010426 000207          RTS PC ;EXIT FOR NEXT GROUP
1426
1427
1428          ;***** TEST 3 *****
1429          ;*TEST OF TRANSMITTER IDLE FUNCTIONS.
1430          ;*TEST THAT THE TRANSMITTER WILL IDLE
1431          ;*MARK STATE (377) WHEN BIT0 IS
1432          ;*SET IN THE DLE/PROTOCOL REGISTER.
1433          ;*THIS TEST IS DONE FOR SYNC LINE CARDS ONLY.
1434          ;*****

```

```

1435          ; TEST 3
1436          ;-----
1437
1438 010430 012737 000003 001226 TST3: MOV #3,TSTNO
1439 010436 012737 011174 001216 MOV #TST4,NEXT
1440 010444 012700 000000 MOV #0,R0 ;PLACE LINE NUMBER INTO R0
1441 010450 113737 001406 001244 MOVB MASK.A,MASKX ;PLACE "MASK" FOR CHARS INTO MASKX
1442 010456 013737 001416 001236 MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
1443 010464 100402          BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
1444 010466 004737 010576 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 1
1445 010472 012700 000004 100$: MOV #4,R0 ;PLACE LINE NUMBER INTO R0
1446 010476 113737 001407 001244 MOVB MASK.B,MASKX ;GET MASK
1447 010504 013737 001420 001236 MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
1448 010512 100402          BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
1449 010514 004737 010576 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 2
1450 010520 012700 000010 101$: MOV #8,R0 ;LOAD LINE NUMBER
1451 010524 113737 001410 001244 MOVB MASK.C,MASKX ;GET MASK
1452 010532 013737 001422 001236 MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
1453 010540 100402          BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED
1454 010542 004737 010576 JSR PC,105$ ;DO THE TEST FOR LINE CARD 3
1455 010546 012700 000014 102$: MOV #12,R0 ;LOAD LINE NO.
1456 010552 113737 001411 001244 MOVB MASK.D,MASKX ;GET MASKK
1457 010560 013737 001424 001236 MOV L12.15,STAT ;LOAD LINE CARD STATUS
1458 010566 100402          BMI 103$ ;BR IF LINE CARD NOT TO BE TESTED
1459 010570 004737 010576 JSR PC,105$ ;DO THE TESTS FOR LINE CARD 4
1460 010574 104400          103$: SCOPE ;SCOPE THIS TEST.
1461 010576          105$: ;TEST ENTRANCE.
1462 010576 032737 004000 001236 BIT #ASYNC,STAT ;#IS THIS AN ASYNC LINE CARD?
1463 010604 001401          BEQ .+4 ;#BR IF NOT ASYNC
1464 010606 000207          RTS PC ;#EXIT TEST. (ASYNC LINE CARD NOT TESTED)
1465 010610 0 2737 010662 001220 MOV #3$,LOCK ;SET FOR RETURN IF SW09=1
1466 010616 104413          RAMCLR ;CLEAR ALL SEC REGISTERS
1467 010620 012705 023560 MOV #TXTAB,R5 ;CLEAR
1468 010624 012704 030160 MOV #RXTAB,R4 ;RECEIVER
1469 010630 005001          CLR R1 ;AND
1470 010632 005025          CLR (R5)+ ;TRANSMITTER
1471 010634 005024          CLR (R4)+ ;CONTROL
1472 010636 105201          INCB R1 ;TABLES
1473 010640 100374          BPL 1$ ;
1474 010642 012737 000001 022560 MOV #1,TXBAP ;LOAD TX
1475 010650 112737 000015 022561 MOVB #15,TXBAP+1 ;DTA
1476 010656 012702 000004          MOV #4,R2 ;SET FOR 4 LINE GROUP
1477 010662 010077 170504 3$: MOV R0,@DVSR5 ;LOAD LINE NUMBER
1478 010666 005037 027560 CLR RXBA ;CLEAR
1479 010672 005037 027562 CLR RXBA+2 ;RECEIVER
1480 010676 005037 027564 CLR RXBA+4 ;BUFFER
1481 010702 032737 004000 001236 BIT #ASYNC,STAT ;#IS THIS AN ASYNC LINE CARD?
1482 010710 001406          BEQ 80$ ;#BR IF NOT ASYNC.
1483 010712 004537 022120 PERFORM ,SETREG ;#ADJUST FOR ASYNC LINE CARD
1484 010716 000 001          .BYTE 000,001 ;#REGISTERS
1485 010720 022560          TXBAP ;LOAD FOR ASYNC
1486 010722 177776          -2 ;LOAD FOR ASYNC
1487 010724 000405          BR 81$ ;#CONTINUE TEST
1488 010726 004537 022120 80$: PERFORM ,SETREG ;
1489 010732 000 001          .BYTE 000,001 ;TX PRINCIPLE BA, PRINCIPLE BC
1490 010734 022556          SYNC ;

```

```

1491 010736 177774          -4
1492 010740 004537 022120 81$: PERFORM ,SETREG          ;
1493 010744      004      005      .BYTE 004,005      ;RX BA, RX BC
1494 010746 0275CJ          RXBA          ;
1495 010750 177772          -6
1496 010752 004537 022120  PERFORM ,SETREG          ;
1497 010756      010      011      .BYTE 010,011      ;TX TABLE, RXTABLE
1498 010760 023560          T) :AB          ;
1499 010762 030160          RXTAB          ;
1500 010764 004537 022120  PERFORM ,SETREG          ;
1501 010770      013      012      .BYTE 013,012      ;LINE STATE, LINE PROCOOL
1502 010772 000004          BIT2          ;TX GOOD
1503 010774 000001          BIT0          ;IDLE MARK ON BYTE CNT=0
1504 010776 032737 004000 001238 BIT      #ASYNC,STAT      ;IS THIS ASYNC LINE CARD?
1505 011004 001412          BEQ      60$          ;#BR IF NO.
1506 011006 004537 022164  PERFORM ,LOAD.MODE      ;#LOAD PARAMETERS.
1507 011012 020000          BIT13         ;#RECEIVER ENABLE
1508 011014 004537 022164  PERFORM ,LOAD.MODE      ;#
1509 011020 015000          <BIT12+BIT11>+BIT9      ;#B BITS/PER/CHAR
1510 011022 004537 022164  PERFORM ,LOAD.MODE      ;#
1511 011026 072000          <BIT14+BIT13+BIT12>+BIT10 ;#9600 BAUD.
1512
1513 011030 000405          BR      4$
1514 011032 004537 022164 60$: PERFORM ,LOAD.MODE      ;LOAD
1515 011036 034000          BIT13+BIT12+BIT11      ;MODE AND RX ENABLE
1516 011040 004537 021706  PERFORM ,SETSYNC      ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
1517 011044 005277 170312 4$: INC      @DVSCR      ;SET MICRO CPU GO
1518 011050 105777 170308  TSTB      @DVSCR      ;WAIT FOR
1519 011054 100375          BPL      -4          ;DVSCRO7=1
1520 011056 005004          CLR      R4          ;
1521 011060 012705 000001  MOV      #1,R5          ;SET EXPECTED
1522 011064 113704 027560  MOVB     RXBA,R4        ;READ 1ST CHAR
1523 011070 020504          CMP      R5,R4        ;OK?
1524 011072 001401          BEQ      5$          ;
1525 011074 104001          HLT      1          ;1ST CHAR S/B=11
1526 011076 112705 000015 5$: MOVB     #15,R5       ;SET EXPECTED
1527 011102 113704 027561  MOVB     RXBA+1,R4     ;GET 2ND CHAR
1528 011106 020504          CMP      R5,R4        ;
1529 011110 001401          BEQ      6$          ;
1530 011112 104001          HLT      1          ;2ND CHAR S/B=15
1531 011114
1532 011114 012705 000377  MOV      #377,R5       ;SET EXPECTED=MARK CHAR
1533 011120 042705 177400  BIC      #"C<377>,R5   ;CLEAR HIGH BYTE
1534 011124 143705 001244  BICB     MASKX,R5      ;CLEAR BITS/PER/CHAR MASK.
1535 011130 012703 000004  MOV      #4,R3          ;SET TO LOOK AT 4 CHARS
1536 011134 012701 027562  MOV      #RXBA+2,R1     ;GET RX DATA POINTER
1537 011140 112104          MOVB     (R1)+,R4      ;GET FOUND DATA
1538 011142 042704 177400  BIC      #"C<377>,R4   ;CLEAN HIGH BYTE
1539 011146 020504          CMP      R5,R4        ;
1540 011150 001401          BEQ      8$          ;
1541 011152 104001          HLT      1          ;TRANSMITTER IDLED WRONG
1542 011154 005303 8$: DEC      R3          ;4 CHARS CHECKED?
1543 011156 001370          BNE     7$          ;BR IF NO
1544 011160 104412          MSTCLR          ;INIT DV11
1545 011162 104401          SCOPE          ;LOCK ON LINE?
1546 011164 005200          INC      R0          ;UPDATE LINE POINTER

```

```

1547 011166 005302          DEC      R2          ;4 LINE GROUP DONE?
1548 011170 001234          BNE     3$          ;BR IF NO
1549 011172 000207          RTS     PC          ;EXIT FOR NEXT GROUP
1550
1551
1552
1553 ;***** TEST 4 *****
1554 ;*TEST OF RECEIVER CONTROL BYTE OPERATIONS.
1555 ;*TEST OF THE "STORE/DISCARD" FUNCTIONS.
1556 ;*TEST THAT CHARS:
1557 ;* 25 STORED
1558 ;* 23 DISCARDED
1559 ;* 31 STORED
1560 ;* 32 DISCARDED
1561 ;*SINCE TWO CHRS SHOULD BE THROWN AWAY;
1562 ;*THE TX LINE IS SET TO GO BACK TO A MARK STATE;
1563 ;*THEREFORE THE RX BUFFER S/B:
1564 ;*RXBA 31,25
1565 ;* 377,377
1566 ;*(AT 8 BITS PER CHAR)
1567 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
1568 ;*****
1569
1570 ; TEST 4
1571 011174 012737 000004 001226 TST4: MOV      #4,TSTNO
1572 011202 012737 012004 001216 MOV      #TSTB,NEXT
1573 011210 012700 000000 MOV      #0,,RO
1574 011214 113737 001406 001244 MOVB     MASK.A,MASKX ;PLACE LINE NUMBER INTO RO
1575 011222 013737 001416 001236 MOV      L00.03,STAT ;PLACE "MASK"FOR CHARS INTO MASKX
1576 011230 100402 BMI      100$ ;LOAD LINE CARD STATUS INTO STAT
1577 011232 004737 011342 JSR      PC,105$ ;BR IF LINE CARD NOT TO BE TESTED
1578 011236 012700 000004 MOV      #4,,RO ;GO DO THE TEST FOR LINE CARD 1
1579 011242 113737 001407 001244 MOVB     MASK.B,MASKX ;PLACE LINE NUMBER INTO RO
1580 011250 013737 001420 001236 MOV      L04.07,STAT ;GET MASK
1581 011256 100402 BMI      101$ ;LOAD LINE CARD STATUS INTO STAT
1582 011260 004737 011342 JSR      PC,105$ ;BR IF LINE CARD NOT TO BE TESTED
1583 011264 012700 000010 MOV      #8,,RO ;GO DO THE TEST FOR LINE CARD 2
1584 011270 113737 001410 001244 MOVB     MASK.C,MASKX ;LOAD LINE NUMBER
1585 011276 013737 001422 001236 MOV      L08.11,STAT ;GET MASK
1586 011304 100402 BMI      102$ ;LOAD LINE CARD STATUS INTO STAT
1587 011306 004737 011342 JSR      PC,105$ ;BR IF LINE CARD NOT TO BE TESTED
1588 011312 012700 000014 MOV      #12,,RO ;DO THE TEST FOR LINE CARD 3
1589 011316 113737 001411 001244 MOVB     MASK.D,MASKX ;LOAD LINE NO.
1590 011324 013737 001424 001236 MOV      L12.15,STAT ;GET MASKK
1591 011332 100402 BMI      103$ ;LOAD LINE CARD STATUS
1592 011334 004737 011342 JSR      PC,105$ ;BR IF LINE CARD NOT TO BE TESTED
1593 011340 104400 103$: SCOPE          ;DO THE TESTS FOR LINE CARD 4
1594 011342 1055:          ;SCOPE THIS TEST.
1595 011342 012737 011442 001220 MOV      #1$,LOCK ;TEST ENTRANCE.
1596 011350 104413 RAMCLR          ;SET RETURN IF SW09=1
1597 011352 105037 023605 CLRB     TXTAB+25 ;CLEAR ALL DV11 SEC REGISTERS
1598 011356 105037 023603 CLRB     TXTAB+23 ;ZERO
1599 011362 105037 023611 CLRB     TXTAB+31 ;      USED
1600 011366 105037 023612 CLRB     TXTAB+32 ;      CONTROL
1601 011372 105037 024157 CLRB     TXTAB+377 ;      BYTES
1602 011376 012705 022560 MOV      #TXBAP,R5 ;FOR TRANSMITTER
;LOAD

```

```
1603 011402 012725          MOV      (PC)+,(R5)+      ;TRANSMITTER
1604 011404          .BYTE 25,23              ;DATA
1605 011406 012715          MOV      (PC)+,(R5)      ;CHARS
1606 011410          .BYTE 31,32              ;
1607 011412 112737 000020 030203 MOVB     #BIT4,RXTAB+23   ;DISCARD
1608 011420 112737 000020 030212 MOVB     #BIT4,RXTAB+32   ;DISCARD
1609 011426 105037 030205          CLR      RXTAB+25        ;DEFAULT-STORE
1610 011432 105037 030211          CLR      RXTAB+31        ;DEFAULT-STORE
1611 011436 012702 000004          MOV      #4,R2           ;SET FOR 4 LINE GROUP
1612 011442 010077 167724          MOV      R0,@DVSR5      ;LOAD LINE NO.
1613 011446 005037 027560          CLR      RXBA           ;MAKE SURE
1614 011452 005037 027562          CLP      RXBA+2         ;RX BUFFER=0
1615 011456 032737 004000 001236 BIT       #ASYNC,STAT     ;IS THIS AN ASYNC LINE CARD?
1616 011464 001406          BEQ      B0$            ;#BR IF NOT ASYNC.
1617 011466 004537 022120          PERFORM ,SETREG        ;ADJUST FOR ASYNC LINE CARD
1618 011472          .BYTE 000,001        ;#REGISTERS
1619 011474 022560          TXBAP          ;LOAD FOR ASYNC
1620 011476 177774          -4                    ;LOAD FOR ASYNC
1621 011500 000405          BR       B1$            ;CONTINUE TEST
1622 011502 004537 022120 80$: PERFORM ,SETREG        ;
1623 011506          .BYTE 000,001        ;TX PRINCIPLE BA, PRINCIPLE BC
1624 011510 022556          SYNC          ;SYNC CHAR
1625 011512 177772          -6                    ;2 SYNC, 4 DATA=6
1626 011514
1627 011514 032737 004000 001236 BIT       #ASYNC,STAT     ;IS THIS AN ASYNC LINE CARD?
1628 011522 001406          BEQ      B2$            ;#BR IF NOT ASYNC.
1629 011524 004537 022120          PERFORM ,SETREG        ;ADJUST FOR ASYNC LINE CARD
1630 011530          .BYTE 004,005        ;#REGISTERS
1631 011532 027560          RXBA          ;LOAD FOR ASYNC
1632 011534 177776          -2                    ;LOAD FOR ASYNC
1633 011536 0 0405          BR       B3$            ;CONTINUE TEST
1634 011540 004537 022120 82$: PERFORM ,SETREG        ;
1635 011544          .BYTE 004,005        ;RXBA, RXBC
1636 011546 027560          RXBA          ;
1637 011550 177774          -4                    ;
1638 011552 004537 022120 83$: PERFORM ,SETREG        ;
1639 011556          .BYTE 010,011        ;TX TABLE, RX TABLE
1640 011560 023560          TXTAB          ;
1641 011562 030160          RXTAB          ;
1642 011564 004537 022120          PERFORM ,SETREG        ;
1643 011570          .BYTE 013,012        ;LINE STATE, LINE PROTOCOL
1644 011572 000004          BIT2          ;TX GO
1645 011574 000001          BIT0          ;IDLE MARK ON BYTE COUNTS=0
1646 011576 032737 004000 001236 BIT       #ASYNC,STAT     ;IS THIS ASYNC LINE CARD?
1647 011604 001412          BEQ      B0$            ;#BR IF NO.
1648 011606 004537 022164          PERFORM ,LOAD.MODE     ;LOAD PARAMETERS.
1649 011612 020000          BIT13          ;#RECEIVER ENABLE
1650 011614 004537 022164          PERFORM ,LOAD.MODE     ;#
1651 011620 015000          <BIT12+BIT11>+BIT9     ;#8 BITS/PER/CHAR
1652 011622 004537 022164          PERFORM ,LOAD.MODE     ;#
1653 011626 002000          <BIT14+BIT13+BIT12>+BIT10 ;#9600 BAUD.
1654
1655 011630 000405          BR       B2$            ;
1656 011632 004537 022164 60$: PERFORM ,LOAD.MODE     ;LOAD
1657 011636 034000          BIT13+BIT12+BIT11     ;MODE+RX ENABLE
1658 011640 004537 021706          PERFORM ,SETSYNC      ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
```

```
1659 011644 005277 167512 2$: INC      @DVSCR        ;SET MICRO CPU GO
1660 011650 005005          CLR      R5             ;DELAY
1661 011652 105777 167504 3$: TSTB     @DVSCR        ;FOR
1662 011656 1004C 1          BMI      4$            ;RX INTERRUPT (BIT 7)
1663 011660 104414          DELAY          ;WASTE TIME
1664 011662 005205          INC      R5             ;KEEP COUNTING.
1665 011664 001372          BNE      3$            ;BR
1666 011666 104000          HLT           ;BIT 7 OF DVSCR NOT SET!
1667 011670
1668 011670 012705 000025          MOV      #25,R5         ;SET EXPECTED
1669 011674 113704 027560          MOVB     RXBA,R4        ;GET FOUND
1670 011700 020504          CMP      R5,R4         ;OK?
1671 011702 001401          BEQ      5$            ;
1672 011704 104002          HLT       2            ;'25' NOT FIRST IN RX BUFFER
1673 011706 012705 000031          MOV      #31,R5         ;NEXT CHAR S/B '31'
1674 011712 113704 027561          MOVB     RXBA+1,R4      ;GET NEXT CHAR
1675 011716 120504          CMPB     R5,R4         ;OK
1676 011720 001401          BEQ      6$            ;
1677 011722 104002          HLT       2            ;'31' NOT SECOND IN RX BUFFER
1678 011724 032737 004000 001236 6$: BIT       #ASYNC,STAT     ;IS THIS AN ASYNC LINE CARD?
1679 011732 001016          BNE      B$            ;#BR IF YES.
1680 011734 112705 000377          MOVB     #377,R5       ;MARK=377 (NEXT CHAR)
1681
1682
1683 011740 143705 001244          BICB     MASKX,R5       ;CLEAR BITS/PER/CHAR MASK.
1684 011744 113704 027562          MOVB     RXBA+2,R4     ;GET FOUND
1685 011750 120504          CMPB     R5,R4         ;OK
1686 011752 001401          BEQ      7$            ;
1687 011754 104002          HLT       2            ;EITHER TX NOT AT MARK (377) OR RX WRONG.
1688 011756 113704 027563 7$: MOVB     RXBA+3,R4     ;NEXT CHAR
1689 011762 120504          CMPB     R5,R4         ;
1690 011764 001401          BEQ      8$            ;
1691 011766 104002          HLT       2            ;IF ABOVE PASSED; RX WRONG!
1692 011770 104412          MSTCLR          ;INIT DV11
1693 011772 104401          SCOP1          ;LOCK ON CURRENT LINE?
1694 011774 005200          INC      R0            ;UPDATE LINE POINTER
1695 011776 005302          DEC      R2            ;4 LINES DONE?
1696 012000 001220          BNE      1$            ;BR IF NO
1697 012002 000207          RTS           ;EXIT FOR NEXT GROUP
```

```
1698
1699
1700
1701 ;***** TEST 5 *****
1702 ;*TEST OF RECEIVER CONTROL BYTE OPERATIONS.
1703 ;*TEST OF THE "INCLUDE IN BCC YES/NO FUNCTION"
1704 ;*TEST THAT THE CHAR "031" IS INCLUDED
1705 ;*IN THE BCC WHEN AT:
1706 ;*LRCB
1707 ;*CRC16
1708 ;*CRC.CCITT
1709 ;*THE RECEIVER BCC STARTS AT 0 AND CALCULATES
1710 ;*ONLY ONE CHAR (31)
1711 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
1712 ;*****
1713
1714 ; TEST 5
```

```

1715 012004 012737 000005 001226 TST5: MOV #5,TSTND
1716 012012 012737 012512 001216 MOV #TST6,NEXT
1717 012020 012700 000000 MOV #0.,RO ;PLACE LINE NUMBER INTO RO
1718 012024 013737 001416 001236 MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
1719 012032 100402 BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
1720 012034 004737 012122 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 1
1721 012040 012700 000004 1005: MOV #4.,RO ;PLACE LINE NUMBER INTO RO
1722 012044 013737 001420 001236 MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
1723 012052 100402 BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
1724 012054 004737 012122 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 2
1725 012060 012700 000010 1015: MOV #8.,RO ;LOAD LINE NUMBER
1726 012064 013737 001422 001236 MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
1727 012072 100402 BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED
1728 012074 004737 012122 JSR PC,105$ ;DO THE TEST FOR LINE CARD 3
1729 012100 012700 000014 1025: MOV #12.,RO ;LOAD LINE NO.
1730 012104 013737 001424 001236 MOV L12.15,STAT ;LOAD LINE CARD STATUS
1731 012112 100402 BMI 103$ ;BR IF LINE CARD NOT TO BE TESTED
1732 012114 004737 012122 JSR PC,105$ ;DO THE TESTS FOR LINE CARD 4
1733 012120 104400 1035: SCOPE ;SCOPE THIS TEST.
1734 012122 1055: ;TEST ENTRANCE.
1735 012122 012737 012156 001220 MOV #1$,LOCK ;SET FOR "LOCK ON LINE".
1736 012130 104413 RAMCLR ;CLEAR ALL SEC REGISTERS
1737 012132 105037 023611 CLRB ;CLEAT CNTRL BYTE (TX)
1738 012136 112737 000010 030211 MOVB #BIT3,RXTAB+31 ;SET "INC/BCC" IN RX CNTRL BYTE
1739 012144 112737 000031 022560 MOVB #31,TXBAP ;SET TX DATA CHAR
1740 012152 012702 000004 MOV #4,R2 ;4 LINE GROUP
1741 012156 15:
1742 012156 104412 64$: MSTCLR ;INIT DV11
1743 012160 010077 MOV RO,@DVSR$ ;LOAD LINE NO.
1744 012164 004737 022224 JSR PC,DV110N ;GOSUB DV110N
1745 012170 004537 022120 PERFORM ,SETREG ;
1746 012174 007 012 .BYTE 007,012 ;RXBCC, LINE PROTOCOL
1747 012176 000000 0 ;START BCC AT 0.
1748 012200 000000 0 ;POLYNOMIAL SELECT
1749 012202 005277 167154 INC @DVSCR ;SET MICRO CPU GO
1750 012206 105777 167150 65$: TSTB @DVSCR ;WAIT FOR
1751 012212 100375 BPL 65$ ;BIT 7 OF DVSCR=1
1752 012214 112777 000007 167152 MOVB #7,@DVSRSH ;SEL RX BCC REG
1753 012222 017704 167150 MOV @DV$RA,R4 ;READ BCC
1754 012226 005037 022116 CLR CALBCC ;SET SOFTWARE BCC=0
1755 012232 012737 000200 022112 MOV #LRCS,XPOLY ;SET SOFTWARE POLYNOMINAL
1756 012240 004537 021740 JSR R5,SIMBCC ;GO GET SOFTWARE BCC
1757 012244 000010 8. ;SHIFTS
1758 012246 000071 31 ;DATA
1759 012250 000000 0 ;PREVIOUS BCC
1760 012252 013705 022118 MOV CALBCC,R5 ;GET SOFTWARE BCC
1761 012256 020504 CMP R5,R4 ;SOFT=HARD?
1762 012260 001401 BEQ .+4 ;
1763 012262 104004 HLT 4 ;RECEIVER BCC INCORRECT!
1764 012264 104412 66$: MSTCLR ;INIT DV11
1765 012266 010077 MOV RO,@DVSR$ ;LOAD LINE NO.
1766 012272 004737 022224 JSR PC,DV110N ;GOSUB DV110N
1767 012276 004537 022120 PERFORM ,SETREG ;
1768 012302 007 012 .BYTE 007,012 ;RXBCC, LINE PROTOCOL
1769 012304 000000 0 ;START BCC AT 0.
1770 012306 000010 BIT3 ;POLYNOMIAL SELECT

```

```

1771 012310 005277 167046 INC @DVSCR ;SET MICRO CPU GO
1772 012314 105777 167042 67$: TSTB @DVSCR ;WAIT FOR
1773 012320 100375 BPL 67$ ;BIT 7 OF DVSCR=1
1774 012322 112777 000007 167044 MOVB #7,@DVSRSH ;SEL RX BCC REG
1775 012330 017704 167042 MOV @DV$RA,R4 ;READ BCC
1776 012334 005037 022116 CLR CALBCC ;SET SOFTWARE BCC=0
1777 012340 012737 120001 022112 MOV #CRC16,XPOLY ;SET SOFTWARE POLYNOMINAL
1778 012346 004537 021740 JSR R5,SIMBCC ;GO GET SOFTWARE BCC
1779 012352 000010 8. ;SHIFTS
1780 012354 000031 31 ;DATA
1781 012356 000000 0 ;PREVIOUS BCC
1782 012360 013705 022118 MOV CALBCC,R5 ;GET SOFTWARE BCC
1783 012364 020504 CMP R5,R4 ;SOFT=HARD?
1784 012366 001401 BEQ .+4 ;
1785 012370 104004 HLT 4 ;RECEIVER BCC INCORRECT!
1786 012372 104412 68$: MSTCLR ;INIT DV11
1787 012374 010077 MOV RO,@DVSR$ ;LOAD LINE NO.
1788 012400 004737 022224 JSR PC,DV110N ;GOSUB DV110N
1789 012404 004537 022120 PERFORM ,SETREG ;
1790 012410 007 012 .BYTE 007,012 ;RXBCC, LINE PROTOCOL
1791 012412 000000 0 ;START BCC AT 0.
1792 012414 000030 BIT4+BIT3 ;POLYNOMIAL SELECT
1793 012416 005277 166740 INC @DVSCR ;SET MICRO CPU GO
1794 012422 105777 166734 69$: TSTB @DVSCR ;WAIT FOR
1795 012426 100375 BPL 69$ ;BIT 7 OF DVSCR=1
1796 012430 112777 000007 166736 MOVB #7,@DVSRSH ;SEL RX BCC REG
1797 012436 017704 166734 MOV @DV$RA,R4 ;READ BCC
1798 012442 005037 022116 CLR CALBCC ;SET SOFTWARE BCC=0
1799 012446 012737 102010 022112 MOV #CRC.CCITT,XPOLY ;SET SOFTWARE POLYNOMINAL
1800 012454 004537 021740 JSR R5,SIMBCC ;GO GET SOFTWARE BCC
1801 012460 010010 8. ;SHIFTS
1802 012462 000031 31 ;DATA
1803 012464 000000 0 ;PREVIOUS BCC
1804 012466 013705 022118 MOV CALBCC,R5 ;GET SOFTWARE BCC
1805 012472 020504 CMP R5,R4 ;SOFT=HARD?
1806 012474 001401 BEQ .+4 ;
1807 012476 104004 HLT 4 ;RECEIVER BCC INCORRECT!
1808 012500 104401 SCDP1 ;LOCK ON SELECTED LINE?
1809 012502 005200 INC RO ;UPDATE LINE NO. POINTER
1810 012504 005302 DEC R2 ;ALL LINES DONE?
1811 012506 001223 BNE 15 ;BR IF NO
1812 012510 000207 RTS PC ;EXIT FOR NEXT GROUP
1813
1814
1815 ;***** TEST 6 *****
1816 ;*TEST OF RECEIVER CONTROL BYTE OPERATIONS.
1817 ;*TEST OF THE "NEXT MODE" FUNCTION.
1818 ;*TEST THAT THE NEXT MODE REGISTER (015)
1819 ;*CAN BE LOADED FROM THE CONTROL BYTES.
1820 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
1821 ;*****
1822
1823 ; TEST 6
1824 ;-----
1825 012512 012737 000006 001226 TST6: MOV #6,TSTND
1826 012520 012737 012772 001216 MOV #TST7,NEXT

```

```

1827 012526 012700 000000      MOV      #0.,R0      ;PLACE LINE NUMBER INTO R0
1828 012532 013737 001416 001236  MOV      L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
1829 012540 100402                BMI      100$       ;BR IF LINE CARD NOT TO BE TESTED
1830 012542 004777 012630                JSR      PC,105$    ;GO DO THE TEST FOR LINE CARD 1
1831 012546 012700 000004                MOV      #4.,R0      ;PLACE LINE NUMBER INTO R0
1832 012552 013737 001420 001236  MOV      L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
1833 012560 100402                BMI      101$       ;BR IF LINE CARD NOT TO BE TESTED
1834 012562 004737 012630                JSR      PC,105$    ;GO DO THE TEST FOR LINE CARD 2
1835 012566 012700 000010                MOV      #8.,R0      ;LOAD LINE NUMBER
1836 012572 013737 001422 001236  MOV      L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
1837 012600 1*0402                BMI      102$       ;BR IF LINE CARD NOT TO BE TESTED
1838 012602 004737 012630                JSR      PC,105$    ;DO THE TEST FOR LINE CARD 3
1839 012606 012700 000014                MOV      #12.,R0     ;LOAD LINE NO.
1840 012612 013737 001424 001236  MOV      L12.15,STAT ;LOAD LINE CARD STATUS
1841 012620 100402                BMI      103$       ;BR IF LINE CARD NOT TO BE TESTED
1842 012622 004737 012630                JSR      PC,105$    ;DO THE TESTS FOR LINE CARD 4
1843 012626 104400                SCOPE     SCOPE     ;SCOPE THIS TEST.
1844 012630                105$:     TEST ENTRANCE.
1845 012630 012737 012662 001220  MOV      #1$,LOCK    ;SET IF SW09=1
1846 012636 104413                RAMCLR    ;CLEAR ALL SEC REGISTERS
1847 012640 005003                CLR      R3          ;SET EXPECT RESULTS OF MODE REGISTER
1848 012642 005001                CLR      R1          ;SET CNTRL BYTE MODE
1849 012644 012702 000004                MOV      #4,R2       ;SET FOR4 LINE GROUP
1850 012650 012737 000031 022560  MOV      #31,TXBAP   ;LOAD TX DATA CHAR
1851 012656 105037 023611                CLRB     TXTAB+31    ;ZERO TX CNTRL BYTE
1852 012662 110137 030211                MOV      R1,RXTAB+31 ;LOAD RX CNTRL BYTE (WITH MODE)
1853 012666 004737 022224                JSR      PC,DV110N   ;GO SETUP ROUTINE THINGS (BA,BC,LS,LP)
1854 012672 004537 022120                PERFORM  ,SETREG    ;ZERO
1855 012676 015 015                .BYTE   015,015     ;RECEIVER
1856 012700 000000                0           ;MODE
1857 012702 000000                0           ;REGISTER
1858 012704 005277 166452                INC      @DVSCR      ;SET MICRO CPU GO
1859 012710 105777 166446                TSTB    @DVSCR      ;WAIT FOR
1860 012714 100375                BPL      -4          ;DVSCR07=1
1861 012716 112777 000015 166450  MOV      #15,@DVSRSH ;SEL RX MODE REGISTER
1862 012724 017704 166446                MOV      @DVSRA,R4   ;READ MODE REGISTER
1863 012730 010305                MOV      R3,R5       ;SET EXPECTED MODE
1864 012732 020504                CMP      R5,R4       ;
1865 012734 001401                BEQ      3$          ;
1866 012736 104002                HLT      2           ;RX MODE REGISTER WRONG
1867 012740 104412                3$:     MSTCLR    ;INIT DV11
1868 012742 005203                INC      R3          ;UPDATE EXPECTED MODE
1869 012744 062701 000040                ADD      #BITS,R1    ;UPDATE LOADED (NEXT) MODE
1870 012750 105701                TSTB    R1          ;ALL DONE?
1871 012752 001743                BEQ      1$          ;BR IF NO
1872 012754 005001                CLR      R1          ;ZERO LOAD MODE
1873 012756 005003                CLR      R3          ;ZERO EXPECTED MODE
1874 012760 104401                SCOP1    ;LOCK ON SELECTED LINE?
1875 012762 005200                INC      R0          ;UPDATE LINE POINTER
1876 012764 005302                DEC      R2          ;4 LINE GROUP DONE?
1877 012766 001335                BNE     1$          ;BR IF NO
1878 012770 000207                RTS      PC          ;EXIT FOR NEXT GROUP OF LINES
1879
1880
1881
1882
;***** TEST 7 *****
;*TEST OF TRANSMITTER CONTROL BYTE OPERATIONS.
    
```

```

1883
1884
1885
1886
1887
1888
1889
1890
1891
1892 012772 012737 000007 001226 TST7:  MOV      #7,TSTNO
1893 013000 012737 013244 001216  MOV      #TST10,NEXT
1894 013006 012700 000000                MOV      #0.,R0      ;PLACE LINE NUMBER INTO R0
1895 013012 013737 001416 001236  MOV      L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
1896 013020 100402                BMI      100$       ;BR IF LINE CARD NOT TO BE TESTED
1897 013022 004737 013110                JSR      PC,105$    ;GO DO THE TEST FOR LINE CARD 1
1898 013026 012700 000004                MOV      #4.,R0      ;PLACE LINE NUMBER INTO R0
1899 013032 013737 001420 001236  MOV      L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
1900 013040 100402                BMI      101$       ;BR IF LINE CARD NOT TO BE TESTED
1901 013042 004737 013110                JSR      PC,105$    ;GO DO THE TEST FOR LINE CARD 2
1902 013046 012700 000010                MOV      #8.,R0      ;LOAD LINE NUMBER
1903 013052 013737 001422 001236  MOV      L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
1904 013060 100402                BMI      102$       ;BR IF LINE CARD NOT TO BE TESTED
1905 013062 004737 013110                JSR      PC,105$    ;DO THE TEST FOR LINE CARD 3
1906 013066 012700 000014                MOV      #12.,R0     ;LOAD LINE NO.
1907 013072 013737 001424 001236  MOV      L12.15,STAT ;LOAD LINE CARD STATUS
1908 013100 100402                BMI      103$       ;BR IF LINE CARD NOT TO BE TESTED
1909 013102 004737 013110                JSR      PC,105$    ;DO THE TESTS FOR LINE CARD 4
1910 013106 104400                SCOPE     SCOPE     ;SCOPE THIS TEST.
1911 013110                105$:     TEST ENTRANCE.
1912 013110 012737 013144 001220  MOV      #1$,LOCK    ;SET IF SW09=1
1913 013116 104413                RAMCLR    ;CLEAR ALL SEC REGISTERS
1914 013120 112737 000002 023611  MOV      #BIT1,TXTAB+31 ;SET "SND/DLE" IN CNTRL BYTE
1915 013126 112737 000031 022560  MOV      #31,TXBAP   ;SET TX DATA CHAR
1916 013134 105037 030211                CLRB     RXTAB+31    ;ZERO RX CNTRL BYTE
1917 013140 012702 000004                MOV      #4,R2       ;SET FOR 4 LINE GROUP
1918 013144 004737 022224                JSR      PC,DV110N   ;SET ROUTINE THING
1919 013150 004537 022120                PERFORM  ,SETREG    ;
1920 013154 012 012                .BYTE   012,012     ;LINE PROTOCOL REG
1921 013156 012400                25*400          ;PUT 25
1922 013160 012400                25*400          ;IN HIGH BYTE
1923 013162 005037 027560                CLR      RXBA        ;ZERO RX BUFFER
1924 013166 005277 166170                INC      @DVSCR      ;SET MICRO CPU GO
1925 013172 105777 166164                TSTB    @DVSCR      ;WAIT FOR
1926 013176 100375                BPL      -4          ;DVSCR07=1
1927 013200 013704 027560                MOV      RXBA,R4     ;GET DATA
1928 013204 012705 000025                MOV      #25,R5       ;LOAD DLE INTO EXPECTED
1929 013210 020504                CMP      R5,R4       ;
1930 013212 001401                BEQ      2$          ;
1931 013214 104003                HLT      3           ;25 (DLE) NOT 1ST IN RX BUFFER
1932 013216 104412                2$:     MSTCLR    ;INIT DV11
1933 013220 1*2777 000012 166146  MOV      #12,@DVSRSH ;SEL LINE PROTOCOL
1934 013226 005077 166144                CLR      @DVSRA      ;ZERO IT.
1935 013232 104401                SCOP1    ;LOCK ON SELECTED LINE?
1936 013234 005200                INC      R0          ;UPDATE LINE POINTER
1937 013236 005302                DEC      R2          ;4 LINE GROUP DONE?
1938 013240 001341                BNE     1$          ;BR IF NO
    
```

```
1939 013242 000207          RTS    PC          ;EXIT FOR NEXT GROUP OF LINES
1940
1941
1942 ;***** TEST 10 *****
1943 ;*TEST OF BOTH BITS 6 AND 5 OF THE LINE PROTOCOL REG.
1944 ;*TEST THAT NEITHER THE TRANSMITTER OR RECEIVER
1945 ;*CONTROL BYTES ARE USED AND THAT
1946 ;*THE CHARS ARE AUTOMATICALLY INCLUDED INTO THE BCC.
1947 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
1948 ;*****
1949
1950 ; TEST 10
1951
1952 013244 012737 000010 001226 TST10: MOV    #10,TSTNO
1953 013252 012737 013554 001216 MOV    #TST11,NEXT
1954 013260 012700 000000 MOV    #0,.RO          ;PLACE LINE NUMBER INTO RO
1955 013264 013737 001416 001236 MOV    L00.03,STAT    ;LOAD LINE CARD STATUS INTO STAT
1956 013272 100402 BMI    100$          ;BR IF LINE CARD NOT TO BE TESTED
1957 013274 004737 013362 JSR    PC,105$        ;GO DO THE TEST FOR LINE CARD 1
1958 013300 012700 000004 100$: MOV    #4,.RO          ;PLACE LINE NUMBER INTO RO
1959 013304 013737 001420 001236 MOV    L04.07,STAT    ;LOAD LINE CARD STATUS INTO STAT
1960 013312 100402 BMI    101$          ;BR IF LINE CARD NOT TO BE TESTED
1961 013314 004737 013362 JSR    PC,105$        ;GO DO THE TEST FOR LINE CARD 2
1962 013320 012700 000010 101$: MOV    #8,.RO          ;LOAD LINE NUMBER
1963 013324 013737 001422 001236 MOV    L08.11,STAT    ;LOAD LINE CARD STATUS INTO STAT
1964 013332 100402 BMI    102$          ;BR IF LINE CARD NOT TO BE TESTED
1965 013334 004737 013362 JSR    PC,105$        ;DO THE TEST FOR LINE CARD 3
1966 013340 012700 000014 102$: MOV    #12,.RO         ;LOAD LINE NO.
1967 013344 013737 001424 001236 MOV    L12.15,STAT    ;LOAD LINE CARD STATUS
1968 013352 100402 BMI    103$          ;BR IF LINE CARD NOT TO BE TESTED
1969 013354 0 4737 013362 JSR    PC,105$        ;DO THE TESTS FOR LINE CARD 4
1970 013360 104400 103$: SCOPE          ;SCOPE THIS TEST.
1971 013362 105$:          ;TEST ENTRANCE.
1972 013362 012737 013412 001220 MOV    #1$,LOCK       ;SET RETURN IF SW09=1
1973 013370 104413 RAMCLR          ;CLEAR ALL SEC REGISTERS
1974 013372 012702 000004 MOV    #4,R2          ;SET FOR 4 LINE GROUP
1975 013376 112737 000340 023575 MOVVB  #BIT7+BIT6+BITS,TXTAB+1$ ;SET RX AND TX NEXT MODE=7
1976 013404 113737 023575 030175 MOVVB  TXTAB+1$,RXTAB+1$ ;SET UP MINOR DETAILS
1977 013412 004737 022224 1$: JSR    PC,DV11CN     ;GET LINE PROTOCOL REGISTER
1978 013416 112777 000012 165750 MOVVB  #12,@DVSRSH    ;SET TX AND RX DDCMP MODE
1979 013424 052777 000140 165744 BIS    #BIT6+BITS,@DVSRA ;LOAD DATA CHAR
1980 013432 112737 000015 022560 MOVVB  #15,TXBAP      ;SET MICRO CPU GO
1981 013440 005277 165716 INC     @DVSCR        ;WAIT FOR
1982 013444 105777 165712 TSTB   @DVSCR        ;DVSCRO7=1
1983 013450 100375 BPL    R5            ;EXPECTED=0
1984 013452 005005 CLR     R5            ;GET TX MODE REG
1985 013454 112777 000014 165712 MOVVB  #14,@DVSRSH    ;READ MODE REG
1986 013462 017704 165710 MOV    @DVSRA,R4     ;S/B=0
1987 013466 001401 BEQ    +4             ;TX MODE REG S/B=0
1988 013470 104001 HLT    1              ;GET RX MODE REG
1989 013472 105277 165676 INCB   @DVSRSH        ;READ RX MODE
1990 013476 017704 165674 MOV    @DVSRA,R4     ;S/B=0
1991 013502 001401 BEQ    3$             ;RX MODE REG S/B=0
1992 013504 104001 HLT    1              ;TX BCC REG.
1993 013506 112777 000006 165660 3$: MOVVB  #6,@DVSRSH    ;READ TXBCC REG.
1994 013514 017704 165656 MOV    @DVSRA,R4
```

```
1995 013520 001001          BNE    4$            ;S/B NOT=0
1996 013522 104000 HLT    0              ;NOTHING IN BCC!! (TX)
1997 013524 105277 165644 4$: INCB   @DVSRSH        ;GET RX BCC REG
1998 013530 017704 165642 MOV    @DVSRA,R4     ;READ INTO R4
1999 013534 001001 BNE    5$            ;
2000 013536 104000 HLT    0              ;NOTHING IN RXBCC!!
2001 013540 104412 5$: MSTCLR          ;INIT DV11
2002 013542 104401 SCOP1          ;LOCK ON SELECTED LINE?
2003 013544 005200 INC     R0            ;UPDATE LINE POINTER
2004 013546 005302 DEC     R2            ;4 LINES DONE?
2005 013550 001320 BNE    1$            ;BR IF NO
2006 013552 000207 RTS     PC            ;EXIT FOR NEXT 4 LINE GROUP
2007
2008 ;***** TEST 11 *****
2009 ;*TEST OF BIT 1 IN LINE PROTOCOL PARAMETER REGISTER.
2010 ;*TEST OF "STRIP LEADING SYNC'S".
2011 ;*TEST TO XMIT 10. SYNC CHARS, 1 NON-SYNC, AND 2 SYNC'S
2012 ;*[13 CHARS TOTAL]
2013 ;*DV11 SHOULD RECEIVE 1 NON-SYNC,AND TWO SYNC'S.
2014 ;*[ 3 CHARS TOTAL].
2015 ;*THE TEN LEADING SYNC'S S/B STRIPPED
2016 ;*THIS TEST IS DONE FOR SYNC LINE CARDS ONLY.
2017 ;*****
2018
2019 ; TEST 11
2020
2021
2022 013554 012737 000011 001226 TST11: MOV    #11,TSTNO
2023 013562 012737 014166 001216 MOV    #TST12,NEXT
2024 013570 012700 000000 MOV    #0,.RO          ;PLACE LINE NUMBER INTO RO
2025 013574 113737 001406 001244 MOVVB  MASK.A,MASKX   ;PLACE "MASK"FOR CHARS INTO MASKX
2026 013602 013737 001416 001236 MOV    L00.03,STAT    ;LOAD LINE CARD STATUS INTO STAT
2027 013610 100402 BMI    100$          ;BR IF LINE CARD NOT TO BE TESTED
2028 013612 004737 013722 JSR    PC,105$        ;GO DO THE TEST FOR LINE CARD 1
2029 013616 012700 000004 100$: MOV    #4,.RO          ;PLACE LINE NUMBER INTO RO
2030 013622 113737 001407 001244 MOVVB  MASK.B,MASKX   ;GET MASK
2031 013630 013737 001420 001236 MOV    L04.07,STAT    ;LOAD LINE CARD STATUS INTO STAT
2032 013636 100402 BMI    101$          ;BR IF LINE CARD NOT TO BE TESTED
2033 013640 004737 013722 JSR    PC,105$        ;GO DO THE TEST FOR LINE CARD 2
2034 013644 012700 000010 101$: MOV    #8,.RO          ;LOAD LINE NUMBER
2035 013650 113737 001410 001244 MOVVB  MASK.C,MASKX   ;GET MASK
2036 013656 013737 001422 001236 MOV    L08.11,STAT    ;LOAD LINE CARD STATUS INTO STAT
2037 013664 100402 BMI    102$          ;BR IF LINE CARD NOT TO BE TESTED
2038 013666 004737 013722 JSR    PC,105$        ;DO THE TEST FOR LINE CARD 3
2039 013672 012700 000014 102$: MOV    #12,.RO         ;LOAD LINE NO.
2040 013676 113737 001411 001244 MOVVB  MASK.D,MASKX   ;GET MASK
2041 013704 013737 001424 001236 MOV    L12.15,STAT    ;LOAD LINE CARD STATUS
2042 013712 100402 BMI    103$          ;BR IF LINE CARD NOT TO BE TESTED
2043 013714 004737 013722 JSR    PC,105$        ;DO THE TESTS FOR LINE CARD 4
2044 013720 104400 103$: SCOPE          ;SCOPE THIS TEST.
2045 013722 105$:          ;TEST ENTRANCE.
2046 013722 032737 004000 001236 BIT    #ASYNC,STAT    ;IS THIS AN ASYNC LINE CARD?
2047 013730 001401 BEQ    +4             ;BR IF NOT ASYNC LINE CARD.
2048 013732 000207 RTS     PC            ;EXIT TEST. (ASYNC LINE CARD NOT TESTED)
2049 013734 012737 014004 001220 MOV    #2$,LOCK       ;SET RETURN IF SW09=1
2050 013742 104413 RAMCLR          ;CLEAR ALL SEC REGISTERS
```

2051	013744	012702	000004		MOV	#4,R2	:4 LINE GROUP
2052	013750	012704	000010		MOV	#8,R4	:LOAD TX BUFFER
2053	013754	012705	022560		MOV	#TXBAP,R5	:WITH
2054	013760	113725	001236	1\$:	MOVB	STAT,(R5)+	:B, SYNC
2055	013764	005304			DEC	R4	:CHARS
2056	013766	001374			BNE	1\$:
2057	013770	112725	000005		MOVB	#5,(R5)+	:LOAD "NON-SYNC" CHAR
2058	013774	113725	001236		MOVB	STAT,(R5)+	:SYNC
2059	014000	113725	001236		MOVB	STAT,(R5)+	:SYNC
2060	014004	004737	022224	2\$:	JSR	PC,DV110N	:MINOR DETAIL SETUP
2061	014010	005037	027560		CLR	RXBA	:CLEAR
2062	014014	005037	027562		CLR	RXBA+2	:RX BUFFER
2063	014020	004537	022120		PERFORM	,SETREG	:
2064	014024	001	005		.BYTE	001,005	:TX PRINCIPLE BC, RX BC
2065	014026	177763			-13,		:
2066	014030	177775			-3		:
2067	014032	112777	000012	165334	MOVB	#12,@DVSRSH	:LINE PROTOCOL REG
2068	014040	012777	000143	165330	MOV	#BIT6+BIT5+BIT1+BIT0,@DVSR	:BIT0,@DVSR
2069	014046	005277	165310		INC	@DVSCR	:LP=TX+RX DDCMP, STRIP SYNC, IDLE MARK
2070	014052	105777	165304		TSTB	@DVSCR	:WAIT FOR
2071	014056	100375			BPL	.-4	:DVSCRO7=1
2072	014060	012705	000005		MOV	#5,R5	:1ST DATA S/B=15
2073	014064	113704	027560		MOVB	RXBA,R4	:GET DATA
2074	014070	042704	177400		BIC	#^C<377>,R4	:STRIP HIGH BYTE
2075	014074	020504			CMP	R5,R4	:OK
2076	014076	001401			BEQ	3\$:
2077	014100	104001			HLT	1	:1ST CHAR NOT=15
2078	014102	113705	001236	3\$:	MOVB	STAT,R5	:LOAD SYNC INTO EXPECTED
2079	014106	042705	177400		BIC	#^C<377>,R5	:STRIP HIGH BYTE
2080	014112	143705	001244		BICB	MASKX,R5	:CLEAR BITS/PER/CHAR MASK.
2081	014116	113704	027561		MOVB	RXBA+1,R4	:GET 2ND CHAR
2082	014122	042704	177400		BIC	#^C<377>,R4	:STRIP HIGH BYTE
2083	014126	020504			CMP	R5,R4	:WERE *ONLY* LOADING- SYNC STRIPPED?
2084	014130	001401			BEQ	4\$:
2085	014132	104001			HLT	1	:JUST *LEADING* SYNC ARE TO BE STRIPPED
2086	014134	113704	027562	4\$:	MOVB	RXBA+2,R4	:GET 3RD CHAR
2087	014140	042704	177400		BIC	#^C<377>,R4	:STRIP HIGH BYTE
2088	014144	020504			CMP	R5,R4	:OK?
2089	014146	001401			BEQ	5\$:
2090	014150	104001			HLT	1	:JUST *LEADING* SYNC ARE TO BE STRIPPED
2091	014152	104412		5\$:	MSTCLR		:INIT DV11
2092	014154	104401			SCOP1		:LOCK ON SELECTED LINES?
2093	014156	005200			INC	R0	:UPDATE LINE POINTER
2094	014160	005372			DEC	R2	:4 LINE GROUP DONE?
2095	014162	001310			BNE	2\$:BR IF NO
2096	014164	000207			RTS	PC	:EXIT FOR NEXT GROUP

2097
2098
2099
2100
2101
2102
2103
2104
2105
2106

```

:***** TEST 12 *****
:TEST OF BIT08 OF DVSCR.
:TEST OF "RECEIVER INTERRUPT RESPONSE COMPLETE"
:TEST TO RECEIVE 6 CHARS INTERRUPTING
:*ON EACH CHAR HAVING IT BEING A "SPECIAL CHAR"
:*AND THE RESTARTING THE MICRO PROCESSOR AFTER EACH
:*INTERUPT FLAG.
:*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.

```

2107							:***** TEST 12 *****	
2108							:***** TEST 12 *****	
2109							:***** TEST 12 *****	
2110							:***** TEST 12 *****	
2111	014166	012737	000012	001226	TST12:	MOV	#12,TSTNO	
2112	014174	012737	014650	001216		MOV	#TST13,NEXT	
2113	014202	012700	000000			MOV	#0,R0	:PLACE LINE NUMBER INTO R0
2114	014206	013737	001416	001236		MOV	L00.03,STAT	:LOAD LINE CARD STATUS INTO STAT
2115	014214	100402				BMI	100\$:BR IF LINE CARD NOT TO BE TESTED
2116	014216	004737	014304			JSR	PC,105\$:GO DO THE TEST FOR LINE CARD 1
2117	014222	012700	000004	100\$:		MOV	#4,R0	:PLACE LINE NUMBER INTO R0
2118	014226	013737	001420	301236		MOV	L04.07,STAT	:LOAD LINE CARD STATUS INTO STAT
2119	014234	100402				BMI	101\$:BR IF LINE CARD NOT TO BE TESTED
2120	014236	004737	014304			JSR	PC,105\$:GO DO THE TEST FOR LINE CARD 2
2121	014242	012700	000010	101\$:		MOV	#8,R0	:LOAD LINE NUMBER
2122	014246	013737	001422	001236		MOV	L08.11,STAT	:LOAD LINE CARD STATUS INTO STAT
2123	014254	100402				BMI	102\$:BR IF LINE CARD NOT TO BE TESTED
2124	014256	004737	014304			JSR	PC,105\$:DO THE TEST FOR LINE CARD 3
2125	014262	012700	000014	102\$:		MOV	#12,R0	:LOAD LINE NO.
2126	014266	013737	001424	001236		MOV	L12.15,STAT	:LOAD LINE CARD STATUS
2127	014274	100402				BMI	103\$:BR IF LINE CARD NOT TO BE TESTED
2128	014276	004737	014304			JSR	PC,105\$:DO THE TESTS FOR LINE CARD 4
2129	014302	104400			103\$:	SCOPE		:SCOPE THIS TEST.
2130	014304				105\$:			:TEST ENTRANCE.
2131	014304	012737	014362	001220		MOV	#66\$,LOCK	
2132	014312	104413				RAMCLR		:CLEAR ALL SEC REGISTERS
2133	014314	005001				CLR	R1	:
2134	014316	012702	000004			MOV	#4,R2	:SET FOR 4 LINE GROUP
2135	014322	005005				CLR	R5	:LOAD
2136	014324	012704	022560			MOV	#TXBAP,R4	:TX DATA
2137	014330	10524			1\$:	MOVB	R5,(R4)+	:
2138	014332	005205				INC	R5	:
2139	014334	022705	000007			CMP	#7,R5	:
2140	014340	001373				BNE	1\$:
2141	014342	012705	000006			MOV	#5,R5	:FILL
2142	014346	012704	030160			MOV	#RXTAB,R4	:THE
2143	014352	112724	000001		2\$:	MOVB	#RTAB,(R4)+	:RX CNTRL TABLE
2144	014356	005305				DEC	R5	:WITH SPECIAL
2145	014360	001374				BNE	2\$:CHAR BITS
2146	014362	004737	022224	66\$:		JSR	PC,DV110N	:SET UP DV11
2147	014366	004537	022120			PERFORM	,SETREG	:
2148	014372	001	005			.BYTE	001,005	:TX PRINCIPLE BC, RX BC
2149	014374	177770				-8,		:
2150	014376	177772				-6		:
2151	014400	112777	000012	164766		MOVB	#12,@DVSRSH	:LINE PROTOCOL
2152	014406	052777	000100	164762		BIS	#BIT6,@DVSR	:TX DDCMP
2153	014414	012737	000340	177776		MOV	#340,PS	:LOCK OUT INTERRUPTS
2154	014422	012777	014472	164722		MOV	#67\$,@DVRVEC	:SET RX INTER VECTOR.
2155	014430	012777	000340	164716		MOV	#340,@DVRVLC	:SET PRIO. LEVEL TO 7
2156	014436	052777	000101	164716		BIS	#BIT6+BIT0,@DVSCR	:
2157								:SET RX IE AND UCPU GO.
2158	014444	005003				CLR	R3	:DATA IMAGE
2159	014446	005037	014646	3\$:		CLR	69\$:STALL COUNTER
2160	014452	005037	177776			CLR	PS	:ENABLE INTERRUPTS
2161	014456	104414				DELAY		:WASTE TIME
2162	014460	005237	014646			INC	69\$:UPDATE STALL


```

2163 014464 001372 BNE .-12 ;BR BACK
2164 014466 104000 HLT ;NO INTERRUPT OCCURED.
2165 014470 024646 CMP -(SP),-(SP) ;FAKE AN INTERRUPT
2166 014472 010003 67$: MOV R0,R5 ;LOAD LINE NO.
2167 014474 000305 SWAB R5 ;PUT IN HIGH BYTE
2168 014476 050305 BIS R3,R5 ;SET DATA
2169 014500 017704 164662 MOV @DVRIC,R4 ;READ FOUND RESULT
2170 014504 020504 CN? R5,R4 ;
2171 014506 001401 BEQ 4$ ;
2172 014510 104001 HLT ;DVRIC WRONG!
2173 014512 052777 000400 164642 4$: BIS #BIT8,@DVSCR ;SET "RECEIVER INT RESP COMP"
2174 014520 005203 INC R3 ;UPDATE DATA IMAGE
2175 014522 022703 000006 CMP #6,R3 ;ALL DONE?
2176 014525 001403 BEQ 70$ ;BR IF YES
2177 014530 012716 014446 MOV #3$, (SP) ;SET RETURN
2178 014534 000002 RTI ;CONTINUE
2179 014536 042777 000100 164616 70$: BIC #BIT6,@DVSCR ;NO MORE INTERRUPTS.
2180 014544 012716 014552 MOV #68$, (SP) ;SET RETURN
2181 014550 000002 RTI ;CONT.
2182 014552 105777 164604 68$: TSTB @DVSCR ;WAIT FOR
2183 014556 100375 BPL .-4 ;DVSCRO7=1
2184 014560 010005 MOV R0,R5 ;LOAD LINE NO.
2185 014562 000305 SWAB R5 ;PUT IN HIGH BYTE
2186 014564 052705 040005 BIS #BIT14+5,R5 ;"BYTE CNT WARNING + DATA"
2187 014570 017704 164572 MOV @DVRIC,R4 ;READ RESULTS
2188 014574 020504 CMP R5,R4 ;
2189 014576 001401 BEQ 5$ ;
2190 014600 104001 HLT ;DVRIC WRONG!
2191 014602 012701 027560 5$: MOV #RXBA,R1 ;GET RX POINTER
2192 014606 005005 CLR R5 ;
2193 014610 005004 CLR R4 ;
2194 014612 112104 6$: MOVB (R1)+,R4 ;GET RX DATA (INCORE)
2195 014614 020504 CMP R5,R4 ;
2196 014616 001401 BEQ 7$ ;
2197 014620 104001 HLT ;RECEIVER PLACED DATA IN CORE WRONG
2198 014622 005205 INC R5 ;UPDATE DATA IMAGE
2199 014624 022705 000006 CMP #6,R5 ;ALL DONE?
2200 014630 001370 BNE 6$ ;BR IF NO
2201 014632 104412 MSTCLR ;INIT DV11
2202 014634 104401 SCOP1 ;LOCK ON CURRENT LINE?
2203 014636 005200 INC R0 ;UPDATE LINE POINTER
2204 014640 005302 DEC R2 ;4 LINE GROUP DONE?
2205 014642 001247 BNE 65$ ;BR IF NO
2206 014644 000207 RTS PC ;EXIT FOR NEXT GROUP OF LINES
2207 014646 000000 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

```

```

2219 ;RECEIVER "EXPECT THE BCC"
2220 ;AND THAT BIT10 IN LINE PROGRESS TELL TX TO SEND BCC.
2221 ;THIS TEST USES CRC.CCITT FOR THE POLYNOMIAL
2222 ;**NOTE*: IF LINE CARD IS SET FOR OTHER THAN "B" BITS
2223 ;THE TEST WILL *NOT* BE EXECUTED ON THAT LINE CARD!!
2224 ;THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
2225 ;*****
2226 ;
2227 ; TEST 13
2228 ;-----
2229 014650 012737 000013 001226 TST13: MOV #13,TSTND
2230 014656 012737 015442 001216 MOV #TST14,NEXT
2231 014664 012700 000000 MOV #0.,R0 ;PLACE LINE NUMBER INTO R0
2232 014670 013737 001413 001236 MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
2233 014676 100402 BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
2234 014700 004737 014766 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 1
2235 014704 012700 000004 MOV #4.,R0 ;PLACE LINE NUMBER INTO R0
2236 014710 013737 001420 001236 MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
2237 014716 100402 BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
2238 014720 004737 014766 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 2
2239 014724 012700 000010 MOV #8.,R0 ;LOAD LINE NUMBER
2240 014730 013737 001422 001236 MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
2241 014736 100402 BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED
2242 014740 004737 014766 JSR PC,105$ ;DO THE TEST FOR LINE CARD 3
2243 014744 012700 000014 MOV #12.,R0 ;LOAD LINE NO.
2244 014750 013737 001424 001236 MOV L12.15,STAT ;LOAD LINE CARD STATUS
2245 014756 100402 BMI 103$ ;BR IF LINE CARD NOT TO BE TESTED
2246 014760 004737 014766 JSR PC,105$ ;DO THE TESTS FOR LINE CARD 4
2247 014764 104400 103$: SCOPE ;SCOPE THIS TEST.
2248 014766 105$: ;TEST ENTRANCE.
2249 014766 012737 015114 001220 MOV #65$,LOCK ;SET RETURN IF SW09=1
2250 014774 032737 001400 001236 BIT #BITS+BITS,STAT ;"B BITS/PER/CHAR ?"
2251 015002 001401 BEQ .+4 ;BR IF YES
2252 015004 000207 RTS PC ;EXIT TEST FOR THIS LINE CARD!
2253 015006 104413 RAMCLR ;CLEAR ALL SECONDARY REGISTERS
2254 015010 012702 000004 MOV #4,R2 ;SET FOR 4 LINE GROUP
2255 015014 012704 000012 MOV #10.,R4 ;LOAD 10 BYTES
2256 015020 012705 023560 MOV #TXTAB,R5 ;WITH
2257 015024 112725 000010 1$: MOVB #BITS,(R5)+ ;INC/BCC
2258 015030 005304 DEC R4 ;
2259 015032 001374 BNE 1$ ;
2260 015034 012705 023560 MOV #TXTAB,R5 ;CLEAR
2261 015040 013704 001236 MOV STAT,R4 ;SYNC
2262 015044 042774 177400 BIC #<37>,R4 ;CONTROL
2263 015050 060406 ADD R4,R5 ;BYTE
2264 015052 105010 CLR R5 ;
2265 015054 012705 022560 MOV #TXBAP,R5 ;LOAD
2266 015060 005004 CLR R4 ;DATA
2267 015062 110425 2$: MOVB R4,(R5)+ ;INTO
2268 015064 005204 INC R4 ;TRANSMITTER BUFFER
2269 015066 012704 000013 CMP #11.,R4 ;
2270 015072 001373 BNE 2$ ;
2271 015074 012705 030160 MOV #RXTAB,R5 ;LOAD
2272 015100 012704 000012 MOV #10.,R4 ;10
2273 015104 112725 000010 3$: MOVB #BITS,(R5)+ ;RECEIVER
2274 015110 005304 DEC R4 ;CONTROL BYTES

```

```

2275 015112 001374          BNE 3$ ;WITH "INC/BCC"
2276 015114 010077 164252    MOV RO,@DVSR5 ;LOAD LINE NO.
2277 015120 032737 004000 001236 BIT #ASYNCR,STAT ;IS THIS AN ASYNC LINE CARD?
2278 015126 001406          BEQ 80$ ;#BR IF NOT ASYNC.
2279 015130 004537 022120    PERFORM ,SETREG ;ADJUST FOR ASYNC LINE CARD
2280 015134 000 001        .BYTE 000,001 ;#REGISTERS
2281 015136 022560          TXBAP ;LOAD FOR ASYNC
2282 015140 077766          <-10.>-BIT15 ;#LOAD FOR ASYNC
2283 015142 000405          BR 81$ ;#CONTINUE TEST
2284 015144 004537 022120    80$: PERFORM ,SETREG ;
2285 015150 000 001        .BYTE 000,001 ;TX PRINCIPLE BA, BC
2286 015152 022556          SYNC ;
2287 015154 077764          <-12.>-BIT15 ;MARKED BC1
2288 015156 004537 022120    81$: PERFORM ,SETREG ;
2289 015162 004 005        .BYTE 004,005 ;RX BA, BC
2290 015164 027560          RXBA ;
2291 015166 077766          <-10.>-BIT15 ;MARKED BC1
2292 015170 004537 022120    PERFORM ,SETREG ;
2293 015174 010 011        .BYTE 010,011 ;TX TABLE, RX TABLE
2294 015176 023560          TXTAB ;
2295 015200 030160          RXTAB ;
2296 015202 004537 022120    PERFORM ,SETREG ;
2297 015206 012 013        .BYTE 012,013 ;LINE PROTOCOL, LINE STATE
2298 015210 000031          BIT4+BIT3+BIT0 ;CRC.CCITT, IDLE MARK
2299 015212 162004          BIT15+BIT14+BIT13+BIT10+BIT2 ;
2300 015214 004537 022120    PERFORM ,SETREG ;MODE 7, TXGO
2301 015220 016 017        .BYTE 016,017 ;LINE PROGRESS REC, REC CNTR STORE
2302 015222 162000          BIT15+BIT14+BIT13+BIT10 ;NEXT MODE=7
2303 015224 000000          0 ;ZERO
2304 015226 032737 004000 001236 BIT #ASYNCR,STAT ;IS THIS ASYNC LINE CARD?
2305 015234 01412 000000    BEQ 60$ ;#BR IF NO.
2306 015236 004537 022164    PERFORM ,LOAD.MODE ;LOAD PARAMETERS.
2307 015242 020000          BIT13 ;#RECEIVER ENABLE
2308 015244 004537 022164    PERFORM ,LOAD.MODE ;#
2309 015250 015000          <BIT12+BIT11>+BIT9 ;#8 BITS/PER/CHAR
2310 015252 004537 022164    PERFORM ,LOAD.MODE ;#
2311 015256 072000          <BIT14+BIT13+BIT12>+BIT10 ;#9600 BAUD.
2312
2313 015260 000405          BR 61$
2314 015262 004537 022164    60$: PERFORM ,LOAD.MODE ;LOAD
2315 015266 034000          BIT13+BIT12+BIT11 ;MODE AND RECV ENABLE
2316 015270 004537 021706    PERFORM ,SETSNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
2317 015274 005277 164082    61$: INC @DVSCR ;SET MICRO CPU GO
2318 015300 105777 164056    TSTB @DVSCR ;WAIT FOR
2319 015304 100375          BPL -.4 ;DVSCR07=1
2320 015306 017704 164054    MOV @DVRIC,R4 ;READ RESULT
2321 015312 010005          MOV RO,R5 ;LOAD LINE NUMBER
2322 015314 000305          SWAB R5 ;PUT IN HIGH BYTE
2323 015316 052705 050000    BIS #BIT14+BIT12,R5 ;SET "BLOCK CHECK COMPLETE"
2324 015322 020504          CMP R5,R4 ;RIC OK
2325 015324 001401          BEQ 4$ ;
2326 015326 104001          HLT 1 ;DVRIC INCORRECT
2327 015330 112777 000014 164036 4$: MOVB #14,@DVSRSH ;GET TX MODE REGISTER
2328 015336 017704 164034    MOV @DVSRA,R4 ;
2329 015342 012705 000007    MOV #BIT2+BIT1+BIT0,R5 ;WAS NEXT MODE PICKED UP?
2330 015346 020504          CMP R5,R4 ;
    
```

```

2331 015350 001401          BEQ 5$ ;
2332 015352 104001          HLT 1 ;NEXT MODE INCORRECT/ S/B=7
2333 015354 105277 164014    5$: INCB @DVSRSH ;SEL RX MODE REG
2334 015360 017774 164012    MOV @DVSRA,R4 ;READ
2335 015364 020504          CMP R5,R4 ;
2336 015366 001401          BEQ 6$ ;
2337 015370 104001          HLT 1 ;RX MODE REGISTER INCORRECT. S/B=7
2338 015372 005005          CLR R5 ;SET EXPECTED=0
2339 015374 112777 000006 163772 MOVB #6,@DVSRSH ;SEL TX BCC REG
2340 015402 017704 163770    MOV @DVSRA,R4 ;READ
2341 015406 001401          BEQ 7$ ;BR IF=0
2342 015410 104001          HLT 1 ;IF BCC WAS SENT; BCC S/B=0
2343 015412 105277 163756    7$: INCB @DVSRSH ;SEL RX BCC REG
2344 015416 017704 163754    MOV @DVSRA,R4 ;READ IT
2345 015422 001401          BEQ 8$ ;
2346 015424 104001          HLT 1 ;IF RX RECVD GOOD BCC; BCC S/B=0
2347 015426 104413          RAMCLR ;CLEAR ALL SEC REG
2348 015430 104401          SCOP1 ;LOCK ON CURRENT LINE?
2349 015432 005200          INC R0 ;UPDATE LINE POINTER
2350 015434 005302          DEC R2 ;4 LINE GROUP DONE?
2351 015436 001226          BNE 65$ ;BR IF NO
2352 015440 000207          RTS PC ;EXIT FOR NEXT 4 LINE GROUP
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372 015442 012737 000014 001226 TST14: MOV #14,TSTNO
2373 015450 012737 016222 001216 MOV #TST15,NEXT
2374 015456 012700 000000    MOV #0.,RO ;PLACE LINE NUMBER INTO RO
2375 015462 013737 001416 001236 MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
2376 015470 100402          BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
2377 015472 004737 015560    JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 1
2378 015476 012700 000004    MOV #4.,RO ;PLACE LINE NUMBER INTO RO
2379 015502 013737 001420 001236 MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
2380 015510 100402          BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
2381 015512 004737 015560    JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 2
2382 015516 012700 000010    MOV #8.,RO ;LOAD LINE NUMBER
2383 015522 013737 001422 001236 MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
2384 015530 100402          BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED
2385 015532 004737 015560    JSR PC,105$ ;DO THE TEST FOR LINE CARD 3
2386 015536 012700 000014    MOV #12.,RO ;LOAD LINE NO.
    
```

```

2387 015542 013737 001424 001236 MOV L12.15,STAT ;LOAD LINE CARD STATUS
2388 015550 100402 BMI 103$ ;BR IF LINE CARD NOT TO BE TESTED
2389 015552 004737 015560 JSR PC,105$ ;DO THE TESTS FOR LINE CARD 4
2390 015556 104400 103$: SCOPE ;SCOPE THIS TEST.
2391 015560 105$: ;TEST ENTRANCE.
2392 015560 012737 015674 001220 MOV #65$,LOCK ;SET RETURN IF SW09=1
2393 015566 104413 RAMCLR ;CLEAR ALL SECONDARY REGISTERS
2394 015570 012702 000004 MOV #4,R2 ;SET FOR 4 LINE GROUP
2395 015574 012704 000012 MOV #10,R4 ;LOAD 10 BYTES
2396 015600 012705 023560 MOV #TXTAB,R5 ;WITH
2397 015604 112725 000010 1$: MOV #BIT3,(R5)+ ;INC/BCC
2398 015610 005304 DEC R4 ;
2399 015612 001374 BNE 1$ ;
2400 015614 012705 023560 MOV #TXTAB,R5 ;CLEAR
2401 015620 013704 001236 MOV STAT,R4 ;SYNC
2402 015624 042704 177400 BIC #<C<377>,R4 ;CONTROL
2403 015630 060405 ADD R4,R5 ;BYTE
2404 015632 105015 CLR (R5) ;
2405 015634 012705 022560 MOV #TXBAP,R5 ;LOAD
2406 015640 005004 CLR R4 ;DATA
2407 015642 110425 2$: MOV #R4,(R5)+ ;INTO
2408 015644 005204 INC R4 ;TRANSMITTER BUFFER
2409 015646 022704 000013 CMP #11,R4 ;
2410 015652 001373 BNE 2$ ;
2411 015654 012705 030160 MOV #RXTAB,R5 ;LOAD
2412 015660 012704 000012 MOV #10,R4 ;10
2413 015664 112725 000010 3$: MOV #BIT3,(R5)+ ;RECEIVER
2414 015670 005304 DEC R4 ;CONTROL BYTES
2415 015672 001374 BNE 3$ ;WITH "INC/BCC"
2416 015674 010077 163472 65$: MOV #0,DVSR5 ;LOAD LINE NO.
2417 015700 032737 004000 001236 BIT #ASYNCR,STAT ;IS THIS AN ASYNCR LINE CARD?
2418 015706 001406 BEQ 80$ ;BR IF NOT ASYNCR.
2419 015710 004537 022120 PERFORM ,SETREG ;ADJUST FOR ASYNCR LINE CARD
2420 015714 000 001 .BYTE 000,001 ;REGISTERS
2421 015716 022560 TXBAP ;LOAD FOR ASYNCR
2422 015720 077766 <-10.>-BIT15 ;/LOAD FOR ASYNCR
2423 015722 000405 BR 81$ ;/CONTINUE TEST
2424 015724 004537 022120 80$: PERFORM ,SETREG ;
2425 015730 000 001 .BYTE 000,001 ;TX PRINCIPLE BA, BC
2426 015732 022566 SYNC ;MARKED BCI
2427 015734 077764 <-12.>-BIT15 ;
2428 015736 004537 022120 81$: PERFORM ,SETREG ;RX BA, BC
2429 015742 004 005 .BYTE 004,005 ;
2430 015744 027510 RXBA ;
2431 015746 077766 <-10.>-BIT15 ;MARKED BCI
2432 015750 004537 022120 PERFORM ,SETREG ;
2433 015754 010 011 .BYTE 010,011 ;TX TABLE, RX TABLE
2434 015756 023560 TXTAB ;
2435 015760 030160 RXTAB ;
2436 015762 004537 022120 PERFORM ,SETREG ;LINE PROTOCOL, LINE STATE
2437 015766 012 013 .BYTE 012,013 ;
2438 015770 000001 BIT0 ;LRCC, IDLE MARK
2439 015772 162004 BIT15+BIT14+BIT13+BIT10+BIT2 ;
2440 015774 004537 022120 PERFORM ,SETREG ;MODE 7, TXGO
2441 016000 016 017 .BYTE 016,017 ;LINE PROGRESS REC, REC CNTR STORE
2442 016002 162000 BIT15+BIT14+BIT13+BIT10 ;NEXT MODE=7
    
```

```

2443 016004 000000 0 ;ZERO
2444 016006 032737 004000 001236 BIT #ASYNCR,STAT ;IS THIS ASYNCR LINE CARD?
2445 016014 001412 BEQ 60$ ;BR IF NO.
2446 016016 004537 022164 PERFORM ,LOAD.MODE ;LOAD PARAMETERS.
2447 016022 020000 BIT13 ;/RECEIVER ENABLE
2448 016024 004537 022164 PERFORM ,LOAD.MODE ;/
2449 016030 015000 <BIT12+BIT11>+BIT9 ;# BITS/PER/CHAR
2450 016032 004537 022164 PERFORM ,LOAD.MODE ;/
2451 016036 072000 <BIT14+BIT13+BIT12>+BIT10 ;#800 BAUD.
2452 ;
2453 016040 000405 BR 81$ ;
2454 016042 004537 022164 80$: PERFORM ,LOAD.MODE ;LOAD
2455 016046 034000 BIT13+BIT12+BIT11 ;MODE AND REC VENABLE
2456 016050 004537 021706 PERFORM ,SETSYNCR ;GET SYNCR CHARS AND ADJUST FOR ONE OR TWO.
2457 016054 005277 163302 81$: INC #DVSCR ;SET MICRO CPU GO
2458 016060 105777 163276 TST #DVSCR ;WAIT FOR
2459 016064 100375 BPL .-4 ;DVSCRO=1
2460 016066 017704 163274 MOV #DVRC,R4 ;READ RESULT
2461 016072 010005 MOV R0,R5 ;LOAD LINE NUMBER
2462 016074 000305 SWAB R5 ;PUT IN HIGH BYTE
2463 016076 052705 050000 BIS #BIT14+BIT12,R5 ;SET "BLOCK CHECK COMPLETE"
2464 016102 020504 CMP R5,R4 ;RIC OK
2465 016104 001401 BEQ 4$ ;
2466 016106 104001 HLT 1 ;DVRIC INCORRECT
2467 016110 112777 000014 163256 4$: MOV #14,DVSRSH ;GET TX MODE REGISTER
2468 016116 017704 163254 MOV #DVSR,R4 ;
2469 016122 012705 000007 MOV #BIT2+BIT1+BIT0,R5 ;WAS NEXT MODE PICKED UP?
2470 016126 020504 CMP R5,R4 ;
2471 016130 001401 BEQ 5$ ;
2472 016132 104001 HLT 1 ;NEXT MODE INCORRECT/ S/B=7
2473 016134 105277 163234 5$: INCB #DVSRSH ;SEL RX MODE REG
2474 016140 017704 163232 MOV #DVSR,R4 ;READ
2475 016144 020504 CMP R5,R4 ;
2476 016146 001401 BEQ 6$ ;
2477 016150 104001 HLT 1 ;RX MODE REGISTER INCORRECT. S/B=7
2478 016152 005005 CLR R5 ;SET EXPECTED=0
2479 016154 112777 000006 163212 6$: MOV #6,DVSRSH ;SEL TX BCC REG
2480 016162 017704 163210 MOV #DVSR,R4 ;READ
2481 016166 001401 BEQ 7$ ;BR IF=0
2482 016170 104001 HLT 1 ;IF BCC WAS SENT; BCC S/B=0
2483 016172 105277 163176 7$: INCB #DVSRSH ;SEL RX BCC REG
2484 016176 017704 163174 MOV #DVSR,R4 ;READ IT
2485 016202 001401 BEQ 8$ ;
2486 016204 104001 HLT 1 ;IF RX RECVD GOOD BCC; BCC S/B=0
2487 016206 104413 RAMCLR ;CLEAR ALL SEC REG
2488 016210 104401 SCOP1 ;LOCK ON CURRENT LINE?
2489 016212 005200 INC R0 ;UPDATE LINE POINTER
2490 016214 005302 DEC R2 ;4 LINE GROUP DONE?
2491 016216 001226 BNE 65$ ;BR IF NO
2492 016220 000207 RTS PC ;EXIT FOR NEXT 4 LINE GROUP
2493 ;
2494 ;
2495 ;***** TEST 15 *****
2496 ;*TEST OF RECIEVER AND TRANSMITTER MODE BITS.
2497 ;*TEST TO TRANSMIT AND RECEIVE
2498 ;*A DIFFERENT CHAR FROM EACH
    
```

```

2499          ;*MODE. THE TX TABLE WILL BE
2500          ;*FILLED WITH "SEND DLE" SO IF CHAR
2501          ;*GOES TO WRONG TABLE RX WILL
2502          ;*RECEIVE A DLE CHAR(31). THE RX
2503          ;*FILLS TABLE WITH "INCLUDE IN BCC"
2504          ;*SO THAT IF RECV GOES TO WRONG
2505          ;*TABLE THE RX BCC REG WILL
2506          ;*BE NON-ZERO!
2507          ;*CHAR    CURRENT MODE    NEXT MODE
2508          ;* 15      0              1
2509          ;* 16      1              2
2510          ;* 21      2              3
2511          ;* 23      3              4
2512          ;* 25      4              5
2513          ;* 7       5              6
2514          ;* 34      6              7
2515          ;* 32      7              7
2516          ;* 36      7              7
2517          ;*
2518          ;*
2519          ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
2520          ;*****
2521          ;
2522          ; TEST 15
2523          ;-----
2524 016222 012737 000015 001226 TST15: MOV #15,TSTNO
2525 016230 012737 017166 001216 MOV #TST16,NEXT
2526 016236 012700 000000 MOV #0.,R0 ;PLACE LINE NUMBER INTO R0
2527 016242 013757 001416 001236 MOV L00.03,STAT ;LOAD LINE CARD STATUS INTO STAT
2528 016250 100402 BMI 100$ ;BR IF LINE CARD NOT TO BE TESTED
2529 016252 004737 016340 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 1
2530 016256 012700 000004 100$: MOV #4.,R0 ;PLACE LINE NUMBER INTO R0
2531 016262 013737 0001420 001236 MOV L04.07,STAT ;LOAD LINE CARD STATUS INTO STAT
2532 016270 100402 BMI 101$ ;BR IF LINE CARD NOT TO BE TESTED
2533 016272 004737 016340 JSR PC,105$ ;GO DO THE TEST FOR LINE CARD 2
2534 016276 012700 000010 101$: MOV #8.,R0 ;LOAD LINE NUMBER
2535 016302 013737 0001422 001236 MOV L08.11,STAT ;LOAD LINE CARD STATUS INTO STAT
2536 016310 100402 BMI 102$ ;BR IF LINE CARD NOT TO BE TESTED
2537 016312 004737 016340 JSR PC,105$ ;DO THE TEST FOR LINE CARD 3
2538 016316 012700 000014 102$: MOV #12.,R0 ;LOAD LINE NO.
2539 016322 013737 0001424 001236 MOV L12.15,STAT ;LOAD LINE CARD STATUS
2540 016330 100402 BMI 103$ ;BR IF LINE CARD NOT TO BE TESTED
2541 016332 004737 016340 JSR PC,105$ ;DO THE TESTS FOR LINE CARD 4
2542 016336 104400 103$: SCOPE ;SCOPE THIS TEST.
2543 016340 105$: ;TEST ENTRANCE.
2544 016340 012737 016604 001220 MOV #12$,LOCK ;LOCK ON LINE RETURN
2545 016346 104413 RAMCLR ;CLEAR ALL SEC REGISTERS
2546 016350 012705 023560 MOV #TXTAB,R5 ;LOAD
2547 016354 012704 030160 MOV #RXTAB,R4 ;
2548 016360 012701 004000 MOV #4000,R1 ;ALL CNTRL BYTES
2549 016364 112725 000002 15: MOV #BIT1,(R5)+ ;WITH "SND/DLE"
2550 016370 112724 000010 25: MOV #BIT3,(R4)+ ;WITH "INCL/BCC"
2551 016374 005301 DEC R1 ;
2552 016376 001372 BNE 1$ ;
2553 016400 004537 021706 PERFORM ,SETSYNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
2554 016404 012702 000004 11$: MOV #4,R2 ;SET FOR 4 LINE GROUP
    
```

```

2555 016410 113705 001236 MOV #STAT,R5 ;CLEAR
2556 016414 042705 177400 BIC #~C<377>,R5 ;SYNC
2557 016420 012704 023560 MOV #TXTAB,R4 ;ENTRY
2558 016424 060504 ADD R5,R4 ;IN
2559 016426 105014 CLRB (R4) ;CONTROL TABLE
2560 016430 112737 000040 023575 MOV #BIT5, TXTAB+15
2561 016436 112737 000100 024176 MOV #BIT6, TXTAB+BIT8+16
2562 016444 112737 000140 024601 MOV #BIT6+BIT5, TXTAB+BIT9+21
2563 016452 112737 000200 025203 MOV #BIT7, TXTAB+BIT9+BIT8+23
2564 016460 112737 000240 025605 MOV #BIT7+BIT5, TXTAB+BIT10+25
2565 016466 112737 000300 026167 MOV #BIT7+BIT6, TXTAB+BIT10+BIT8+7
2566 016474 112737 000340 026614 MOV #BIT7+BIT6+BIT5, TXTAB+BIT10+BIT9+34
2567 016502 112737 000340 027212 MOV #BIT7+BIT6+BIT5, TXTAB+BIT10+BIT9+BIT8+32
2568 016510 112737 000340 027216 MOV #BIT7+BIT6+BIT5, TXTAB+BIT10+BIT9+BIT8+36
2569
2570 016516 112737 000040 030175 MOV #BIT5, RXTAB+15
2571 016524 112737 000100 030576 MOV #BIT6, RXTAB+BIT8+16
2572 016532 112737 000140 031201 MOV #BIT6+BIT5, RXTAB+BIT9+21
2573 016540 112737 000200 031603 MOV #BIT7, RXTAB+BIT9+BIT8+23
2574 016546 112737 000240 032205 MOV #BIT7+BIT5, RXTAB+BIT10+25
2575 016554 112737 000300 032567 MOV #BIT7+BIT6, RXTAB+BIT10+BIT8+7
2576 016562 112737 000340 033214 MOV #BIT7+BIT6+BIT5, RXTAB+BIT10+BIT9+34
2577 016570 112737 000340 033612 MOV #BIT7+BIT6+BIT5, RXTAB+BIT10+BIT9+BIT8+32
2578 016576 112737 000340 033616 MOV #BIT7+BIT6+BIT5, RXTAB+BIT10+BIT9+BIT8+36
2579 016604 012705 027560 12$: MOV #RXBA,R5 ;SET RX POINTER
2580 016610 005025 CLR (R5)+ ;Z
2581 016612 005025 CLR (R5)+ ;E
2582 016614 005025 CLR (R5)+ ;R
2583 016616 005025 CLR (R5)+ ;D
2584 016620 005025 CLR (R5)+ ;BUFFER!
2585 016622 012705 022560 MOV #TXBAP,R5 ;L
2586 016626 012725 MOV (PC)+,(R5)+ ;O
2587 016630 015 016 .BYTE 15,16 ;A
2588 016632 012725 MOV (PC)+,(R5)+ ;D
2589 016634 021 023 .BYTE 21,23 ;T
2590 016636 012725 MOV (PC)+,(R5)+ ;R
2591 016640 025 007 .BYTE 25,7 ;A
2592 016642 012725 MOV (PC)+,(R5)+ ;N
2593 016644 034 032 .BYTE 34,32 ;S
2594 016646 112725 000036 MOV #36,(R5)+ ;BUFFER
2595 016652 010077 162514 MOV R0,@DVSRS ;LOAD LINE NO.
2596 016656 032737 004000 001236 BIT #ASYNC,STAT ;IS THIS AN ASYNC LINE CARD?
2597 016664 001406 BEQ 80$ ;#BR IF NOT ASYNC.
2598 016666 004537 022120 PERFORM ,SETREG ;ADJUST FOR ASYNC LINE CARD
2599 016672 000 001 .BYTE 000,001 ;#REGISTERS
2600 016674 022560 TXBAP ;#LOAD FOR ASYNC
2601 016676 177767 -9. ;#LOAD FOR ASYNC
2602 016700 000405 BR 81$ ;#CONTINUE TEST
2603 016702 004537 022120 80$: PERFORM ,SETREG ;
2604 016706 000 001 .BYTE 000,001 ;PRINCIPLE BA, BC
2605 016710 025566 SYNC ;
2606 016712 177765 -11. ;
2607 016714 004537 022120 81$: PERFORM ,SETREG ;
2608 016720 004 005 .BYTE 004,005 ;RX BA, BC
2609 016722 027560 RXBA ;
2610 016724 177767 -9. ;
    
```

```

2611 016726 004537 022120      PERFORM ,SETREG      ;
2612 016732 010      .BYTE 010,011      ;TX TABLE, RX TAB
2613 016734 023560      TXTAB                ;
2614 016736 030160      RXTAB                ;
2615 016740 004537 022120      PERFORM ,SETREG      ;
2616 016744 012      .BYTE 012,013      ;LINE PROTOCOL, LINE STATE
2617 016746 014400      31*400              ;31 IN HIGH BYTE
2618 016750 000004      BIT2                 ;TX GO
2619 016752 032737 004000 001236  BIT      #ASYNC,STAT  ;#IS THIS ASYNC LINE CARD?
2620 016760 001412      SEQ 60$              ;#BR IF NO.
2621 016762 004537 022164      PERFORM ,LOAD.MODE  ;#LOAD PARAMETERS.
2622 016766 020000      BIT13                ;#RECEIVER ENABLE
2623 016770 004537 022164      PERFORM ,LOAD.MODE  ;#
2624 016774 015000      <BIT12+BIT11>+BIT9  ;#B BITS/PER/CHAR
2625 016776 004537 022164      PERFORM ,LOAD.MODE  ;#
2626 017002 072000      <BIT14+BIT13+BIT12>+BIT10 ;#9600 BAUD.
2627
2628 017004 000403      BR 61$              ;
2629 017006 004537 022164      60$: PERFORM ,LOAD.MODE ;LOAD
2630 017012 034000      BIT13+BIT12+BIT11  ;MODE AND RX ENABLE
2631 017014 005277 162342      61$: INC @DVSCR        ;SET MICRO CPU GO
2632 017020 105777 162336      TSTB @DVSCR        ;WAIT FOR
2633 017024 100375      BPL ,-4              ;DVSCR0=1
2634 017026 012701 022560      MOV #TXBAP,R1       ;SET TX POINTER
2635 017032 012703 027560      MOV #RXBA,R3        ;SET RX POINTER
2636 017036 012737 000011 001248  MOV #9.,TEMP1       ;CHECK 9. CHAR
2637 017044 005005      CLR R5              ;
2638 017046 005004      CLR R4              ;
2639 017050 112105      3$: MOVB (R1)+,R5     ;SET EXPECTED
2640 017052 112304      MOVB (R3)+,R4       ;SET FOUND
2641 017054 010504      CMP R5,R4           ;GOOD?
2642 017056 001401      BEQ 4$              ;
2643 017060 104001      HLT 1               ;
2644 017062 005337 001246      4$: DEC TEMP1       ;DATA COMPARE ERROR-(IS IT IDLE)?
2645 017066 001370      BNE 3$              ;ALL CHARS DONE?
2646 017070 005005      CLR R5              ;BR IF NO
2647 017072 112777 000007 162274  MOVB #7,@DVSRSH     ;SEL RX BCC REG
2648 017100 017704 162272      MOV @DVSR,R4        ;READ IT
2649 017104 001401      SEQ 5$              ;IF RX WENT TO GOOD CNTRL BYTE;
2650 017106 104001      HLT 1               ;RX BCC S/B=0
2651 017110 012705 000007      MOV #7,R5           ;SET MODE=0
2652 017114 112777 000014 162252  MOVB #14,@DVSRSH    ;SEL TX MODE REG
2653 017122 017704 162250      MOV @DVSR,R4        ;READ TX MODE REG
2654 017126 020504      CMP R5,R4           ;
2655 017130 001401      BEQ 6$              ;
2656 017132 104001      HLT 1               ;TX MODE NOT=7?
2657 017134 105277 162234      6$: INCB @DVSRSH     ;SEL RX MODE REG
2658 017140 017704 162232      MOV @DVSR,R4        ;READ IT
2659 017144 020504      CMP R5,R4           ;
2660 017146 001401      BEQ 7$              ;
2661 017150 104001      HLT 1               ;RX MODE NOT=7?
2662 017152 104412      7$: MSTCLR          ;INIT DV11
2663 017154 104401      SCOP1              ;LOCK ON CURRENT LINE.
2664 017156 005200      INC R0              ;INC LINE POINTER
2665 017160 005302      DEC R2              ;4 LINE GROUP DONE?
2666 017162 001210      BNE 12$            ;BR IF NO
    
```

```

2667 017164 000207      RTS PC              ;EXIT FOR NEXT GROUP OF LINES
2668
2669
2670      ;***** TEST 16 *****
2671      ;*TEST OF RECEIVER AND TRANSMITTER MULTIPLE FUNCTIONS.
2672      ;*TEST OF RECV BCC AND TRANS BCC.
2673      ;*CHAR      RX FUNC.      TX FUNC.
2674      ;* 0      INC/BCC      INC/BCC
2675      ;* 1      INC/BCC/DSCARD  INC/BCC
2676      ;* 2      INC/BCC      INC/BCC/SND/DLE
2677      ;* 3      INC/BCC      INC/BCC
2678      ;* 4      NO FUNC      SND/DLE
2679      ;* 5      INC/BCC/DSCARD  INC/BCC
2680      ;* 6      INC/BCC/EXP/BCC  INC/BCC/SND/BCC
2681      ;*
2682      ;*      NEXT MODE =7      NEXT MODE =7
2683      ;*
2684      ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
2685      ;*****
2686
2687      ; TEST 16
2688      ;-----
2689      TST16: MOV #16,TSTNO
2690      MOV #TST17,NEXT
2691      MOV #0.,R0
2692      MOV L00.03,STAT ;PLACE LINE NUMBER INTO R0
2693      BMI 100$        ;LOAD LINE CARD STATUS INTO STAT
2694      JSR PC,105$     ;BR IF LINE CARD NOT TO BE TESTED
2695      MOV #4.,R0     ;GO DO THE TEST FOR LINE CARD 1
2696      MOV L04.07,STAT ;PLACE LINE NUMBER INTO R0
2697      BMI 101$        ;LOAD LINE CARD STATUS INTO STAT
2698      JSR PC,105$     ;BR IF LINE CARD NOT TO BE TESTED
2699      MOV #8.,R0     ;GO DO THE TEST FOR LINE CARD 2
2700      MOV L08.11,STAT ;LOAD LINE NUMBER
2701      BMI 102$        ;LOAD LINE CARD STATUS INTO STAT
2702      JSR PC,105$     ;BR IF LINE CARD NOT TO BE TESTED
2703      MOV #12.,R0    ;DO THE TEST FOR LINE CARD 3
2704      MOV L12.15,STAT ;LOAD LINE NO.
2705      BMI 103$        ;LOAD LINE CARD STATUS
2706      JSR PC,105$     ;BR IF LINE CARD NOT TO BE TESTED
2707      MOV #13.,R0    ;DO THE TESTS FOR LINE CARD 4
2708      MOV L13.17,STAT ;SCOPE THIS TEST.
2709      BMI 104$        ;SCOPE THIS TEST.
2710      JSR SCOPE
2711      MOV #3$,LOCK    ;RETURN IF SW09=1
2712      BIT #BIT9+BIT8,STAT ;#8 BITS/PER/CHAR ?"
2713      BEQ .+4         ;BR IF YES
2714      PC             ;EXIT TEST FOR THIS LINE CARD!
2715      RAMCLR        ;CLEAR ALL SEC REGISTERS
2716      MOV #RXTAB,R5 ;CLEAR
2717      MOV #RXTAB,R3 ;TRANSMITTER
2718      CLR R4         ;AND
2719      CLR (R5)+      ;RECEIVER
2720      CLR (R3)+      ;CONTROL
2721      INCB R4        ;TABLES
2722      BPL 1$         ;
2723      MOV #BIT3,R5   ;INC/BCC IS IN R5
2724      MOV R5,TXTAB  ;INC/BCC
2725      MOV R5,TXTAB+1 ;INC/BCC
2726      MOV R5,TXTAB+2 ;INC/BCC
    
```

2723	017370	152737	000002	023562	BISB	#BIT1,XTAB+2	; SND/DLE
2724	017376	110537	023563		MOV8	R5,XTAB+3	;INC/BCC
2725	017402	112737	000002	023564	MOV8	#BIT1,XTAB+4	;SND/DLE
2726	017410	110537	023565		MOV8	R5,XTAB+5	;INC/BCC
2727	017414	110537	023566		MOV8	R5,XTAB+6	;INC/BCC
2728	017420	052737	000344	023566	BIS	#BIT7+BIT6+BIT5+BIT2,XTAB+6	;INC/BCC SND/BCC MOD=7
2729	017426	110537	030160		MOV8	R5,XTAB	;INC/BCC
2730	017432	110537	030161		MOV8	R5,XTAB+1	;INC/BCC
2731	017436	152737	000020	030161	BISB	#BIT4,RXTAB+1	; DSCARD
2732	017444	110537	030162		MOV8	R5,XTAB+2	;INC/BCC
2733	017450	110537	030163		MOV8	R5,XTAB+3	;INC/BCC
2734	017454	105037	030164		CLRB	RXTAB+4	;NO FUNC.
2735	017460	110537	030165		MOV8	R5,XTAB+5	;INC/BCC
2736	017464	152737	000020	030165	BISB	#BIT4,RXTAB+5	; DSCARD.
2737	017472	112737	000354	030166	MOV8	#BIT7+BIT6+BIT5+BIT3+BIT2,RXTAB+6	;INC/BCC EXP/BCC MODE=7
2738	017500	012702	000004		MOV	#4,R2	;SET FOR 4 LINE GROUP
2739	017504	005037	027560	3\$:	CLR	RXBA	;ZERO
2740	017510	005037	027562		CLR	RXBA+2	;RX
2741	017514	005037	027564		CLR	RXBA+4	;BUFFER
2742	017520	005037	027566		CLR	RXBA+6	;AREA
2743	017524	010077	161642		MOV	R0,@DVSRS	;LOAD LINE NO.
2744	017530	032737	004000	001236	BIT	#ASYNC,STAT	;IS THIS AN ASYNC LINE CARD?
2745	017536	001406			BEQ	80\$;#BR IF NOT ASYNC.
2746	017540	004537	022120		PERFORM	,SETREG	;#ADJUST FOR ASYNC LINE CARD
2747	017544	000	001		.BYTE	000,001	;#REGISTERS
2748	017546	022560			TXBAP		;#LOAD FOR ASYNC
2749	017550	177771			-.		;#LOAD FOR ASYNC
2750	017552	000405			BR	81\$;#CONTINUE TEST
2751	017554	004537	022120	80\$:	PERFORM	,SETREG	
2752	017560	000	001		.BYTE	000,001	;PRINCIPLE BA, BC
2753	017562	022556			SYNC		
2754	017564	177767			-9.		
2755	017566	004537	022120	81\$:	PERFORM	,SETREG	
2756	017572	004	005		.BYTE	004,005	;RX BA, BC
2757	017574	027560			RXBA		
2758	017576	177766			-10.		
2759	017600	004537	022120		PERFORM	,SETREG	
2760	017604	010	011		.BYTE	010,011	;TX TAB, RXTAB
2761	017606	023560			TXTAB		
2762	017610	030160			RXTAB		
2763	017612	004537	022120		PERFORM	,SETREG	
2764	017616	013	012		.BYTE	013,012	;LINE STATE, LINE PROTOCOL
2765	017620	000004			BIT2		;TX GO
2766	017622	010051			<20*400>+BIT4+BIT3+BIT0		;DLE(20 HIGH BYTE),CRC.CCITT, IDLE MARK
2767	017624	032737	004000	001236	BIT	#ASYNC,STAT	;IS THIS ASYNC LINE CARD?
2768	017632	001412			BEQ	60\$;#BR IF NO.
2769	017634	004537	022164		PERFORM	,LOAD.MODE	;#LOAD PARAMETERS.
2770	017640	020000			BIT13		;#RECEIVER ENABLE
2771	017642	004537	022164		PERFORM	,LOAD.MODE	
2772	017646	015000			<BIT12+BIT11>+BIT9		;#8 BITS/PER/CHAR
2773	017650	004537	022164		PERFORM	,LOAD.MODE	
2774	017654	072000			<BIT14+BIT13+BIT12>+BIT10		;#9600 BAUD.
2775							
2776	017656	000405			BR	4\$	
2777	017660	004537	022164	60\$:	PERFORM	,LOAD.MODE	;LOAD
2778	017664	034000			BIT13+BIT12+BIT11		;MODE AND RX ENABLE

2779	017666	004537	021706		PERFORM	,SETSYNC	;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
2780	017672	005004		4\$:	CLR	R4	;LOAD
2781	017674	012705	022560		MOV	#TXBAP,R5	;TX
2782	017700	110425		5\$:	MOV8	R4,(R5)+	;DATA
2783	017702	005204			INC	R4	
2784	017704	020427	000007		CMP	R4,#7	
2785	017710	001373			BNE	5\$	
2786	017712	005277	161444		INC	@DVSCR	;SET MICRO CODE GO
2787	017716	105777	161440		TSTB	@DVSCR	;WAIT FOR
2788	017722	100375			BPL	-	;DVSCRO7=1
2789	017724	012701	027560		MOV	#RXBA,R1	;GET RX POINTER
2790	017730	012703	020134		MOV	#50\$,R3	;GET DATA EXPECTED POINTER
2791	017734	012737	000007	001252	MOV	#7.,TEMP3	;CHECK 7 CHARS
2792	017742	112104		6\$:	MOV8	(R1)+,R4	;GET RECEIVED CHAR
2793	017744	112305			MOV8	(R3)+,R5	;GET EXPECTED CHAR
2794	017746	020504			CMP	R5,R4	
2795	017750	001401			BEQ	7\$;YES
2796	017752	104001			HLT	1	;DATA IS WRONG!!
2797	017754	005337	001252	7\$:	DEC	TEMP3	;ALL CHARS DONE?
2798	017760	001370			BNE	6\$;BR IF NO
2799	017762	112777	000014	161404	MOV8	#14,@DVSRSR	;GET TX MODE REG.
2800	017770	017704	161402		MOV	@DVSRA,R4	
2801	017774	042704	177770		BIC	#^C<BIT2+BIT1+BIT0>,R4	;CLEAR JUNK
2802	020000	012703	000007		MOV	#7,R5	;SET EXPECTED=7
2803	020004	020504			CMP	R5,R4	
2804	020006	001401			BEQ	8\$	
2805	020010	104001			HLT	1	;TX MODE REG NOT=7
2806	020012	105277	161356	8\$:	INCB	@DVSRSR	;RX MODE REG
2807	020016	017704	161354		MOV	@DVSRA,R4	
2808	020022	042704	177770		BIC	#^C<BIT2+BIT1+BIT0>,R4	
2809	020026	010504			CMP	R5,R4	
2810	020030	001401			BEQ	9\$	
2811	020032	104001			HLT	1	;RX MODE REG NOT=7
2812	020034	112777	000006	161332	MOV8	#6,@DVSRSR	;TX BCC REG
2813	020042	017704	161330		MOV	@DVSRA,R4	
2814	020046	001402			BEQ	10\$	
2815	020050	005005			CLR	R5	
2816	020052	104001			HLT	1	;TX BCC REG S/B=0
2817	020054	105277	161314	10\$:	INCB	@DVSRSR	;TXBCC
2818	020060	017704	161312		MOV	@DVSRA,R4	
2819	020064	001402			BEQ	11\$	
2820	020066	005005			CLR	R5	
2821	020070	104001			HLT	1	
2822	020072	010005		11\$:	MOV	R0,R5	;RX BCC REG S/B=0
2823	020074	000305			SWAB	R5	;LOAD LINE NO.
2824	020076	052705	050000		BIS	#BIT14+BIT12,R5	;PUT IN HIGH BYTE
2825	020102	017704	161260		MOV	@DVRIC,R4	;SET BCC COMPLETE
2826	020106	020504			CMP	R5,R4	;READ RIC
2827	020110	001401			BEQ	12\$	
2828	020112	104001			HLT	1	
2829	020114	104413		12\$:	RAMCLR		;DVRIC INCORRECT
2830	020116	104401			SCOPI		;CLEAR ALL SEC REGS
2831	020120	005200			INC	R0	;RETURN WITH SAME LINE
2832	020122	005302			DEC	R2	;UPDATE LINE POINTER
2833	020124	001402			BEQ	.+6	;4 LINES DONE?
2834	020126	000137	017504		JMP	3\$;BR IF NO

```

2835 020132 000207          RTS      PC          ;EXIT
2836 020134      000      020    50S:  .BYTE  0,20
2837 020136      002      003      .BYTE  2,3
2838 020140      020      004      .BYTE 20,4
2839 020142      006      000      .BYTE  6,0
2840
2841
2842 ;***** TEST 17 *****
2843 ;*TEST OF RECEIVER RESYNC
2844 ;*TEST TO TRANSMIT A BLOCK OD
2845 ;*DATA (SYN,SYN,1,2,3,4,5)
2846 ;*HAVING CHAR "1" BEING A "SPECIAL CHAR" TO THE RECEIVER
2847 ;*AT WHICH TIME A "RE-SYNC" PULSE WILL BE ISSUED
2848 ;*AND A RESTART CHAR PROC. (DVSCROB=1) WILL BE DONE.
2849 ;*WHEN THE TRANSMITTER IS DONE (DVSCR15=1) A SECOND
2850 ;*BLOCK OF DATA (SYN,SYN,SYN,SYN,6,7,10)
2851 ;*WILL BE SENT EXPECTING THAT THE NEXT TIME DVSCRO7=1
2852 ;*THAT THE DVRC WILL HAVE!
2853 ;*14=1 11:08=LINE NO. 07:00= "10"
2854 ;*RXBUFFER (CDRE) S/B= 1,6,7,10.
2855 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
2856 ;*****
2857
2858 ; TEST 17
2859 020144 012737 000017 001226 TST17: MOV #17,TSTND
2860 020152 012737 021142 001216 MOV #TST20,NEXT
2861 020160 012700 000000 MOV #0.,RO
2862 020164 013737 001416 001236 MOV L00.03,STAT
2863 020172 100402 BMI 100S
2864 020174 004737 020282 JSR PC,105S
2865 020200 012700 000004 100S: MOV #4.,RO
2866 020204 013737 001420 001236 MOV L04.07,STAT
2867 020212 100402 BMI 101S
2868 020214 004737 020272 JSR PC,105S
2869 020220 012700 000010 101S: MOV #8.,RO
2870 020224 013737 001422 001236 MOV L08.11,STAT
2871 020232 100402 BMI 102S
2872 020234 004737 020282 JSR PC,105S
2873 020240 012700 000014 102S: MOV #12.,RO
2874 020244 013737 001424 001236 MOV L12.15,STAT
2875 020252 100402 BMI 103S
2876 020254 004737 020282 JSR PC,105S
2877 020260 104400 103S: SCOPE
2878 020262 105S: ;TEST ENTRANCE.
2879 020262 012737 020314 001220 MOV #1$,LOCK
2880 020270 104413 RAMCLR ;CLEAR ALL SEC. REGS
2881 020272 112737 000001 030181 MOVB #BIT0,RXTAB+1
2882 020300 005037 030186 CLR RXTAB+6
2883 020304 005037 030170 CLR RXTAB+10
2884 020310 012702 000004 MOV #4,R2
2885 020314 010077 161082 1S: MOV RO,@DVSR
2886 020320 032737 004000 001236 BIT #ASYN,STAT
2887 020326 001406 BEQ 80S
2888 020330 004537 022120 PERFORM ,SETREG
2889 020334 000 001 .BYTE 000,001
2890 020336 022560 TXBAP ;LOAD FOR ASYNC
    
```

```

2891 020340 177773          -5
2892 020342 000405          BR      B1$
2893 020344 004537 022120 80S: PERFORM ,SETREG
2894 020350 000 001 .BYTE 000,001
2895 020352 022566          SINC
2896 020354 177771          -7
2897 020356 004537 022120 81S: PERFORM ,SETREG
2898 020362 004 005 .BYTE 004,005
2899 020364 027560          RXBA
2900 020366 177774          -4
2901 020370 004537 022120 PERFORM ,SETREG
2902 020374 010 011 .BYTE 010,011
2903 020376 023560          TXTAB
2904 020400 030160          RXTAB
2905 020402 004537 022120 PERFORM ,SETREG
2906 020406 013 012 .BYTE 013,012
2907 020410 000004          BIT2
2908 020412 000101          BIT6+BIT0
2909 020414 032737 004000 001236 BIT #ASYN,STAT
2910 020422 001412          BEQ 60S
2911 020424 004537 022164 PERFORM ,LOAD.MODE
2912 020430 020000          BIT13
2913 020432 004537 022164 PERFORM ,LOAD.MODE
2914 020436 015000          <BIT12+BIT11>+BIT9
2915 020440 004537 022164 PERFORM ,LOAD.MODE
2916 020444 072000          <BIT14+BIT13+BIT12>+BIT10
2917
2918 020446 000403          BR      61$
2919 020450 004537 022164 60S: PERFORM ,LOAD.MODE
2920 020454 034000          BIT13+BIT12+BIT11
2921 020456 005037 027580 61S: CLR RXBA
2922 020462 005037 027582 CLR RXBA+2
2923 020466 012705 022560 MOV #TXBAP,R5
2924 020472 005004          CLR R4
2925 020474 005204          INC R4
2926 020476 110425          MOVB #4,(R5)+
2927 020500 022704 000005 CMP #5,R4
2928 020504 001373          BNE 2S
2929 020506 004537 021708 PERFORM ,SETSYNC
2930 020512 005277 160844 INC RS
2931 020516 005005          CLR RS
2932 020520 105777 160836 TSTB @DVSCR
2933 020524 100404          BMI .+12
2934 020526 104414          DELAY
2935 020530 005205          INC R5
2936 020532 001372          BNE .-12
2937 020534 104000          HLT
2938 020536 005005          CLR R5
2939 020540 005777 160816 TST @DVSCR
2940 020544 100404          BMI .+12
2941 020546 104414          DELAY
2942 020550 005205          INC R5
2943 020552 001372          BNE .-12
2944 020554 104000          HLT
2945 020556 012705 022560 MOV #TXBAP,R5
2946 020562 113725 001236 MOVB STAT,(R5)+
    
```

```

2947 020566 113725 001236      MOV# STAT,(R5)+      ;SYNC
2948 020572 012704 000006      MOV #6,R4            ;SET 1ST DATA TO 6
2949 020576 110425                3$: MOV# R4,(R5)+      ;LOAD
2950 020600 005204                INC R4               ;DATA
2951 020602 022704                CMP #11,R4          ;ALL DONE?
2952 020606 001373                BNE 3$              ;BR IF NO
2953 020610 032737 004000 001236  BIT #ASYNC,STAT     ;#IS THIS AN ASYNC LINE CARD?
2954 020616 001406                BEQ 82$             ;#BR IF NOT ASYNC.
2955 020620 004537 022120        PERFORM ,SETREG     ;#ADJUST FOR ASYNC LINE CARD
2956 020624 000 001             .BYTE 000,001      ;#REGISTERS
2957 020626 022562                TXBAP+2             ;#LOAD FOR ASYNC
2958 020630 177775                -3                  ;#LOAD FOR ASYNC
2959 020632 000405                BR 83$              ;#CONTINUE TEST
2960 020634 004537 022120        82$: PERFORM ,SETREG
2961 020640 000 001             .BYTE 000,001      ;TX BA P, TX BC P
2962 020642 022556                SYNC                ;
2963 020644 177771                -7                  ;
2964 020646 032737 004000 001236 83$: BIT #ASYNC,STAT     ;#ASYNC LINE CARD?
2965 020654 001403                BEQ .+10            ;#BR IF NO
2966 020656 004537 022164        PERFORM ,LOAD.MODE ;#CLEAR RX ENABLE
2967 020662 000000                0                   ;#
2968 020664 112777 000013 160502  MOV# #13,@DVSRSH    ;LINE STATE
2969 020672 042777 000200 160476  BIC #BIT7,@DVSRSH  ;CLEAR "USE SEC TABLES"
2970 020700 052777 000002 160470  BIS #BIT1,@DVSRSH  ;SET RE-SYNC
2971 020706 112777 000012 160460  MOV# #12,@DVSRSH    ;SEL LINE PROTOCOL PARAM.
2972 020714 052777 000002 160454  BIS #BIT1,@DVSRSH  ;SET STRIP LEADING SYNC
2973 020722 012737 006000 020732  MOV #6000,84$      ;GIVE UCPU TIME
2974 020730 005327                DEC (PC)+           ;TO RESYNC SILO
2975 020732 000000                84$: 0              ;
2976 020734 001375                BNE -.4             ;
2977 020736 027377 004000 001236  BIT #ASYNC,STAT     ;#ASYNC LINE CARD?
2978 020744 001403                BEQ .+10            ;#BR IF NOT ASYNC LINE CARD.
2979 020746 004537 022164        PERFORM ,LOAD.MODE ;#SET RX ENABLE FOR ASYNC LINE CARD
2980 020752 020000                BIT13               ;#RX ENABLE
2981 020754 112777 000013 160412  MOV# #13,@DVSRSH    ;SEL LINE STATE.
2982 020762 052777 000004 160406  BIS #BIT2,@DVSRSH  ;SET TX GD.
2983 020770 052777 000400 160364  BIS #BIT8,@DVSCR    ;RESTART CPU
2984 020776 005004                CLR R4              ;SET FOR TIME OUT.
2985 021000 105777 160356        TSTB @DVSCR         ;RX DONE?
2986 021004 100404                BMI -.12            ;BR IF YES
2987 021006 104414                DELAY               ;WASTE TIME
2988 021010 005204                INC R4              ;LOOP DONE?
2989 021012 001372                BNE -.12            ;BR IF NO
2990 021014 104000                HLT                 ;DVSCRO7 NOT SET AFTER RESYNC.
2991 021016 017704 160344        MOV @DVRIC,R4       ;READ DVRIC
2992 021022 010005                MOV R0,R5           ;LOAD LINE NO
2993 021024 000305                SWAB R5             ;PLACE IN HIGH BYTE
2994 021026 052705 040010        BIS #BIT14+10,R5    ;SET BC WARNING + CHAR 10
2995 021032 020504                CMP R5,R4           ;RIC OK
2996 021034 001401                BEQ 4$              ;
2997 021036 104001                HLT 1               ;DVRIC WRONG
2998 021040 012703 027560        4$: MOV #RXBA,R3     ;CHECK RX DATA
2999 021044 112304                MOV# (R3)+,R4       ;
3000 021046 012705 000001        MOV #1,R5           ;
3001 021052 020504                CMP R5,R4           ;
3002 021054 001401                BEQ 5$              ;
    
```

```

3003 021056 104001                HLT 1               ;1ST CHAR NOT "1"!
3004 021060 112304                5$: MOV# (R3)+,R4    ;
3005 021062 012705 000006        MOV #6,R5           ;
3006 021066 020504                CMP R5,R4           ;
3007 021070 001401                BEQ 6$              ;
3008 021072 104001                HLT 1               ;2ND CHAR NOT "6"!
3009 021074 112304                6$: MOV# (R3)+,R4    ;
3010 021076 012705 000007        MOV #7,R5           ;
3011 021102 020504                CMP R5,R4           ;
3012 021104 001401                BEQ 7$              ;
3013 021106 104001                HLT 1               ;3RD CHAR NOT "7"!
3014 021110 005205                7$: INC R5          ;
3015 021112 112304                MOV# (R3)+,R4       ;
3016 021114 020504                CMP R5,R4           ;
3017 021116 001401                BEQ 8$              ;
3018 021120 104001                HLT 1               ;4TH CHAR NOT "10"!
3019 021122 104412                8$: MSTCLR          ;RESET DV11
3020 021124 104401                SCOP1               ;LOCK ON CURRENT LINE?
3021 021126 005200                INC R0              ;UPDATE LINE NO.
3022 021130 005302                DEC R2              ;4 LINES DONE
3023 021132 001402                BEQ .+6             ;BR IF YES
3024 021134 000137 020314        JMP 1$              ;JMP IF NO
3025 021140 000207                RTS PC              ;EXIT FOR NEXT 4 LINE GROUP
3026
3027 ;***** TEST 20 *****
3028 ;*TEST OF RECEIVER OVERRUN.
3029 ;*TEST TO TXMIT 134. CHARS AND RECVD 129
3030 ;*SERVICING THE FIRST CHAR AS A SPECIAL CHAR
3031 ;*AND STOPPING THE CHAR PROCESSOR.
3032 ;*WHEN THE TRANSMITTER FINISHES ALL 134. CHARS
3033 ;*THE RECEIVER IS RESTARTED AND THE NEXT ENTRY
3034 ;*IN THE RIC REG 5/B OVER RUN ON CHAR 202(8).
3035 ;*THIS TEST IS DONE FOR BOTH ASYNC AND SYNC LINE CARDS.
3036 ;*****
3037
3038 ; TEST 20
3039
3040 021142 012737 000020 001228  TST20: MOV #20,TSTNO
3041 021150 012737 002436 001216  MOV #.EDP,NEXT
3042 021156 012700 000000                MOV #0.,R0          ;PLACE LINE NUMBER INTO R0
3043 021162 113737 001406 001244  MOV# MASK.A,MASKX    ;PLACE "MASK"FOR CHARS INTO MASKX
3044 021170 013737 001416 001236  MOV L00.03,STAT     ;LOAD LINE CARD STATUS INTO STAT
3045 021176 100402                BMI 100$            ;BR IF LINE CARD NOT TO BE TESTED
3046 021200 004737 021310        JSR PC,105$         ;GO DO THE TEST FOR LINE CARD 1
3047 021204 012700 000004        MOV #4.,R0          ;PLACE LINE NUMBER INTO R0
3048 021210 113737 001407 001244  MOV# MASK.B,MASKX    ;GET MASK
3049 021216 013737 001420 001236  MOV L04.07,STAT     ;LOAD LINE CARD STATUS INTO STAT
3050 021224 100402                BMI 101$            ;BR IF LINE CARD NOT TO BE TESTED
3051 021226 004737 021310        JSR PC,105$         ;GO DO THE TEST FOR LINE CARD 2
3052 021232 012700 000010        MOV #8.,R0          ;LOAD LINE NUMBER
3053 021236 113737 001410 001244  MOV# MASK.C,MASKX    ;GET MASK
3054 021244 013737 001422 001236  MOV L08.11,STAT     ;LOAD LINE CARD STATUS INTO STAT
3055 021252 100402                BMI 102$            ;BR IF LINE CARD NOT TO BE TESTED
3056 021254 004737 021310        JSR PC,105$         ;DO THE TEST FOR LINE CARD 3
3057 021260 012700 000014        MOV #12.,R0         ;LOAD LINE NO.
3058 021264 113737 001411 001244  MOV# MASK.D,MASKX    ;GET MASK
    
```



```

3059 021272 013737 001424 001236      MOV    L12,15,STAT ;LOAD LINE CARD STATUS
3060 021300 100402                BMI    103$        ;BR IF LINE CARD NOT TO BE TESTED
3061 021302 004737 021310                JSR    PC,105$    ;DO THE TESTS FOR LINE CARD 4
3062 021306 104400                SCOPE                ;SCOPE THIS TEST.
3063 021310                SCOPE                ;TEST ENTRANCE.
3064 021310 012737 021348 001220      MOV    #1$,LOCK    ;RETURN FOR SW09
3065 021316 104413                RAMCLR                ;CLEAR ALL SEC REGISTERS
3066 021320 005004                CLR    R4           ;CLEAR
3067 021322 012705 030180                MOV    #RXTAB,R5   ;THE
3068 021326 005025                CLR    (R5)+       ;RECEIVER
3069 021330 105204                INCB  R4           ;CONTROL
3070 021332 100375                BPL    -4          ;TABLE
3071 021334 112737 000001 030181      MOV    #BIT0,RXTAB+1 ;SET "SPECIAL CHAR"(1)
3072 021342 012702 000001                MOV    #4,R2       ;4 LINE GROUP
3073 021346 010077 160020                MOV    R0,@DVSR5  ;LOAD LINE NO.
3074 021352 032737 004000 001236      BIT    #ASYNC,STAT ;IS THIS AN ASYNC LINE CARD?
3075 021360 001406                BEQ    80$         ;BR IF NOT ASYNC.
3076 021362 004537 022120                PERFORM ,SETREG    ;ADJUST FOR ASYNC LINE CARD
3077 021366 000000 001                .BYTE 000,001     ;REGISTERS
3078 021370 022550                TXBAP                ;LOAD FOR ASYNC
3079 021372 177573                -133.              ;LOAD FOR ASYNC
3080 021374 000405                BR     81$         ;CONTINUE TEST
3081 021378 004537 022120 80$: PERFORM ,SETREG    ;
3082 021402 000000 001                .BYTE 000,001     ;TX BA P, TX BC P
3083 021404 022556                SYNC                ;
3084 021406 177572                -134.              ;
3085 021410                81$: BIT    #ASYNC,STAT ;IS THIS AN ASYNC LINE CARD?
3086 021416 001406                BEQ    82$         ;BR IF NOT ASYNC.
3087 021420 004537 022120                PERFORM ,SETREG    ;ADJUST FOR ASYNC LINE CARD
3088 021424 000405                .BYTE 004,005     ;REGISTERS
3089 021426 027560                RXBA                ;LOAD FOR ASYNC
3090 021430 177576                -130.              ;LOAD FOR ASYNC
3091 021432 000405                BR     83$         ;CONTINUE TEST
3092 021434 004537 022120 82$: PERFORM ,SETREG    ;
3093 021440 000405 005                .BYTE 004,005     ;RX BA, RX BC
3094 021442 027560                RXBA                ;
3095 021444 177577                -129.              ;
3096 021446 004537 022120 83$: PERFORM ,SETREG    ;
3097 021452 010011 011                .BYTE 010,011     ;TX TAB, RX TAB
3098 021454 023560                TXTAB                ;
3099 021456 030160                RXTAB                ;
3100 021460 004537 022120                PERFORM ,SETREG    ;
3101 021464 000303 012                .BYTE 013,012     ;LINE STATE, LINE PROTOCOL PARAM
3102 021466 000004                BIT2                ;TX GO
3103 021470 000101                BIT6+BIT0           ;TX DDCMP + IDLE MARK
3104 021472 032737 004000 001236      BIT    #ASYNC,STAT ;IS THIS ASYNC LINE CARD?
3105 021500 001412                BEQ    80$         ;BR IF NO.
3106 021502 004537 022164                PERFORM ,LOAD.MODE ;LOAD PARAMETERS.
3107 021506 020000                BIT13               ;RECEIVER ENABLE
3108 021510 004537 022164                PERFORM ,LOAD.MODE ;#
3109 021514 015000                <BIT12+BIT11>+BIT9 ;#8 BITS/PER/CHAR
3110 021516 004537 022164                PERFORM ,LOAD.MODE ;#
3111 021522 072000                <BIT14+BIT13+BIT12>+BIT10 ;#8600 BAUD.
3112 021524 000403                BR     81$
3113
3114

```

```

3115 021526 004537 022164 80$: PERFORM ,LOAD.MODE ;LOAD
3116 021532 034000                BIT13+BIT12+BIT11 ;MODE
3117 021534 012705 022560 81$: MOV    #TXBAP,R5   ;LOAD
3118 021540 005004                CLR    R4           ;TX
3119 021542 105204                INCB  R4           ;DATA
3120 021544 001402                BEQ    21$         ;BUFFER
3121 021546 110425                MOV    R4,(R5)+   ;
3122 021550 000774                BR     2$          ;
3123 021552 004537 021706 21$: PERFORM ,SETSYNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
3124 021556 005277 157600                INC    @DVSCR      ;SET UCPU GO
3125 021562 105777 157574                TSTB  @DVSCR      ;DVSCR07=1?
3126 021566 100375                BPL    -4          ;BR IF NO
3127 021570 005777 157566                TST  @DVSCR      ;DVSCR15=1?
3128 021574 100375                BPL    -4          ;BR IF NO
3129 021576 112777 000012 157570      MOV    #12,@DVSR5H ;LINE PROTOCOL PARAM.
3130 021604 052777 000040 157564      BIS    #BITS,@DVSR ;SET RX DDCMP
3131 021612 052777 000400 157542      BIS    #BIT8,@DVSCR ;RESTART
3132 021620 105777 157536                TSTB  @DVSCR      ;DVSCR07=1?
3133 021624 100375                BPL    -4          ;BR IF NO
3134 021626 017704 157534                MOV    @DVRIC,R4  ;READ RIC
3135 021632 010005                MOV    R0,R5       ;LINE
3136 021634 000305                SWAB  R5           ;HIGH BYTE
3137 021636 052705 020202                BIS    #BIT13+202,R5 ;130.
3138 021642 032737 004000 001236      BIT    #ASYNC,STAT ;IS THIS AN ASYNC LINE CARD?
3139 021650 001401                BEQ    -4          ;BR IF NOT ASYNC
3140 021652 005205                INC    R5          ;ADJUST FOR ASYNC. DOUBLE BUFFER CAUSES
3141                                ;CHAR TO BE ONE MORE THAN SYNC LINE CARD.
3142 021654 143705 001244                BICB  MASKX,R5    ;CLEAR UNUSED BITS
3143 021660 020504                CMP    R5,R4       ;RIC OK?
3144 021662 001401                BEQ    3$         ;
3145 021664 1.4001                HLT    1           ;NO OVER-RUN; OR ON WRONG CHAR!
3146 021666 104412                3$: MSTCLR                ;RESET DVA
3147 021670 104401                SCOPI                ;LOCK ON CURRENT LINE?
3148 021672 005200                INC    R0          ;UPDATE LINE NO.
3149 021674 005302                DEC    R2          ;4 LINES DONE
3150 021676 001402                BEQ    -4          ;BR IF YES
3151 021700 000137 021348                JMP    1$         ;JMP IF YES
3152 021704 000207                RTS                ;EXIT
3153
3154 021706                SETSYNC:
3155 021708 113737 001236 022556      MOV    STAT,SYNC   ;SET SYNC FOR THIS LINE.
3156 021714 113737 022556 022557      MOV    SYNC,SYNC+1 ;PLACE SYNC IN HIGH BYTE
3157 021722 032737 010000 001236      BIT    #TWO,STAT   ;ONE SYNC OR TWO?
3158 021730 001402                BEQ    1$         ;BR IF JUMPED FOR TWO.
3159 021732 105037 022556                CLRB  SYNC         ;SET FIRST SYNC TO NON-SYNC
3160 021736 000205                1$: EXIT
3161 021740 010046                SIMBCC: MOV    R0,-(SP)
3162 021742 010146                MOV    R1,-(SP)
3163 021744 010246                MOV    R2,-(SP)
3164 021746 012537 001246                MOV    (R5)+,TEMP1
3165 021752 012537 001250                MOV    (R5)+,TEMP2
3166 021756 012537 001252                MOV    (R5)+,TEMP3
3167 021762 005037 022114 1$: CLR    BCCFBK
3168 021766 013700 001252                MOV    TEMP3,R0
3169 021772 006037 001250                ROR   TEMP2
3170 021776 005500                ADC   R0

```

```

3171 022000 032700 000001 BIT #BIT0,R0
3172 022004 001402 BEQ 2$
3173 022006 005137 022114 COM BCCFBK
3174 022012 013700 022112 2$: MOV XPOLY,R0
3175 022016 005100 COM R0
3176 022020 040037 022114 R0,BCCFBK
3177 022024 000241 CLC
3178 022026 006037 001252 RCR TEMP3
3179 022032 013700 022114 MOV BCCFBK,R0
3180 022036 013701 001252 MOV TEMP3,R1
3181 022042 010102 MOV R1,R2
3182 022044 040100 BIC R1,R0
3183 022046 043702 022114 BIC BCCFBK,R2
3184 022052 050200 BIC R2,R0
3185 022054 043737 022112 001252 BIT XPOLY,TEMP3
3186 022062 050037 001252 BIS R0,TEMP3
3187 022066 005337 001246 DEC TEMP1
3188 022072 001333 BNE 1$
3189 022074 013737 001252 022116 MOV TEMP3,CALBCC
3190 022102 012602 MOV (SP)+,R2
3191 022104 012601 MOV (SP)+,R1
3192 022106 012600 MOV (SP)+,R0
3193 022110 000205 RTS R5
3194 022112 000000 XPOLY: 0
3195 022114 000000 BCCFBK: 0
3196 022116 000000 CALBCC: 0
3197 000200 LRCB=200
3198 120001 CRC16=120001
3199 102010 CRC.CCITT=102010
3200
3201
3202 022120 010046 SETREG: MOV R0,-(SP)
3203 022122 010146 MOV R1,-(SP)
3204 022124 112500 MOVB (R5)+,R0
3205 022126 112501 MOVB (R5)+,R1
3206 022130 110077 157240 MOVB R0,@DVSRSH
3207 022134 012577 157236 MOV (R5)+,@DVSR
3208 022140 042777 000060 157214 BIC #BIT5+BIT4,@DVSCR
3209 022146 110177 157222 MOVB R1,@DVSRSH
3210 022152 012577 157220 MOV (R5)+,@DVSR
3211 022156 012601 MOV (SP)+,R1
3212 022160 012600 MOV (SP)+,R0
3213 022162 000205 EXIT
3214
3215 022164 LOAD.MODE:
3216 022164 012577 157200 MOV (R5)+,@DVLCR
3217 022170 052777 100000 157172 BIS #BIT15,@DVLCR
3218 022176 010046 MOV R0,-(SP)
3219 022200 005000 CLR R0
3220 022202 005777 157162 1$: TST @DVLCR
3221 022206 100004 BPL 2$
3222 022210 104414 DELAY
3223 022212 005200 INC R0
3224 022214 001372 BNE 1$
3225 022216 104000 HLT 0 ;BIT 15 FAILED TO CLEAR
3226 022220 012600 2$: MOV (SP)+,R0
    
```

```

3227 022222 000205 EXIT
3228
3229 ;SUBROUTINE.
3230 ;CORE TABLES ALREAY SET UP
3231 ;XMIT 3 CHARS 2SYNC+ 1 DATA
3232 ;RCV 1 CHAR
3233 022224 010077 157142 DV11DN: MOV R0,@DVSR
3234 022230 032737 004000 001236 BIT #ASYNC,STAT ;#IS THIS AN ASYNC LINE CARD?
3235 022236 001406 BEQ 60$ ;#BR IF NOT ASYNC.
3236 022240 004537 022120 PERFORM ,SETREG ;#ADJUST FOR ASYNC LINE CARD
3237 022244 000 001 .BYTE 000,001 ;#REGISTERS
3238 022246 022560 TXBAP ;#LOAD FOR ASYNC
3239 022250 177777 -1 ;#LOAD FOR ASYNC
3240 022252 000405 -1 BR 81$ ;#CONTINUE TEST
3241 022254 004537 022120 80$: PERFORM ,SETREG
3242 022260 000 001 .BYTE 000,001
3243 022262 022556 SYNC
3244 022264 177775 -3
3245 022266 004537 022120 81$: PERFORM ,SETREG
3246 022272 004 005 .BYTE 004,005
3247 022274 027560 RXBA
3248 022276 177777 -1
3249 022300 004537 022120 PERFORM ,SETREG
3250 022304 010 011 .BYTE 010,011
3251 022306 023560 TXTAB
3252 022310 030160 RXTAB
3253 022312 004537 022120 PERFORM ,SETREG
3254 022316 013 012 .BYTE 013,012
3255 022320 000004 BIT2
3256 022322 000001 BIT0
3257 022324 032737 004000 001236 BIT #ASYNC,STAT ;#IS THIS ASYNC LINE CARD?
3258 022332 001412 BEQ 60$ ;#BR IF NO.
3259 022334 004537 022164 PERFORM ,LOAD.MODE ;#LOAD PARAMETERS.
3260 022340 020000 BIT13 ;#RECEIVER ENABLE
3261 022342 004537 022164 PERFORM ,LOAD.MODE ;#
3262 022346 015000 <BIT12+BIT11>+BIT9 ;#8 BITS/PER/CHAR
3263 022350 004537 022164 PERFORM ,LOAD.MODE ;#
3264 022354 072000 <BIT14+BIT13+BIT12>+BIT10 ;#9600 BAUD.
3265
3266 022356 000405 BR 61$
3267 022360 004537 022164 60$: PERFORM ,LOAD.MODE
3268 022364 030000 BIT13+BIT12+BIT11
3269 022366 004537 021706 PERFORM ,SETSYNC ;GET SYNC CHARS AND ADJUST FOR ONE OR TWO.
3270 022372 000267 61$: RTS PC
3271
3272
3273 022374 SETSCAN:
3274 022374 010346 MCV R3,-(SP)
3275 022376 052777 000010 156756 BIS #BIT3,@DVSCR
3276 022404 012503 MOV (R5)+,R3
3277 022406 014144 BEQ 2$
3278 022410 012777 050102 156762 1$: MOV #BIT14+BIT12+BIT6+BIT1,@DVSR
3279 022416 104415 ROMCLK
3280 022420 005201 INC R1
3281 022422 012777 050102 156750 MOV #BIT14+BIT12+BIT6+BIT1,@DVSR
3282 022430 104415 ROMCLK
    
```

```

3283 022432 005201          INC      R1
3284 022434 005303          DEC      R3
3285 022436 001364          BNE     1$
3286 022440 012603          2$:    MOV     (SP)+,R3
3287 022442 010100          MOV     R1,R0
3288 022444 000241          CLC
3289 022446 006000          ROR     R0
3290 022450 000205          EXIT
3291
3292 022452 000042          REGBUF: .BLKW 34.
3293 022556 000001          SYNC:  .BLKB 1
3294 022560 000400          TXBAP: .BLKB 400
3295 023160 000400          TXBAS: .BLKB 400
3296 023560 000400          XTAB:  .BLKB 400
3297 024160 000400          .BLKB 400
3298 024560 000400          .BLKB 400
3299 025160 000400          .BLKB 400
3300 025560 000400          .BLKB 400
3301 026160 000400          .BLKB 400
3302 026560 000400          .BLKB 400
3303 027160 000400          .BLKB 400
3304 027560 000400          RXBA:  .BLKB 400
3305 030160 000400          RXTAB: .BLKB 400
3306 030560 000400          .BLKB 400
3307 031160 000400          .BLKB 400
3308 031560 000400          .BLKB 400
3309 032160 000400          .BLKB 400
3310 032600 000400          .BLKB 400
3311 033160 000400          .BLKB 400
3312 033560 000400          .BLKB 400
3313 034160 000000          DATA: 0
3314 034162 043377 042522 020105 EM1:  .ASCIZ <377>/FREE RUNNING ROM TESTS/
      034212 051377 041505 044505 EM2:  .ASCIZ <377>/RECEIVER CONTROL BYTE TEST./
      034247 377 051124 047101 EM3:  .ASCIZ <377>/TRANSMITTER CONTROL BYTE TEST./
      034307 377 042522 042503 EM4:  .ASCIZ <377>/RECEIVER BCC ERROR/
      034333 377 054105 042520 DH1:  .ASCIZ <377>/EXPECTED FOUND LINE(8)/
      .EVEN
      DT1: 3
3315 034370 006 004          .BYTE 6,4
3316 034372 001272          SAVR5
3317 034374 006 002          .BYTE 6,2
3318 034376 001270          SAVR4
3319 034400 002 001          .BYTE 2,1
3320 034402 001260          SAVR0
3321
3322 034404          .ERRTAB:
3323 034404 000000          0
3324 034408 000000          0
3325 034410 000000          0
3326 034412 034162          EM1
3327 034414 034333          DH1 ;HALT 1
3328 034416 034366          DT1
3329 034420 034212          EM2
3330 034422 034333          DH1 ;HALT 2
3331 034424 034366          DT1
3332 034426 034247          EM3

```

```

3333 034430 034333          DH1 ;HALT 3
3334 034432 034366          DT1
3335 034434 034307          EM4
3336 034436 034303          DH1 ;HALT 4
3337 034440 034366          DT1
3338
3339 034442          !*****
      CORMAX:
3340 000001          .END

```


ERRORS DETECTED: 0

CZDVDC,CZDVDC/SOL/CRF=CZDVDC.MAC,CZDVDC.P11
RUN-TIME: 9 13 1 SECONDS
RUN-TIME RATIO: 112/24=4.6
CORE USED: 25K (49 PAGES)